Solutions to Supplementary Problems

S10.1 Prove the following connections between computable functions and languages:

(a) A language $A \subseteq \Sigma^*$ is decidable, if and only if its characteristic function

$$\chi_A : \Sigma^* \to \{0,1\}, \qquad \chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

is a computable function.

(b) A language $A \subseteq \Sigma^*$ is semi-decidable, if and only if either $A = \emptyset$ or there exists a computable function $g : \{0,1\}^* \to \Sigma^*$ such that

$$A = \{g(x) \mid x \in \{0,1\}^*\}.$$

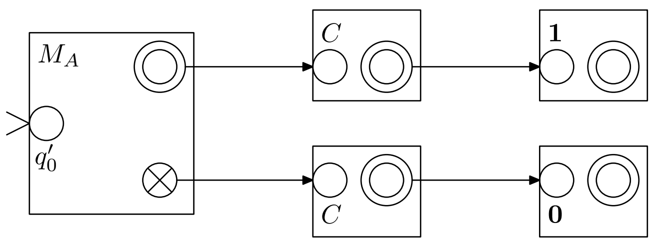
Note: This is the characterisation underlying the historic term "recursively enumerable language": a language A is r.e. if and only if its elements can be "enumerated" by a recursive (= computable) function g.

Solution. We start by defining five simple helper machines:

- **1** is a 1-tape machine that writes a '1' on its tape, moves the tape head right and stops.
- **0** is a 1-tape machine that writes a '0' on the tape, moves the tape head right and stops.
- C is a 1-tape machine that clears its tape (i.e. writes some special 'empty' symbols over its former contents), moves the head to the beginning of the tape and stops.
- NEXT^i is a multitape machine that replaces the string on its tape i with its shortlex successor.
- $\text{CMP}^{i,j}$ is a multitape machine that compares the contents on its tapes i and j and accepts if they are identical.

Since the machines are simple, they are not presented here.

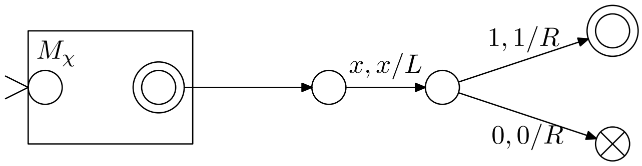
(a) [⇒] Let $A \subseteq \Sigma^*$ be a decidable language and M_A a total 1-tape Turing machine that recognises it. We construct a 1-tape machine M by combining M_A with machines **1**, **0**, C as follows



where the double circle denotes an accept state and a crossed circle a reject state.

If $w \in L$, then M_A accepts w. After that M clears the tape and writes a 1 at its beginning. Correspondingly, if M_A rejects w, then a 0 is written. Since M_A is total then so is M, and it computes the function $\chi(w) = \begin{cases} 1 & \text{if } w \in A \\ 0 & \text{if } w \notin A \end{cases}$ as desired.

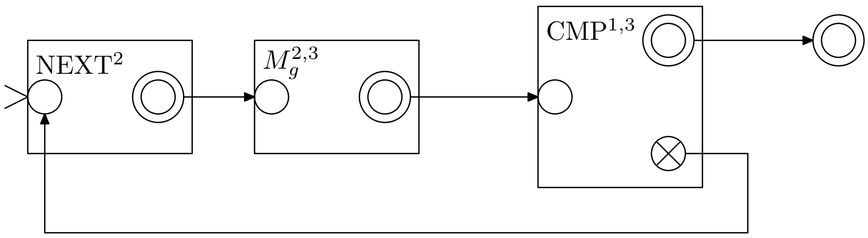
[⇐] Assume that the function $\chi(w) = \chi_A(w)$ is computable. Then there exists a total Turing machine M_χ that computes it. Let us construct a new total machine M as follows:



Now M accepts w when $\chi(w) = 1$ and rejects it when $\chi(w) = 0$, and thus is a decider for the language A.

(b) [⇐] If $A = \emptyset$, then A is trivially semi-decidable by a machine that rejects all inputs.

