Administrative instructions: Please turn in sheets to Richard Kueng before the exercise session. A complete solution will be worth ten points. Solutions should be submitted in groups of up to three. Check the web site for occasional hints and updates.

[P1] [*Turing undecidability*] The aim of this exercise is to prove that there are decision problems that no Turing machine is able to answer. This can be done by employing a counting argument along the lines outlined below.

(1) Recall *Cantor's diagonal argument* for the fact that the set of real numbers is uncountable (find it online, if you haven't encountered it during your first-year math classes). The *power set* $\mathcal{P}(S)$ of a set S is the set of all subsets of S. Prove that $\mathcal{P}(\mathbb{N})$ is uncountable. (2 P.)

(2) Recall that a *language* $\mathcal{L}$ is a subset of the set $\{0,1\}^*$ of binary strings (e.g. we have encountered the language of strings of odd parity). A Turing machine T *decides* the language $\mathcal{L}$ if (*i*) T takes bit strings as input, (*ii*) for every input x, T halts after finitely many steps, (*iii*) T will write "1" to a designated cell of its output tape if $x \in \mathcal{L}$, and it will write "0" to that cell if $x \notin \mathcal{L}$. Using just the results of (1), give a very short proof that there are Turing-undecidable languages. (2 P.)

[P2] [*Programming with Turing machines*] Here, the goal is to design simple Turing machines, in order to get a feel for this computational model.

(1) Design a Turing machine T_{+1} that computes the function $x \mapsto x+1$. I.e. when the tape initially contains x in binary representation, the machine should halt after having written the binary representation of $x+1$ to the tape. Thus, the alphabet Σ must certainly contain $\{\Box, 0, 1\}$, but you can add more symbols if you wish. Completely specify the states in Q and the transition function δ. (2 P.)

(2) Likewise, design a Turing machine T_{-1} that substracts minus one from a positive number given in *decimal representation*. That is, both the input x and the output $x-1$ are represented by decimal numbers. When the input is 0, the output should be an empty tape. There should be no leading zeros in the output – e.g. $T_{-1}(10) = 9$ and not 09. (2 P.)

(3) *Decimal to binary conversion*: Argue—in words—how the two Turing machines constructed above could be combined to one that converts any positive decimal number input into binary representation (e.g. $3 \mapsto 11$, or $6 \mapsto 110$). (2 P.)

Hint: Plenty of interactive Turing machine simulators can be found online (see e.g. www.morphett.info/turing/ for one that is particularly easy to use). You can test and debug your code by having it run on one of them.