CM50260 Foundations of Computation Coursework: Written Questions

...

November 20, 2023

Contents

1 Question 5

Given $L \subseteq \{0,1\}^*$ is a regular language, there exists a finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ recognizing L.

(a) Let's prove that the language L^{-0}, formed by taking strings from L and removing one 0 from each, is regular. For example, if L contains 010, then 10 would be in L^{-0}.

Since L is regular, there's a finite automaton that recognizes it. We create a new automaton that works like the original but with a modification: it has the ability to ignore exactly one 0 in the string.

We need to ensure our new automaton skips exactly one 0, not more. So, we design it to change its behavior after skipping a 0, ensuring it won't skip any more.

This modified automaton can now recognize L^{-0}, proving that L^{-0} is regular.

Consider the construction of a modified NFA M' alongside the original NFA M that recognizes language L. A workable construction is one where we create a clone of the original NFA M, each state in the clone corresponds to a state in M. Importantly, we introduce a mechanism to indicate whether 0 has been deleted or not at each state; whenever 0 can be consumed in the original NFA M, we establish transitions from the original via the empty string (ε) to the same state on the clone. This allows us to simulate the behavior of M while enabling the deletion of precisely one 0 when required. It's important to not that once transitioned to the clone there's no going back to the original.

A workable construction formulated more precisely:

- States: $Q' = Q \times \{0,1\}$, where the second component indicates if 0 has been deleted.
- Transitions: For each $\delta(q,a)$ in M:
  - $\delta'((q,0),a) = (\delta(q,a),0)$ for $a \in \{1\}$ (normal transition).
  - $\delta'((q,0),a) = (\delta(q,a),1)$ for $a \in \{0\}$ (delete 0).
  - $\delta'((q,1),a) = (\delta(q,a),1)$ for $a \in \{0,1\}$ (after deletion).
- Start State: $(q_0, 0)$.
- Accept States: $F' = \{(q,1) \mid q \in F\}$.

(b) Let's now prove that the language L^{+0}, created by taking strings from L and inserting an extra 0 somewhere in each string, is regular. For example, if 10 is in L, then 100, 010, and 100 would all be in L^{+0}.

Since L is regular, there's a finite automaton that recognizes it. We design a new automaton that simulates the original but can add an extra 0 at any point in the string.

The new automaton is set up so that it can only add this extra 0 once. After adding a 0, it continues processing the rest of the string normally.

This new automaton can recognize any string from L with an extra 0 added. This shows that L^{+0} is regular.

Consider the construction of a modified NFA M' alongside the original NFA M that recognizes language L. A workable construction is one where we create a clone of the original NFA M, each state in the clone corresponds to a state in M. Importantly, we introduce a mechanism to indicate whether 0 has been inserted or not at each state; we establish transitions from the original via the empty string (ε) to the same state on the clone. This allows us to simulate the behavior of M while enabling the addition of precisely one 0 when required. It's important to not that once transitioned to the clone there's no going back to the original.

A workable construction formulated more precisely:

- States: $Q' = Q \times \{0, 1\}$, with the second component indicating if 0 has been inserted.
- Transitions: For each $\delta(q, a)$ in M:
  - $\delta'((q, 0), a) = (\delta(q, a), 0)$ for $a \in \{1\}$ (before insertion).
  - $\delta'((q, 0), \varepsilon) = (q, 1)$ (insert 0).
  - $\delta'((q, 1), a) = (\delta(q, a), 1)$ for $a \in \{0, 1\}$ (after insertion).
- Start State: $(q_0, 0)$.
- Accept States: $F' = \{(q, 1) \mid q \in F\}$.

2 Question 6

Question: For each of the following statements, say whether it is true or false, and justify your answer.

(a) The language $\{a^{2n} \mid n \geq 0\} \subseteq \{a\}^*$ is regular.

Answer: False.

We prove that the language $L = \{a^{2^n} \mid n \geq 0\}$ is not regular using the Pumping Lemma.

Assume, for the sake of contradiction, that L is regular. By the Pumping Lemma, there exists a pumping length p such that any string s in L with $|s| \geq p$ can be divided into three parts $s = xyz$, satisfying the following conditions:

1 For each $i \geq 0$, the string $xy^iz \in L$.

2 $|y| > 0$.

3 $|xy| \leq p$.

Consider the string $s = a^{2^m}$, where m is chosen such that $2^m \geq p$. According to the Pumping Lemma, s can be split into xyz where $|xy| \leq p$ and $|y| > 0$. Let $y = a^k$ for some $k > 0$.

