CSE 311: Foundations of Computing I

Section 10: Final Exam Review Solutions

0. Induction

Let f_n and g_n be defined as:

$$f_0 = 0$$
$$f_1 = 1$$
$$f_2 = 1$$
$$f_n = f(n-1) + f(n-2) + f(n-3)$$

$$g_0 = 0$$
$$g_1 = 1$$
$$g_n = g(n-1) + g(n-2)$$

Prove that $f_n \leq 3^n$ for all $n \geq 0$ and $f_n \geq g_n$ for all $n \geq 2$ by strong induction.

Solution:

$f_n \leq 3^n$:

Case $n = 0$**:** Note, $f_0 = 0 \leq 1 = 3^0$

Case $n = 1$**:** Note, $f_1 = 1 \leq 3 = 3^1$

Case $n = 2$**:** Note, $f_2 = 1 \leq 9 = 3^2$

Case $n > 2$**:** Suppose that $f_i \leq 3^i$ for all $0 \leq i \leq k$ for some $k \geq 2$. Note that:

$$\begin{aligned} f_{k+1} &= f_k + f_{k-1} + f_{k-2} && \text{[By Definition of f]} \\ &\leq 3^k + 3^{k-1} + 3^{k-2} && \text{[By IH]} \\ &\leq 3^k + 3^k + 3^k \\ &= 3^{k+1} \end{aligned}$$

Thus, the claim is true for all $n \geq 0$ by structural induction.

$f_n \geq g_n$: For this proof, we take a moment to prove that f_n is non-negative.

Case $n = 0$**:** $f_0 = 0 \geq 0$

Case $n = 1$**:** $f_1 = 1 \geq 0$

Case $n = 2$**:** $f_2 = 1 \geq 0$

Case $n > 2$**:** Suppose that $f_i \geq 0$ for all $0 \leq i \leq k$ for some $k \geq 2$. Note that $f_{k+1} = f_k + f_{k-1} + f_{k-2}$ by definition of f. We can also note $f_k \geq 0$, $f_{k-1} \geq 0$, and $f_{k-2} \geq 0$ by IH, so therefore, $f_{k+1} \geq 0$.

Thus, the claim that f_n is non-negative is true by strong induction. Now we move on to the real claim we want to prove.

Case $n = 0$**:** Note, $f_0 = 0 \geq 0 = g_0$

Case $n = 1$**:** Note, $f_1 = 1 \geq 1 = g_1$

Case $n = 2$**:** Note, $f_2 = 1 \geq 0 + 1 = g_0 + g_1 = g_2$

Case $n > 2$**:** Suppose that $f_i \geq g_i$ for all $0 \leq i \leq k$ for some $k \geq 2$. Note that:

$$\begin{aligned} f_{k+1} &= f_k + f_{k-1} + f_{k-2} && \text{[By Definition of f]} \\ &\geq g_k + g_{k-1} + f_{k-2} && \text{[By IH]} \\ &\geq g_k + g_{k-1} + 0 && \text{[By earlier proof]} \\ &= g(k+1) && \text{[By Definition of g]} \end{aligned}$$

Thus, the claim is true.

1. Structural Induction

Consider some new programs on lists:

$$\begin{aligned} &\text{in}(a, [\,]) && = \text{false} \\ &\text{in}(a, b :: S) && = \text{if } a = b \text{ then true else in}(a, S) \\ \\ &\text{set}([\,]) && = [\,] \\ &\text{set}(a :: S) && = \text{not in}(a, S) \text{ and set}(S) \\ \\ &\text{append}(a, [\,]) && = a :: [\,] \\ &\text{append}(a, b :: L) && = b :: \text{append}(a, L) \\ \\ &\text{rev}([\,]) && = [\,] \\ &\text{rev}(a :: L) && = \text{append}(a, \text{rev}(L)) \\ \\ &\text{getAll}(a, [\,]) && = [\,] \\ &\text{getAll}(a, b :: L) && = \text{if } a = b \text{ then } b :: \text{getAll}(a, L) \text{ else getAll}(a, L) \end{aligned}$$

Suppose that for arbitrary p, q, and list L, $\text{getAll}(p, q :: L)) = \text{getAll}(p, \text{append}(q, L))$. Prove that if $\text{set}(L)$, then $\text{getAll}(a, L) = \text{getAll}(a, \text{rev}(L))$ for all lists L and elements a.

