CSE 311: Foundations of Computing I Winter 2026

Problem Set 6

Due: Monday, March 9th by 6:00 PM

Instructions

Write up carefully argued solutions to the following problems. Each solution should be clear enough that it can explain (to someone who does not already understand the answer) why it works.

Collaboration policy. You are required to submit your own solutions. You are allowed to discuss the homework with other students. However, the **write-up** must clearly be your own, and moreover, you must be able to explain your solution at any time. We reserve ourselves the right to ask you to explain your work at any time in the course of this class.

Solutions submission. Submit your solution via Gradescope. In particular:

– Each numbered task should be solved on its own page (or pages). Do not write your name on the individual pages. (Gradescope will handle that.)

– When you upload your pages, make sure each one is **properly rotated**. If not, you can use the Gradescope controls to turn them to the proper orientation.

– Follow the Gradescope prompt to **link tasks to pages**.

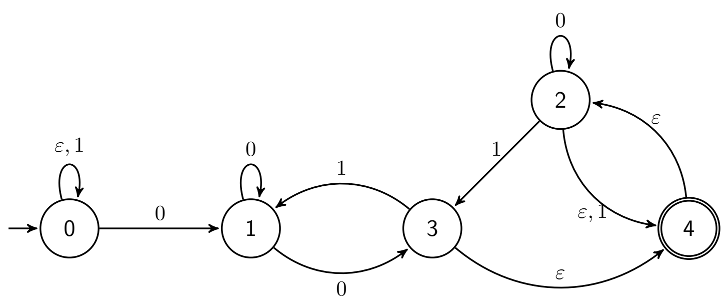
– You are not required to typeset your solution, but your submission must be **legible**. It is your responsibility to make sure solutions are readable — we will *not* grade unreadable write-ups.

– Extra practice problems are included at the bottom of the assignment. These will not be graded, so don't submit solutions to them.

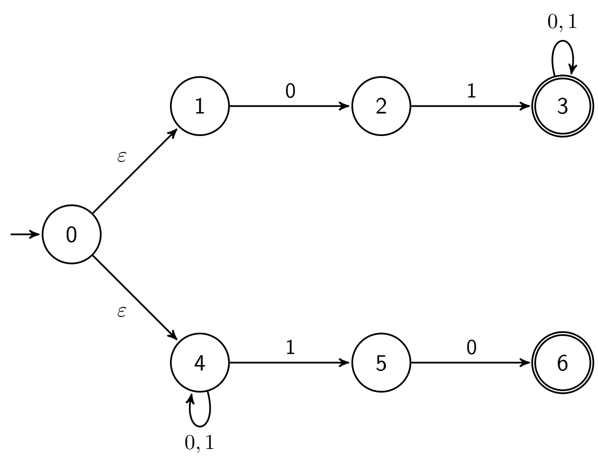
Task 1 – Final Study Mission [10 pts]

Use the algorithm from lecture to convert each of the following NFAs to DFAs. Label each DFA state with the set of NFA states it represents in the powerset construction.

a) The NFA below, which accepts strings starting with any number of 1's, immediately followed by "00", and then any binary string:



b) The NFA below, which accepts strings starting with "01"or ending in "10":



Submit and check your answers to all parts here:

http://grin.cs.washington.edu

Think carefully about your answer to make sure it is correct before submitting. While you have practically unlimited chances (999 allowed attempts) to submit a correct answer in this homework, you will only have one chance on the quiz.

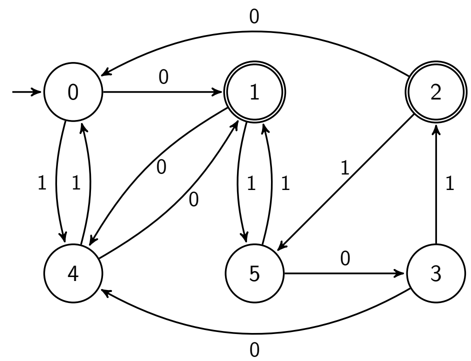
Important: Grin does not validate whether your state labels are correct. But do not skip them – correct state labels on DFAs will be required on quiz/exam with NFA-DFA conversion problems.

Task 2 – A Whole New Small Game [16 pts]

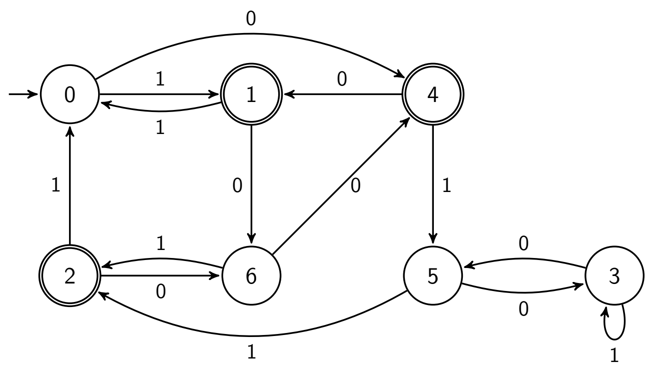
Use the algorithm from lecture to minimize the each of the following DFAs.

For each step of the algorithm, write down the groups of states, which group was split in that step and the reason for splitting that group. At the end, write down the minimized DFA, with each state named by the set of states of the original machine that it represents (e.g., “$\{B, C\}$”).

a) The following machine:



b) The following machine:



Submit and check your answers to all parts here:

http://grin.cs.washington.edu

Think carefully about your answer to make sure it is correct before submitting. While you have practically unlimited chances (999 allowed attempts) to submit a correct answer in this homework, you will only have one chance on the quiz.