Pumping y to create a new string xy^2z results in a string of length $2^m + k$. Since $k \leq p$ and $k > 0$, and given that $2^m \geq p$, we have $2^m < 2^m + k < 2^{m+1}$. This implies that $2^m + k$ is not a power of 2, and hence, xy^2z does not conform to the form a^{2^n} for any n.

Additionally, considering further pumping of y, such as in xy^3z or xy^5z, leads to strings of lengths $2^m + 3k$ and $2^m + 5p$, respectively. These lengths are not powers of 2 either,.

Therefore, there exist strings formed by pumping y that do not belong to L. This contradicts the Pumping Lemma's condition that all such strings must be in L. Consequently, this contradiction implies that the assumption of L being regular is false. Thus, the language $L = \{a^{2^n} \mid n \geq 0\}$ is not regular.

(b) The language $\{0^m1^n \mid m \neq n\} \subseteq \{0, 1\}^*$ is regular.

Answer: False.

We prove that the language $L = \{0^m1^n \mid m \neq n\}$ is not regular using the Pumping Lemma.

Assume, for the sake of contradiction, that L is regular. By the Pumping Lemma, there exists a pumping length p such that any string w in L with $|w| \geq p$ can be divided into three parts $w = xyz$, satisfying the following conditions:

1 For each $i \geq 0$, the string $xy^iz \in L$.

2 $|y| > 0$.

3 $|xy| \leq p$.

Consider the string $w = 0^p 1^{p+p!}$, where $|w| \geq p$ and hence the Pumping Lemma applies. According to the Pumping Lemma, w can be split into xyz where $|xy| \leq p$ and $|y| > 0$. Let $y = 0^a$ where $0 < a \leq p$, without loss of generality.

By the Pumping Lemma, $xy^iz \in L$ for all $i \geq 0$. Pump y to create a new string by choosing $i = 1 + \frac{p!}{a}$. Note that i is an integer since $p!$ is divisible by any a, where $0 < a \leq p$. The resulting string is $xy^iz = 0^{p-a}0^{a(1+\frac{p!}{a})}1^{p+p!} = 0^{p+p!}1^{p+p!}$.

The new string $0^{p+p!}1^{p+p!}$ is not in L as it does not satisfy the condition $m \neq n$. This contradicts the assumption that L is regular. Hence, the language $L = \{0^m 1^n \mid m \neq n\}$ is not regular.

(c) All subsets of the language $\{0^n 1^n \mid n \in \mathbb{N}\} \subseteq \{0,1\}^*$ are nonregular.

Answer: False.

While the language $\{0^n 1^n \mid n \in \mathbb{N}\}$ itself is not regular, some of its subsets are regular.

Consider these examples:

(a) Finite sets, like $\{0^1 1^1\}$ or $\{0^2 1^2, 0^3 1^3\}$, are regular. Finite sets of strings can be recognized by a finite automaton and described by a regular expression.

(b) The empty set $\emptyset$ is a regular language. As far as set theory is concerned the empty set is a subset of the language under consideration.

(c) The set with just the empty string $\{\varepsilon\}$, where ε is the string of length 0, is regular. This is a subset for the language under consideration if we take $\mathbb{N}$ to start from 0.

A language is regular if a finite automaton can recognize it. Simple or finite languages, even as subsets of a nonregular language, can be recognized by such automata.

(d) All infinite subsets of the language $\{0^n 1^n \mid n \in \mathbb{N}\} \subseteq \{0,1\}^*$ are nonregular.

Answer: True.

The language $\{0^n 1^n | n \in \mathbb{N}\}$, consists of strings where there are n 0s followed by n 1s, for any natural number n. This language is nonregular because it requires a memory of how many 0s there are to ensure an equal number of 1s follow, which is something a finite state automaton cannot do. This can also be proved using the pumping lemma.

Any infinite subset that maintains the structure where the number of 0s must equal the number of 1s, regardless of what n is, will also be nonregular. For example, a subset where n is an even number or a prime number still requires counting and matching, so such subsets would be nonregular.

In exploring the regularity of infinite subsets of the language $\{0^n 1^n | n \in \mathbb{N}\}$, I encounter a challenge in rigorously proving that all infinite subsets retain the inherent counting requirement of the original language. My current approach, though not entirely rigorous, hinges on the fact that in any infinite subset, the strings must have the potential to grow indefinitely. Given this unbounded growth, it seems plausible to apply the pumping lemma to demonstrate that such subsets are nonregular. The lemma suggests that if a language is regular, there exists a way to pump parts of its strings without violating the language's rules. However, for these subsets, any attempt to pump the strings would disrupt the delicate balance of 0s and 1s, leading to strings that fall outside the subset. While this reasoning aligns with the characteristics of nonregular languages, it still lacks the rigor needed for a definitive proof.