Solution:

Let L be an arbitrary list. We prove the claim via structural induction.

Case: $L = [\,]$: Let a be an arbitrary element. Note that:

$$\begin{aligned} \text{getAll}(a, [\,]) &= [\,] && \text{[Definition of getAll]} \\ &= \text{getAll}(a, [\,]) && \text{[Definition of getAll]} \\ &= \text{getAll}(a, \text{rev}([\,])) && \text{[Definition of rev]} \end{aligned}$$

So, the claim holds for $L = [\,]$.

Case: $L = b :: L'$. Let a be an arbitrary element. Suppose the claim holds for L'. Suppose $\text{set}(b :: L')$. Note that by definition of set, not $\text{in}(b, L')$ and $\text{set}(L')$. Since $\text{set}(L')$, our IH can simplify to $\text{getAll}(a, L') = \text{getAll}(a, \text{rev}(L'))$. We go by cases:

Case: $a = b$: Note that:

$$\begin{aligned} \text{getAll}(a, b :: L') &= b :: \text{getAll}(a, L') && \text{[Definition of getAll]} \\ &= b :: \text{getAll}(a, \text{rev}(L')) && \text{[IH]} \\ &= \text{getAll}(a, b :: \text{rev}(L')) && \text{[Definition of getAll]} \\ &= \text{getAll}(a, \text{append}(b, \text{rev}(L'))) && \text{[Supposition]} \\ &= \text{getAll}(a, \text{rev}(b :: L')) && \text{[Definition of rev]} \end{aligned}$$

So the claim holds when $a = b$.
Case: $a \neq b$: Note that:

$$\begin{aligned}
\text{getAll}(a, b :: L') &= \text{getAll}(a, L') && \text{[Definition of getAll]} \\
&= \text{getAll}(a, \text{rev}(L')) && \text{[IH]} \\
&= \text{getAll}(a, b :: \text{rev}(L')) && \text{[Definition of getAll, } a \neq b\text{]} \\
&= \text{getAll}(a, \text{append}(b, \text{rev}(L'))) && \text{[Supposition]} \\
&= \text{getAll}(a, \text{rev}(b :: L')) && \text{[Definition of rev]}
\end{aligned}$$

So the claim holds for $a \neq b$. So, we've shown the claim holds for all cases of L, so we have proven the claim for all lists L and elements a by structural induction.

2. Regular Expressions, CFGs, and FSMs

Let $\Sigma = \{\texttt{H}, \texttt{J}, \texttt{K}, \texttt{L}\}$. Let the language L be defined for Σ^* such that $w \in L$ iff w:

- starts with `K` and ends with `L` or starts with `L` and ends with `K`
- has exactly one `J` between any two (not necessarily consecutive) occurrences of `K`
- has exactly one `H` between any two (not necessarily consecutive) occurrences of `L`

(a) Write a regular expression that matches L.

Solution:

$$\begin{gathered}
\texttt{K}(\texttt{JK} \cup \texttt{LH} \cup \texttt{JLKH} \cup \texttt{JLHK} \cup \texttt{LJHK} \cup \texttt{LJKH} \cup \varepsilon)^*\texttt{L} \\
\cup \\
\texttt{L}(\texttt{KJ} \cup \texttt{HL} \cup \texttt{KHJL} \cup \texttt{KHLJ} \cup \texttt{HKLJ} \cup \texttt{HKJL} \cup \varepsilon)^*\texttt{K}
\end{gathered}$$

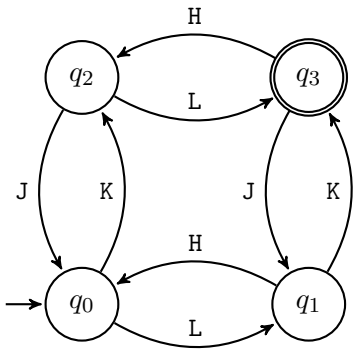
(b) Construct a CFG that generates L.

