ICCS310: Assignment 4

Possawat Sanorkam
possawat2017@hotmail.com
February 24, 2021

1: Eh? They Have The Same Cardinality?

Prove the following statements using rigorous mathematical reasoning:

(1) $|[0, \frac{1}{2})| = |[0, 1)|$

Proof: We want to show that $|[0, \frac{1}{2})| = |[0, 1)|$ by direct proof. By definition, let A and B be sets. Say A and B have the same cardinality (size), denoted by $|A| = |B|$, if there exists a bijection between them.

We want to show that there exist a function f such that f is a bijection. Let $f : A \to B$, $A = [0, \frac{1}{2})$, and $B = [0, 1)$. Then, let $f(x) = 2x$. Let $a \in A$ and $a' \in A$. From observation, $f(a)$ is unique for any arbitrary a. We have that $(\forall a \neq a')[f(a) \neq f(a')]$, so f is injective. Let $b \in B$. From observation, every b can be obtain from some $f(a)$. We have that $(\forall b)(\exists a)[f(a) = b]$, so f is also surjective. According to the definition we stated before, f is a bijective function. Hence, $|A| = |B|$. Therefore, $|[0, \frac{1}{2})| = |[0, 1)|$. $\square$

(2) $|[0, 1)| = |(-1, 1)|$

Proof: We want to show that $|[0, 1)| = |(-1, 1)|$ by direct proof. By definition, let A and B be sets. Say A and B have the same cardinality (size), denoted by $|A| = |B|$, if there exists a bijection between them. In addition, $|A| = |B|$ if and only if $|A| \leq |B|$ and $|B| \leq |A|$.

We want to show that there exist functions f that is injective and g that is also injective. Let $f : A \to B$, $g : B \to A$, $A = [0, 1)$, and $B = (-1, 1)$. Then, let $f(x) = x$. Let $a \in A$ and $a' \in A$. From observation, $f(a)$ is unique for any arbitrary a. We have that $(\forall a \neq a')[f(a) \neq f(a')]$, so f is injective. Then, let $g(x) = \frac{(x+1)}{2}$. Let $b \in B$ and $b' \in B$. From observation, $g(b)$ is unique for any arbitrary b. We have that $(\forall b \neq b')[f(b) \neq f(b')]$, so g is injective. Hence, $|A| \leq |B|$ and $|B| \leq |A|$ which implies that $|A| = |B|$. Therefore, $|[0, 1)| = |(-1, 1)|$. $\square$

(3) $|[0, 1)| = |\mathbb{R}|$

Proof: We want to show that $|[0, 1)| = |\mathbb{R}|$ by direct proof. Besides, we have that $|[0, 1)| = |(-1, 1)|$ which means we can show that $|\mathbb{R}| = |(-1, 1)|$ instead. Say A and B have the same cardinality (size), denoted by $|A| = |B|$, if there exists a bijection between $|A| = |C|$, we have $|B| = |C|$ also.

We want to show that there exist a function f such that f is a bijection. Let $f : A \to \mathbb{R}$, and $A = (-1, 1)$. Then, let $f(x) = \frac{x}{1-x^2}$. This function is continuous on domain A when $x \neq 1$ and $x \neq -1$. Let $a \in A$ and $a' \in A$. From observation, $f(a)$ is unique for any arbitrary a. We have that $(\forall a \neq a')[f(a) \neq f(a')]$, so f is injective. Let $b \in B$. From observation, every b can be obtain from some $f(a)$. The upper bound of $f(x)$ is $\lim_{x \to 1} f(x) \approx \infty$ and the lower bound of $f(x)$ is $\lim_{x \to -1} f(x) \approx -\infty$. So, we can cover all element in $\mathbb{R}$. We have that $(\forall b)(\exists a)[f(a) = b]$, so f is also surjective. According to the definition we stated before, f is a bijective function. So, $|A| = |\mathbb{R}|$ or $|\mathbb{R}| = |(-1, 1)|$. Hence, $|\mathbb{R}| = |(-1, 1)|$ implies that $|\mathbb{R}| = |[0, 1)|$ also. Therefore, $|[0, 1)| = |\mathbb{R}|$. $\square$

2: The Power Set of A

(1) Prove that $|2^A| = |\{0,1\}^A|$.

