Exercise Sheet 1:
Mathematical Foundations, Decidability, and Recognisability

David Carral

October 16, 2019

Exercise Sheet 1

Exercise 1. Consider a non-empty set M and a function $f : M \to 2^M$. Show that f is NOT surjective.

Definition. A function $f : X \to Y$ is *surjective* iff, for every $y \in Y$, there is some $x \in X$ with $f(x) = y$.

Solution.

1. Suppose for a contradiction that there is a surjective function such as f. That is, f is a function such that, for every $y \in 2^M$, there is some $x \in M$ with $f(x) = y$.
2. Let $D = \{x \mid x \in M \text{ and } x \notin f(x)\}$. Note that $D \subseteq M$ and hence, $D \in 2^M$.
3. By (1) and (2): There is some element $x' \in M$ with $f(x') = D$.
4. By (2): If $x' \in D$, then $x' \notin f(x')$.
5. By (2) and (3): If $x' \notin f(x')$, then $x' \in D$.
6. By (3–5): $x' \in D$ iff $x' \notin D$—a contradiction!

Exercise Sheet 1

Exercise 2. Show the following claims.

1. $|\mathbb{N}| = |\mathbb{N} \times \mathbb{N}|$.
2. $|\mathbb{N}| = |\mathbb{Q}|$.
3. $|\mathbb{N}| \neq |\mathbb{R}|$.

Exercise Sheet 1

Exercise 2. Show that $|\mathbb{N}| = |\mathbb{N} \times \mathbb{N}|$.

Definition.

- Two sets X and Y have the same cardinality iff there exists a *bijection* from A to B. That is, a function $f : X \to Y$ that is both injective and surjective.
- Consider a function $f : X \to Y$.
  - f is *injective* iff, for every $x, y \in X$ with $x \neq y$, we have that $f(x) \neq f(y)$.
  - f is *surjective* iff, for every $y \in Y$, there is some $x \in X$ with $f(x) = y$.

Solution. $f(n_1, n_2) = \frac{(n_1+n_2)(n_1+n_2+1)}{2} + n_1$ is a bijection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$.

$$
\begin{array}{llllll}
f(0,0) = 0 & f(0,1) = 1 & f(1,0) = 2 & f(0,2) = 3 & f(1,1) = 4 & f(2,0) = 5 \\
f(0,3) = 6 & f(1,2) = 7 & f(2,1) = 8 & f(3,0) = 9 & f(0,4) = 10 & f(1,3) = 11
\end{array}
$$

(Yes, I assume that $0 \in \mathbb{N}$.)

Exercise Sheet 1

Exercise 2. Show that $|\mathbb{N}| = |\mathbb{Q}|$.

Solution.

- Every positive integer can be written uniquely as $p_1^{a_1} \cdot p_2^{a_2} \cdot \ldots$, where the $p_1 = 2, p_2 = 3, p_3 = 5, \ldots$ is the sequence of all primes, and the a_i are non-negative integers, and are non-zero for only finitely may is.
- Every positive rational number can be written uniquely as $p_1^{b_1} \cdot p_2^{b_2} \cdot \ldots$, where the b_i are (possibly negative) integers and only finitely many of the b_i are non-zero.
- Let $s : \mathbb{N} \to \mathbb{Z}$ (where we take $\mathbb{N}$ to include 0):

  $$s(n) = (-1)^n \left\lfloor \frac{n+1}{2} \right\rfloor$$

  The sequence $s(0), s(1), s(2), s(3), \ldots$ would be $0, -1, 1, -2, 2, \ldots$
- The function s is a bijection from $\mathbb{N}$ to $\mathbb{Z}$.
- For any $n = p_1^{a_1} \cdot p_2^{a_2} \cdot \ldots$, let $h(n) = p_1^{s(a_1)} \cdot p_2^{s(a_2)} \cdot \ldots$
- h is a bijection from $\mathbb{N}$ to $\mathbb{Q}$.

Exercise Sheet 1

Exercise 2. Show that $|\mathbb{N}| \neq |\mathbb{R}|$.

Solution.

1. By Exercise 1: $|\mathbb{N}| < |2^{\mathbb{N}}|$
2. $|2^{\mathbb{N}}| \leq |[0, 1]|$
3. $|[0, 1]| \leq |\mathbb{R}|$
4. By (1–3): $|\mathbb{N}| < |\mathbb{R}|$

Exercise Sheet 1

Exercise 3. Show the following claims.

1. There exist non-regular languages.
2. There exist undecidable languages.
3. There exist non-Turing-recognisable languages.

Solution 1.

(a) A language is regular iff it is recognised by a finite automaton (FA). A FA is a tuple $\langle Q, \Sigma, \delta, q_s, Q_f \rangle$ consisting of a finite set of states Q, a finite set of input symbols Σ, a transition function $\delta : Q \times \Sigma \to Q$, a start state $q_s \in Q$, and a set of final states $Q_f \subseteq Q$.

