ASSIGNMENT 2

FORMAL PROOFS AND THE LIMITS OF COMPUTATION
SPRING 2022

Florestan Brunck & Yunzhe Li

Turing Machines Can Play with Themselves

The goal of this exercise is to show that Turing machine can obtain their own description and compute on it.

Exercise 1 (Turing Machines can Self-Replicate).

(i) *Show that there exists a computable function*

$$q : \begin{cases} \Sigma^* & \longrightarrow \Sigma^* \\ w & \longmapsto \langle M_w \rangle \end{cases}$$

where M_w is a Turing machine that outputs w and halts.

(ii) *We want to build a Turing machine that ignores its input and prints a copy of its own description. You might think first of something analogous to the sentence:*

Print this sentence.

While this sentence makes perfect sense to us, what is "this" referring to? In order to accommodate for this self-reference problem, we propose the following alternative formulation $(\star)$*:*

Print the following sentence twice, the second time with quotes around it: "Print the following sentence twice, the second time with quotes around it:"

Following this example, we want to build a Turing machine SELF, whose description decomposes in two parts: $\langle SELF \rangle = \langle A \rangle \langle B \rangle$. A runs first, outputs a description of B and passes control to B. Show how to use (*i*) *to build A.*

(iii) *Show that if B is given $\langle B \rangle$ as input, it can return $\langle A \rangle$ as output.*

(iv) *Note that B can obtain $\langle B \rangle$.*

(v) *Using A and B, build a Turing machine SELF, that ignores its input and prints out $\langle SELF \rangle$.*

(vi) *Which part of* $(\star)$ *corresponds to A, respectively to B?*

Exercise 2 (The Recursion Theorem). *We want to show that, not only can Turing machines output their own description, they can also obtain their own description* *and* *compute with it. To formalise this, let* $T : \Sigma^* \times \Sigma^* \to \Sigma^*$ *be a Turing machine that takes as input two arguments (which you can think of as being concatenated on its input tape), the description of a Turing machine and a string. Show that there exists a Turing machine* R *which computes as* T *does on input* $(\langle R \rangle, w)$*, i.e :*

$$R(w) = T(\langle R \rangle, w)$$

Suggestion. Build R the same way we built $SELF$, except that R now requires three parts, A, B and T, where T is the Turing machine in the statement of the theorem.

An Example of a True, but non Provable Sentence

Exercise 3 (Liar, liar, Turing machine on fire!). *Construct a* Π_1 *formula that corresponds to the sentence "This sentence is not provable" (the logical equivalent of the Liar's Paradox: "This sentence is a lie") and show that it is true, but not provable.*

Suggestions. Recall the following useful facts we have seen in class:

1. Given a Turing Machine M and a string w, we can construct (using a Turing machine) a sentence $\phi_{M,w}$ containing a single free variable s, such that the Σ_1 formula $\exists s \phi_{M,w}$ is true if and only if M accepts w.

2. The negation of a Σ_1 formula is a Π_1 formula.

3. Turing machines can obtain their own description and compute on it.

4. The set of provable sentences $\{\phi \in \mathrm{Th}(\mathbb{N}, +, \times) \mid \phi \text{ is provable}\}$ is Turing recognisable, *i.e* there exists a Turing machine P listing all the provable sentences.

Hint: Using these 4 facts, you may want to build a Turing machine that constructs the Σ_1 sentence for itself (and some arbitrary string) and calls onto P.

Exercise 4 (The Complexity Barrier). *Fix a universal Turing machine U with input alphabet Σ. We implicitly identify every natural number with a unique string over Σ via base-$|\Sigma|$ encoding.*

1. *Show that for any $L > 0$ the language $\{s \mid H_U(s) \leq L\}$ is Turing recognisable. Use the representation theorem to deduce that there exists a Π_1 arithmetical formula ψ_L such that*

$$\text{for all } s \in \Sigma^*, \quad H_U(s) > L \iff \psi_L(s) \text{ is true.}$$

2. *Let $\mathcal{A}$ be a sound axiomatic system of $\mathbb{N}$. Show that there exists a constant L such that $\mathcal{A}$ can't prove any strings Kolmogorov complexity is greater than L. In other words, for all $s \in \Sigma^*$, the sentence $\psi_L(s)$ is not provable in $\mathcal{A}$.*

Suggestion. Construct a Turing machine which does the following.

- Given an input $L \in \mathbb{N}$,
- List the set of provable sentences in $\mathcal{A}$.
- If $\psi_L(n)$ appears as a provable sentence for some n, then output n and halt.

Then consider $L >> 1$ such that $|\langle M, \langle L \rangle\rangle| < L$.