Proof: We want to show that $|2^A| = |\{0,1\}^A|$ by direct proof. Say K and B have the same cardinality (size), denoted by $|K| = |B|$, if there exists a bijection between them. In addition, $|K| = |B|$ if and only if $|K| \leq |B|$ and $|B| \leq |K|$.

We want to show that there exist a function f such that f is a bijection. Let $f : K \to B$, $g_k : K \to \{1,0\}$, $K = 2^A$, $a \in A, k \in K$ and $B = \{0,1\}^A$. Then, let

$$f(x) = \text{binary array of length } |A| \text{ where each bit represents the presence of } a \text{ in } x$$

Besides, we say that binary array is just a tuple of $g_k(x)$.

$$f(x) = (g_k(x)|k \in A)$$

Also, let

$$g_k(x) = \begin{cases} 1 & \text{if } k \in x \\ 0 & \text{if } k \notin x \end{cases}$$

So, we have the function that map a set a and represents it in a tuple of bits like $(1, 0, ...)$ of length $|A|$. Let $i \in K$ and $i' \in K$. From observation, $f(i)$ is unique for any arbitrary i. Besides, we can map all the permutation of binary string to each set. We have that $(\forall i \neq i')[f(i) \neq f(i')]$, so f is injective. Let $b \in B$. From observation, every b can be obtain from some $f(a)$. The upper bound of $f(x)$ is $f(A) = (1, 1, ..., 1)$ which is a tuple of 1s with length of $|A|$ and the lower bound of $f(x)$ is $f(\emptyset) = (0, 0, ..., 0)$ which is a tuple of 0s with length of $|A|$. So, we can cover all element in B. We have that $(\forall b)(\exists a)[f(a) = b]$, so f is also surjective. According to the definition we stated before, f is a bijective function. So, $|A| = |B|$. Therefore, $|2^A| = |\{0,1\}^A|$. $\square$.

(2) Prove that $|A| < |\{0,1\}^A|$ and conclude that $|A| < |2^A|$.

Proof: Let A be a nonempty set, though it is potentially countably infinite. We want to show that $|\{0,1\}^A|$ is not countable and then show that $|A| < |\{0,1\}^A|$.

Assume for the sake of contradiction that $|\{0,1\}^A|$ is countable, so $|A| \geq |\{0,1\}^A|$. This means, there exists a surjective function $f : A \to \{0,1\}^A$. Define the following string $d \in \{0,1\}^A$ so that for $i = 0, 1, 2, ...,$

$$d[i] = 1 - f(i)[i]$$

that is, $d[i]$ is taking the i-th bit of the string given by $f(i)$ and negating it. We'll show that d differs from $f(k)$ for every $k \in N$. In particular, they disagree on the k-th position, i.e., $d[k] \neq f(k)[k]$. Hence, f cannot possibly be a surjective function from $A \to \{0,1\}^A$, so it is a contradiction to our assumption.

Next, we want to show that there exist $g : A \to \{0,1\}^A$ such that g is injective. We have that $A \subset 2^A$. Since $A \subset 2^A$, we have that $g : A \to \{0,1\}^A$ is

$$g(x) = \text{binary array of length } |A| \text{ where each bit represents the presence of } a \text{ in } x$$

where $a \in A$.

Besides, we say that binary array is just a tuple of $s_k(x)$, where $s_k : A \to \{1, 0\}$, and $k \in A$.

$$f(x) = (s_k(x)|k \in A)$$

Also, let

$$g_k(x) = \begin{cases} 1 & \text{if } k \neq x \\ 0 & \text{if } k = x \end{cases}$$

So, we have the function that map an element a and represents it in a tuple of bits like $(1, 0, ...)$ of length $|A|$. Let $i \in A$ and $i' \in A$. From observation, $f(i)$ is unique for any arbitrary i. Since we only take on element of A into the function, we will always get a tuple with only single bit of 1, meaning that only one element existed. We have that $(\forall i \neq i')[f(i) \neq f(i')]$, so f is injective. From earlier, we showed that $|A| \not\geq |\{0, 1\}^A|$. Hence, $|A| < |\{0, 1\}^A|$.

Since we showed that $|2^A| = |\{0, 1\}^A|$, it implies that $|A| < |2^A|$ also.

Therefore, $|A| < |2^A|$. $\square$

3: Hamming Code

Consider applying the Hamming coding scheme to send 8 bits of data. This will require 4 parity bits, so an encoded code word in this scheme is 12 bits long.

