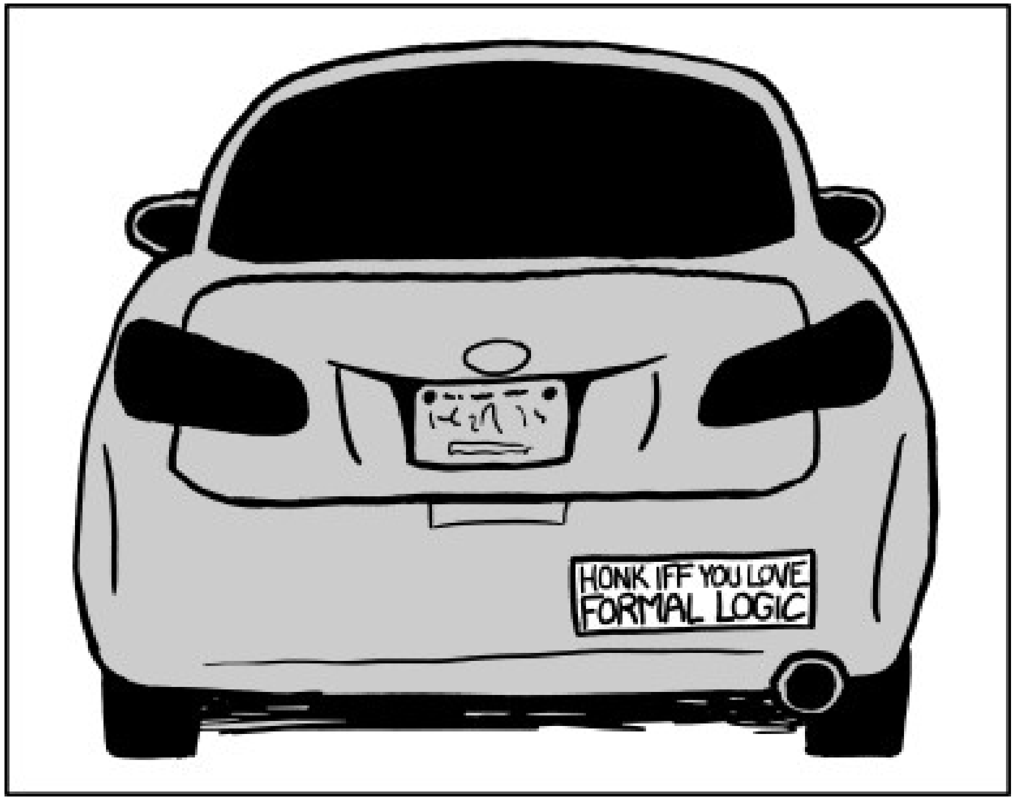


Final Exam Review

CSE 311: Foundations of Computing I
Lecture 24

Translations

- All dogs are cute
- There is only one dog who is the best dog

Translations

- All dogs are cute

$$\forall x(dog(x) \rightarrow cute(x))$$

- There is only one dog who is the best dog

$$\exists x(dog(x) \wedge best(x) \wedge \forall y((dog(y) \wedge best(y)) \rightarrow x = y))$$

Set Theory

- Prove that $(A \cap B) \cup (A \setminus C) \subseteq A$.

Set Theory

- Prove that (A ∩ B) ∪ (A \ C) ⊆ A.

Let $x \in (A \cap B) \cup (A \ \setminus C)$ be arbitrary. By definition of union, $x \in (A \cap B)$ or x $\in (A \ \setminus C)$. We will use cases:

1. If $x \in (A \cap B)$, then $x \in A$ by definition of intersection.
2. If x $\in (A \ \setminus C)$, then $x \in A$ and $x \notin C$, so clearly $x \in A$

In both cases $x \in A$, thus if $x \in (A \cap B) \cup (A \ \setminus C)$, $x \in A$.

