COM S 5310 Theory of Computing, Homework 3

Mobina Amrollahi

Due: October 20^{th}, 11:59pm on Gradescope.

Problem 1

For $A \subseteq \Sigma^*$ and $n \in \mathbb{N}$, we define the n^{th} *slice* of A to be the language

$$A_n = \{y \in \Sigma^* \mid <n, y> \in A\},$$

where $<n, y> = <s_n, y>$ and $s_0, s_1, \ldots$ is the standard enumeration of Σ^*.

Let $\mathcal{C}$ and $\mathcal{D}$ be classes of languages.

1. $\mathcal{C}$ *parametrizes* $\mathcal{D}$ (or $\mathcal{C}$ is *universal* for $\mathcal{D}$) if there exists $A \in \mathcal{C}$ such that $\mathcal{D} = \{A_n | n \in \mathbb{N}\}$.
2. $\mathcal{D}$ is $\mathcal{C}$-*countable* if there exists $A \in \mathcal{C}$ such that $\mathcal{D} \subseteq \{A_n | n \in \mathbb{N}\}$.

(a) Prove: A class $\mathcal{D}$ of languages is countable if and only if $\mathcal{D}$ is $\mathcal{P}(\Sigma^*)$-countable.

(b) Prove that DEC is not DEC-countable.

Answer

(a) ($\Rightarrow$) class $\mathcal{D}$ of languages is countable $\implies$ $\mathcal{D}$ is $\mathcal{P}(\Sigma^*)$-countable:
Suppose $\mathcal{D}$ is countable. Then

$$\mathcal{D} = \{d_1, d_2, d_3, \ldots\}.$$

Define

$$A = \{\langle n, x\rangle \mid n \in \mathbb{N},\ x \in d_n\} \subseteq \Sigma^*.$$

Then $A \in \mathcal{P}(\Sigma^*)$ and, by construction, the n^{th} slice satisfies $A_n = d_n$. Hence

$$\mathcal{D} = \{d_1, d_2, d_3, \ldots\} = \{A_1, A_2, A_3, \ldots\},$$

so $\mathcal{D}$ is $\mathcal{P}(\Sigma^*)$-countable.

($\Leftarrow$) $\mathcal{D}$ is $\mathcal{P}(\Sigma^*)$-countable $\implies$ class $\mathcal{D}$ of languages is countable:
Suppose $\mathcal{D}$ is $\mathcal{P}(\Sigma^*)$-countable. Then there exists $A \in \mathcal{P}(\Sigma^*)$ such that

$$\mathcal{D} \subseteq \{A_n \mid n \in \mathbb{N}\}.$$

If $\mathcal{D} = \emptyset$, it is trivially countable. Otherwise, let $d \in \mathcal{D}$. Define the function $f : \mathbb{N} \to \mathcal{D}$ by

$$f(n) = \begin{cases} A_n, & \text{if } A_n \in \mathcal{D}, \\ d, & \text{if } A_n \notin \mathcal{D}. \end{cases}$$

This function is onto, since for any $D \in \mathcal{D}$, there exists n such that $A_n = D$. Hence $\mathcal{D}$ is countable.

(b) Suppose, for contradiction, that DEC is DEC-countable. Then there exists $A \in \text{DEC}$ such that

$$\text{DEC} \subseteq \{A_n \mid n \in \mathbb{N}\}.$$

Let M_A be the Turing machine that decides A. For each n, construct a Turing machine M_n deciding A_n as follows:

M_n**:** On input x:

(a) Run M_A on input $<n, x>$.

  i. If M_A accepts, then **accept**.

  ii. If M_A rejects, then **reject**.

Each A_n is therefore decidable.

Now define

$$B = \{\, s_n \in \{0,1\}^* \mid s_n \notin A_n \,\}.$$

Note that $B \neq A_n$ for all n, since

$$s_n \in B \Leftrightarrow s_n \notin A_n.$$

Construct a Turing machine M_B deciding B as follows:

M_B**:** On input $x \in \{0,1\}^*$:

(a) Determine the index n such that $x = s_n$ in the standard enumeration.

