Section 10: Solutions

1. Irregularity

(a) Let $\Sigma = \{0, 1\}$. Prove that $\{0^n 1^n 0^n \ : \ n \geq 0\}$ is not regular.
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. Say these strings are $0^i 1^i$ 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:

$a = 0^i 1^i 0^i$ Note that $a \in L$.

$b = 0^j 1^j 0^i$ Note that $b \notin L$, 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 accept and reject 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 $L = \{0^n (12)^m \ : \ n \geq m \geq 0\}$. Let D be an arbitrary DFA, and suppose for contradiction that D accepts L. Consider $S = \{0^n : n \geq 0\}$. 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. Say these strings are 0^i and 0^j for some $i, j \geq 0$ such that $i > j$. Append the string $(12)^i$ to both of these strings. The two resulting strings are:

$a = 0^i (12)^i$ Note that $a \in L$.

$b = 0^j (12)^i$ Note that $b \notin L$, since $i > j$.

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

2. Cardinality

(a) You are a pirate. You begin in a square on a 2D grid which is infinite in all directions. In other words, wherever you are, you may move up, down, left, or right. Some single square on the infinite grid has treasure on it. Find a way to ensure you find the treasure in finitely many moves.
Solution:

Explore the square you are currently on. Explore the unexplored perimeter of the explored region until you find the treasure (your path will look a bit like a spiral).

(b) Prove that $\{3x \ : \ x \in \mathbb{N}\}$ is countable.
Solution:

We can enumerate the set as follows:

$$f(0) = 0$$
$$f(1) = 3$$
$$f(2) = 6$$
$$f(i) = 3i$$

Since every natural number appears on the left, and every number in S appears on the right, this enumeration spans both sets, so S is countable.

(c) Prove that the set of irrational numbers is uncountable.
Hint: Use the fact that the rationals are countable and that the reals are uncountable.
Solution:

We first prove that the union of two countable sets is countable. Consider two arbitrary countable sets C_1 and C_2. We can enumerate $C_1 \cup C_2$ by mapping even natural numbers to C_1 and odd natural numbers to C_2.
Now, assume that the set of irrationals is countable. Then the reals would be countable, since the reals are the union of the irrationals (countable by assumption) and the rationals (countable). However, we have already shown that the reals are uncountable, which is a contradiction. Therefore, our assumption that the set of irrationals is countable is false, and the irrationals must be uncountable.

(d) 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 $\mathcal{P}$.
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 $x[i] = \ s_i[i]$ (that is, $x[i]$ is 1 if $s_i[i]$ is 0, and $x[i]$ is 0 otherwise).
Note that X is not any of S_i, since it differs from S_i on the ith natural number. However, X still represents a valid subset of the natural numbers, which means our enumeration is incomplete, which is 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. Countable Unions

(a) Show that $\mathbb{N} \times \mathbb{N}$ is countable.

Hint: How did we show the rationals were countable?

Solution:

We use dovetailing to create a sequence of elements of $\mathbb{N} \times \mathbb{N}$ that includes the entirety of $\mathbb{N} \times \mathbb{N}$.

For a fixed integer $k \geq 2$, consider subset S_k of $\mathbb{N} \times \mathbb{N}$ consisting of the elements (a, b) such that $a + b = k$. There can be at most $k - 1$ such elements because for each value of $a = 1, 2, \ldots, k - 1$, there can only be one possible value for b, namely $k - a$. Thus, if we create a sequence consisting of all the elements of S_2, then S_3, then S_4, etc. because each set is of finite size, any pair $(a, b) \in \mathbb{N} \times \mathbb{N}$ will eventually show up in this sequence in S_{a+b}.

Thus, because we can enumerate the elements of $\mathbb{N} \times \mathbb{N}$, it must be countable.

(b) Show that the countable union of countable sets is countable. That is, given a collection of sets $S_1, S_2, S_2, \ldots$ such that S_i is countable for all $i \in \mathbb{N}$, show that

$$S = S_1 \cup S_2 \cup \cdots = \{x : x \in S_i \text{ for some } i\}$$

is countable.

Hint: Find a way labeling the elements and see if you can apply the previous part to construct an onto function from $\mathbb{N}$ to S.