(1) If the data bits are $d_1, d_2, d_3...d_8$, what is β_2 in terms of d_i's?

Solution: $\beta_2 = p_2 \oplus d_1 \oplus d_3 \oplus d_4 \oplus d_6 \oplus d_7$

In case that β_2 is a parity bit (I am not sure if you mean p_2), $\beta_2 = d_1 \oplus d_3 \oplus d_4 \oplus d_6 \oplus d_7$

(2) Encode the following 8-bit data: 01101010.

Solution: Hamming Code = $(p_1p_2d_1p_4d_2d_3d_4p_8d_5d_6d_7d_8)$

Encode 01101010 by adding the parity bits as followed

$$p_1 = d_1 \oplus d_2 \oplus d_4 \oplus d_5 \oplus d_7 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 1 \quad (1)$$
$$p_2 = d_1 \oplus d_3 \oplus d_4 \oplus d_6 \oplus d_7 = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 0 \quad (2)$$
$$p_4 = d_2 \oplus d_3 \oplus d_4 \oplus d_8 = 1 \oplus 1 \oplus 0 \oplus 0 = 0 \quad (3)$$
$$p_8 = d_5 \oplus d_6 \oplus d_7 \oplus d_8 = 1 \oplus 0 \oplus 1 \oplus 0 = 0 \quad (4)$$

Therefore, encoded bits are 100011001010

(3) Assuming that at most a single single bit flip, decide the following codewords (indicate also whether there was any error):

Solution: Hamming Code = $(p_1p_2d_1p_4d_2d_3d_4p_8d_5d_6d_7d_8)$

(i) 010011111000

So, $p_1 = 0, p_2 = 1, p_4 = 0$,and $p_8 = 1$.

$$\beta_1 = p_1 \oplus d_1 \oplus d_2 \oplus d_4 \oplus d_5 \oplus d_7 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 1 \quad (5)$$
$$\beta_2 = p_2 \oplus d_1 \oplus d_3 \oplus d_4 \oplus d_6 \oplus d_7 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1 \quad (6)$$
$$\beta_4 = p_4 \oplus d_2 \oplus d_3 \oplus d_4 \oplus d_8 = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 1 \quad (7)$$
$$\beta_8 = p_8 \oplus d_5 \oplus d_6 \oplus d_7 \oplus d_8 = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0 \quad (8)$$

Error Position is $0111_2 = 7$. Corrected Data is 010011011000.

(ii) 011101010010

So, $\beta_1 = 0, \beta_2 = 1, \beta_4 = 1$,and $\beta_8 = 1$.

$$\beta_1 = p_1 \oplus d_1 \oplus d_2 \oplus d_4 \oplus d_5 \oplus d_7 = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 0 \quad (9)$$

$$\beta_2 = p_2 \oplus d_1 \oplus d_3 \oplus d_4 \oplus d_6 \oplus d_7 = 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 0 \quad (10)$$

$$\beta_4 = p_4 \oplus d_2 \oplus d_3 \oplus d_4 \oplus d_8 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0 \quad (11)$$

$$\beta_8 = p_8 \oplus d_5 \oplus d_6 \oplus d_7 \oplus d_8 = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0 \quad (12)$$

Data is already correct.

4: Same Number of 0s and 1s

Consider the language $L = \{w \in \{0,1\}^* |$ w contains an equal number of 0s and 1s $\}$. Show that L is (Turing) decidable by providing a TM that decides it (a medium-level detail is preferred).

Proof: We will show that L is Turing decidable. The proof is by construction: we will give a $TM\ M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ that decides L:

On input string w:

1. Sweep left to right across the tape, check whether 0 and 1 are both existed, reject if it is not.

2. Return the head to the left-hand end of the tape then scan to the right. If the tape contains only $\sqcup$, it accepts.

3. If in stage 2 a 0 is found, then replace the left most 0 with $\sqcup$. Keep scanning to the right, if there is no 1s, it rejects.

4. If in stage 3 a 1 is found, then replace the left most 1 with $\sqcup$. Then, go to stage 2.

5. If in stage 2 a 1 is found, then replace the left most 1 with $\sqcup$. Keep scanning to the right, if there is no 0s, it rejects.