(b) A FA can be represented with a finite string of 0s and 1s.

(c) There is only countably many different FA and therefore, only countably many regular languages.

(d) Since languages are sets of words, there are uncountably many different languages.

(e) There is some non-regular language.

Solutions 2 and 3. We can make an analogous argument to show these claims.

Exercise Sheet 1

Exercise 4. Let $G = \{V, E\}$ be a SIMPLE undirected graph with $|V| \geq 2$. Show that G contains two or more nodes that have equal degree. That is, show that that there is a pair of nodes that occur in the same number of edges.

Solution.

(a) Every node in G has degree at most $|V| - 1$.

(b) The degree of a node in V can take only $|V| - 1$ different values. Since there are $|V|$ nodes, there must be at least two with equal degree.

Pidgeonhole Principle. If n items are put into m containers, with $n > m$, then at least one container must contain more than one item.

Exercise Sheet 1

Exercise 5. Let $\mathcal{A} = \{s\}$, where

$$s = \begin{cases} 0 & \text{if life will never be found on Mars,} \\ 1 & \text{if life will be found on Mars someday.} \end{cases}$$

Is $\mathcal{A}$ decidable? For the purpose of this problem, assume that the question whether life will be found on Mars has an unambiguous yes or no answer.

Solution.

- **Definition:** A language $\mathcal{L}$ is decidable iff there is some TM that decides $\mathcal{L}$.
- By the assumption above, we have that either $\mathcal{A} = \{0\}$ or $\mathcal{A} = \{1\}$.
- In either case, there exists a TM that decides $\mathcal{L}$.
- Language $\mathcal{A}$ is decidable.

Remark. Every finite language is decidable.

Exercise Sheet 1

Exercise 6. Show that the class of Turing-decidable languages is closed under (1) union, (2) intersection, (3) concatenation, and (4) star.

Solution 1.

- Let $\mathcal{L}$ and $\mathcal{L}'$ be some Turing decidable languages.
- There are some (standard) TM $\mathcal{M}$ and $\mathcal{M}'$ that decide $\mathcal{L}$ and $\mathcal{L}$, respectively.
- Let $\mathcal{M}''$ be a 2-tape TM machine designed as follows:
  - Copy the input to the second tape.
  - Run $\mathcal{M}$ on the first tape and $\mathcal{M}'$ on the second.
  - $\mathcal{M}''$ accepts iff either run accepts.
- $\mathcal{M}''$ decides $\mathcal{L} \cup \mathcal{L}'$.

Solution 2. Small mod: $\mathcal{M}''$ accepts iff both run accepts. Then, $\mathcal{M}''$ decides $\mathcal{L} \cap \mathcal{L}'$.

Exercise Sheet 1

Exercise 6. Show that the class of Turing-decidable languages is closed under (1) union, (2) intersection, (3) concatenation, and (4) star.

Solution 3.

- We show that the concatenation of the decidable languages $\mathcal{L}$ and $\mathcal{L}'$, i.e., the language $\mathcal{L} \circ \mathcal{L}' = \{wv \mid w \in \mathcal{L} \text{ and } v \in \mathcal{L}'\}$, is decidable.
- There are some (standard) TM $\mathcal{M}$ and $\mathcal{M}'$ that decide $\mathcal{L}$ and $\mathcal{L}$, respectively.
- Let $\mathcal{M}''$ be a 2-tape, non-deterministic TM machine designed as follows:
  - On input $a_1, \ldots, a_n$, "non-deterministically" copy some (possibly empty) substring $a_1, \ldots, a_k$ on to the second tape and remove this substring from the input tape.
  - Run $\mathcal{M}$ on the first tape and $\mathcal{M}'$ on the second.
  - $\mathcal{M}''$ accepts iff both runs accept.
- $\mathcal{M}''$ decides $\mathcal{L} \circ \mathcal{L}'$.

Solution 4. Discussion.

Exercise Sheet 1

Exercise 7. Show that the class of Turing-recognisable languages is closed under homomorphism.

Solution. Let $f : \Sigma^* \to \Delta^*$ be a homomorphism and let $\mathcal{L}$ be a Turing-recognisable language. We show that, if $\mathcal{L}$ is recognisable, then so is $f(\mathcal{L})$.

- There is an enumerator $\mathcal{M}$ for the language $\mathcal{L}$.
- Let $\mathcal{M}'$ be the enumerator that performs the following computation:
  (a) Using the "code" from $\mathcal{M}$, produces the first/next word $w \in \mathcal{L}$.
  (b) Applies f to w and the prints the resulting word (i.e., $f(w)$). Note that homomorphisms are always computable.
  (c) Go to (a).

