| Discrete Structures | Prelim 2 sample questions |
|---|---|
| CS2800 | Questions selected for Spring 2019 |

1. Prove that $7^m - 1$ is divisible by 6 for all positive integers $m$.

2. Suppose that Alice sends the message $a$ to Bob, encrypted using RSA. Suppose that Bob's implementation of RSA is buggy, and computes $k^{-1} \mod 4\phi(m)$ instead of $k^{-1} \mod \phi(m)$. What decrypted message does Bob see? Justify your answer.

3. (a) What are the units of $\mathbb{Z}$ mod 12?

   (b) What are their inverses?

   (c) What is $\phi(12)$?

4. Use Euler's theorem and repeated squaring to efficiently compute $8^n \mod 15$ for $n = 5$, $n = 81$ and $n = 16023$. Hint: you can solve this problem with 4 multiplications of single digit numbers. Please fully evaluate all expressions for this question (e.g. write 15 instead of $3 \cdot 5$).

5. In this problem, we are working mod 7, i.e. $\equiv$ denotes congruence mod 7 and $[a]$ is the equivalence of $a$ mod 7.

   (a) What are the units of $\mathbb{Z}_7$? What are their inverses?

   (b) Compute $[2]^{393}$.

6. (a) Recall Bézout's identity from the homework: for any integers $n$ and $m$, there exist integers $s$ and $t$ such that $gcd(n, m) = sn + tm$. Use this to show that if $gcd(k, m) = 1$ then $[k]$ is a unit of $\mathbb{Z}_m$.

   (b) Use part (a) to show that if $p$ is prime, then $\phi(p) = p - 1$.

   (c) Use Euler's theorem to compute $3^{38} \mod 37$ (note: 37 is prime).

7. Bob the Bomber wishes to receive encrypted messages from Alice the Accomplice. He generates a public key pair $m = 21$ and $k = 5$. Luckily, you have access to an NSA supercomputer that was able to factor 21 into $7 \cdot 3$.

   (a) Use this information to find the decryption key $k^{-1}$.

   (b) Without changing $m$, what other possible keys $k$ could Bob have chosen? Find the decryption keys for those keys as well.

   (c) Alice encrypts a secret message $msg$ using Bob's public key ($k = 5$), and sends the ciphertext $c = 4$. What was the original message?

8. Which of the following does RSA depend on? Explain your answer briefly.

   (a) Factoring is easy and testing primality is hard.

   (b) Factoring is hard and testing primality is easy.

   (c) Both factoring and testing primality are hard.

   (d) Both factoring and testing primality are easy.

9. (a) Let $m$ and $n$ be integers greater than 1. Show that the function $f : \mathbb{Z}_m \times \mathbb{Z}_n \to \mathbb{Z}_m$ given by $f : ([a]_m, [b]_n) \to [a + b]_m$ is not necessarily well defined. [Hint: you just need an example here.]

   (b) Show that $f$ is well defined if $m|n$.

10. We define a set $S$ of functions from $\mathbb{Z}$ to $\mathbb{Z}$ inductively as follows:

    **Rule 1.** For any $n \in \mathbb{Z}$, the translation (or offset) function $t_n : x \mapsto x + n$ is in $S$.

    **Rule 2.** For any $k \neq 0 \in \mathbb{Z}$, the scaling function $r_k : x \mapsto kx$ is in $S$.

**Rule 3.** If $f$ and $g$ are elements of $S$, then the composition $f \circ g \in S$.

**Rule 4.** If $f \in S$ and $f$ has a right inverse $g$, then $g$ is also in $S$.

In other words, $S$ consists of functions that translate and scale integers, and compositions and right inverses thereof.

> **Note:** This semester, we made a bigger distinction between the elements of an inductively defined set and the meaning of an inductively defined set. We probably would have phrased this question as follows: Let $S$ be given by
>
> $$s \in S ::= t_n \mid r_k \mid s_1 \circ s_2 \mid \mathit{rinv}\ s$$
>
> and inductively, let the function defined by $s$ (written $F_s : \mathbb{Z} \to \mathbb{Z}$) be given by the rules $F_{t_n}(x) ::= x + n$, $F_{r_k}(x) ::= ks$, $F_{s_1 \circ s_2}(x) ::= F_{s_1} \circ F_{s_2}$ and let $F_{\mathit{rinv}\ s} ::= g$ where $g$ is a right inverse of $F_s$.