Solution:

Because each S_i is countable, the elements can be enumerated. Let the elements of S_i be $a_{i,1}, a_{i,2}, a_{i,3}, \ldots$. Next, because $\mathbb{N} \times \mathbb{N}$ is countable, there exists an onto function $f : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$. Then define the function $g : \mathbb{N} \to S$ as follows. For each $n \in \mathbb{N}$, let $(i_n, j_n) = f(n)$. The define $g(n)$ to be a_{i_n,j_n}.

I claim g is onto. Indeed, let $a_{i,j}$ be an arbitrary element of S. Because f is onto, there exists an n such that $f(n) = (i, j)$. Then $g(n) = a_{i,j}$. This shows g is onto and thus S is countable.

4. Uncomputability

(a) Let $\Sigma = \{0, 1\}$. Prove that the set of palindromes is decidable.
Solution:

We can implement the function that takes a string as input and reverses that string, using the recursive definition of string reverse given in class. So on input x we run that reversing program to create the string $y = x^R$. Then we compare x against y character by character and output yes iff we find that $x = y$.

(b) Prove that the set $\{(\texttt{CODE}(R), x, y) : R \text{ is a program and } R(x) \neq R(y)\}$ is undecidable where $R(x)$ is the output string that R produces on input x if R halts and we write $R(x) = \uparrow$ if R runs forever.
Solution:

Let S be the set $\{(\texttt{CODE}(R), x, y) : R \text{ is a program and } R(x) \neq R(y)\}$. Assume for the sake of contradiction that S is decidable. Then there exists some program `Q(String input, String x, String y)` which returns `true` iff $(\texttt{CODE}(R), x, y) \in S$.

Let `P()` be some arbitrary program. We will show that we can use `Q` to determine if `P` halts.

We first write a program `I(String input)` that incorporates the code of `P`:

```
String I(String input) {
   if (input.equals("kittens")) {
       // Run forever
       while (true) {
       }
   } else {
       // Execute P
       <Code of P>
   }
}
```

Note that this program will always run forever when the input is "kittens" OR P runs forever, but will otherwise return whatever P returns.
Now, we can write DOESHHALT():

```
boolean DOESHHALT() {
   return Q(CODE(I),"kittens","bunnies");
}
```

If Q(CODE(I),"kittens","bunnies") returns true, then I("kittens") $\neq$ I("bunnies"), so P does not run forever, so P halts.

If Q(CODE(I),"kittens","bunnies") returns false, then I("kittens") = I("bunnies"), so P runs forever, so P does not halt.

Since P was arbitrary, we can construct a program using Q() like DOESHHALT() for *any* program, which allows us to decide the halting set. Since we can use Q to decide the halting set, but the halting set is undecidable, Q cannot exist.
Since Q was an arbitrary function that decides S, no function that decides S can exist, and therefore S is undecidable.

5. Review: 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 Student, Course, CseCourse to do the domain restriction.

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

(a) Every student is taking some course. **Solution:**

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

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

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

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

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

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

Every course taught by Robbie contains theory.

(e) $\exists x\, \texttt{CseCourse}(x) \wedge \texttt{RobbieTeaches}(x) \wedge \texttt{ContainsTheory}(x) \wedge \forall y((\texttt{CseCourse}(y) \wedge \texttt{RobbieTeaches}(y)) \rightarrow x = y)$
Solution:

There is only one cse course that Robbie teaches and that course contains theory.

6. Review: Functions

Let $f : X \to Y$ be a function. For a subset C of X, define $f(C)$ to be the set of elements that f sends C to. In other words, $f(C) = \{f(c) : c \in C\}$.

Let A, B be subsets of X. Prove that $f(A \cap B) \subseteq f(A) \cap f(B)$.

Solution:

Let $y \in f(A \cap B)$ be arbitrary. Then there exists some element $x \in A \cap B$ such that $f(x) = y$. Then by the definition of intersection, $x \in A$ and $x \in B$. Then $f(x) \in f(A)$ and $f(x) \in f(B)$. Thus $y \in f(A)$ and $y \in f(B)$. By definition of intersection, $y \in f(A) \cap f(B)$. Since y was arbitrary, $f(A \cap B) \subseteq f(A) \cap f(B)$.

