Practice Final Exam

Task 1 – Regularly Irregular

Let $\Sigma = \{0, 1\}$. For a string s, denote $\#_0(s), \#_1(s)$ to be the number of 0s and 1s in s, respectively. Prove that $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ is irregular.

Suppose, for the sake of contradiction, that there exists a DFA M that recogizes L.

Define $S = \{0^n : n \geqslant 0\}$. Since S has infinitely many strings and M has finitely many states, there must be 2 distinct strings $0^a, 0^b \in S$, where $a, b \geqslant 0$ and $a \neq b$, that M takes to the same (intermediate) state p.

We go by cases, since knowing $a \neq b$ means $a < b$ or $a > b$.

Case 1: Suppose that $a < b$. Then consider appending 1^b to both $0^a, 0^b$.

Then $0^a 1^b \in L$ because $\#_0(0^a 1^b) = a < b = \#_1(0^a 1^b)$.

But $0^b 1^b \notin L$ because $\#_0(0^b 1^b) = b = \#_1(0^b 1^b)$.

Since M takes $0^a, 0^b$ to the same state p, M takes $0^a 1^b, 0^b 1^b$ to the same (final) state q. Since $0^a 1^b \in L$, q must be accepting. But $0^b 1^b \notin L$ so M incorrectly accepts a string not in L, which is a contradiction!

Case 2: Suppose that $b < a$. The consider appending 1^a to both $0^a, 0^b$.

Then $0^a 1^a \notin L$ because $\#_0(0^a 1^a) = a = \#_1(0^a 1^a)$.

But $0^b 1^a \in L$ because $\#_0(0^b 1^a) = b < a = \#_1(0^b 1^a)$.

Since M takes $0^a, 0^b$ to the same state p, M takes $0^a 1^a, 0^b 1^a$ to the same (final) state q. Since $0^b 1^a \in L$, q must be accepting. But $0^a 1^a \notin L$ so M incorrectly accepts a string not in L, which is a contradiction!

In both cases, we've derived a contradiction, so we have a contradiction overall. So there's no DFA that recognizes L. Thus, L is irregular.

Task 2 – Recurrences, Recurrences

Define

$$T(n) = \begin{cases} n & \text{if } n = 0, 1 \\ 4T(\lfloor \frac{n}{2} \rfloor) + n & \text{otherwise} \end{cases}$$

Prove that $T(n) \leqslant n^3$ for all $n \geqslant 3$. You may cite the fact that for positive integers $x \geqslant 0$, $\lfloor x \rfloor \leqslant x$ as a floor function property.

Let $P(n)$ be "$T(n) \leqslant n^3$". We prove $P(n)$ holds for all $n \geqslant 3$ by strong induction on n.

Base cases:

For $n = 3$, $T(3) = 4T(\lfloor \frac{3}{2} \rfloor) + 3 = 4T(1) + 3 = 4(1) + 3 = 7 \leqslant 27 = 3^3$. So $P(3)$ holds.

For $n = 4$, $T(4) = 4T(\lfloor \frac{4}{2} \rfloor) + 4 = 4T(2) + 4 = 4(4T(\lfloor \frac{2}{2} \rfloor) + 2) + 4 = 4(4T(1) + 2) + 4 = 4(4(1) + 2) + 4 = 28 \leqslant 64 = 4^3$. So $P(4)$ holds.

For $n = 5$, $T(5) = 4T(\lfloor \frac{5}{2} \rfloor) + 5 = 4T(2) + 5 = 4(4T(\lfloor \frac{2}{2} \rfloor) + 2) + 5 = 4(4T(1) + 2) + 5 = 4(4(1) + 2) + 5 = 29 \leqslant 125 = 5^3$. So $P(5)$ holds.

Inductive Hypothesis: Suppose $P(3) \wedge P(4) \wedge \cdots \wedge P(k)$ holds for some arbitrary integer $k \geqslant 5$.

Inductive Step: We show $P(k+1)$ holds.

