Homework 8: Finite Automata; Fundamentals Review

Due date: Wednesday March 8th at 10 PM
If you work with others (and you should!), remember to follow the collaboration policy outlined in the syllabus.

In general, you are graded on both the clarity and accuracy of your work. Your solution should be clear enough that someone in the class who had not seen the problem before would understand it.

We sometimes describe approximately how long our explanations are. These are intended to help you understand approximately how much detail we are expecting. You are allowed to have longer explanations, but explanations significantly longer than necessary may receive deductions.

To help with formatting of English proofs, we've published a style guide on the website containing some tips. **Unless otherwise noted in a problem, all proofs must be English proofs.** You should not have numbered steps (e.g., you should not be doing an inference proof.)

Finally, be sure to read the grading guidelines for more information on what we're looking for.

This is the last homework :D. Late days work the same on this homework as they have with others (e.g., you can submit up to Saturday at 10 PM by using three late days), but keep in mind that you need time to study for the exam. Unused late days do not have any effect on your grade.

We will not be able to return feedback on HW8 until after the final.

1. Like Divides, but more complicated [10 points]

Define the following relation R on the positive integers. $(a, b) \in R$ if and only if a/b is an even integer.

So $(2, 4) \notin R$, because $2/4 = 1/2$ is not an integer.
$(30, 5) \in R$ because $30/5 = 6 = 2 \cdot 3$.

(a) Prove that R is a transitive relation. Be sure to explicitly introduce arbitrary variables as arbitrary. [7 points]

(b) Prove that R is not a symmetric relation (Hint: since 'symmetric' is a $\forall$ statement, you're proving an $\exists$ statement here!) [3 points]

2. Com-pair-isons [15 points]

Let $\mathcal{A} = \{S : S \subseteq \mathbb{N} \wedge |S| = 2\}$. I.e., an element of $\mathcal{A}$ is a set containing two natural numbers.

We define the relation $\preceq$ on $\mathcal{A}$ as follows.

Let $X = \{x_1, x_2\}$ with $x_1 < x_2$
Let $Y = \{y_1, y_2\}$ with $y_1 < y_2$

We will say $X \preceq Y$ if and only if $x_1 \leq y_1$ and $x_2 \leq y_2$.

In English, we say $X \preceq Y$ if and only if the smaller element of X is at most the smaller element of Y **and** the larger element of X is at most the larger element of Y.

For example:

- $\{5, 10\} \preceq \{7, 12\}$
- $\{5, 10\} \not\preceq \{4, 20\}$ because $5 > 4$
- $\{1, 2\} \preceq \{1, 3\}$
- $\{10, 5\} \preceq \{7, 12\}$ (we compare elements based on their sizes, not what order they are listed in).

(a) Prove that $\preceq$ is antisymmetric on $\mathcal{A}$. You may not use proof by contradiction for this problem. Remember to introduce arbitrary variables as arbitrary. [7 points]
Hint: Remember that we had two equivalent definitions of antisymmetry. The definition $\forall x \forall y([(x, y) \in R \wedge (y, x) \in R] \rightarrow x = y)$ is probably going to give you a cleaner proof than the other definition in this problem.

(b) Convince yourself that $\preceq$ is a partial order on $\mathcal{A}$ (you just showed antisymmetry; you should convince yourself of reflextivity and transitivity). You do not have to write anything for this part [0 point]

(c) A partial order is called a **total order** if for all $X, Y \in \mathcal{A}$, ($X \preceq Y$ or $Y \preceq X$). Is $\preceq$ a total order on $\mathcal{A}$? Prove your answer. [4 points]

(d) Imagine we change the "and" in the definition of $\preceq$ to an "or." More formally, define a new relation R on the set $\mathcal{A}$, where $(X, Y) \in R$ if and only if $x_1 \leq y_1$ or $x_2 \leq y_2$ (as before $X = \{x_1, x_2\}, Y = \{y_1, y_2\}, x_1 < x_2$, and $y_1 < y_2$). The new relation R is not transitive. Prove it is not transitive. [4 points]

3. Build DFAs (Online) [15 points]

For each of the following languages, construct a DFA that accepts exactly the given set of strings. You should submit (and check!) your answers online at https://grin.cs.washington.edu/

Think carefully before entering your DFA; you have a limited number of guesses. Because these are auto-graded, we will not award partial credit.

(a) Binary strings with at least four 1s.

(b) Binary strings that have at least one 1 and an even number of 0s.

(c) Binary strings such that none of their runs of 0s have an even length $\geq$2.
A "run" of 0s is a string of consecutive 0s with edges at the start of the string, end of the string, or adjacent to a 1. "00" contains exactly one run of 0s (of length 2).
"000010" contains two runs of 0s (one of length 4, the other of length 1).
For example, "0001100" is not accepted because the last two characters are an even-length run of 0s. But "01110001" is accepted because the first run of 0s has length 1, and the second run of 0s has length 3. "11111" is also accepted because it contains no run of 0s.

4. Build NFAs (Online) [15 points]

For each of the following languages, construct an **NFA** that accepts exactly the given set of strings. You should submit (and check!) your answers online at https://grin.cs.washington.edu/

Think carefully before entering your NFA; you have a limited number of guesses. Because these are auto-graded, we will not award partial credit.

(a) The set of binary strings that contain 00 **and** do not contain 11. [7 points]

(b) The set of binary strings that contain 00 **or** do not contain 11. [8 points]

5. Primal Translations [10 points]

We define a **twin prime** as a prime number which is separated from another prime number by a distance of only 2. For example, 3 and 5 are twin primes. Formally, we say an integer x is prime if there exists a prime number y such that $x - y = 2$ or $y - x = 2$.

