Section 09: Solutions

1. Irregular Languages

(a) Let $\Sigma = \{0, 1\}$. Prove that $\{0^n 1^n 0^n \ : n \geq 0\}$ is irregular.

Solution:

Let $L = \{0^n 1^n 0^n : n \geq 0\}$. Let D be an arbitrary DFA, and suppose for contradiction that D accepts L. Consider $S = \{0^n 1^n : n \geq 0\}$. Since S contains infinitely many strings and D has a finite number of states, two strings in S must end up in the same state. Let these strings be $0^i 1^i$ and $0^j 1^j$ for some $i, j \geq 0$ such that $i \neq j$. Append the string 0^i to both of these strings. The two resulting strings are:

- $a = 0^i 1^i 0^i$, which is in L.
- $b = 0^j 1^j 0^i$, which is not in L since $i \neq j$.

Since a and b end up in the same state, but $a \in L$ and $b \notin L$, that state must be both an accepting and a rejecting state, which is a contradiction. Since D was arbitrary, there is no DFA that recognizes L, so L is not regular.

(b) Let $\Sigma = \{0, 1, 2\}$. Prove that $\{0^n (12)^m : n \geq m \geq 0\}$ is not regular.

Solution:

Let $L = \{0^n (12)^m : n \geq m \geq 0\}$. Assume, for the sake of contradiction, that there exists a deterministic finite automaton (DFA) D that accepts L.

Consider the set $S = \{0^n : n \geq 0\}$. Since S contains infinitely many strings and D has a finite number of states, by the Pigeonhole Principle, there must be two distinct strings 0^i and 0^j (with $i > j$) in S that are mapped to the same state by D.

Now, append the string $(12)^i$ to both of these strings. We obtain two new strings:

$$a = 0^i (12)^i$$

$$b = 0^j (12)^i$$

Note that $a \in L$ because $i \geq i$.

However, $b \notin L$ because $i > j$ implies $n = j$ and $m = i$ does not satisfy the condition $n \geq m$.

Since a and b end up in the same state in D, but $a \in L$ and $b \notin L$, it follows that the state reached by a and b must be both an accepting state and a rejecting state, which is a contradiction.

Therefore, no such DFA D exists that accepts L. Consequently, L is not a regular language.

2. Countability

(a) Prove that the set $\{5x : x \in \mathbb{N}\}$ is countable.

Solution:

Define the function $f : \mathbb{N} \to \{5x : x \in \mathbb{N}\}$ as follows:

$$f(0) = 0, f(1) = 5, f(2) = 10, f(3) = 15, \ldots, f(n) = 5n.$$

This function is onto since for every element s in the set, $s = 5x$ for some $x \in \mathbb{N}$ and by the definition of f, $f(x) = s$, which means that every element in the set is reached.

(b) Prove that the set of irrational numbers is uncountable. You may use the fact that the real numbers are uncountable and the rationals are countable. **Hint:** Every real number is either rational or irrational.

Solution:

Suppose for contradiction that the irrational numbers were countable. Then we have a function g which maps the naturals to the irrationals. We also know that the rationals are countable, so there exists a function h which maps the naturals to the irrationals. Then define a new function f which alternates between g and h like so:

$$f(0) = g(0), f(1) = h(0), f(2) = g(1), f(3) = h(1), f(4) = g(2), \ldots, f(n) = \begin{cases} g(n/2) & \text{if } n \text{ even} \\ h((n-1)/2) & \text{if } n \text{ odd} \end{cases}$$

This function is onto the reals since g and h are onto the irrationals and rationals, which would mean that the reals are countable. But this is a contradiction since we know the reals are uncountable. Therefore the irrational numbers must be uncountable.

(c) Prove that $\mathcal{P}(\mathbb{N})$ is uncountable.

Solution:

Assume for the sake of contradiction that $\mathcal{P}(\mathbb{N})$ is countable. This means we can define an enumeration of elements S_i in $\mathcal{P}(\mathbb{N})$. Let s_i be the binary set representation of S_i in $\mathbb{N}$. For example, for the set $\{0, 1, 2\}$, the binary set representation would be $111000\ldots$.

