1 Expressiveness

1.1 Finite Languages

Finite Languages

Definition 1. A language is finite if it has finitely many strings.

Example 2. $\{0, 1, 00, 10\}$ is a finite language, however, $(00 \cup 11)^*$ is not.

Proposition 3. *If L is finite then L is regular.*

Proof. Let $L = \{w_1, w_2, \dots w_n\}$. Then $R = w_1 \cup w_2 \cup \cdots \cup w_n$ is a regular expression defining L. □

1.2 Non-Regular Languages

Are all languages regular?

Proposition 4. *The language $L_{\text{eq}} = \{w \in \{0,1\}^* \mid w$ has an equal number of 0s and 1s$\}$ is not regular.*

Proof? No DFA has enough states to keep track of the number of 0s and 1s it might see. □

Above is a weak argument because $E = \{w \in \{0,1\}^* | w$ has an equal number of 01 and 10 substrings$\}$ is regular!

---

2 Proving Non-regularity

2.1 Lower Bound Method

Proving Non-Regularity

Proposition 5. *The language $L_{\text{eq}} = \{w \in \{0,1\}^* \mid w$ has an equal number of 0s and 1s$\}$ is not regular.*

Proof. Suppose (for contradiction) L_{eq} is recognized by DFA $M = (Q, \{0,1\}, \delta, q_0, F)$.

Let $W = \{0\}^*$. For any $w_1, w_2 \in W$ with $w_1 \neq w_2$, $\hat{\delta}_M(q_0, w_1) \neq \hat{\delta}_M(q_0, w_2)$. Let us observe that if the claim holds, then M has infinitely many states, and so is not a finite automaton, giving the desired contradiction.
Claim: For any $w_1, w_2 \in W$ with $w_1 \neq w_2$, $\hat{\delta}_M(q_0, w_1) \neq \hat{\delta}_M(q_0, w_2)$.
Proof of Claim: Suppose (for contradiction) there is w_1 and w_2 such that $\hat{\delta}_M(q_0, w_1) = \hat{\delta}_M(q_0, w_2) = \{q\}$. Without loss of generality we can assume that $w_1 = 0^i$ and $w_2 = 0^j$, with $i < j$. Then, $\hat{\delta}_M(q_0, w_1 1^i = 0^i 1^i) = \hat{\delta}_M(q, 1^i) = \hat{\delta}_M(q_0, w_2 1^i = 0^j 1^i)$. Thus, M either accepts both $0^i 1^i$ and $0^j 1^i$, or neither. But $0^i 1^i \in L_{\text{eq}}$ but $0^j 1^i \notin L_{\text{eq}}$, contradicting the assumption that M recognizes L_{eq}. □

Example I

Proposition 6. $L_{0n1n} = \{0^n1^n \mid n \geq 0\}$ *is not regular.*

Proof. Suppose L_{0n1n} is regular and is recignized by DFA $M = (Q, \{0, 1\}, \delta, q_0, F)$.

- Let $W = \{0\}^*$. For any $w_1, w_2 \in W$ with $w_1 \neq w_2$, $\hat{\delta}_M(q_0, w_1) \neq \hat{\delta}_M(q_0, w_2)$.
  - Suppose (for contradiction) $\hat{\delta}_M(q_0, w_1) = \hat{\delta}_M(q_0, w_2) = \{q\}$, where $w_1 = 0^i$ and $w_2 = 0^j$, with $i < j$.
  - Then, $\hat{\delta}_M(q_0, w_1 1^i = 0^i 1^i) = \hat{\delta}_M(q, 1^i) = \hat{\delta}_M(q_0, w_2 1^i = 0^j 1^i)$.
  - But $0^i 1^i \in L_{0n1n}$ but $0^j 1^i \notin L_{0n1n}$, contradicting the assumption that M recognizes L_{0n1n}.
- Because of the claim, M has infinitely many states, and so is not a finite automaton! □

2.2 Using Closure Properties

Example II
Closure Properties

Proposition 7. $L_{anban} = \{a^nba^n \mid n \geq 0\}$ *is not regular.*

Proof. We could prove this proposition the way we demonstrated the other languages to be not regular. We could show that for any two (different) strings in $W = \{a\}^*b$, any DFA M recognizing L_{anban} must go to different states, thus showing that M cannot have finitely many states. However, we choose to demonstrate a different technique to prove non-regularity of languages. This relies on closure properties.

The idea behind the proof is to show that if we had an automaton M accepting L_{anban} then we can construct an automaton accepting $L_{0n1n} = \{0^n1^n \mid n \geq 0\}$. But since we know L_{0n1n} is not regular, we can conclude L_{anban} cannot be regular. This is the idea of *reductions*, where one shows that one problem (namely, L_{0n1n} in this case) can be solved using a modified version of an algorithm solving another problem (L_{anban} in this case), which plays a central role in showing impossibility results. We will see more examples of this as the course goes on.