Solution:

$$\begin{aligned}
\texttt{S} &\to \texttt{K}\mathbf{T}_1\texttt{L} \mid \texttt{L}\mathbf{T}_2\texttt{K} \\
\mathbf{T}_1 &\to \texttt{JK}\mathbf{T}_1 \mid \texttt{LH}\mathbf{T}_1 \mid \texttt{JLKH}\mathbf{T}_1 \mid \texttt{JLHK}\mathbf{T}_1 \mid \texttt{LJHK}\mathbf{T}_1 \mid \texttt{LJKH}\mathbf{T}_1 \mid \varepsilon \\
\mathbf{T}_2 &\to \texttt{KJ}\mathbf{T}_2 \mid \texttt{HL}\mathbf{T}_2 \mid \texttt{KHJL}\mathbf{T}_2 \mid \texttt{KHLJ}\mathbf{T}_2 \mid \texttt{HKLJ}\mathbf{T}_2 \mid \texttt{HKJL}\mathbf{T}_2 \mid \varepsilon
\end{aligned}$$

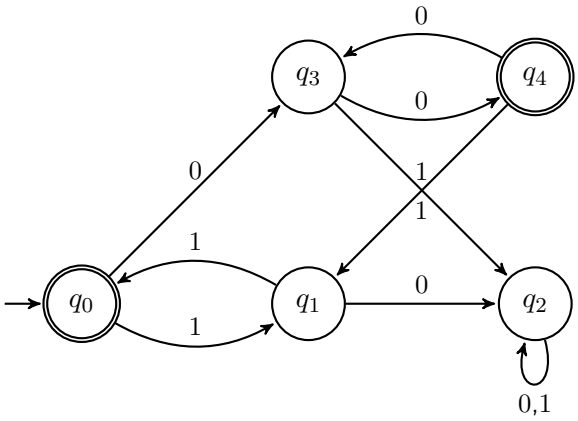
(c) Construct an NFA that accepts L.

Solution:

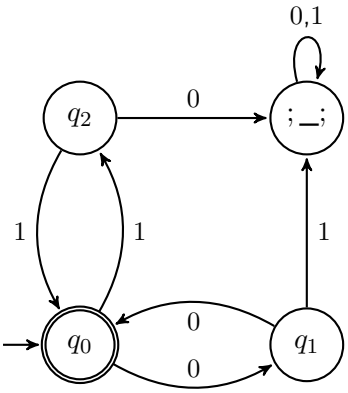


3. DFA Minimization

Minimize the following DFA:



Solution:



4. Irregularity

Let $\Sigma = \{A, C, G, T\}$. Let $\overline{w}$ be defined for a string $w \in \Sigma*$ such that for each character w_i, the character $\overline{w_i}$ is the complement of w_i. C and G are complements of each other, as are A and T. Prove that $\{w\overline{w} \in \Sigma^*\}$ is not regular.

Solution:

Let $L = \{w\overline{w} \ : \ w, \overline{w} \in \Sigma^*\}$. Let D be an arbitrary DFA, and suppose for contradiction that D accepts L. Consider $S = \{A^n : n \geq 0\}$. 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 A^i and A^j for some $i, j \geq 0$ such that $i \neq j$. Append the string T^i to both of these strings. The two resulting strings are:

$a = A^iT^i$ Note that $a \in L$, since $T^i = \overline{A^i}$, thus we can choose $w = A^i$ and write a as $w\overline{w}$.

$b = A^jT^i$ Note that $b \notin L$, since the length of $A^j \neq$ the length of T^i, thus $T^i \neq \overline{A^j}$.

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.

5. Diagonalization

Here is a "proof" that the positive rationals are uncountable.

> Suppose for contradiction that the positive rationals $\mathbb{Q}^+$ are countable. Then there exists some listing of all elements $\mathbb{Q}^+ = \{q_1, q_2, q_3, \dots\}$. Note that each of these rationals q_i can also be written as an infinite decimal expansion. We define a new number $X \in \mathbb{Q}^+$ by flipping the diagonals of $\mathbb{Q}^+$; we set the ith digit of X to 7 if the ith digit of q_i is a 4, otherwise we set the digit to 4. This means that X differs from every q_i on the ith digit, so X *cannot* be one of q_i. Therefore our listing for $\mathbb{Q}^+$ was incomplete, which is a contradiction. Since the above proof works for any listing of the positive rationals $\mathbb{Q}^+$, *no* listing can be created for $\mathbb{Q}^+$, and therefore $\mathbb{Q}^+$ is uncountable.

What is the key error in this proof?

Solution:

X is not guaranteed to be a rational number (in fact, it almost certainly isn't), so X does not need to be in our listing of $\mathbb{Q}^+$ for our listing to be complete, so there is no contradiction.