First, we show that $3 \leqslant \lfloor \frac{k+1}{2} \rfloor \leqslant k$ so it is valid to apply the IH. Since $k \geqslant 5$, it follows that $k + 1 \geqslant 6$, so $\lfloor \frac{k+1}{2} \rfloor$ is at least 3. Using the bounds on the floor function and that $k + 1 \leqslant 2k$ for $k \geqslant 1$, we have that $\lfloor \frac{k+1}{2} \rfloor \leqslant \frac{k+1}{2} \leqslant k$.

$$\begin{aligned}
T(k+1) &= 4T\left(\left\lfloor \frac{k+1}{2} \right\rfloor\right) + (k+1) && \text{By definition of } T, k+1 \geqslant 2 \\
&\leqslant 4\left(\left\lfloor \frac{k+1}{2} \right\rfloor\right)^3 + (k+1) && \text{By IH} \\
&\leqslant 4\left(\frac{k+1}{2}\right)^3 + (k+1) && \text{By floor function property} \\
&= 4\left(\frac{(k+1)^3}{2^3}\right) + (k+1) \\
&= \frac{(k+1)^3}{2} + (k+1) \\
&= \frac{(k+1)((k+1)^2 + 2)}{2} \\
&\leqslant \frac{(k+1)((k+1)^2 + (k+1)^2)}{2} && \text{since } (k+1)^2 \geqslant 2 \text{ for } k \geqslant 1 \\
&= (k+1)^3
\end{aligned}$$

Conclusion: Thus, $P(n)$ is true for all integers $n \geqslant 3$ by strong induction.

Task 3 – All The Machines!

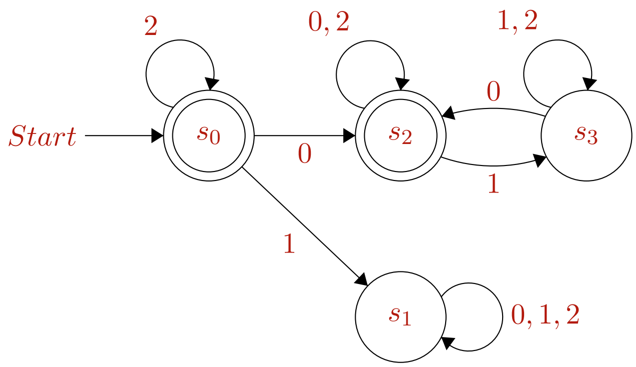
Let $\Sigma = \{0, 1, 2\}$ and define the following language:

$L = \{w \in \Sigma^* :$ Every 1 in the string has at least one 0 before and after it (not necessarily immediately)$\}$

a) Give a regular expression that represents L.

$$(0 \cup 2)^*(\varepsilon \cup 0(0 \cup 1 \cup 2)^*0)(0 \cup 2)^*$$

b) Give a DFA that recognizes L.



The key idea is that the 2s we see don't matter and we must see a 0 before the first 1.

s_0: Haven't seen 0 or 1.

s_1: Saw a 1 before any 0s.

s_2: Saw a 0 before any 1s and has at least one 0 after every 1.

s_3: Saw a 1 but haven't seen a 0 after it.

c) Give a CFG that generates L.

$S \to ABA$

$A \to 0A \mid 2A \mid \varepsilon$

$B \to 0C0 \mid \varepsilon$

$C \to 0C \mid 1C \mid 2C \mid \varepsilon$

A generates the $(0 \cup 2)^*$ part.

B generates the $\varepsilon \cup 0 \ldots 0$ part of the regular expression.

C generates $(0 \cup 1 \cup 2)^*$ part.

Task 4 – Structural CFGs

Consider the following CFG: **S** $\to \varepsilon$|**SS**|**S**1|**S**01. Another way of writing the recursive definition of this set, Q, is as follows:

- $\varepsilon \in Q$
- If $S \in Q$, then $S1 \in Q$ and $S01 \in Q$
- If $S, T \in Q$, then $ST \in Q$

Prove, by structural induction that if $w \in Q$, then w has at least as many 1's as 0's.

Let $P(w)$ be "$\#_0(w) \leqslant \#_1(w)$". We show $P(w)$ holds for all $w \in Q$ by structural induction on w.