Since x was arbitrary, (A ∩ B) ∪ (A \ C) ⊆ A

Number Theory

Let p be a prime number > 3. Show that $p \equiv_6 1$ or $p \equiv_6 5$

Number Theory

Let p be a prime number > 3. Show that $p \equiv_6 \mathbf{1}$ or $p \equiv_6 \mathbf{5}$

Let p be arbitrary. We know that by the division theorem, $p = 6q + r$ for some q and r. this means that $p \equiv_6 r$. Consider the following cases:

$p \equiv_6 \mathbf{0}$: This means that 6|p, making p not prime

$p \equiv_6 2$: This means that p is even, and since p>3, it must be not prime

$p \equiv_6 3$: This means 3|p, making p not prime

$p \equiv_6 4$: This means that p is even and not prime

Therefore, r = 1 or 5, so $p \equiv_6 \mathbf{1}$ or $p \equiv_6 \mathbf{5}$

Since p was arbitrary, the claim holds

Structural Induction

We define the set **Trees** as follows:

Basis Elements: for any $x \in \mathbb{Z}$, Leaf(x) $\in$ **Trees**.

Recursive Step: for any $x \in \mathbb{Z}$, if $L, R \in$ **Trees**, then Branch(x, L, R) $\in$ **Trees**.

This defines a set of *binary trees* just like those defined in lecture except that we now include data in the nodes.

The following functions count the number of each type of node in the tree:

$$\begin{aligned} \text{leaves}(\text{Leaf}(x)) &= 1 && \text{for any } x \in \mathbb{Z} \\ \text{leaves}(\text{Branch}(x, L, R) &= \text{leaves}(L) + \text{leaves}(R) && \text{for any } x \in \mathbb{Z} \text{ and } L, R \in \textbf{Trees} \end{aligned}$$

$$\begin{aligned} \text{branches}(\text{Leaf}(x)) &= 0 && \text{for any } x \in \mathbb{Z} \\ \text{branches}(\text{Branch}(x, L, R) &= 1 + \text{branches}(L) + \text{branches}(R) && \text{for any } x \in \mathbb{Z} \text{ and } L, R \in \textbf{Trees} \end{aligned}$$

Prove that, for any $T \in$ **Trees**, we have leaves(T) = branches(T) + 1.

Structural Induction

Let P(t) mean that "leaves(t) = branches(t) + 1". We will prove this by structural induction for all $t \in T$.

Base Case: Leaf(x). Leaves(leaf(x)) = 1, and branches(leaf(x)) + 1 =1, so P(leaf(x)) holds.

Inductive Hypothisis: Suppose P(L), P(R) for arbitrary L, R $\in T$.

Inductive Step: let x be an arbitrary integer

Leaves(branch(x, L, R)) = Leaves(L) + Leaves(R)

= Branches(L) + Branches(R)+1 + 1 (by IH)

= Branches(Branch(x, L, R)) + 1

Which shows P(branch(x, L, R))

Therefore we have shown P(t) for all $t \in T$ by structural induction.

Regex

- Binary Strings with an even number of 1s

- Binary strings with an odd number of 1s

Regex

- Binary Strings with an even number of 1s

(0*10*10*)*0*

- Binary strings with an odd number of 1s

(0*10*10*)*0*1(0*10*10*)*0*

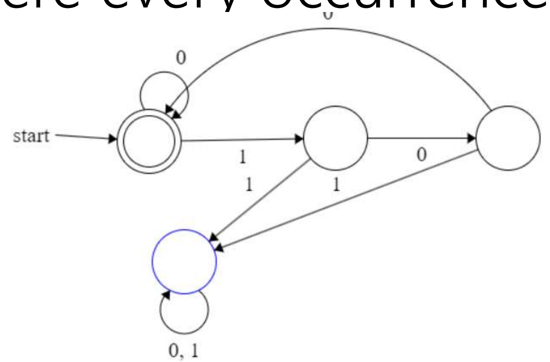
DFA, NFA

- Binary Strings where every occurrence of a 1 is followed by 00

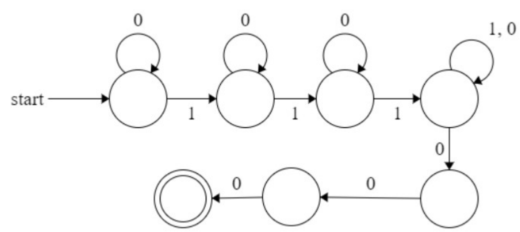
- NFA: Binary Strings with at least 3 1s and ends with 000