How do we show that a DFA recognizing L_{anban} can be modified to obtain a DFA for L_{0n1n}? We will use closure properties for this. More formally, we will show that by applying a sequence of "regularity preserving" operations to L_{anban} we can get L_{0n1n}. Then, since L_{0n1n} is not regular, L_{anban} cannot be regular. The proof is as follows.

- Consider homomorphism $h_1 : \{a, b, c\}^* \to \{a, b\}^*$ defined as $h_1(a) = a$, $h_1(b) = b$, $h_1(c) = a$.
  - $L_1 = h_1^{-1}(L_{anban}) = \{(a \cup c)^n b (a \cup c)^n \mid n \geq 0\}$
- Let $L_2 = L_1 \cap \mathbf{L}(a^*bc^*) = \{a^nbc^n \mid n \geq 0\}$
- Homomorphism $h_2 : \{a, b, c\}^* \to \{0, 1\}^*$ is defined as $h_2(a) = 0$, $h_2(b) = \epsilon$, and $h_2(c) = 1$.

- $L_3 = h_2(L_2) = \{0^n 1^n \mid n \geq 0\} = L_{0n1n}$

- Now if L_{anban} is regular then so are L_1, L_2, L_3, and L_{0n1n}. But L_{0n1n} is not regular, and so L is not regular. □

Example III

Proposition 8. $L_{\text{neq}} = \{w_1 w_2 \mid w_1, w_2 \in \{0,1\}^*,\ |w_1| = |w_2|,\ \textit{but}\ w_1 \neq w_2\}$ *is not regular.*

Proof. As before there are two ways to show this result. First we can show that if M with initial state q_0 is a DFA recognizing L_{ww}, then on any two (different) strings in $W = \{0,1\}^*$, M must be in different states. This is because, suppose on $x, y \in \{0,1\}^*$, $\hat{\delta}_M(q_0, x) = \hat{\delta}_(q_0, y)$ then $\hat{\delta}_M(q_0, xy) = \hat{\delta}_M(q_0, yy)$. But $xy \in L_{\text{neq}}$ and $yy \notin L_{\text{neq}}$, giving us the desired contradiction. Thus, M must have infinitely many states (since $|W|$ is infinite), contradicting the fact that M is a finite automaton.

Another proof uses closure properties. Consider the following sequence of languages.

- Let $h_1 : \{0, 1, \#\}^* \to \{0,1\}^*$ be a homomorphism such that $h_1(0) = 1$, $h_1(1) = 1$ and $h_1(\#) = \epsilon$. Consider

  $$L_1 = h_1^{-1}(L_{\text{neq}}) \cap \mathbf{L}((0 \cup 1)^* \# (0 \cup 1)^*) = \{w_1 \# w_2 | w_1, w_2 \in \{0,1\}^*, |w_1| + |w_2| \text{ is even, and } w_1 \neq w_2\}$$

- $L_2 = \{0, 1, \#\}^* \setminus L_1$

- $L_3 = L_1 \cap \mathbf{L}((0 \cup 1)^* \# (0 \cup 1)^*) \cap ((\{0,1,\#\}\{0,1,\#\})^* \{0,1,\#\}) = \{w_1 \# w_2 \mid w_1, w_2 \in \{0,1\}^*,\ \text{and}\ w_1 = w_2\}$

- Let $h_2 : \{0, 1, \bar{0}, \bar{1}, \#\}^* \to \{0, 1, \#\}^*$ be a homomorphism where $h_2(0) = h_2(\bar{0}) = 0$, $h_2(1) = h_2(\bar{1}) = 1$ and $h_2(\#) = \#$. Let $L_4 = h_2^{-1}(L_3) \cap \mathbf{L}((\bar{0} \cup \bar{1})^* \# (0 \cup 1)^*)$. Observe that

  $$L_4 = \{w_1 \# w_2 \mid w_1 \in \{\bar{0}, \bar{1}\}^*,\ w_2 \in \{0,1\}^* \text{ and } w_1 \text{ is same as } w_2 \text{ except for the bars}\}$$

- Let $h_3 : \{0, 1, \bar{0}, \bar{1}, \#\}^* \to \{0,1\}^*$ be the homomorphism where $h_3(\bar{0}) = 0$, $h_3(\bar{1}) = h_3(\#) = h_3(1) = \epsilon$, and $h_3(0) = 1$. Observe that $h_3(L_4) = L_{0n1n}$.

Due the closure properties of the regular languages, if L_{neq} is regular, then so are $L_1, L_2, L_3, L_4, h_3(L_4 = L_{0n1n}$. But since L_{0n1n} is not regular, L_{neq} is not regular. □

Lessons on Expressivity

Limits of Finite Memory
Finite automata cannot

- "keep track of counts": e.g., L_{0n1n} not regular.
- "compare far apart pieces" of the input: e.g. L_{xx} not regular.
- do "computations that require it to look at global properties" of the input.