We then construct a new subset $X \subset \mathbb{N}$ such that $x[i] = 1$ if $s_i[i] = 0$, and $x[i] = 0$ otherwise. Note that X is not any of the S_i, since it differs from S_i on the i-th natural number. However, X still represents a valid subset of the natural numbers, which means our enumeration is incomplete, leading to a contradiction. Since the above proof works for any listing of $\mathcal{P}(\mathbb{N})$, no listing can be created for $\mathcal{P}(\mathbb{N})$, and therefore $\mathcal{P}(\mathbb{N})$ is uncountable.

3. Translations

Translate the following sentences into logical notation if the English statement is given or to an English statement if the logical statement is given, taking into account the domain restriction.

Let the domain of discourse be students and courses.

Use predicates $\mathsf{Student}$, Course, $\mathsf{CseCourse}$ to do the domain restriction.

You can use $\mathsf{Taking}(x, y)$ which is true if and only if x is taking y. You can also use $\mathsf{JacobTeaches}(x)$ if and only if Jacob teaches x and $\mathsf{ContainsTheory}(x)$ if and only if x contains theory.

(a) Every student is taking some course.

Solution:

$$\forall x \exists y(\mathsf{Student}(x) \rightarrow [\mathsf{Course}(y) \wedge \mathsf{Taking}(x, y)])$$

(b) There is a student that is not taking every CSE course. **Solution:**

$$\exists x \exists y[\mathsf{Student}(x) \wedge (\mathsf{CseCourse}(y) \rightarrow \neg\mathsf{Taking}(x, y))]$$

(c) Some student has taken only one CSE course. **Solution:**

$\exists x \exists y[\mathsf{Student}(x) \wedge \mathsf{CseCourse}(y) \wedge \mathsf{Taking}(x, y) \wedge \forall z((\mathsf{CseCourse}(z) \wedge \mathsf{Taking}(x, z)) \rightarrow y = z)]$

(d) $\forall x[(\mathsf{Course}(x) \wedge \mathsf{JacobTeaches}(x)) \rightarrow \mathsf{ContainsTheory}(x)]$ **Solution:**

Every course taught by Jacob contains theory.

(e) $\exists x(\mathsf{CseCourse}(x) \wedge \mathsf{JacobTeaches}(x) \wedge \mathsf{ContainsTheory}(x) \wedge \forall y((\mathsf{CseCourse}(y) \wedge \mathsf{JacobTeaches}(y)) \rightarrow x = y)$
Solution:

There is only one CSE course that Jacob teaches and that course contains theory.

4. Induction

(a) A Husky Tree is a tree built by the following definition:

Basis: A single gold node is a Husky Tree.

Recursive Rules:

(i) Let T_1 and T_2 be two Husky Trees, both with root nodes colored gold. Make a new purple root node and attach the roots of T_1 and T_2 to the new node to make a new Husky Tree.

(ii) Let T_1 and T_2 be two Husky Trees, both with root nodes colored purple. Make a new purple root node and attach the roots of T_1 and T_2 to the new node to make a new Husky Tree.

(iii) Let T_1 and T_2 be two Husky Trees, one with a purple root and the other with a gold root. Make a new gold root node and attach the roots of T_1 and T_2 to the new node to make a new Husky Tree.

Use structural induction to show that for every Husky Tree:

- If it has a purple root, then it has an even number of leaves.
- If it has a gold root, then it has an odd number of leaves.

Solution:

Let $P(T)$ be "if T has a purple root, then it has an even number of leaves, and if T has a gold root, then it has an odd number of leaves."

We show $P(T)$ holds for all Husky Trees T by structural induction.

Base Case: Let T be a Husky Tree made from the basis step. By the definition of a Husky Tree, T must be a single gold node. That node is also a leaf node (since it has no children), so there are an odd number (specifically, 1) of leaves, as required for a gold root node.

