Léonard Brice, Mathieu Sassolas
leonard.brice@ulb.be, mathieu.sassolas@ulb.be

Session 2:
Countability

Solutions

1 Countable sets

1. *Show that the set of odd integers $\{2n + 1 \mid n \in \mathbb{N}\}$ is countable.*

   ☛ The mapping $f : n \mapsto 2n+1$ is a bijection. The corresponding enumeration is $1, 3, 5, \ldots$

2. *Show that the set $\mathbb{Z}$ is countable.*

   ☛ The mapping:
   $$f : \begin{cases} \mathbb{Z} & \to & \mathbb{N} & \\ z & \mapsto & 2z & \text{if } z \geq 0 \\ z & \mapsto & 2|z| - 1 & \text{if } z < 0 \end{cases}$$
   is a bijection. The corresponding enumeration is $0, -1, 1, -2, 2, -3, \ldots$

3. *Show that the set of finite words Σ^* over a finite alphabet Σ is countable.*

   ☛ Since there are finitely many words over Σ of each size, the words over Σ can be enumerated by size. For example, with $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, a possible enumeration is: ε, `a`, `b`, `aa`, `ab`, `ba`, `bb`, `aaa` ...

2 Uncountable sets

Let Σ be a finite alphabet with at least two symbols. We write Σ^ω the set of infinite words $w_0 w_1 \ldots$ over Σ. We want to prove that Σ^ω is uncountable.

1. *Consider a mapping $f : \mathbb{N} \to \Sigma^\omega$, and write $w^{(n)} = w_0^{(n)} w_1^{(n)} \cdots = f(n)$ for each $n \geq 0$. Consider a word $w = w_0 w_1 \ldots$ where for every k, we have $w_k \neq w_k^{(k)}$. Is there an integer n such that $w = f(n)$?*

   ☛ No. Let us assume such an integer n exists. Then, in particular, we have $w_n^{(n)} = w_n$; but by definition of w, we have $w_n^{(n)} \neq w_n$. Contradiction.

2. *Conclude.*

   ☛ The set Σ^ω is uncountable. Indeed, if it was countable, then there would exist a bijection f from $\mathbb{N}$ to Σ^ω. But, since Σ has at least two elements, we can construct a word w that satisfies the hypotheses of the previous question: then, there is no n such that $f(n) = w$, which is a contradiction.

3. *Use that result to prove that the set $[0, 1]$ is uncountable.*

   ☛ The set $[0, 1]$ is in bijection with the set $\{\mathtt{0}, \mathtt{1}\}^\omega$, with the usual binary encoding (for example, the number $0.110100\ldots$ is mapped to the word `110100`...). By the previous question, that set is not countable.

4. *Let A be an infinite countable set. Use similar techniques to prove that 2^A, the set of subsets of A, is uncountable.*

   ☛ Let us write $A = \{a_0, a_1, \ldots\}$.

   Let us assume there exists a bijection $F : \mathbb{N} \to 2^A$. Then, let us enumerate $F(0), F(1), \ldots$ by the following table, where a *yes* means that the element belongs to the set, and a *no* means that is does not:

	a_0	a_1	a_2	a_3	a_4
$F(0)$	no	yes	no	yes	no
$F(1)$	no	yes	no	no	yes
$F(2)$	yes	yes	no	no	yes
$F(3)$	...	...	...	...	...

Let us now define X as the complement of the set defined by the diagonal:

	a_0	a_1	a_2	a_3	a_4
$F(0)$	~~no~~ **yes**	yes	no	yes	no
$F(1)$	no	~~yes~~ **no**	no	no	yes
$F(2)$	yes	yes	~~no~~ **yes**	no	yes
$F(3)$	...	...	...	...	...

I.e. in our example, $X = \{a_0, a_2, \ldots\}$. Then, with the same argument as in Question 1, this set is not reached by the mapping F, hence the contradiction.

Formally: we define $X = \{n \in \mathbb{N} \mid a_n \notin F(n)\}$. Then, there should exist $n \in \mathbb{N}$ such that $X = F(n)$. But then, do we have $a_n \in X$? If it is the case, then by definition of X, we have $a_n \notin X$: contradiction. But if it is not the case, then by definition of X, we have $a_n \in X$, which is also a contradiction.

3 Counting Turing machines

Let $\Sigma \subseteq \Gamma$ be two finite alphabets. In all this exercise, by language*, we mean a language of finite words over Σ, and by* Turing machine*, we mean a Turing machine with input alphabet Σ and tape alphabet Γ.*

1. *Explain briefly why a Turing machine can be encoded as a finite word over a finite alphabet.*

   ☛ A Turing machine can be described by:

   - the list of states, each of them encoded by a word over $\{0, 1\}$;
   - the list of letters belonging to Σ, each of them encoded by a word over $\{0, 1\}$;
   - the list of letters belonging to Γ, each of them encoded by a word over $\{0, 1\}$;
   - the list of tuples (p, a, q, b, d), where $a, b \in \Gamma$, $p, q \in Q$ and $d \in \{\mathsf{L}, \mathsf{R}\}$ (encoded by 0 and 1), such that $\delta(p, a) = (q, b, d)$;
   - which state is the initial one,
   - which state is the accepting one,
   - and which state is the rejecting one,

   successively listed, with appropriate separating symbols.

2. *Which ones of these sets are countable?*

   (a) The set of Turing machines;
   (b) the set of decidable languages ($\mathbf{R}$);
   (c) the set of recursively enumerable languages ($\mathbf{RE}$);
   (d) the set of languages (2^{Σ^*});
   (e) the set of undecidable languages ($2^{\Sigma^*} \setminus \mathbf{R}$);
   (f) the set of languages that are neither recursively enumerable nor co-recursively enumerable ($2^{\Sigma^*} \setminus \mathbf{RE} \setminus \mathbf{coRE}$).

   ☛

(a) Countable: by the previous question, a Turing machine can be seen as a finite word over, for example, the alphabet $\{\mathtt{0}, \mathtt{1}, \#\}$. But by Question 1.3, we know that the set $\{\mathtt{0}, \mathtt{1}, \#\}^*$ is countable. Therefore, the set of Turing machines over Σ, as a subset of a countable set, is countable.

(b) Countable: each Turing machine decides at most one language.

(c) Countable: same argument.

(d) Uncountable by Question 2.4.

(e) Uncountable, since 2^{Σ^*} is uncountable, while $\mathbf{R}$ is countable.

(f) Uncountable: the set $\mathbf{RE}$ is countable. The set $\mathbf{coRE}$ is the set of complements of elements of $\mathbf{RE}$, and is therefore in bijection with $\mathbf{RE}$: as a consequence, it is also countable. On the other hand, the set 2^{Σ^*} is uncountable.

3. *Assume one finds a computation model that is strictly more expressive than Turing machines (challenging therefore the Church-Turing thesis). Is there a hope that such a computation model "decides" all the languages?*

☛ We have proven that there exist undecidable problems, and even an uncountable number of them, using only the fact that Turing machines can be described by a finite word. Therefore, an alternative computation model that could "decide" all the languages would necessarily consist in machines, or other objects, that cannot be described by finite words, which would have few interest in practice.