(a) Define `TWIN-PRIME(x)` in predicate logic by translating the formal definition above. Let your domain of discourse be integers, and let `PRIME(p)` be the predicate which is true iff p is prime. You may use standard arithmetic notation (including $+, -, =, \leq$, etc.) in all parts of this problem [2 points]

(b) The **twin prime conjecture** states that there are infinitely many twin primes (it is not known to the 311 staff if the conjecture is true). We can rephrase the conjecture as follows: For any integer N, no matter how large, there always exists a twin prime larger than N. Write in predicate logic a proposition which is true if and only if the **twin prime conjecture** holds. Your answer should use your predicate from part (a). [2 points]

(c) Negate the translation you wrote in part (b). Show your work, but leave your answers as symbols. You don't have to name rules as you use them. Your final answer must have negations applied only to single predicates. You do not need to simplify to "take advantage of domain restriction." [3 points]

(d) Translate your answer from (c) into English. You may **NOT** use the words *finite* or *infinite* or any variation of them. [3 points]

6. Proof By Contradiction [11 points]

In this problem you'll show

Claim: The sum of a rational number and an irrational number is always irrational.

(a) Translate the claim above into predicate logic. Let your domain of discourse be all real numbers. Use the predicate `Rational` (note that "x is irrational" is $\neg$ `Rational`(x), so you don't need another predicate; you can use $+$ for summation).

(b) Write the negation of the claim above in predicate logic.

(c) Now, write an (English) proof of the claim. You must use proof by contradiction for this problem.

7. Buggy [4 points]

Recall that we say a set is closed under an operation if whenever you apply the operation to members of the set you get another member of the set.

For example, "integers are closed under addition" means that whenever you add two integers together, you get another integer. "Positive integers are not closed under subtraction" means that there are positive integers x, y such that $x - y$ is not a positive integer.

Consider the following claim. "Irregular languages are closed under intersection."

In symbolic logic, we can write the claim as:

$$\forall L_1 \forall L_2([\neg \texttt{regular}(L_1) \wedge \neg \texttt{regular}(L_2)] \rightarrow \neg \texttt{regular}(L_1 \cap L_2))$$

(a) Your friend writes the following proof, aiming to show the claim above. Their proof is incorrect. Identify the bug. State both the step where the biggest bug is, and why it is incorrect. [4 points]

Proof:
① We rephrase the original claim as "if L_1 and L_2 are irregular then $L_1 \cap L_2$ is irregular."

② We argue by contradiction. Suppose that if L_1 and L_2 are irregular, then $L_1 \cap L_2$ is regular.
③ Consider $L_1 = \{a^n b^n c^n : n \geq 0\}$ and $L_2 = \{a^n b^n c^n : n \geq 0\}$ (i.e. the same language).
④ We recall from class that L_1 (and thus L_2) is irregular.
⑤ Applying the statement in step ②, we get that $L_1 \cap L_2 = L_1$ must be regular.
⑥ But we know L_1 is irregular! That's a contradiction.
⑦ Thus we have irregular languages are closed under intersection.

8. Feedback [1 point]

Please keep track of how much time you spend on this homework and answer the following questions. This can help us calibrate future assignments and future iterations of the course, and can help you identify which areas are most challenging for you.

- How many hours did you spend working on this assignment (excluding any extra credit questions, if applicable)? Report your estimate to the nearest hour.
- Which problem did you spend the most time on?
- Any other feedback for us?

9. Extra Credit: Going Back to the Well [0 point]

In class, we'll use proof by contradiction as a method of showing languages are not regular. Another common way to prove languages are not regular is "The Pumping Lemma."

Formally, the pumping lemma says: If L is a regular language, then there exists an integer p such that for all $w \in L$, if $\texttt{len}(w) \geq p$, then there is a way to divide w into substrings x, y, z such that:

- $w = xyz$ (i.e. x, y, z break w into pieces)
- $\texttt{len}(xy) \leq p$
- $\texttt{len}(y) \geq 1$
- $\forall i \in \mathbb{N}$, $xy^i z \in L$

The lemma says you can break any string of L into pieces, where the "middle" piece is length at least 1, such that you can "pump" the string w by inserting extra copies of y into the string while still having the new string also be in L

For some intuition: roughly, p corresponds to the number of states in a hypothetical DFA for L. Once a string w has more than p characters, you have to have a cycle when the machine processes w, but then you could go around the cycle as many times as you want (including 0 times) and still end up in the same accepting state at the end. So some y in the string (what's read when you go around the cycle) could be duplicated any number of times and still be accepted. The difficulty of using the lemma to prove irregularity is we don't know which part of the string the cycle would be in (we're doing proof by contradiction – the language is irrecular, so there isn't even a DFA in real life!) so these proofs often have cases based on all the places y could be in the string.

Use the pumping lemma to show $\{a^{311} b^n c^m : n = 311 + m\}$ is not regular. The hardest thing about using the pumping lemma is getting the quantifiers right (there are a lot of them...) make sure you're declaring and using the arbitrary variables as arbitrary, and saying what the existential variables are. Usually this proof is done by contradiction (suppose L is regular and use the pumping lemma's guarantee to derive a contradiction) or contrapositive (show the conclusion of the pumping lemma does not hold, and apply the contrapositive to get that the language is not regular). You should feel free to look online or in the textbook for an example proof or two to get a sense of how they usually go.

10. Extra Credit: Finishing Strong [0 point]

Let L be a regular language over the alphabet Σ. Define $\mathsf{ends}(L) = \{v \in \Sigma^* : \exists u \in \Sigma^*(uv \in L)\}$. Prove that $\mathsf{ends}(L)$ is regular.