Deterministic Finite Automata are useful (but sometimes a pain to write)

A new model: Nondeterministic Finite Automata (NFAs)

“Lambda ($\lambda$) transitions”: don’t consume input. They just change state.

Key feature: An NFA does not need to be deterministic (not all transitions need to be defined).

Key feature: An NFA can be in more than one state at a time.

WHAT DOES THIS NFA DO?

NFAs are a model of computation:

Data: (bit)strings

Behavior: Take every transition that applies. Any transition that is not a $\lambda$ transition consumes input. The machine stops when the input is empty or when it cannot make progress. The machine accepts if it is in at least one accepting state.

Let’s practice

Draw these NFAs:

(1) $L = \{w \mid w$ is 101 or $w$ contains an even number of 0s$\}

(2) $L = \{w \mid w$ starts and ends with 01$\}$

Some definitions

An alphabet ($\Sigma$, pronounced “Sigma”) is a set of characters.

A string is a sequence of characters

A language is a set of strings

An automaton is a computational model that accepts or “recognizes” a language.
Regular expressions (REs)

A shorthand notation for a language.

Regular expressions support these operations:

- concatenation
- union (+)
- Kleene star (*)

Let's practice

Write regular expressions that describe these languages:

1. \( L = \{w | w \text{ starts and ends with } a \text{ and has 0 or more bs in the middle}\} \)
2. \( L = \{w | w \text{ is 010 or the length of } w \text{ is even}\} \)

Equivalent models of computation

What makes for a “good” DFA?

DISCUSSION: What do we mean by “good”?

Distinguishability: state ~ fate

**distinguishability of strings:** Two strings are *distinguishable* (with respect to a given language \( L \)) if we can append a common suffix \( z \), causing a DFA for \( L \) to accept one string and reject the other.

**pairwise-distinguishability of a set of strings:** A set of strings is *pairwise-distinguishable* (with respect to a given language \( L \)) if every unique pair of strings in the set is distinguishable with respect to \( L \).

**distinguishability theorem:** If a set of \( n \) strings is pairwise-distinguishable (with respect to a given language \( L \)), then a DFA that accepts \( L \) must contain at least \( n \) states.

Let’s practice

Use the distinguishability theorem to prove that any DFA that accepts the following language requires at least 5 states: \( L = \{w | \text{the third bit in } w \text{ is a 1}\} \).

Irregularity

Myhill-Nerode theorem: A language \( L \) is regular if and only if we can define a set \( S \) of pairwise distinguishable strings such that |\( S \)| (i.e., the size of \( S \)) is finite.

Next time: more models of computation + “unsolvable” problems