(b) Run M_n on s_n.

  i. If M_n accepts, then **reject**.

  ii. If M_n rejects, then **accept**.

Hence B is decidable, i.e., $B \in \text{DEC}$. However, since $B \neq A_n$ for all n, it follows that $B \notin \{A_n \mid n \in \mathbb{N}\}$, contradicting the assumption that $\text{DEC} \subseteq \{A_n \mid n \in \mathbb{N}\}$. Therefore, DEC is not DEC-countable.

Problem 2

(a) Assume that $\mathcal{C}$ and $\mathcal{D}$ are sets of languages and $g : \mathcal{C} \xrightarrow{\text{onto}} \mathcal{D}$. Prove: if $\mathcal{C}$ is countable, then $\mathcal{D}$ is countable.

(b) Prove: if $\mathcal{C}$ is a countable set of languages, then $\exists\mathcal{C}$ and $\forall\mathcal{C}$ are countable.

Answer

(a) Suppose $\mathcal{C}$ is countable. Then there exists a function

$$f : \mathbb{N} \xrightarrow{\text{onto}} \mathcal{C}.$$

By assumption, there also exists a function

$$g : \mathcal{C} \xrightarrow{\text{onto}} \mathcal{D}.$$

Define the function h as following:

$$h : \mathbb{N} \to \mathcal{D}, \quad h(n) = g(f(n)).$$

The function h is well-defined since $f(n) \in \mathcal{C}$ for all $n \in \mathbb{N}$, and dom $g = \mathcal{C}$.
We now show that h is onto. Let $d \in \mathcal{D}$ be arbitrary. Since g is onto, there exists $c \in \mathcal{C}$ such that $g(c) = d$. Because f is onto, there exists $n_0 \in \mathbb{N}$ such that $f(n_0) = c$. Hence,

$$h(n_0) = g(f(n_0)) = g(c) = d.$$

Therefore:

$$h : \mathbb{N} \xrightarrow{\text{onto}} \mathcal{D}.$$

Thus, $\mathcal{D}$ is countable.

(b) Since $\mathcal{C}$ is countable, there exists a function

$$f : \mathbb{N} \xrightarrow{\text{onto}} \mathcal{C}.$$

By definition,

$$\exists\mathcal{C} = \{\exists B \mid B \in \mathcal{C}\} \quad \text{and} \quad \forall\mathcal{C} = \{\forall B \mid B \in \mathcal{C}\}.$$

Each language $B \in \mathcal{C}$ corresponds uniquely to the languages $\exists B$ and $\forall B$. Now define the functions

$$f_\exists : \mathbb{N} \xrightarrow{\text{onto}} \exists\mathcal{C}, \qquad f_\exists(n) = \exists f(n),$$

and

$$f_\forall : \mathbb{N} \xrightarrow{\text{onto}} \forall\mathcal{C}, \qquad f_\forall(n) = \forall f(n).$$

These functions are well defined since for each $n \in \mathbb{N}$, $f(n) \in \mathcal{C}$, and both $\exists(\cdot)$ and $\forall(\cdot)$ map languages to languages.

We show that $f_\exists$ is onto (similar argument can be used for $f_\forall$). Let $L \in \exists\mathcal{C}$. Then $L = \exists B$ for some $B \in \mathcal{C}$. Since f is onto, there exists $n_0 \in \mathbb{N}$ such that $f(n_0) = B$. Hence,

$$f_\exists(n_0) = \exists f(n_0) = \exists B = L.$$

Therefore, $f_\exists$ is onto, and by the same reasoning, $f_\forall$ is onto. Thus, both $\exists\mathcal{C}$ and $\forall\mathcal{C}$ are countable.

Problem 3

Prove that the class of countable classes of languages (defined as CTBL in class) is a σ-ideal on $\mathcal{P}(\Sigma^*)$.

