PART II AUTOMATA AND FORMAL LANGUAGES
MICHAELMAS 2018-19
EXAMPLE SHEET 2

As on Example Sheet 1, you may use the phrase "using Church's thesis" to mean "we give an informal proof using the conventions of Church's thesis". You may use this phrase in your proofs, provided you clearly state it.
Unless explicitly asked to, you need not *prove* that any machine you construct defines the language you say it does.

$*$ denotes a harder problem.

(1) Let A be a recursive set, and define the set

$$B = \{2n \mid n \in A\} \cup \{2n + 1 \mid n \in \mathbb{K}\}$$

(a) Is B recursive? If not, which of B and $\mathbb{N} \setminus B$ are r.e. (if any), and why?

(b) By replacing A in the construction of B with a suitably-chosen set, construct a set $C \subseteq \mathbb{N}$ such that neither C nor $\mathbb{N} \setminus C$ are r.e.

(2) Give examples, with proofs, of infinite collections of recursive sets whose union:

(a) Is recursive.

(b) Is r.e. but not recursive.

(c) Is not r.e., and its complement is not r.e. either.

(3) Construct DFA's, via transition diagrams, which accept the following languages:

(a) $\{w \in \{0, 1\}^* \mid |w| > 2\}$.

(b) $\{w \in \{0, 1\}^* \mid w$ is an alternating sequence of 1's and 0's $\}$.

(c) $\{w \in \{0, 1\}^* \mid w$ is a multiple of 3 when interpreted in binary $\}$.

(d) $\{w \in \{a, \ldots, z\}^* \mid w$ contains *ababa* as a substring $\}$.

(4) Construct NFA's, via transition diagrams, which accept the following languages:

(a) $\{w \in \{a, \ldots, z\}^* \mid w$ contains *ababa* as a substring $\}$.

(b) $\{w \in \{a, \ldots, z\}^* \mid w$ contains *dpmms* and/or *damtp* as a substring $\}$.

(c) $\{w \in \{a, \ldots, z\}^* \mid w \in \{what, where, when\}\}$.

Use the subset construction to convert (4a) to a DFA.

Date: October 22, 2018.

(5) Construct an ϵ-NFA which accepts the *union* of the three languages from question (4), and has only *one* accept state.

(6) Let L be a regular language over Σ. Prove that the complement of L, $\Sigma^* \setminus L$, is also a regular language over Σ.

(7) Let Σ be a finite alphabet, and L be a regular language over Σ.

(a) Fix an ordering of Σ, and use this to describe a well-ordering $\{w_1, w_2, \ldots\}$ of Σ^* by extending the idea of the shortlex ordering of $\mathbb{N}^m$.

(b) Using this ordering of Σ^*, show that $\{n \in \mathbb{N} \mid w_n \in L\}$ is a recursive set.

(8) Prove that the set $\{n \in \mathbb{N} \mid n \text{ codes a program and } |W_n| > 5\}$ is r.e., but not recursive.

(9) Show that if $g : \mathbb{N}^2 \to \mathbb{N}$ is partial recursive, then there is some $e \in \mathbb{N}$ such that

$$f_{e,1}(y) = g(e, y) \quad \forall y \in \mathbb{N}.$$

Use this to show there exists some $m \in \mathbb{N}$ such that W_m has exactly m elements.

(10) We define the following sets:
Tot $:= \{n \in \mathbb{N} \mid n \text{ codes a program and } f_{n,1} \text{ is total}\}$, the indices of ***tot***al P.R. functions.
Inf $:= \{n \in \mathbb{N} \mid n \text{ codes a program and } |W_n| = \infty\}$, the indices of ***inf***inite r.e. sets.
Fin $:= \{n \in \mathbb{N} \mid n \text{ codes a program and } |W_n| < \infty\}$, the indices of ***fin***ite r.e. sets.

(a) Show that **Tot** $= \{n \in \mathbb{N} \mid n \text{ codes a program and } W_n = \mathbb{N}\}$

(b) Show that $\mathbb{K} \leq_m$ **Tot**, and that $\mathbb{K} \leq_m$ **Inf**.

(c) Are any of **Tot**, **Inf**, **Fin** recursive? Give reasons.

(d) Show that **Inf** $\leq_m$ **Tot**.
Hint: Try adapting the proof of Theorem 1.27 *from the notes.*

(e*) Show that $\mathbb{K} \leq_m$ **Fin**.

(f) Using the above results, show that **Tot** is not r.e., and neither is $\mathbb{N} \setminus$ **Tot**.

(11) Given an explicit code m for a register machine P_m, consider the set

$$T_m := \{n \in \mathbb{N} \mid n \text{ codes a program and } W_m \subseteq W_n\}$$

Construct two explicit codes m, m' such that T_m is recursive, and $T_{m'}$ is not r.e.
Hint: Use the results of question (10).

(12) Let g be a total recursive function on k variables. Show that, with **Tot** as in (10), we have **Tot** $\leq_m \{n \in \mathbb{N} \mid n \text{ codes a program and } f_{n,k} = g \text{ as functions }\}$. Hence show there is no partial algorithm to verify if answers to question (1) of Example Sheet 1 are correct, nor a partial algorithm to verify if they are incorrect. (That is, the set of correct answers is not r.e., and neither is the set of incorrect answers). What does this mean for someone trying to write an antivirus program?

(13*) Give an example of an infinite collection of recursive sets $\{W_n\}_{n \in I}$, whose indexing set I is r.e., for which $\bigcap_{n \in I} W_n$ is not r.e.