Com S 331 Name KEY

Spring, 2019

Exam 1

This is a closed-book, closed-notes, no-devices, individual-effort examination. All answers should be explained, at least briefly. Please do all your work on these pages.

Remember: Blank space or a clear statement that you do not know how to do a problem receives 30% credit.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

- Design a DFA or Turing machine (in state diagram) that decides a given language L.
  $\forall x \in L$, machine accepts x
  $\forall x \notin L$, machine rejects x.
- Prove a language is not regular
- Find equivalence classes of a given language.
- Closure properties of Regular languages and Turing-decidable, Turing-acceptable languges.
- Cross-product construction.
  given DFA M_A deciding A, construct $M_{\bar{A}}$ deciding $\bar{A}$
- using standard Turing machines to simulate other machines.

Name KEY

1. (70 points) The <u>number of immediate bit repetitions</u> in a string $x \in \{0,1\}^*$ is the number $\#IBR(x)$ of bits in x that are the same as the bit immediately preceding them. Thus, for example,

$$\#IBR(1\underset{\uparrow}{1}010\underset{\uparrow}{0}\underset{\uparrow}{0}1) = 3,$$

because the 3 bits marked with arrows are the same as the bits immediately before them.

Assignment Project Exam Help

Design a DFA M such that

$$L(M) = \{x \in \{0,1\}^* \mid \#IBR(x) \text{ is } \underset{\sim}{\text{even}}\}.$$

https://powcoder.com

mimic the design of #(1,x) is even

<u>Idea</u>

① Analysis: one occurrence of 00 or 11 is counted as 1.

Add WeChat powcoder

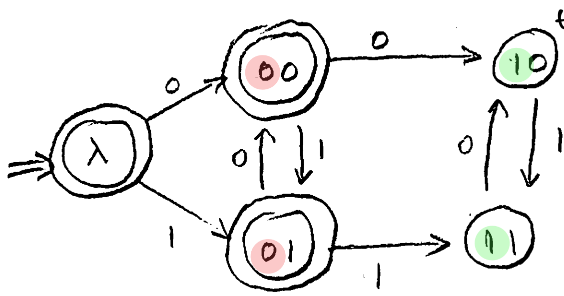
- state λ means that we haven't seen any bits yet.
- If $a, b \in \{0,1\}$, then state ab means that the number of immediate bit repetitions so far is equivalent to a mod 2 and that b is the last bit that we've seen.

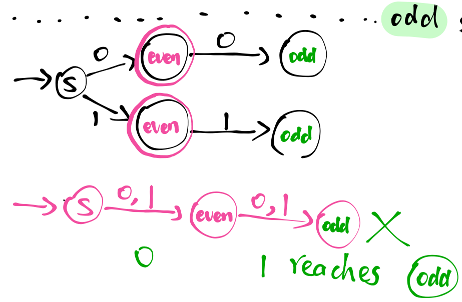
② start with what we want to accept and assign meanings to states.

This gives the 5-state DFA

the # of immediate bit repetition is even so far

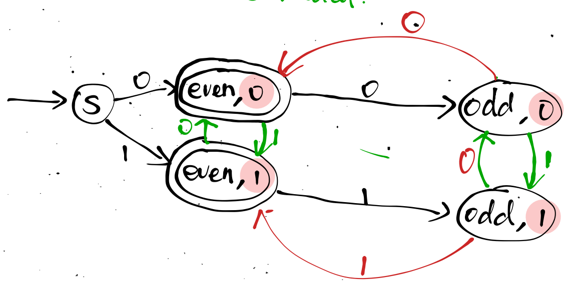
the . . . . odd so far





Name KEY

③ Finish the missing transitions, and maybe further refine the meanings of states, and add more states as needed.



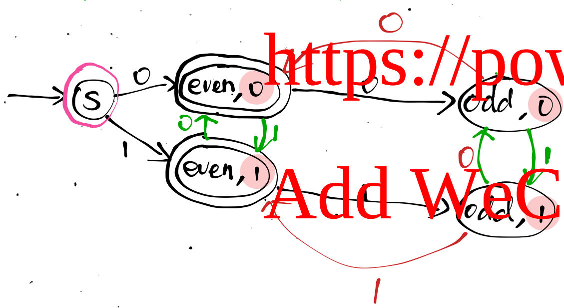
Since one occurrence of 00 or 11 is counted 1 for #IBR(X), the last digit read so far affect which state to it go to while reading the next 0 (or 1)

④ Mark the final states.

Always check the edge case λ:

if $\lambda \in L$, start state $\in F$

if $\lambda \notin L$, start state $\notin F$



Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Name KEY

2. (70 points) Design a DFA M such that

$$L(M) = \{x \in \{0,1\}^* \mid \underbrace{\#(0,x) \text{ is odd and } \#(1,x) \neq 2}_{\text{Independent requirements}}\},$$

where $\#(b,x)$ is the number of times that b appears in x.

Note that we want M to decide the language $\overset{A \cap \overline{B}}{A \sim B}$, where

$$A = \{x \in \{0,1\}^* \mid \#(0,x) \text{ is odd}\}$$

and

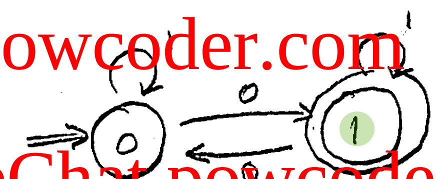
$$B = \{x \in \{0,1\}^* \mid \#(1,x) = 2\}.$$

Assignment Project Exam Help

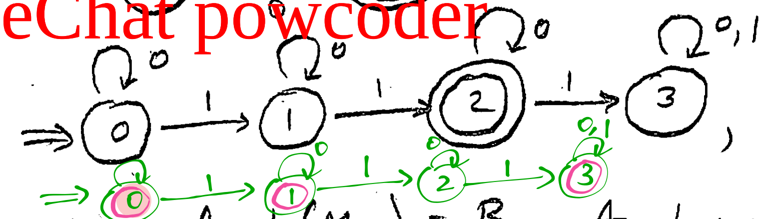
https://powcoder.com

Add WeChat powcoder

If

$M_A \equiv$ 

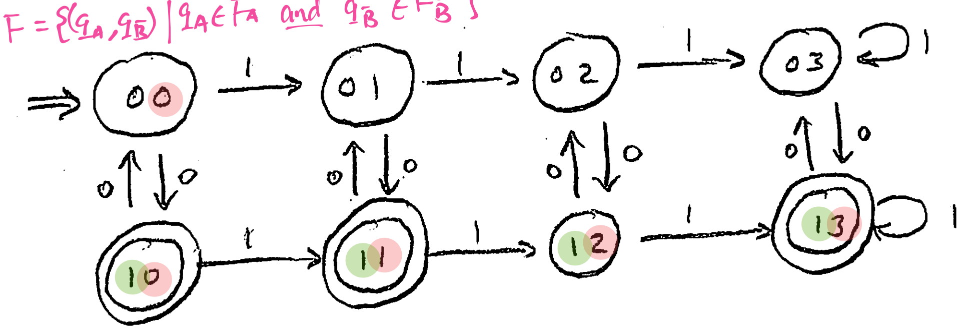
and

$M_B \equiv$ ,

$M_{\overline{B}} \equiv$

then $L(M_A) = A$ and $L(M_B) = B$. Applying the product construction to these gives

$S = (s_A, s_{\overline{B}})$, $F = \{(q_A, q_{\overline{B}}) \mid q_A \in F_A \text{ and } q_{\overline{B}} \in F_{\overline{B}}\}$

$M \equiv$ 

$$\delta((q_A, q_{\overline{B}}), a) = (\delta_A(q_A, a), \delta_{\overline{B}}(q_{\overline{B}}, a))$$

Name KEY

2. (70 points) Prove that the language

$$A = \{ 110^n 1 0^n \mid n \in \mathbb{N} \}$$

is not regular.

Proof: For each $n \in \mathbb{Z}^+$ (positive integer), the first extension of the string 110^n10 is 0^{n-1}, since $A^{(1)} \supseteq \{0^{n-1} \mid n \in \mathbb{Z}^+\}$, $|A^{(1)}| = \infty$. By ordinal extensions theorem, A is not regular.

Assignment Project Exam Help

① You don't need to get all strings in $A^{(1)}$. Showing it contains an infinite subset suffices.

https://powcoder.com

② How to get $A^{(1)}$?