(a) [1 point] Show that the function $f : x \mapsto 3x + 17$ is in $S$.

(b) Use structural induction to prove that for all $f \in S$, $f$ is injective. You may use without proof the fact that the composition of injective functions is injective.

(c) Give a surjection $\phi$ from $S$ to $\mathbb{Z}$ (proof of surjectivity not necessary). Remember that this surjection must map a *function* to an *integer*, and for every integer there must be a function that maps to it.

11. Given DFAs $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$, we can construct a machine $M_{12}$ with $L(M_{12}) = L(M_1) \cap L(M_2)$ as follows:

    - Let $Q = Q_1 \times Q_2 =$ the set of all ordered pairs $(q_1, q_2)$, where $q_1 \in Q_1$ and $q_2 \in Q_2$.
    - Let $q_0 \in Q = (q_{01}, q_{02})$.
    - Let $F = F_1 \times F_2 = \{(q_1, q_2) \mid q_1 \in F_1 \text{ and } q_2 \in F_2\}$.
    - Let $\delta_{12}((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$.
    - Let $M_{12} = (Q, \Sigma, \delta_{12}, q_0, F)$.

    Use structural induction to prove that for all $x \in \Sigma^*$, $\widehat{\delta}_{12}((q_1, q_2), x) = \left(\widehat{\delta}_1(q_1, x), \widehat{\delta}_2(q_2, x)\right)$.

12. Draw a finite automaton (DFA, NFA or $\epsilon$-NFA) with alphabet $\{0, 1\}$ to recognize the language

$$\{x \in \{0,1\}^* \mid x \text{ contains the substring } 010\}$$

13. Draw a finite automaton (DFA, NFA or $\epsilon$-NFA) with alphabet $\{a, b\}$ to recognize strings of the form $x_1 x_2 x_3 \cdots$ where each $x_i$ is either "$ab$" or "$ba$".

14. Build a deterministic finite automaton that recognizes the set of strings of 0's and 1's, that only contain a single 0 (and any number of 1's). Describe the set of strings that lead to each state.

15. Given a string $x$, we can define the "character doubling" of $x$ to be $x$ with every character doubled: for example $cd(abc) = aabbcc$. Formally, $cd(\epsilon) = \epsilon$, and $cd(xa) = cd(x)aa$. We can then define the "character doubling" of a language $L$ to be the set of all strings formed by doubling the characters of strings in $L$; formally $cd(L) = \{cd(x) \mid x \in L\}$.

    Given a DFA $M = (Q, \Sigma, \delta, q_0, F)$, we can construct a new DFA $M_{cd}$ that recognizes $cd(L(M))$ by adding a new state $q'_{qa}$ to the middle of every transition from $q$ on character $a$:

$$\begin{tikzcd} q_1 \arrow[r, "a"] & q_2 & \text{becomes} & q_1 \arrow[r, "a"] & q'_{q_1 a} \arrow[r, "a"] & q_2 \end{tikzcd}$$

(a) Formally describe the components $(Q_{cd}, \Sigma_{cd}, \delta_{cd}, q_{0cd}, F_{cd})$ of $M_{cd}$ in terms of the components of $M$. Be sure to describe $\delta_{cd}$ on all inputs (you may need to add one or more additional states).

(b) Use structural induction on $x$ to prove that for all $x$, $\widehat{\delta}(q_0, x) = \widehat{\delta}_{cd}(q_{0cd}, cd(x))$.

(c) We can also define the "string doubling" of $x$ to be $xx$. For example, $sd(abc) = abcabc$. Show that the set of regular languages is *not* closed under string doubling. In other words, give a regular language $L$ and prove that $sd(L) = \{sd(x) \mid x \in L\}$ is not regular.