Answer

Consider

$$\mathrm{CTBL} = \{\, E \subseteq \mathcal{P}(\Sigma^*) \mid E \text{ is countable}\,\}.$$

To show that CTBL is a σ-ideal on $\mathcal{P}(\Sigma^*)$, we must prove: (i) if $E \subseteq F$ and $F \in \mathrm{CTBL}$, then $E \in \mathrm{CTBL}$; and (ii) CTBL is closed under countable unions.

(i)

(ii)

(i) Suppose $E \subseteq F$ and $F \in \mathrm{CTBL}$. Then F is countable, so there exists a function

$$f : \mathbb{N} \xrightarrow{\text{onto}} F.$$

If $E = \emptyset$, then it is vacuously countable and hence $E \in \mathrm{CTBL}$. Otherwise, fix $e \in E$ and define $f_e : \mathbb{N} \to E$ by

$$f_e(n) \;=\; \begin{cases} f(n), & \text{if } f(n) \in E, \\ e, & \text{if } f(n) \notin E. \end{cases}$$

This function is well-defined. Moreover, it is onto: for any $y \in E$, since $E \subseteq F$ and f is onto F, there exists n_0 with $f(n_0) = y$, hence $f_e(n_0) = y$. Therefore E is countable, so $E \in \mathrm{CTBL}$.

(ii) Let $A_1, A_2, A_3, \ldots \in \mathrm{CTBL}$. Each A_n is a countable subset of $\mathcal{P}(\Sigma^*)$. By the standard result that a countable union of countable sets is countable, the union $\bigcup_{n\geq 1} A_n$ is countable. Hence $\bigcup_{n\geq 1} A_n \in \mathrm{CTBL}$.

Therefore, CTBL is a σ-ideal on $\mathcal{P}(\Sigma^*)$.

Problem 4

Prove that there is a function $g : \mathbb{N} \to \mathbb{N}$ with the following properties.

(i) g is *nondecreasing*, i.e., $g(n) \leq g(n+1)$ holds for all $n \in \mathbb{N}$.

(ii) g is *unbounded*, i.e., for every $m \in \mathbb{N}$ there exists $n \in \mathbb{N}$ such that $g(n) > m$.

(iii) For every computable, nondecreasing, unbounded function $f : \mathbb{N} \to \mathbb{N}$, $f(n) > g(n)$ holds for all but finitely many $n \in \mathbb{N}$.

Answer

Consider

$$G(n) = 1 + \max\{\zeta_k(\ell) \mid 0 \le k, \ell \le n,\ \zeta_k(\ell)\downarrow\} \quad (\text{with } \max\emptyset = 0),$$

as defined in the class, and define

$$g(n) = \min\{x \in \mathbb{N} \mid G(x) \ge n\}.$$

We will show the proposed g satisfies the three conditions $(i), (ii)$, and (iii):

(i) If $n \le m$, then $\{x \mid G(x) \ge m\} \subseteq \{x \mid G(x) \ge n\}$, hence $\min\{x \mid G(x) \ge n\} \le \min\{x \mid G(x) \ge m\}$. Thus $g(n) \le g(m)$.

(ii) Assume for contradiction that g is bounded by some natural number M, that is, $g(n) \le M$ for all $n \in \mathbb{N}$. By the definition of g, this means that for every n, there exists some $x \le M$ such that $G(x) \ge n$.

Since G is nondecreasing and takes only natural values, the condition

$$\min\{x \in \mathbb{N} \mid G(x) \ge n\} \le M \quad \forall n \in \mathbb{N}$$

implies that no matter how large n becomes, one of the finitely many values $G(1), G(2), \ldots, G(M)$ must still be at least n in order for the inequality to hold.