Exercise Sheet 1

Exercise 8. A *Turing machine with two-sided unbounded tape* is a single-tape Turing machine where the tape is unbounded on both sides. Argue that such machines can be simulated by ordinary Turing machines.

Solution. Discussion.

Exercise Sheet 1

Exercise 9. Let $\text{ALL}_{\text{DFA}} = \{\langle A \rangle \mid A \text{ is a DFA that accepts every word}\}$. Show that ALL_{DFA} is decidable.

Solution.

- Let $\mathcal{L}$ be some language. The complement of $\mathcal{L}$, denoted with $\overline{\mathcal{L}}$, is the language containing a word w iff $w \notin \mathcal{L}$
- A language is decidable iff its complement is decidable.
- It suffices to show that $\overline{\text{ALL}_{\text{DFA}}}$ is decidable to solve the exercise.

Exercise Sheet 1

Exercise 9. Let $\overline{\text{ALL}_{\text{DFA}}} = \{\langle A \rangle \mid A \text{ is a DFA that rejects some word}\}$. Show that $\overline{\text{ALL}_{\text{DFA}}}$ is decidable.

Solution.

- Let $A = \langle Q, \Sigma, \delta, q_s, Q_f \rangle$ be a DFA (where Q is a finite set of states Q, Σ is a finite set of input symbols, $\delta : Q \times \Sigma \to Q$ is a deterministic transition function, $q_s \in Q$ is the start state, and $Q_f \subseteq Q$ is the set of final states).
- A state $q \in Q$ is *reachable* iff there is some sequence $\{\langle q_1, a_1 \rangle \mapsto q_2, \ldots, \langle q_n, a_n \rangle \mapsto q_n\} \subseteq \delta$ with $q_1 = q_s$ and $q_n = q$.
- $\langle A \rangle \in \overline{\text{ALL}_{\text{DFA}}}$ iff there is some reachable, non-final state $q \in Q$.
- To show that $\overline{\text{ALL}_{\text{DFA}}}$ is decidable, we define a TM $\mathcal{M}$ that will detect whether a DFA contains a reachable, non-final state.

Exercise Sheet 1

Exercise 10. Let $\mathsf{E}_{\mathsf{TM}} = \{\langle M \rangle \mid M \text{ is a TM such that } \mathcal{L}(M) = \emptyset\}$. Show that $\overline{\mathsf{E}_{\mathsf{TM}}}$ is Turing-recognisable.

Definition. A language $\mathcal{L}$ is Turing-recognisable iff there is a TM $\mathcal{M}$ such that,

- $\mathcal{M}$ accepts on any input $w \in \mathcal{L}$, and
- $\mathcal{M}$ does not accept on any input $w \notin \mathcal{L}$.

Solution. Discussion.

Exercise Sheet 1

Exercise 11. Let C be a language. Prove that C is Turing-recognisable iff a decidable language D exists such that $C = \{x \mid \exists y.\langle x, y\rangle \in D\}$.

Solution. If C is Turing-recognisable, then there is a decidable language D such that $C = \{x \mid \exists y.\langle x, y\rangle \in D\}$

(a) There is a TM $\mathcal{M}$ that recognises C.

(b) Let D be the language containing a word $\langle x, y\rangle$ iff $x \in D$ and y is the encoding of a finite sequence of configurations $C_0 \vdash_{\mathcal{M}} C_1 \vdash_{\mathcal{M}} \ldots \vdash_{\mathcal{M}} C_n$ with C_0 the start configuration for x and $\mathcal{M}$ and C_n an accepting configuration.

(c) The language D satisfies all of the above requirements.

- A word $x \in C$ iff there is some finite sequence of configurations such as the one described in (b).
- The language D is decidable by a TM $\mathcal{M}'$ which, on input $\langle x, y\rangle$, checks that y is a finite sequence of configurations such as the one described in (b).

Exercise Sheet 1

Exercise 11. Let C be a language. Prove that C is Turing-recognisable iff a decidable language D exists such that $C = \{x \mid \exists y.\langle x, y\rangle \in D\}$.

Solution. We show that, if there is a decidable language D such that $C = \{x \mid \exists y.\langle x, y\rangle \in D\}$, then C is Turing-recognisable.

1. For the sake of simplicity, we assume that both D and C are languages defined using only three symbols: "a", "b", and ",".
2. There is a TM $\mathcal{M}$ that decides D.
3. The language C is recognised by a 2-tape TM $\mathcal{M}'$ which, on input x, performs the following computation:
   (a) Adds "," at the end of the input.
   (b) Copy the string in the input tape onto the second tape and then check (using only the second tape) whether this word is in D (use the "code" in $\mathcal{M}$). If this is the case, *accept*. Otherwise, erase the second tape.
   (c) Non-deterministically, append an "a" or a "b" at the end of the word of the input tape. Go to (b).