Base Case: $\#_0(\varepsilon) = 0$ and $\#_1(\varepsilon) = 0$. Therefore, $\#_0(\varepsilon) = \#_1(\varepsilon)$, and $P(\varepsilon)$ holds.

Inductive Hypothesis: Suppose that $P(x), P(y)$ hold for some arbitrary $x, y \in Q$.

Inductive Step:

We consider all the recursive steps as cases:

For the first recursive rule, we show $P(x1), P(x01)$ hold.

Observe that

$$\begin{aligned}
\#_0(x1) &= \#_0(x) \\
&\leqslant \#_1(x) && \text{by IH} \\
&< \#_1(x) + 1 \\
&= \#_1(x1)
\end{aligned}$$

and that

$$\begin{aligned}
\#_0(x01) &= \#_0(x) + 1 \\
&\leqslant \#_1(x) + 1 && \text{by IH} \\
&= \#_1(x1)
\end{aligned}$$

For the second recursive rule, we show $P(xy)$ holds.

$$\begin{aligned}
\#_0(xy) &= \#_0(x) + \#_0(y) \\
&\leqslant \#_1(x) + \#_1(y) && \text{by IH} \\
&= \#_1(xy)
\end{aligned}$$

Conclusion: Therefore, $P(w)$ holds for all $w \in Q$ by structural induction.

Task 5 – Tralse!

a) True/False: Any subset of a regular language is also regular.

False. Consider $\{0,1\}^*$ and $\{0^n 1^n : n \geqslant 0\}$. Note that the first is regular (we can create a regular expression representation for it) and the second is irregular, but the second is a subset of the first.

b) True/False: The set of programs that loop forever on at least one input is decidable.

False. If we could solve this problem, we could solve HaltNoInput. Intuitively, a program that solves this problem would have to try all inputs, but, since the program might infinte loop on some of them, it won't be able to.

c) If $\mathbb{R} \subseteq A$ for some set A, then A is uncountable.

True. Diagonalization would still work; alternatively, if A were countable, then we could find an surjective (i.e., onto) function between $\mathbb{N}$ and $\mathbb{R}$ by skipping all the elements in A that aren't in R.

d) True/False: If the domain of discourse is people, the logical statement

$$\exists(P(x) \to \forall y(x \neq y \to \neg P(y)))$$

can be correctly translated as "There exists a unique person who has property P".

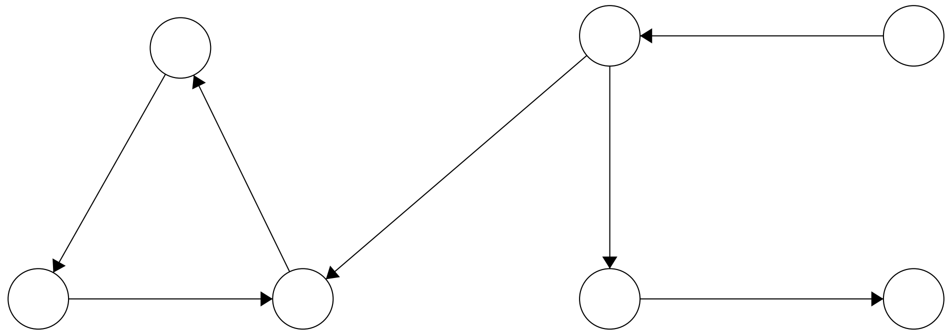
False. Any x for which $P(x)$ is false makes the entire statement true. This is not the same as there existing a unique person with property P.

e) True/False: $\exists x(\forall y P(x,y)) \to \forall y(\exists x P(x,y))$ is true regardless of what the predicate P is.

True. The left part of the implication is saying that there is a single x that works for all y; the right one is saying that for every y, we can find an x that depends on it, but the single x that works for everything will still work.

