452 FINAL-Spring 2024-Solutions

GO TO NEXT PAGE

1. (20 points-5 points for each part) Let the alphabet be $\Sigma = \{a, b\}$. Let $n \in \mathsf{N}$ and $n \geq 1000$.

   Recall that $\#_a(w)$ is the number of a's in w.

   Recall that $\#_b(w)$ is the number of b's in w. Let

$$L = \{w : \#_a(w) \equiv 0 \pmod{n} \text{ OR } \#_b(w) \equiv 0 \pmod{n+1}\}.$$

   Note that L is regular.

   (a) Give either the transition table OR draw the DFA for L. You may use DOT DOT DOT. Try to use as few states as possible.

   (b) How many states does the DFA in Part 1 have?

   (c) Give either the transition table OR draw the NFA for L. You may use DOT DOT DOT. Try to use as few states as possible.

   (d) How many states does the NFA in Part 3 have? (It should be smaller than the DFA.)

   BEGIN SOL

   Solutions omitted- this is really easy and you know how to do it.

   END SOL

2. (20 points) In this problem we use the WS1S conventions.

   (a) (10 points) Give a DFA for

   $$\{(x,y) : x+2=y\}.$$

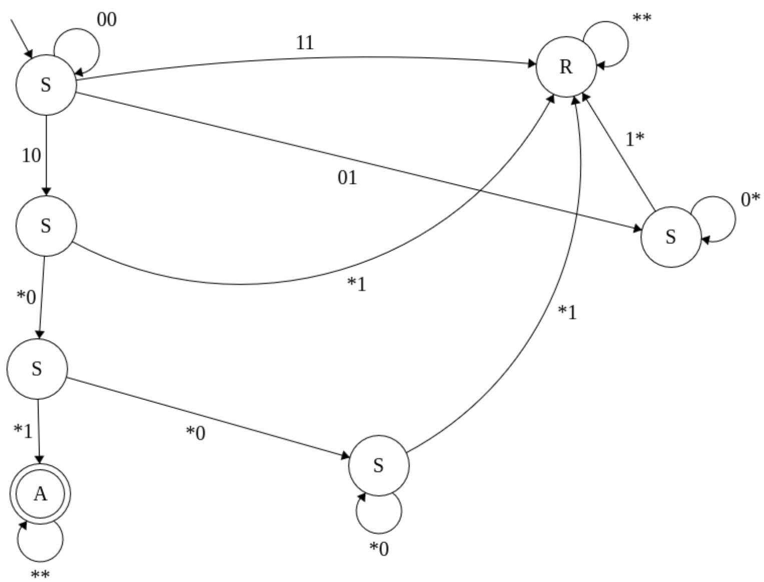
   How many states does your DFA have?
   (All states are labelled A for accept or R for reject or S for stupid.)
   comments.

   (b) (10 points) Let $n \in \mathsf{N}$. Consider the language

   $$\{(x,X) : x+n=y\}.$$

   How many states would a DFA for this language have? How may are ACCEPT? How many are REJECT? How many are STUPID?

BEGIN SOL



END SOL

3. (20 points) Show that the following set is Diophantine by giving the relevant polynomial.

$$\{x : x \text{ is a square and } x \not\equiv 3 \pmod 4\}.$$

BEGIN SOL

We do this in pieces. $\{x : x \text{ is a square}\}$ is

$$\{x : (\exists y)[x - y^2 = 0]\}.$$

$\{x \not\equiv 3 \pmod 4)\}$ is

$$\{x : (\exists y_1, y_2, y_3)[(x - 4y_1)(x - (4y_2 + 1))(x - (4y_3 + 2)) = 0]$$

So the final answer is

$$\{x : (\exists y, y_1, y_2, y_3)[(x-y^2)^2+((x-4y_1)(x-(4y_2+1))(x-(4y_3+2)))^2 = 0]$$

END SOL

4. (20 points) For this problem you will assume that P $\neq$ NP.

   Recall the following abbreviations.

   - REG be the set of Regular Languages.
   - CFG be the set of Context Free Languages.
   - P be the set of problems in polynomial time.
   - NP be ... YOU KNOW WHAT THAT IS.
   - DEC be the set of decidable sets
   - Σ_1 be the set of Σ_1 sets.

   Let CFG $-$ REG be languages that are CFG but not REG. Same with other differences.

   Recall that

$$\text{REG} \subset \text{CFG} \subset \text{P} \subset \text{NP} \subset \text{DEC} \subset \Sigma_1.$$

   For each of the sets on the next page say where they are, e.g., CFG $-$ REG. NO explanation required.

   $M_0, M_1, \ldots$ is a list of Turing Machines.

   Important 5 points for a right answer but -2 for a wrong answer. **For our own benefit do not guess!** I really mean it! **For your own benefit Do not guess!**

   Note Mercy rule: If you earn a negative score on this problem you will get a 0 on this problem.

(a) CLIQ_{1000}, the set of graphs that have a clique of size 1000. (Recall that a Clique of a graph $G = (V, E)$ is a set $U \subseteq V$ such that ever $x, y \in U$, $(x, y) \in E$.

(b) DNF_{1000}, satisfiable for formulas in DNF form, $D_1 \vee \cdots \vee D_m$ where each D_i is a $\wedge$ of exactly 1000 variables.

(c) The set of DFA's M such that M accepts at least one string.

(d) The set of CFG's G in Chomsky Normal Form such that G generates at least one string.

BEGIN SOL

THEY ARE ALL in P − CFG.

I TOLD YOU NOT TO GUESS.

SOME OF YOU GUESSED ANYWAY.

THAT WAS A MISTAKE.

HEADS UP FOR 2025 class: THE LAST PROBLEM WE DID NOT COVER IN CLASS. IF CURIOUS SEE SOLUTION HERE:

`https://cs.stackexchange.com/questions/92305/solving-the-emptiness-problem-fo`

END SOL

5. For this problem $\Sigma = \{0,1\}^*$.

Let $b, c \in \mathsf{N}$. X is a *(b,c)-Clyde Set* if the following happens.

- There exists $B \subseteq \Sigma^* \times \Sigma^*$ such that $(x,y) \in B$ can be computed in time $|x|^b$.
- $X = \{x : (\exists y, |y| = \lfloor c \log_2(|x|) \rfloor)[(x,y) \in B]\}$.

Show that if X is a (b,c)-Clyde Set then $X \in \mathrm{P}$. Give the time bound for your poly time algorithm for X in terms of c and b.

BEGIN SOL

(a) Input x. $|x| = n$.

(b) For all y, $|y| = c \lfloor \log_2(n) \rfloor$ test $(x,y) \in B$ (this takes time n^b). If so then output YES and HALT. If not then go to the next y.

(c) (If you got here then no y worked.) Output NO and HALT.

The loop goes through $2^{c\lfloor \log_2(n) \rfloor} \le n^c$ iterations. Each iteration takes n^b steps. Hence the run time is

$O(n^b n^c) = O(n^{b+c})$.

END SOL

Scratch Paper