Inductive Hypothesis: Let T_1 and T_2 be arbitrary Husky Trees, and suppose $P(T_1)$ and $P(T_2)$ hold.

Inductive Step: We will have separate cases for each possible rule.

- **Rule 1:** Suppose T_1 and T_2 both have gold roots. By the recursive rule, T has a purple root. By the inductive hypothesis on T_1, since T_1's root is gold, it has an odd number of leaves. Similarly, by the inductive hypothesis, T_2 has an odd number of leaves. T's leaves are exactly the leaves of T_1 and T_2, so the total number of leaves in T is the sum of two odd numbers, which is even. Thus, T has an even number of leaves, as is required for a purple root. Thus, $P(T)$ holds.
- **Rule 2:** Suppose T_1 and T_2 both have purple roots. By the recursive rule, T has a purple root. By the inductive hypothesis on T_1, since T_1's root is purple, it has an even number of leaves. Similarly, by the inductive hypothesis, T_2 has an even number of leaves. T's leaves are exactly the leaves of T_1 and T_2, so the total number of leaves in T is the sum of two even numbers, which is even. Thus, T has an even number of leaves, as is required for a purple root. Thus, $P(T)$ holds.
- **Rule 3:** Suppose T_1 and T_2 have opposite colored roots. Let T_1 be the one with a gold root, and T_2 the one with a purple root. By the recursive rule, T has a gold root. By the inductive hypothesis on T_1, since T_1's root is gold, it has an odd number of leaves. Similarly, by the inductive hypothesis, T_2 has an even number of leaves since it has a purple root. T's leaves are exactly the leaves of T_1 and T_2, so the total number of leaves in T is the sum of an odd number and an even number, which is odd. Thus, T has an odd number of leaves, as is required for a gold root. Thus, $P(T)$ holds.

By the principle of induction, we have that for every Husky Tree T, $P(T)$ holds.

(b) Use induction to prove that for every positive integer n,

$$1 + 5 + 9 + \cdots + (4n - 3) = n(2n - 1).$$

Solution:

For $n \in \mathbb{Z}^+$ let $P(n)$ be ``$1+5+9+\cdots+(4n-3) = n(2n-1)$''. We show $P(n)$ for all $n \in \mathbb{Z}^+$ by induction on n.

Base Case: We have $1 = (1) = 1(2-1)$ which is $P(1)$ so the base case holds.

Inductive Hypothesis: Suppose $P(k)$ holds for some arbitrary integer $k \geq 1$.

Inductive Step: We have

$$\begin{aligned}
1+5+9+\cdots+(4(k+1)-3) &= 1+5+9+\cdots+(4k-3)+(4(k+1)-3) \\
&= k(2k-1)+(4(k+1))-3) && \text{[Inductive Hypothesis]} \\
&= k(2k-1)+(4k+1) \\
&= 2k^2+3k+1 \\
&= (k+1)(2k+1) \\
&= (k+1)(2(k+1)-1).
\end{aligned}$$

which proves $P(k+1)$.

Conclusion: $P(n)$ holds for all $n \in \mathbb{Z}^+$ by the principle of induction.

5. Languages

(a) Construct a regular expression that represents binary strings where no occurence of 11 is followed by a 0.

Solution:

$$(0^*(10)^*)^*1^*$$

(b) Construct a CFG that represents the following language: $\{1^x2^y3^y4^x : x, y \geq 0\}$.

Solution:

$$\mathbf{S} \to 1\mathbf{S}4 \mid \mathbf{T}$$
$$\mathbf{T} \to 2\mathbf{T}3 \mid \varepsilon$$

(c) Construct a DFA that recognizes the language of all binary strings which,when interpreted as a binary number, are divisible by 3. e.g. 11 is 3 in base-10, so should be accepted while 111 is 7 in base-10, so should be rejected. The first bit processed will be the most-significant bit. Hint: you need to keep track of the remainder mod 3. What happens to a binary number when you add a 0 at the end? A 1? It's a lot like a shift operation...

Solution: