HW1

Graded

Student

Vidit Dharmendra Pokharna

Total Points

35 / 40 pts

Question 1

Q1 — **9** / 10 pts

1.1 **Q1a** — **5** / 5 pts

- ✔ **– 0 pts** Correct
- **– 1 pt** Minor error
- **– 2.5 pts** Major error
- **– 5 pts** Completely incorrect/no response

1.2 **Q1b** — Resolved — **4** / 5 pts

- **– 0 pts** Correct
- ✔ **– 1 pt** Minor error
- **– 2.5 pts** Major error
- **– 5 pts** Completely incorrect/no response

You have the right idea, but you just need to specify the overall structure of the DIA (e.g. how you said it would be a union of DFAs is a DFA in part a)

Regrade Request — **Submitted on: Jan 23**

I believe my answer accurately addresses the core concept of the question by explaining how a DIA, with a state for each possible string and transitions based on language rules, can decide any language. This approach inherently implies an underlying structure similar to a union of DFAs, as each state and transition effectively represents a distinct DFA for each possible string. By ensuring that the DIA reaches a final state for any given string, indicating its presence or absence in the language, my answer demonstrates the decidability of all languages by a DIA.

I disagree. The correctness of your DIA is not communicated.

Reviewed on: Jan 28

Question 2

Q2 **17** / 18 pts

2.1 Q2a **6** / 6 pts

- ✔ **+ 2 pts** Correct NFA
- ✔ **+ 2 pts** Correct DFA
- ✔ **+ 2 pts** Correct Regex
- **+ 0 pts** Incorrect

2.2 Q2b Resolved **6** / 6 pts

- ✔ **− 0 pts** Correct
- **− 1 pt** Minor Error
- **− 3 pts** NFA is identical to DFA
- **− 3 pts** Missing DFA or NFA
- **− 3 pts** Major Error
- **− 6 pts** Completely incorrect/missing question

1 incorrect NFA

Regrade Request **Submitted on: Jan 23**

The design of my NFA includes an accepting state q3 that is reached after processing a string that, when converted to an integer in base 2, leaves a remainder of 3 when divided by 4. This is achieved by defining the transition states q0, q1, q2, and q3 to handle the binary string's remainder upon division by 4. Each state transition is designed to reflect the addition of a binary digit (0 or 1), corresponding to multiplying the current total by 2 (a left shift in binary terms) and adding the new digit's value, thus keeping track of the remainder modulo 4.

	0	1
remainder 0	0	1
remainder 1	2	3
remainder 2	0	1
remainder 3	2	3

Thus the NFA and DFA are accurate, and the NFA only does not have the transitions that end up in the same state where it started. Therefore, the NFA accurately recognizes the language specified, and I kindly request a reevaluation of my answer in light of this explanation. Additionally, I think you marked missing DFA or NFA, but the DFA is present on the next page. Thank you so much, and apologies for the long explanation.

Fixed!

Reviewed on: Jan 23

2.3 **Q2c** 5 / 6 pts

- **– 0 pts** Correct
- ✔ **– 1 pt** Minor Error
- **– 1 pt** Minor Error
- **– 3 pts** Major Error
- **– 6 pts** Missing DFA

q3 should be the accepting state

Question 3

Q3 Resolved 5 / 6 pts

- **– 0 pts** Correct
- ✔ **– 1 pt** Minor Error
- **– 2 pts** Missing details
- **– 3 pts** Major Error
- **– 5 pts** Not building a NFA/DFA
- **– 6 pts** Missing

This can create xaay which should not be accepted

Regrade Request **Submitted on: Jan 23**

I believe that my construction of the DFA D′ specifically addresses this issue. In my approach, the new states qa are designed to handle the insertion of a single symbol a, and the transitions from these states are intended to mimic the beginning of a new string processing. This means that once a symbol is added, the machine effectively restarts, thus preventing the acceptance of additional inserted symbols beyond the first one. Therefore, the construction ensures that only one symbol is added to any string from L, aligning with the definition of L′.

It can still add xaay, you can visit office hours if you need to!

Reviewed on: Jan 23

Question 4

Q4

Resolved **4 / 6 pts**

– 0 pts Correct

✔ **– 2 pts** Incorrect transition states/states/start state.

– 4 pts Did not show understanding of how to encode configurations as states.

– 1 pt slight error/Unlabeled Diagram

Regrade Request **Submitted on: Jan 23**

I don't understand why both -2 and -1 were applied. I believe I wrote the states properly, but just did not label the transitions. Shouldnt that only apply to one of the two incorrect boxes?

Regraded. There is no way to decipher if you have correct transitions, so I need to dock points for that on top of it being unlabeled.

Reviewed on: Jan 23

Questions assigned to the following page: 1.1 and 1.2

CS 4510 Automata and Complexity 1/10/2024

Homework 1: Regular Languages

Due:1/17/2024

You should submit a typeset or *neatly* written pdf on Gradescope. The grading TA should not have to struggle to read what you've written; if your handwriting is hard to decipher, you will be asked to typeset your future assignments. Four bonus points if you use LaTeX, and our template. You may collaborate with other students, but any written work should be your own.

1. One perspective of this class is that we are developing a theory of representation, or definability. For finite sets, this is easy, but it becomes more challenging for infinite languages. This class is really about the ability of different finite objects to understand and describe infinite sets.

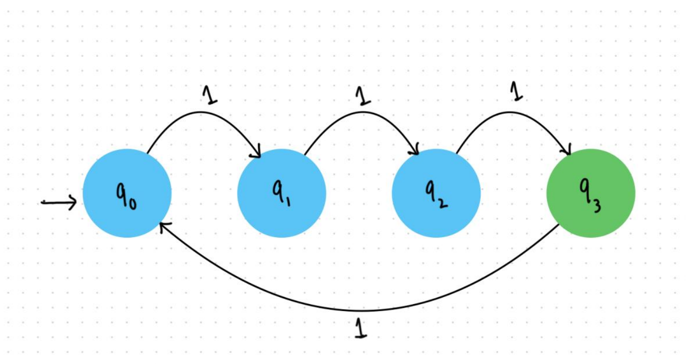
   (a) (5 points) Prove every finite language is regular.
   A finite language consists of a limited number of strings. For each string in this finite set, we can construct a DFA that recognizes just that string. By taking the union of these individual DFAs, we can create a DFA that recognizes the entire finite language. Since DFAs are used to recognize regular languages, this proves that every finite language is regular.

   (b) (5 points) Suppose we relax our definition of DFA too much. We define a DIA to be exactly like a DFA except $|Q| = |\mathbb{N}|$, and δ, F are adjusted appropriately. Prove that every language is decidable by a DIA.
   For any language, no matter how complex, we can theoretically construct a DIA that has a state for every possible string in the language and transitions according to the language's rules. This means that for any given string, the DIA will reach a final state that indicates whether the string is in the language, thus proving that every language is decidable by a DIA.

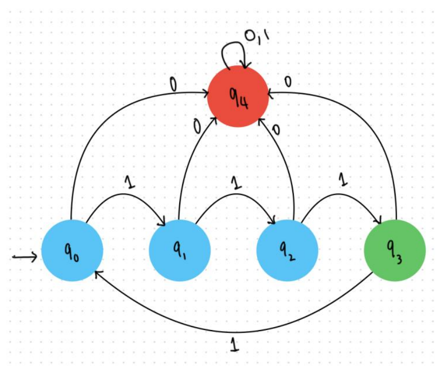
Questions assigned to the following page: 2.1, 2.2, and 2.3

2. Give an NFA, DFA, and regular expression for the following languages. Your solution doesn't have to be minimal, but it may assist the TA in grading. These may seem similar, but I promise they have different solutions. Here $int(w, 2)$ converts the binary string w to a number.