In principle, check each string $x \in \Sigma^*$, and analyze how to make it in A by appending y to it. (make $xy \in A$ and y is the shortest such string).

Add WeChat powcoder

If you just want to get a subset of $A^{(1)}$, then analyze strings with a pattern. eg. $110^{n}10$ in terms of n, or some strings x such that y is in terms of n

③ If $A^{(1)}$ does not work, try $A^{(2)}$.

Name KEY

3. (70 points) Say that a 0 is isolated within a string $x \in \{0,1\}^*$ if it has 1s immediately before and after it. Hence, the first 0, and only the first 0 is isolated in the string

$x = 11010010.$ 011

leading 0's are not isolated.

Design a Turing machine M that accepts an input string $x \in \{0,1\}^*$ if an even number of 0s are isolated in x, and rejects x otherwise.

Assignment Project Exam Help

The key thing to note here is that the

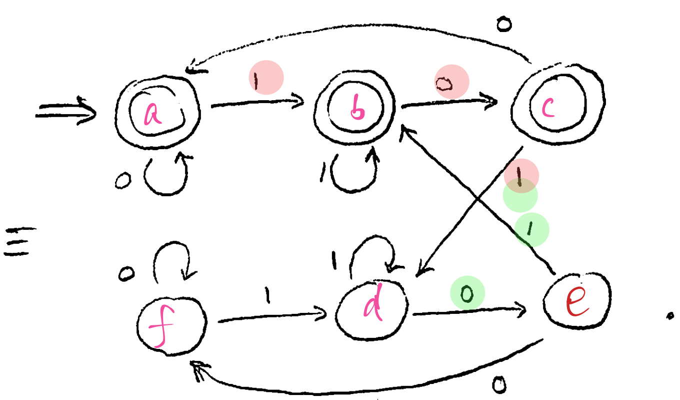
https://powcoder.com

language we want M to decide is regular,

decided by the DFA

Add WeChat powcoder

M' ≡

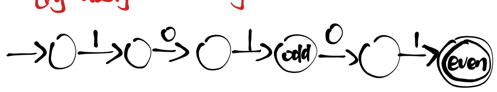


① Analysis:

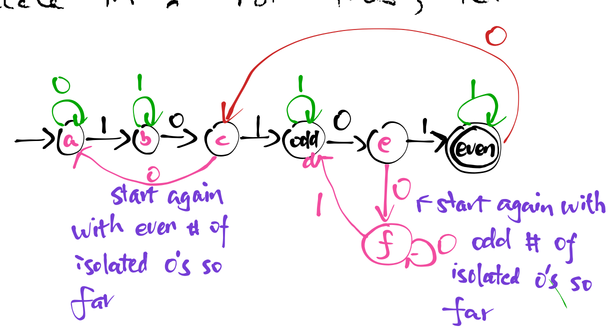
Count the # of substrings 101 allowing overlapping 1's.

Hence we need only design a TM M to simulate M'. For this, let

② start with what we want to accept, try assign meanings to states



③ Complete the missing transitions, add more states as needed, refine meaning of states if needed.



④ Assign final states. check λ always.

F = {a, b, c, even}

We can merge state b and even to minimize the DFA.

Even though a TM can simulate a DFA, if you are asked to give a Turing Machine, then you need to draw the state diagram following the semantics/rules of a TM.

Name KEY

$$M' = (Q', \{0,1\}, \delta', s', F').$$

Then

$$M = (Q, \{0,1\}, \Gamma, \vdash, \sqcup, \delta, s, t, r),$$

where

$$Q = Q' \cup \{s, t, r\},$$

$$\Gamma = \{0, 1, \vdash, \sqcup\},$$

and

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$$

is defined by

$\vdash \to \vdash, R$

$\to s$ This is how a TM starts working.

$$\delta(q,a) = \begin{cases} (q, a, R) & \text{if } q = s' \text{ and } a = \vdash \\ (\delta'(q,a), \underline{a}, R) & \text{if } q \in Q \text{ and } a \in \{0,1\} \\ (t, \underline{a}, R) & \text{if } \underline{a = \sqcup} \text{ and } q \in F \\ (r, \underline{a}, R) & \text{if } \underline{a = \sqcup} \text{ and } q \in Q \setminus F. \end{cases}$$

Do not change tape content.