7. Review: Induction

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

Basis: A single gold node is a Husky Tree.

Recursive Rules:

1. Let T_1, 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, T_2 to the new node to make a new Husky Tree.
2. Let T_1, 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, T_2 to the new node to make a new Husky Tree.
3. Let T_1, T_2 be two Husky Trees, one with a purple root, the other with a gold root. Make a new gold root node, and attatch the roots of T_1, 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 and 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 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, T_2 be arbitrary Husky Trees, and suppose $P(T_1)$ and $P(T_2)$.

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 inductive hypothesis on T_1, since T_1's root is gold, it has an odd number of leaves. Similarly by IH, 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 inductive hypothesis on T_1, since T_1's root is purple, it has an even number of leaves. Similarly by IH, 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 the purple root. By the recursive rule, T has a gold root. By inductive hypothesis on T_1, since T_1's root is gold, it has an odd number of leaves. Similarly, by IH, 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 $1 = 1(1) = 1(2-1)$ which is $P(1)$ so the base case holds.

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

Inductive Step: $\boxed{\text{Goal: Show } 1 + 5 + 9 + \cdots + (4(k+1) - 3) = (k+1)(2(k+1) - 1)}$.

We have:

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

This proves $P(k+1)$.

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

8. Review: Languages

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

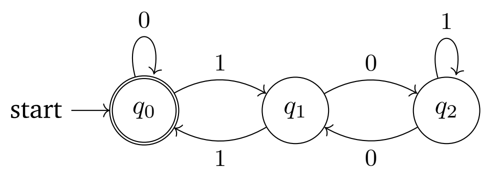
$$(0^*(10)^*)^*1^*$$

(b) Construct a CFG that represents the following language: $\{1^x 2^y 3^y 4^x : x, y \geq 0\}$ **Solution:**

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

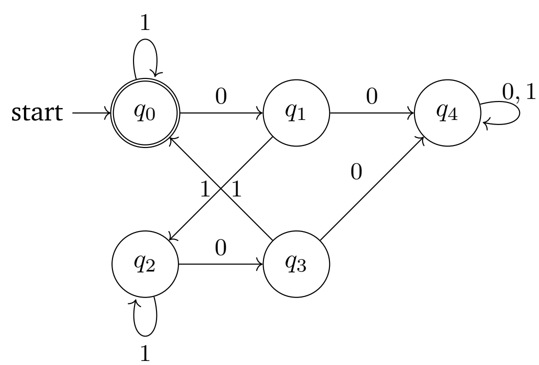
(c) Construct a DFA that recognizes the language of all binary strings which, when interpreted as a binary number, are divisible by 3. e.g. 101 is 3 in base-10, so should be accepted while 111 is 7 in base-10, so should be rejected. The first bit processed will be the **most**-significant bit.
Hint: you need to keep track of the remainder %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:



(d) Construct a DFA that recognizes the language of all binary strings with an even number of 0's and each 0 is (immediately) followed by at least one 1.

Solution:



q_0: even number of 0's, with final 0 followed by at least one 1

q_1: odd number of 0's, with final 0 not yet followed by at least one 1

q_2: odd number of 0's, with final 0 followed by at least one 1

q_3: even number of 0's, with final 0 not yet followed by at least one 1

q_4: garbage state where at least one 0 is not followed by at least one 1

9. Review: Uncountability

Let S be the set of all real numbers in $[0, 1)$ that only have 0s and 1s in their decimal representation. Prove that S is uncountably infinite.

Solution:

Suppose for the sake of contradiction that S is countable. Then there exists a surjection $f : \mathbb{N} \to S$. So for each natural number i, we have some decimal sequence of 0s and 1s that i maps to.

We now construct an element x. We start x with $0.$ (a zero and decimal point). Then for all $i \in \mathbb{N}$, let the ith digit after the decimal point be 1 if $f(i) = 0$, and 0 if $f(i) = 1$.

Note that by our construction, for any $i \in N$, $f(i)$ differs from x on the i-th digit after the decimal point. Furthermore by our construction, x contains only 0s and 1s in its decimal expansion and $x \in [0, 1)$, so $x \in S$. Since $x \in S$ but is not in the range of f, f is not surjective. This is a contradiction. Therefore S is uncountable.