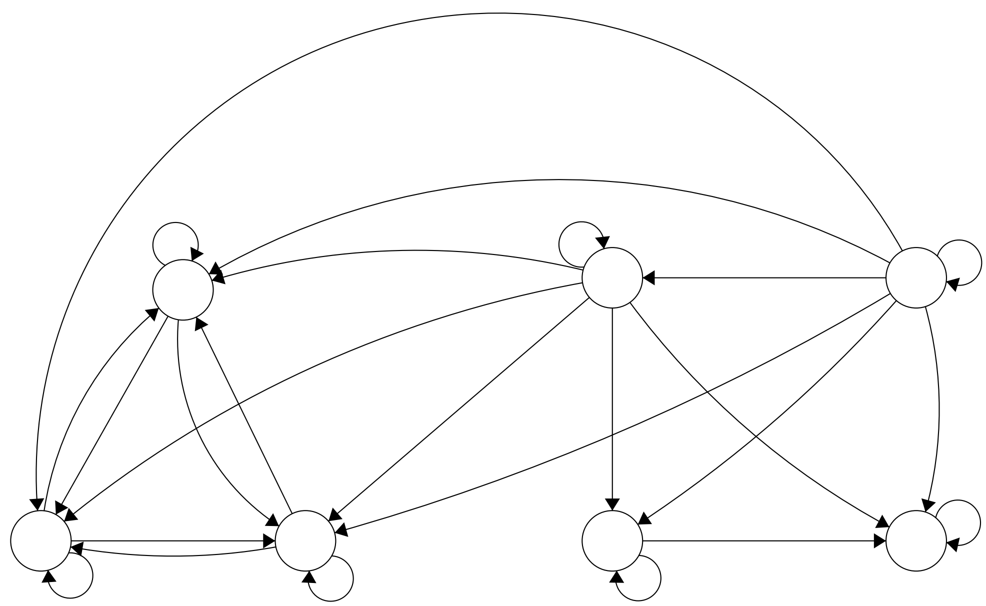
Task 6 – Relationships!

The following is the graph of a binary relation R.



a) Draw the transitive-reflexive closure of R.

Reflexive closure shown by "self-loops" for each vertex.
Transitive closure shown by repeatedly adding edges so that the start and end of paths of length 2 also have a path of length 1 (i.e., one direct edge from start to end).



b) Let $S = \{(X, Y) : X \subseteq Y\}$ be a binary relation on $P(\mathbb{N})$.

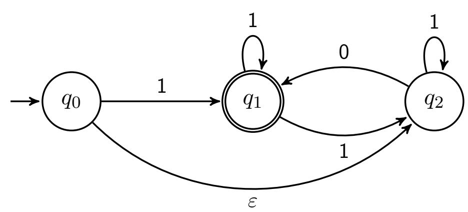
Recall that R is antisymmetric iff $\forall a \forall b((a, b) \in R \land a \neq b \rightarrow (b, a) \notin R)$.

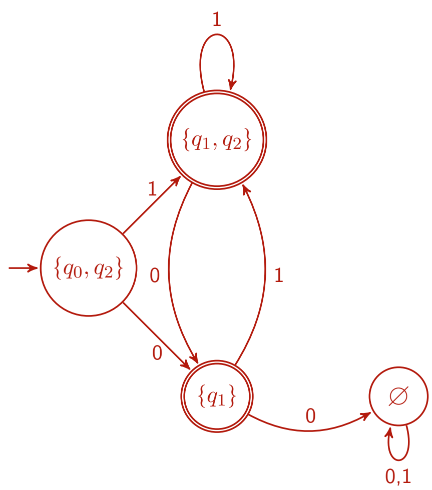
Prove that S is antisymmetric.

Let $X, Y \in \mathcal{P}(\mathbb{N})$ be arbitrary. Suppose $X \neq Y$ and $(X, Y) \in S$. By definition of S, $X \subseteq Y$. Since $X \neq Y$, we must have some element in Y that is not in X (as every element of X is in Y from the definition of subset, so it can't be the other way around). Therefore, $Y \nsubseteq X$, and $(Y, X) \notin S$. SInce X, Y were arbitrary, we've shown that S is antisymmetric.

Task 7 – Construction Paper!

Convert the following NFA into a DFA using the algorithm from lecture.





Task 8 – Modern DFAs

Let $\Sigma = \{0, 1, 2\}$. Construct a DFA that recognizes exactly strings with a 0 in all positions i where i mod $3 = 0$. Consider the first character to be at position 0. E.g., $\varepsilon, 0$ are strings in the language, but $1, 0112$ are not.