Have reached the end of input string

(Other transitions cannot occur, so can be defined in any manner.)

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Name KEY

4. (70 points) In class we have defined the canonical equivalence relation $\equiv_A$ of a language $A \subseteq \Sigma^*$ by

$$x \equiv_A y \iff (\forall z \in \Sigma^*)[xz \in A \iff yz \in A]$$

for all $x, y \in \Sigma^*$.

Let

$$A = \{x \in \{0,1\}^* \mid \#(1,x) \text{ is a multiple of } 3\}.$$

Assignment Project Exam Help

(a) What are the $\equiv_A$-equivalence classes?

$\#(1,x) \bmod 3 = r \bmod 3$

They are the sets

https://powcoder.com

$$[r] = \{x \in \{0,1\}^* \mid \#(1,x) \equiv r \bmod 3\}$$

for each $r \in \{0,1,2\}$.

Add WeChat powcoder

(b) Prove that your answer to (a) is correct.

We do this by proving the following two things.

$\#(1,x) \bmod 3 = \#(1,y) \bmod 3$

(I) For all $x, y \in \{0,1\}^*$,

$$\#(1,x) \equiv \#(1,y) \bmod 3 \Rightarrow x \equiv_A y.$$

(II) $\lambda \not\equiv_A 1$, $\lambda \not\equiv_A 11$, and $1 \not\equiv_A 11$.

strings in different equivalence class are not A-equivalent

Name KEY

These suffice, because (I) tells us that all the elements of each $[r]$ are $\equiv_A$-equivalent, and (II) tells us that no two of the strings $\lambda \in [0]$, $1 \in [1]$, and $11 \in [2]$ are $\equiv_A$-equivalent.

Proof of (I). Assume that $\#(1,x) \equiv \#(1,y) \bmod 3$. It suffices to show that, for all $z \in \{0,1\}^*$, $\#(1,xz) \equiv \#(1,yz) \bmod 3$. (This suffices because it implies that $xz \in A \Leftrightarrow yz \in A$.) We do this by induction on z.

- The case $z = \lambda$ follows immediately from the assumption.
- Assume that $\#(1,xz) \equiv \#(1,yz) \bmod 3$, and let $b \in \{0,1\}$. Then $\#(1,xzb) \equiv_3 [\#(1,xz)+b]$ $\equiv_3 [\#(1,yz)+b] \equiv_3 [\#(1,yzb)]$, so the claim holds for zb. (Here we have used the shorthand $\equiv_3$ for congruence mod 3.) $\square$

extend by one more symbol

Proof of (II). Since $\lambda \in A$, $1 \notin A$, and $11 \notin A$, the string $z = \lambda$ testifies that $\lambda \not\equiv_A 1$ and $\lambda \not\equiv_A 11$. Since $11 \notin A$ and $111 \in A$, the string $z = 1$ testifies that $1 \not\equiv_A 11$. $\square$

$x \equiv_A y \Leftrightarrow \forall z \in \Sigma^* \, (xz \in A \Leftrightarrow yz \in A)$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Name KEY

5. (70 points) Let $A, B \subseteq \Sigma^*$.

Prove or disprove: If A is regular and $A \cap B$ is regular, then B is regular. always consider Σ^*, ϕ for disprove/counterexamples.

This is false. For example, let

$A = \emptyset$, when A is ϕ, $A \cap B$ being regular does not give any information about B.

$B = \{0^n 1^n \mid n \in \mathbb{N}\}$.

Then A is regular and $A \cap B = \emptyset$ is regular, but B is not regular.

Assignment Project Exam Help

A common wrong proof:

https://powcoder.com

Proof by contradiction:

Assume A is Regular and $A \cap B$ is regular, and B is non-regular.

Add WeChat powcoder

Using the closure property $A \cap B$ should be non-regular, which contradicts the assumption that $A \cap B$ is regular.

This proof used the closure property in a wrong way. Closure properties of regular languages can only be applied on regular languages, but B is non-regular.

$A \in Reg$ and $B \in Reg \Rightarrow A \cap B$ is Reg.

$A \in Reg$ and $B \in Non\text{-}Reg \Rightarrow$ we don't know about $A \cap B$.