However, G is a well-defined function that assigns each of the inputs $1, 2, \ldots, M$ a finite value. Let

$$N = \max\{G(1), G(2), \ldots, G(M)\} + 1.$$

Then for this N, we have $G(x) < N$ for all $x \le M$. Hence, there is no $x \le M$ such that $G(x) \ge N$, which means

$$g(N) > M.$$

This contradicts the assumption that $g(n) \le M$ for all n. Therefore, g is unbounded.

(iii) I don't know how to prove this part.

Problem 5

Prove that a partial function $f : \subseteq \Sigma^* \to \Sigma^*$ is computable if and only if its *graph*

$$G_f = \{<x, f(x)> \mid x \in \text{dom } f\}$$

is c.e.

Answer

($\Rightarrow$) The partial function $f : \subseteq \Sigma^* \to \Sigma^*$ is computable $\implies$ its corresponding *graph* is c.e. Suppose $f :\subseteq \Sigma^* \to \Sigma^*$ is computable by a Turing machine M i.e.,

$$\text{dom } f = \{x \in \{0,1\}^* \mid M(x)\downarrow\}$$

Define a machine M_G that recognizes G_f as follows:

> M'**:** On input $\langle x, y\rangle \in \{0,1\}^*$:
>
> 1. Simulate M on input x
>   (a) If M halts with output y, then **accept**
>   (b) if M halts with output $z \ne y$, then **reject**;

Hence $L(M_G) = G_f$, so G_f is c.e. Please note, that this doesn't make G_f necessarily decidable, as M might never halt on an input string.

($\Leftarrow$) G_f is c.e $\implies$ the partial function $f : \subseteq \Sigma^* \to \Sigma^*$ is computable. Since G_f is c.e, then there exists a Turing machine M_G such that $L(M_G) = G_f$. Let E be an enumerator for G_f (Based on HW2, problem 5.3 such an enumerator exists). Construct the Turing machine M_f as follows:

M_f**:** On input $x \in \Sigma^*$:

1. Simulate the enumerator E for strings of G_f
   (a) If the output of E is in the form of $\langle u, v\rangle$:
      – If $u = x$, then:
         i. Erase the work tape and write the v on the output tape and **Halt**.

If $x \in \operatorname{dom} f$, then $\langle x, f(x)\rangle \in G_f$, so E eventually outputs $\langle x, f(x)\rangle$ and M_f halts with output $f(x)$. If $x \notin \operatorname{dom} f$, then no pair with first component x ever appears and M_f does not halt. Thus M_f computes the partial function f,

$$\operatorname{dom} f = \{x \in \{0,1\}^* \mid M_f(x) \downarrow\}$$

so f is computable.

Problem 6

Let $A \subseteq \Sigma^*$ be c.e., and let B be an infinite decidable subset of A. Prove: If A is undecidable, then $A \setminus B$ is undecidable.

Answer

We will prove by contradiction. Assume that $A \setminus B$ is decidable. Hence, suppose B is decided by a Turing machine M, and $A \setminus B$ is decided by a Turing machine M'. Now, construct the following Turing machine M_A:

M_A**:** On input $x \in \{0,1\}^*$:

1. Simulate M on input x.
   (a) If M **accepts**, then **accept**.
   (b) If M **rejects**, simulate M' on input x.
      i. If M' **accepts**, then **accept**.
      ii. If M' **rejects**, then **reject**.

Now, consider an arbitrary string $a \in \{0,1\}^*$. If this string is in A, then since $(A \setminus B) \cap B = \emptyset$, it must be in either B or $A \setminus B$. The proposed Turing machine M_A will halt and accept a if it is in B. If M rejects, it will run M' on a. Since $A \setminus B$ is decidable, M' halts on every input. If M' accepts a, then $a \in A \setminus B$, so M_A accepts. If M' rejects, then $a \notin A$, and M_A rejects.
Therefore, M_A halts on all inputs and decides A. This means A is decidable, which contradicts our assumption that A is undecidable. Hence, $A \setminus B$ must be undecidable.