6. 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.

7. Relations

Recall the following definitions of a relation R on A:

R is reflexive iff $(a, a) \in R$ for every $a \in A$.

R is symmetric iff $(a, b) \in R$ implies $(b, a) \in R$.

R is anti-symmetric iff $(a, b) \in R$ and $(b, a) \in R$ implies $a = b$.

R is transitive iff $(a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$.

(a) Suppose that R is reflexive. Prove that $R \subseteq R^2$.

Solution:

Suppose $(a, b) \in R$. Since R is reflexive, we know $(b, b) \in R$ as well. Since there is a b such that $(a, b) \in R$ and $(b, b) \in R$, it follows that $(a, b) \in R^2$. Thus, $R \subseteq R^2$.

(b) Consider the relation $R = \{(x, y) : x = y + 1\}$ on $\mathbb{N}$. Is R reflexive? Transitive? Symmetric? Anti-symmetric?

Solution:

It isn't reflexive, because $1 \neq 1 + 1$; so, $(1, 1) \notin R$. It isn't symmetric, because $(2, 1) \in R$ (because $2 = 1 + 1$), but $(1, 2) \notin R$, because $1 \neq 2 + 1$. It isn't transitive, because note that $(3, 2) \in R$ and $(2, 1) \in R$, but $(3, 1) \notin R$. It is anti-symmetric, because consider $(x, y) \in R$ such that $x \neq y$. Then, $x = y + 1$ by definition of R. However, $(y, x) \notin R$, because $y = x - 1 \neq x + 1$.

R is transitive iff $(a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$.

(c) Suppose that R is reflexive. Prove that $R \subseteq R^2$.

Solution:

Suppose $(a, b) \in R$. Since R is reflexive, we know $(b, b) \in R$ as well. Since there is a b such that $(a, b) \in R$ and $(b, b) \in R$, it follows that $(a, b) \in R^2$. Thus, $R \subseteq R^2$.

(d) Consider the relation $R = \{(x, y) : x = y + 1\}$ on $\mathbb{N}$. Is R reflexive? Transitive? Symmetric? Anti-symmetric?

Solution:

It isn't reflexive, because $1 \neq 1+1$; so, $(1,1) \notin R$. It isn't symmetric, because $(2,1) \in R$ (because $2 = 1+1$), but $(1,2) \notin R$, because $1 \neq 2+1$. It isn't transitive, because note that $(3,2) \in R$ and $(2,1) \in R$, but $(3,1) \notin R$. It is anti-symmetric, because consider $(x,y) \in R$ such that $x \neq y$. Then, $x = y+1$ by definition of R. However, $(y,x) \notin R$, because $y = x-1 \neq x+1$.

(e) Consider the relation $S = \{(x,y) \ : \ x^2 = y^2\}$ on $\mathbb{R}$. Prove that S is reflexive, transitive, and symmetric.

Solution:

Consider $x \in \mathbb{R}$. Note that by definition of equality, $x^2 = x^2$; so, $(x,x) \in R$; so, R is reflexive.

Consider $(x,y) \in R$. Then, $x^2 = y^2$. It follows that $y^2 = x^2$; so, $(y,x) \in R$. So, R is symmetric.

Suppose $(x,y) \in R$ and $(y,z) \in R$. Then, $x^2 = y^2$, and $y^2 = z^2$. Since equality is transitive, $x^2 = z^2$. So, $(x,z) \in R$. So, R is transitive.

8. Uncomputability

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

Solution:

The following CFG recognizes all binary palindromes:

$$\mathbf{S} \rightarrow 0\mathbf{S}0 \mid 1\mathbf{S}1 \mid 1 \mid 0 \mid \varepsilon$$

Since a CFG exists which recognizes the set of binary palindromes, this set is at most context-free, and therefore decidable.

(b) Prove that the set $\{(\texttt{CODE}(P), x, y) \ : \ P \text{ is a program and } P(x) \neq P(y)\}$ is undecidable.

Solution:

Let S be the set $\{(\texttt{CODE}(P), x, y) \ : \ P \text{ is a program and } P(x) \neq P(y)\}$. 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 $(\texttt{CODE}(P), x, y) \in S$.

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

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

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

Note that this program will always run forever when the input is "kittens" OR `H` runs forever, but will otherwise return whatever H 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 `H` does not run forever, so `H` halts.

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

Since `H` 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.