Otherwise, let g be an enumeration function that fulfils the given conditions, and let M_g be a 1-tape total Turing machine that computes it. We can easily modify M_g into a 2-tape machine $M_g^{1,2}$ that computes g, but stores the result on the second tape instead of the first. We now construct a 3-tape machine M_A that has the following structure:



Initially, the input string w for M_A is placed on tape 1, where it stays untouched throughout the computation. Tapes 2 and 3 initially contain the empty string. The computation of M_A proceeds by iteration over all possible argument values x for the enumeration function $g(x)$. In each iteration round, M_A first replaces the bit string x_{prev} on tape 2 by its shortlex successor x and computes, using machine $M_g^{2,3}$ the string $g(x)$ on tape 3. M_A then compares the contents of tapes 1 and 3 and if they match, it accepts the input string $w = g(x)$; otherwise it proceeds to the next iteration round.

It is clear that if an input string w is a word in A and thus $w = g(x)$ for some $x \in \{0,1\}^*$, then M_A accepts w once the enumeration process reaches the first x (in the shortlex ordering) which maps to w. On the other hand, if w is not in A, then there is no x for which $g(x) = w$, and so M_A rejects w by not terminating. Thus, M_A is a (semi-)recogniser for A.

[⇒] Let $A \neq \emptyset$ be a semi-decidable language, and let M_A be a single-tape Turing machine M_A that recognises it. For simplicity and without loss of generality, let us assume that $A \subseteq \{0,1\}^*$ and that M_A always rejects by not halting. Let us define an encoding for any word $w = a_1 \ldots a_n \in \{0,1\}^*$ as $\widehat{w} = a_1 a_1 \ldots a_n a_n$.

We now modify machine M_A to a 3-tape machine $\widehat{M}$ that reads an input of the form $x = \widehat{w}101^t$ on tape 1, decodes $\widehat{w}$ as w on tape 2, copies 1^t on tape 3, and then simulates M_A on tape 2 for t steps, using the string 1^t on tape 3 as a time counter. If M_A accepts w in this time, then $\widehat{M}$ accepts its input $x = \widehat{w}101^t$, otherwise rejects (also in the case that x was not of the appropriate form). Note that $\widehat{M}$ as defined is a total machine, i.e. it halts on all inputs.

Let then w_0 be some fixed string in $A \neq \emptyset$. We define the enumeration function $g(x)$ as follows:

$$g(x) = \begin{cases} w, & \text{if } x = \widehat{w}101^t \text{ and } \widehat{M} \text{ accepts } x; \\ w_0, & \text{otherwise.} \end{cases}$$

It is now clearly the case that:

(i) the function $g(x)$ is computable;

(ii) for every $x \in \{0,1\}^*$, $g(x) \in A$;

(iii) for every $w \in A$, $g(x) = w$ for some $x \in \{0,1\}^*$ (choose $x = \widehat{w}101^t$ where M_A accepts w in t steps).

The function $g(x)$ thus satisfies the required conditions.

S10.2

(a) Prove that any decision problem that has only finitely many possible inputs is decidable.

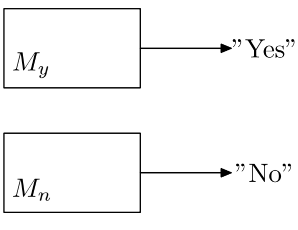
(b) Prove that the problem "Does the decimal expansion of π contain 1000 consecutive zeros?" is decidable. What does this result tell you about (i) the decimal expansion of π, (ii) the notions of decidability and undecidability?

Solution.

(a) If a problem has only a finite number of possible inputs, one can construct a simple table look-up Turing machine that has all the inputs and their correct answers encoded into its states. In processing, the machine just checks the string on the input tape against the tabulated ones and provides the corresponding result.

(b) Because the decision problem "Does the decimal expansion of π contain 1000 consecutive zeros?" concerns a specific fixed number π, it actually *has no inputs* and only one fixed answer: either π has 1000 consecutive zeros and the answer is "Yes", or it does not have and the answer is "No".

Thus, one of the following Turing machines decides the problem:



We just do not know at present which one of these machines is the correct one. (Though M_y would seem to be more likely.)[1]

Answers to the continuation questions: (i) Nothing. (ii) The concepts of decidability and undecidability are only relevant for problems that have infinitely many possible inputs: for finite input spaces, and for questions that concern one specific problem instance, the results that one obtains within this framework are noninformative.[2].

[1] Cf. https://bellard.org/pi/pi2700e9/pidigits.html, https://mathoverflow.net/questions/23547/does-pi-contain-1000-consecutive-zeroes-in-base-10.

[2] However, see https://doi.org/10.1137/S0097539794268789.