You can use any theorem proved in class to help prove this result.

16. Happy Cat has been shown the following proof, and has promptly turned into Grumpy Cat. Briefly but clearly identify the error which has induced grumpiness.

*To prove:* The language of the regular expression $0^*1^*$ is, in fact, not DFA-recognizable.

*Proof.* Let $L$ be the language of $0^*1^*$. Assume there is some DFA $M$ with $n$ states that recognizes $L$. Let $x = 0^{n-1}11$. Clearly, $x \in L$ and $|x| \geq n$. Therefore according to the Pumping Lemma, we can split $x$ into three parts $u$, $v$ and $w$, such that $|uv| \leq n$, $|v| \geq 1$, and $uv^iw \in L$ for all natural numbers $i$. Let $|v| = n$. Since $|uv| \leq n$, it must be the case that $u = \epsilon$, and $v = 0^{n-1}1$. Then $uv^2w = 0^{n-1}10^{n-1}11$, which is clearly not in $L$. This contradicts our assumption that there is a DFA which recognizes the language.

Prove that $L = \{0^n1^m \mid m \geq n^2\}$ is not DFA-recognizable.

17. (a) In lecture, we proved that if $[a]_m$ is a unit, then $[a]_m^{\varphi(m)} = [1]$.

Use this to show that $a^{[b]_{\varphi(m)}} := [a^b]_m$ is well defined.

(b) Use Euler's theorem to prove that if $p$ is prime, then $[a]_p^p = [a]_p$ (whether $[a]_p$ is a unit or not).

18. Give a DFA that accepts strings in $\{0,1\}^*$ if and only if they contain at most one 0 **and** an even number of 1's. For each state, describe the strings that reach that state.

19. In this question we formalize the usual algorithm for adding numbers represented in base $b$. Let $\Sigma = \{0, 1, \ldots, b-1\}$.

(a) Give an *inductive* definition of the base $b$ interpretation function $n : \Sigma^* \to \mathbb{N}$. Check that if $b = 2$ then $n(110) = 6$.

(b) Consider the following inductive definition of a function

$$add : \Sigma^* \times \Sigma^* \times \{0,1\} \to \Sigma^*$$

given by

$$\begin{aligned}
add(\varepsilon, \varepsilon, c) &:= \varepsilon c \\
add(xd, \varepsilon, c) &:= add(x, \varepsilon, q)r &&\text{where } q = quot(d+c, b) \text{ and } r = rem(d+c, b) \\
add(\varepsilon, xd, c) &:= add(x, \varepsilon, q)r &&\text{where } q = quot(d+c, b) \text{ and } r = rem(d+c, b) \\
add(xd, ye, c) &:= add(x, y, q)r &&\text{where } q = quot(d+e+c, b) \text{ and } r = rem(d+e+c, b)
\end{aligned}$$

Note: you know this algorithm well; $c$ stands for "carry".

Prove that $n(add(x, y, c)) = n(x) + n(y) + c$. If multiple cases are substantially similar, you may say so instead of repeating the proof.

20. In this question, we focus on a subset $A$ of the set of regular expressions, given inductively by

$$r \in A ::= a \mid r_1 + r_2 \mid r_1r_2$$

(a) Give the inductive definition of the function $L : A \to 2^{\Sigma^*}$ that gives the language of a regular expression.

(b) Let $rev(x)$ denote the reverse of $x$ (for example, $rev("1011") = "1101"$). For an arbitrary language $L$, let $\overleftarrow{L} := \{rev(x) \mid x \in L\}$. Use **structural induction** to show that for all $r \in A$, $\overleftarrow{L(r)}$ is regular. You may use clearly stated facts about $rev(x)$ without proof, as long as they are true.

21. Give a regular expression that matches strings of 0's and 1's that, when interpreted as binary numbers, are *not* divisible by 3. *Hint:* start with an automaton with $Q = \mathbb{Z}_3$.

    *Note:* if you clearly indicate your process, partial credit may be given. If you simply write down an answer, credit will be awarded on an all-or-nothing basis.