Important: Grin does not validate whether your DFA is correctly minimized or whether your state labels are correct. You do not need to separately submit your algorithm steps or state labels for this homework. But do not skip these steps — the full algorithm run and correct state labels will be required on quiz/exam with DFA minimization problems.

Task 3 – The Great Expression [10 pts]

Use the algorithm from lecture to convert each of the following regular expressions into NFAs that accept the same language. You should **precisely** follow the construction from lecture.

a) $10(01 \cup 1)^*11 \cup 0110$

(Note: We translated this into a CFG in HW5 Task 2(a). Here, we translate it into an NFA.)

b) $(1(01 \cup 10)^*)^*$

Task 4 – Puedo ir(regular) al baño [20 pts]

Use the method described in lecture to prove that each of the following languages is **not regular**. You may need to get creative when constructing your prefix set and suffix!

a) The set of all strings with properly balanced parentheses and brackets. For example, the following are balanced: "([])()([]())", "([(())])", "(()([]))". The following are unbalanced: "()([])(", "([)(])", "](((()](". Here is a CFG which recognizes this language: $\mathbf{S} \to (S)|[S]|SS|\varepsilon$.

Side note: Practically speaking, this is a basic language that any syntax checker (e.g. for Java, Python, even arithmetic expressions with parentheses, etc.) should be able to "recognize" to determine if your code is syntactically invalid! Whatever model of computation which does this must recognize more than regular languages.

b) All binary strings in the set $\{0^m1^n : m, n \in \mathbb{N} \text{ and } m \mid n\}$ (that is, all binary strings in which the number of 0's divides the number of 1's). *Hint: Think about the prime numbers, of which there are infinitely many.*

c) All binary strings in the set $\{0^m1^n : m, n \in \mathbb{N} \text{ and } m \leqslant n^2\}$ (that is, all binary strings in which the number of 0's is less than or equal to the square of the number of 1's).

Task 5 – Last Task of the Quarter; Count Yourself Blist! [10 pts]

In this task, we'll investigate various sets of lists whose elements are in $\{0, 1\}$. For each part, give a proof of whether the given set is countable or uncountable. In your justification, you should use one of the following techniques, discussed in lecture:

- **Enumeration.** If the elements of the given set can be listed in order, e.g. by labeling them with $0, 1, 2, \ldots$, then the given set is countable.
- **Subset.** Show that the given set is a subset of another set, known to be countable. Then the given set is also countable.
- **Dovetailing.** Show that the given set can be partitioned into countably many subsets, each of which is finite. Putting the subsets in order, the elements can then be enumerated in order by group, so the given set is countable.
- **Injection.** Show that there is a one-to-one function $f : A \to B$. Then A is no larger than B.
- **Surjection.** Show that there is an onto function $f : A \to B$. Then A is no smaller than B.
- **Diagonalization.** Suppose for the sake of contradiction that the given set is countable. Then, all of its elements can be listed in order. Use the list to construct an element that is not in the list. By contradiction, the given set is uncountable.

a) The set A of all lists whose elements are in $\{0, 1\}$. Recall that these lists can be arbitrarily long, but each particular list has finite length.

b) The set B of all lists with odd length and whose elements are in $\{0, 1\}$.

c) The set C of all "infinite lists" whose elements are in $\{0, 1\}$. As an analogy, you can think of one of these lists as an endless stream of 0s and 1s, such as a `BitInputStream` in Java.

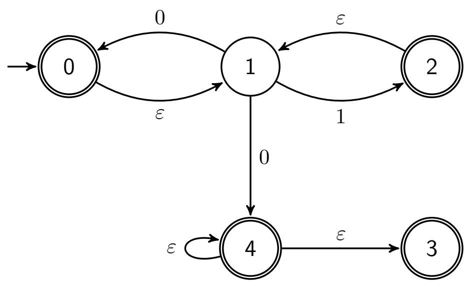
Task 6 – Extra Credit: Rice and Dice [0 pts]

In the lecture slides, it was mentioned that the problem $\mathrm{E\small{QUIV}}(P, Q)$ is undecidable. That is, there is no program that takes as input $(\mathrm{C\small{ODE}}(P), \mathrm{C\small{ODE}}(Q))$ and outputs $\mathrm{T\small{RUE}}$ if $P(x)$ and $Q(x)$ have the same behavior on every input x, and $\mathrm{F\small{ALSE}}$ otherwise. Prove that this problem is undecidable by reducing from the halting problem. You can use Java syntax to describe your programs.

Task 7 – Optional Practice Problems (Ungraded) [0 pts]

The problems below are **optional practice problems** that are **not required and will not be graded**. They are provided to help you practice; you do not need to submit solutions to these problems. The difficulty of these problems has not been vetted as thoroughly, so don't fret if they seem challenging.

a) Convert the following NFA to a DFA.



b) Determine whether set of functions $f : \mathbb{N} \to \{0, 1, 2, 3\}$ is countable or uncountable. Prove your answer.

c) Determine whether $\mathbb{N} \times \mathbb{R}$ is countable or uncountable. Prove your answer.

d) Determine whether the set of odd integers is countable or uncountable. Prove your answer.

e) Prove that the set of all lists *over the integers* is countable. That is, the elements of each list can be any integer in $\mathbb{Z}$, which is not finite! *Hint:* First, prove that for each $k \geqslant 0$, the set of all lists of length k over $\mathbb{Z}$ is countable. Then, use the fact proved in Task 3d of Section 6.

f) Determine whether the set of rooted binary trees with no data is countable or uncountable. Prove your answer.