Problem 7

Let $A = L(U)$ be the universal c.e. language defined in class lectures, and let $B \subseteq \Sigma^*$. Prove: If $A \leq_m B$ and $\Sigma^* \setminus A \leq_m B$, then B is neither c.e. nor co-c.e.

Answer

Knowing that the classes of c.e. (CE) and co-c.e. (coCE) languages are closed under $\leq_m$-*reductions*, and that A is the universal c.e. language while its complement $\Sigma^* \setminus A$ is co-c.e. and not decidable (Corollary 37 from class):
Suppose B is c.e., meaning $B \in$ CE. Since $\Sigma^* \setminus A \leq_m B$ and the class CE is closed under $\leq_m$-reductions, this implies that $\Sigma^* \setminus A$ would also be c.e. However, we already know that $\Sigma^* \setminus A$ is co-c.e. but not decidable, Knowing that a language that is both c.e. and co-c.e. must be decidable, this contradicts the fact that $\Sigma^* \setminus A$ is undecidable. Therefore, B cannot be c.e.
Similarly, suppose B is co-c.e., meaning $B \in$ coCE. Since $A \leq_m B$ and coCE is closed under $\leq_m$-reductions, this implies that A would be co-c.e. However, we know that A is c.e. but not decidable. Knowing that a language that is both c.e. and co-c.e. must be decidable, which again leads to a contradiction. Therefore, B cannot be co-c.e.
Hence, B is neither c.e. nor co-c.e.

Problem 8

(a) Prove that $\leq_m$ is a reflexive, transitive relation on $\mathcal{P}(\Sigma^*)$.

(b) Prove that $\equiv_m$ is an equivalence relation on $\mathcal{P}(\Sigma^*)$.

Answer

1. (a) Reflexivity. To show $\leq_m$ is reflexive, we must show $\forall A \in \mathcal{P}(\Sigma^*)$, $A \leq_m A$. By definition, $A \leq_m A$ if there exists a total computable $f : \Sigma^* \to \Sigma^*$ such that

$$x \in A \iff f(x) \in A \quad \text{for all } x \in \Sigma^*.$$

Take $f(x) = x$ which is computable. Then $x \in A \iff f(x) = x \in A$ holds. Hence $\leq_m$ is reflexive.

(b) Transitivity. To show $\leq_m$ is transitive, suppose $A, B, C \in \mathcal{P}(\Sigma^*)$ with $A \leq_m B$ and $B \leq_m C$. So there exist computable $f_{AB}, f_{BC} : \Sigma^* \to \Sigma^*$ with

$$x \in A \iff f_{AB}(x) \in B \quad \text{and} \quad y \in B \iff f_{BC}(y) \in C.$$

Define $f_{AC} : \Sigma^* \to \Sigma^*$ by

$$f_{AC}(x) \;=\; f_{BC}\big(f_{AB}(x)\big).$$

Then f_{AC} is computable, and

$$x \in A \iff f_{AB}(x) \in B \iff f_{BC}\big(f_{AB}(x)\big) \in C$$

so $x \in A \iff f_{AC}(x) \in C$. Hence $A \leq_m C$, and $\leq_m$ is transitive.

2. (a) Reflexivity. For every A, we have $A \leq_m A$ by part (a), so $A \equiv_m A$.

(b) Symmetry. If $A \equiv_m B$, then by definition $A \leq_m B$ and $B \leq_m A$. This is equivalent to $B \leq_m A$ and $A \leq_m B$, hence $B \equiv_m A$.

(c) Transitivity. Suppose $A \equiv_m B$ and $B \equiv_m C$. Then

$$A \leq_m B \text{ and } B \leq_m A, \qquad B \leq_m C \text{ and } C \leq_m B.$$

Using transitivity of $\leq_m$ from (a), $A \leq_m B \leq_m C$ gives $A \leq_m C$, and $C \leq_m B \leq_m A$ gives $C \leq_m A$. Therefore $A \equiv_m C$.