6. If in stage 5 a 0 is found, then replace the left most 0 with $\sqcup$. Then, go to stage 2.

5: Infinite DFA

Show that the following language is (Turing) decideable:

$$\text{IDFA} = \{\langle M \rangle | \text{ M is a DFA and L(M) is an infinite language } \}.$$

Proof: We want to show that IDFA is decidable. The proof is by construction: So, we will construct a TM T that decides IDFA. For all DFAs M, we want $T(\langle M \rangle)$ to accepts if $L(M)$ is infinite language, else it will reject.

By directly check whether it is infinite language will lead to an eternity loop that iterate to test whether $L(M)$ is an language using every $w \in \Sigma^*$. In fact, it will not halt.

So, we want to claim that if A is a DFA with p states and alphabet Σ. Let

$$B_A = \{w \in \Sigma^* \ : \ w \in L(A) \text{ and } p \leq |w| \leq 2p\}$$

Then $L(A)$ is infinite if and only if $B_A \neq \emptyset$.

We want to show that if $B_A \neq \emptyset$, then $L(A)$ is infinite. Recall the Pumping Lemma, we have p states in DFA to be the same p in the Pumping Lemma and it always apply to all regular language $L(A)$.

Since B_A is non-empty, there is some string $s \in B_A$. By definition of B_A we have $s \in L(A)$ and $|s| \geq p$, so there exist x, y, z such that $s = xyz$ and the three conditions of the Pummping Lemma hold. The first condition says that $(\forall i \geq 0)[xy^i z \in L(A)]$. Hence, $L(A)$ is infinite.

We also want to show that if $L(A)$ is infinite, then $B_A \neq \emptyset$. Let s be a shortest string in $L(A)$ such that $|s| \geq p$. (Such an s exists since $L(A)$ is infinite.) If $|s| \leq 2p$, then $s \in B_A$.

Then, we want to apply the Pumping Lemma to the regular language $L(A)$. Assume for the sake of contradiction that $|s| > 2p$. Let $s = xyz$ such that the three conditions of the Pumping Lemma hold. Setting $i = 0$ in the first condition implies that $xz \in L(A)$. The third condition implies $|y| \leq p$. So, $|xz| = |s| - |y| > 2p - p = p$, meaning xz is a string in $L(A)$ with length strictly greater than p, but is shorter than s. Hence, it contradicts to the assumption that s was the shortest string. So, we have shown that $B_A \neq \emptyset$.

Therefore, we have claimed the above statement.

Referring to our claim, this means that to test whether or not $L(A)$ is infinite, we need only test whether or not B_A is empty. The latter can be done using the machine M_{dfa} since B_A is a finite set. We can provide the construction.

On input $\langle A \rangle$, where A is a DFA:

1. Let p be the number of states in DFA $\langle A \rangle$ and let Σ be the alphabet of DFA A. Store these variables somewhere in the tape.

2. Repeat the following stages for each string $w \in \Sigma^*$ such that $p \leq |w| \leq 2p$:

    3. Run $M_{\text{dfa}}(\langle A, w \rangle)$ and if M_{dfa}, then it accepts.

3. It rejects when nothing can be accepted.

6: Lucky 9

---

(1) Let $L_1 \subseteq \Sigma^*$ be defined as

$$L_1 = \begin{cases} \varnothing & \text{if } 2^{74207281} - 1 \text{ is prime} \\ \{99\} & \text{if } 2^{74207281} - 1 \text{ is not prime} \end{cases}$$

Prove that L_1 is (Turing) decidable.

Proof:

We will show that L is Turing decidable. The proof is by construction: we will give a TM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ that decides L:

On input string w:

1. Perform primality test on $2^{74207281} - 1$ to check whether it is a prime number or not.

2. If $2^{74207281} - 1$ is prime, it rejects.

3. If $2^{74207281} - 1$ is not prime, then check if w is 99 or not. If it is 99, then it accepts else it rejects. □

PS. I have looked up for $2^{74207281} - 1$ whether it is a prime or not, and it is a prime. Here is the url, https://www.mersenne.org/primes/?press=M74207281.

(2) Let $L_2 \subseteq \Sigma^*$ be defined as

$w \in L_2 \iff$ w appears somewhere (not necessarily consecutively) in the decimal expansion of π

Prove that L_2 is (Turing) decidable.