   (a) (6 points) NFA, DFA, Regex for $L_1 = \{1^n \mid n \equiv 3 \pmod 4\}$
   Green is an accepting state, red is a rejection state



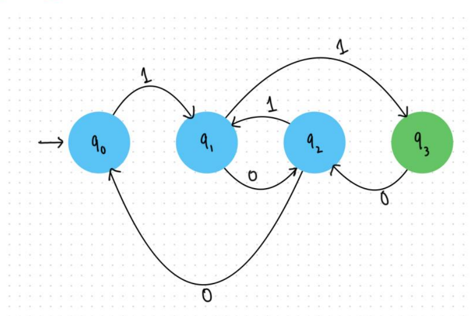
NFA:



DFA:

Regex: $\varepsilon \in 111(1111)^*$

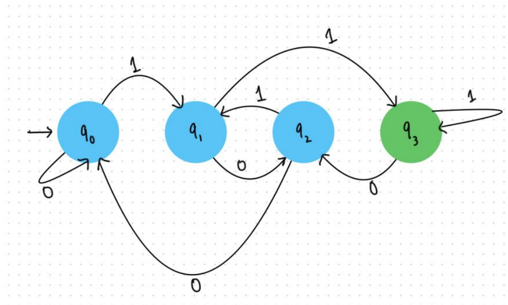
   (b) (6 points) NFA and DFA for $L_2 = \{w \in \Sigma^* \mid int(w, 2) \equiv 3 \pmod 4\}$
   Green is an accepting state



NFA:

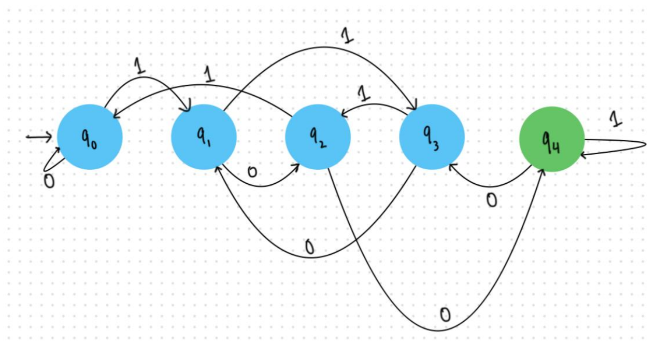
Questions assigned to the following page: 2.1, 2.2, and 2.3

DFA:



(c) (6 points) DFA for $L_3 = \{w \in \Sigma^* \mid int(w,2) \equiv 3 \pmod 5\}$
Green is an accepting state

DFA:



Question assigned to the following page: 3

3. (6 points) Let L be a language. Define $L' = \{xay \in \Sigma^* \mid xy \in L, a \in \Sigma\}$. We take the strings in L, and add a single symbol anywhere in the string, even possibly at the beginning or end. Prove that if L is regular, then so is L'.
   Since L is regular, there exists a DFA D that recognizes L. We can construct a DFA for L', calling it D', by modifying D as follows:

   1. For each state q in D, and for each symbol $a \in \Sigma$, add a transition from q to a new state q_a, which represents the insertion of a after the string recognized by q.

   2. Each new state q_a will have transitions similar to the initial state of D, as if we are starting to process a new string after adding a.

   3. If a state q in D is an accepting state, then for each q_a in D', add transitions back to the initial state of D for every symbol in Σ, since adding more symbols after xy would still keep the string in L'.

   By this construction, D' recognizes L', thus proving that if L is regular, then L' is also regular.

Question assigned to the following page: 4

4. (6 points) Consider the following poorly drawn gravity powered device. It has two input channels which may receive marbles and two output channels. There are three levers in the shown positions initialized as such. As a marble rolls down, it follows the path the level shows. After a marble hits a lever, the lever flips such that the next marble will go down the opposing channel. Consider a sequence of marbles like the letters of a word. Give a DFA to decide the language of sequences in which the last marble accepts. For example your DFA should accept a, aa, aaa but not $aaaa$. It should accept b, ba but not baa. (Hint: There are more than ten states)
Green is an accepting state, red is a rejecting state