DFA, NFA

- Binary Strings where every occurrence of a 1 is followed by 00

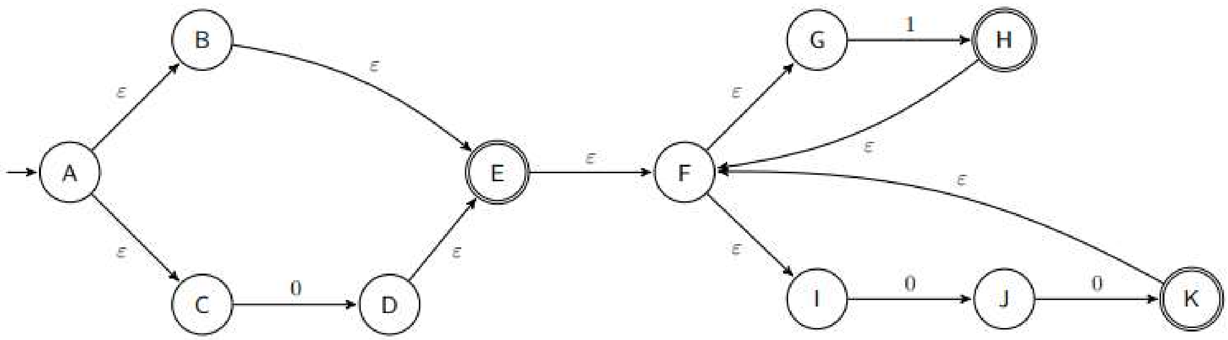


- NFA: Binary Strings with at least 3 1s and ends with 000

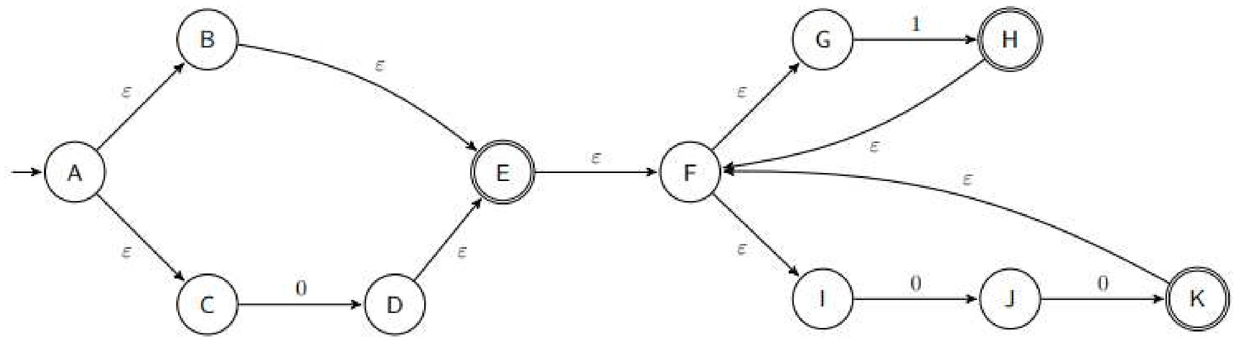


NFA to DFA

Convert the NFA below, which we get by applying the construction described in class[1] to the regular expression $(\varepsilon \cup 0)(1 \cup 00)^*$:



NFA to DFA



Irregularity

- Prove that the language 1^a01^a is irregular

Irregularity

- Prove that the language $L = 1^a01^a$ is irregular

Suppose there is some DFA called M that accepts L. Consider S = $\{1^a0\}$. Since there are finitely many states in M and infinitely many strings in S, we know there exists strings 0^a1 and 0^b1 that go to the same state in M where a != b. Now consider appending 0^a onto both strings. Since 0^a1 and 0^b1 end in the same state, $0^a10^a$and 0^a10^b go to the same state let's call q. q needs to be accepting since $0^a10^a \in L$, but it needs to also be rejecting since $0^a10^b \notin L$ since a != b. This is a contradiction.

Thus no DFA recognizes L