Proof:

We will show that L is Turing decidable. The proof is by construction: we will give a TM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ that decides L:

By the characteristic of π, since it is irrational, it has an infinite number of digits in its decimal representation, and does not settle into an infinitely repeating pattern of digits. This implies that for all $w \in L_2$, M can halt and always accept any $s \in \Sigma^*$ since $\Sigma = 9$. Besides, Σ^* is a set of 9s and $|s|$ is countably infinite.

On input string w:

1. Store input w in the left-most end of the tape and mark a space using $\sqcup$ to seperate between w and π value.

2. Repeat the following stage until all stored characters are $\sqcup$. When all stored characters are $\sqcup$, it accepts.

    3. Keep listing the value of π to the right until 9 is found.

    4. Move the head back to the left-most end 9 in stored w and mark it $\sqcup$ $\square$

7: β-reduction

(1) $(\lambda z.z)(\lambda z.zz)(\lambda z.zy)$

Solution:

$$\begin{aligned}
(\lambda z.z)(\lambda z.zz)(\lambda z.zy) &\to_1 (\lambda z.zz)(\lambda z.zy) && (13)\\
&\to_1 (\lambda z.zy)(\lambda z.zy) && (14)\\
&\to_1 (\lambda z.zy)y && (15)\\
&\to_1 yy && (16)
\end{aligned}$$

(2) $(((\lambda x.\lambda y.(xy))(\lambda y.y))w)$

Solution:

$$\begin{aligned}
(((\lambda x.\lambda y.(xy))(\lambda y.y))w) &\to_1 (((\lambda x.\lambda y.(xy))(\lambda y'.y'))w) && (17)\\
&\to_1 (\lambda y.((\lambda y'.y')y)w) && (18)\\
&\to_1 (\lambda y.(y)w) && (19)\\
&\to_1 w && (20)
\end{aligned}$$

8: Fibonacci

Using the functions we have developed (e.g., `pred, if_then_else, mult, add`, etc.), write down an explicit λ-term fib such that $\overline{\texttt{fib}}\,\overline{n} =_\beta \overline{f(n)}$.

Solution: Let $\texttt{fib}(n) = \texttt{plain\_fib}(\texttt{fib}, n)$ for all n

Define $T = \lambda zw.z$

We say `iszero` is a function that return T if the input is zero, else F.

Define $\texttt{iszero} := \lambda nxy.n(\lambda z.y)x$

Define $\texttt{or} := \lambda xy.x(T)(y)$

Define $\texttt{plain\_fib} := \lambda fn.\texttt{if\_then\_else}(\texttt{or}\ (\texttt{iszero}\ n)(\texttt{iszero}\ (\texttt{pred}\ n)))(\overline{1})$
$(\texttt{add}(f\ \texttt{pred}\ n)(f\ \texttt{pred}\ (\texttt{pred}\ n)))$

$$\begin{aligned} \overline{\texttt{fib}}\ \overline{n} &\to_* (\texttt{plain\_fib}\ \overline{\texttt{fib}})\overline{n} && (21)\\ &\to_* \texttt{if\_then\_else}(\texttt{or}\ (\texttt{iszero}\ \overline{n})(\texttt{iszero}\ (\texttt{pred}\ \overline{n}))) && (22)\\ & \quad (\overline{1})(\texttt{add}(\overline{\texttt{fib}}\ \texttt{pred}\ \overline{n})(\overline{\texttt{fib}}\ \texttt{pred}\ (\texttt{pred}\ \overline{n}))) \end{aligned}$$

Therefore,

$$\overline{\texttt{fib}}\ \overline{n} = \texttt{if\_then\_else}(\texttt{or}\ (\texttt{iszero}\ \overline{n})(\texttt{iszero}\ (\texttt{pred}\ \overline{n})))(\overline{1})(\texttt{add}(\overline{\texttt{fib}}\ \texttt{pred}\ \overline{n})(\overline{\texttt{fib}}\ \texttt{pred}\ (\texttt{pred}\ \overline{n})))$$

9: Power Of 2

Implement a λ-term for the $pow(n) = 2^n$

Solution: $\overline{pow}\ \overline{n} = (pow\ \overline{n})\overline{2} = ((\lambda pq.pq)\overline{n})\overline{2}$

Substitute all church numerals, we get $\overline{pow}\ \overline{n} =$

$$((\lambda pq.pq)\lambda fx.f^n x))\lambda fx.f(fx)$$