Last time:
- Nondeterminism
- NFA → DFA
- Closure under • and *
- Regular expressions → finite automata

Today:
- Finite automata → regular expressions
- Proving languages aren’t regular
- Context free grammars

We start counting Check-ins today.
Review your email from Canvas.
**Recall Theorem:** If \( R \) is a regular expr and \( A = L(R) \) then \( A \) is regular

**Proof:** Conversion \( R \rightarrow \text{NFA } M \)

**Today’s Theorem:** If \( A \) is regular then \( A = L(R) \) for some regular expr \( R \)

**Proof:** Give conversion \( \text{DFA } M \rightarrow R \)

WAIT! Need new concept first.
Defn: A Generalized Nondeterministic Finite Automaton (GNFA) is similar to an NFA, but allows regular expressions as transition labels.

For convenience we will assume:
- One accept state, separate from the start state
- One arrow from each state to each state, except
  a) only exiting the start state
  b) only entering the accept state

We can easily modify a GNFA to have this special form.
**Lemma:** Every GNFA $G$ has an equivalent regular expression $R$

**Proof:** By induction on the number of states $k$ of $G$

**Basis** ($k = 2$):

Let $R = r$

**Induction step** ($k > 2$): Assume Lemma true for $k - 1$ states and prove for $k$ states

**IDEA:** Convert $k$-state GNFA to equivalent $(k - 1)$-state GNFA

![Diagram](attachment:image.png)
$k$-state GNFA $\rightarrow (k-1)$-state GNFA

1. Pick any state $x$ except the start and accept states.
2. Remove $x$.
3. Repair the damage by recovering all paths that went through $x$.
4. Make the indicated change for each pair of states $q_i, q_j$.

Thus DFAs and regular expressions are equivalent in power.
Non-Regular Languages

How do we show a language is not regular?
- Remember, to show a language is regular, we give a DFA.
- To show a language is not regular, we must give a proof.
- It is not enough to say that you couldn’t find a DFA for it, therefore the language isn’t regular.

Two examples: Here \( \Sigma = \{0,1\} \).

1. Let \( B = \{w | w \text{ has equal numbers of 0s and 1s}\} \)
   Intuition: \( B \) is not regular because DFAs cannot count unboundedly.

2. Let \( C = \{w | w \text{ has equal numbers of 01 and 10 substrings}\} \)
   Intuition: \( C \) is not regular because DFAs cannot count unboundedly. However \( C \) is regular!

Moral: You need to give a proof.
Method for Proving Non-regularity

**Pumping Lemma:** For every regular language $A$, there is a number $p$ (the “pumping length”) such that if $s \in A$ and $|s| \geq p$ then $s = xyz$ where

1) $xy^i z \in A$ for all $i \geq 0$ \hspace{1cm} $y^i = yy \cdots y$
2) $y \neq \varepsilon$
3) $|xy| \leq p$

Informally: $A$ is regular $\rightarrow$ every long string in $A$ can be pumped and the result stays in $A$.

**Proof:** Let DFA $M$ recognize $A$. Let $p$ be the number of states in $M$. Pick $s \in A$ where $|s| \geq p$. 

\[
\begin{align*}
  s &= x \quad y \quad z \\
  M \text{ will repeat a state } q_j \text{ when reading } s \text{ because } s \text{ is so long.}
\end{align*}
\]

\[
\begin{align*}
  x &\quad y \quad y \quad z \\
  q_j &\quad q_j \quad q_j
\end{align*}
\]

The path that $M$ follows when reading $s$. Is also accepted
Example 1 of Proving Non-regularity

Let \( D = \{0^k1^k \mid k \geq 0\} \)

Show: \( D \) is not regular

Proof by Contradiction:
Assume (to get a contradiction) that \( D \) is regular.
The pumping lemma gives \( p \) as above. Let \( s = 0^p1^p \in D \).
Pumping lemma says that can divide \( s = xyz \) satisfying the 3 conditions.

But \( xyz \) has excess 0s and thus \( xyz \notin D \) contradicting the pumping lemma.
Therefore our assumption (\( D \) is regular) is false. We conclude that \( D \) is not regular.

**Pumping Lemma:** For every regular language \( A \), there is a \( p \)
such that if \( s \in A \) and \( |s| \geq p \) then \( s = xyz \) where

1) \( xy^iz \in A \) for all \( i \geq 0 \) \( \quad y^i = yy \cdots y \)
2) \( y \neq \epsilon \)
3) \( |xy| \leq p \)
Example 2 of Proving Non-regularity

**Pumping Lemma:** For every regular language $A$, there is a $p$ such that if $s \in A$ and $|s| \geq p$ then $s = xyz$ where

1) $xy^iz \in A$ for all $i \geq 0$ \hspace{1cm} $y^i = yy \cdots y$

2) $y \neq \varepsilon$

3) $|xy| \leq p$

Let $F = \{ww \mid w \in \Sigma^*\}$. Say $\Sigma^* = \{0,1\}$.

**Show:** $F$ is not regular

**Proof by Contradiction:**

Assume (for contradiction) that $D$ is regular.

The pumping lemma gives $p$ as above. Need to choose $s \in F$. Which $s$?

Try $s = 0^p0^p \in F$.

Try $s = 0^p10^p1 \in F$. Show cannot be pumped $s = xyz$ satisfying the 3 conditions. $xyyz \notin F$ Contradiction! Therefore $F$ is not regular.
Example 3 of Proving Non-regularity

**Variant:** Combine closure properties with the Pumping Lemma.

Let \( B = \{w | w \text{ has equal numbers of } 0\text{s and } 1\text{s}\} \)

**Show:** \( B \) is not regular

**Proof by Contradiction:**
Assume (for contradiction) that \( B \) is regular.

We know that \( 0^*1^* \) is regular so \( B \cap 0^*1^* \) is regular (closure under intersection).

But \( D = B \cap 0^*1^* \) and we already showed \( D \) is not regular. Contradiction!

Therefore our assumption is false, so \( B \) is not regular.
Context Free Grammars

\[ G_1 \]
\[
S \rightarrow 0S1 \\
S \rightarrow R \\
R \rightarrow \varepsilon
\]

(Substitution) Rules

**Rule:** Variable \( \rightarrow \) string of variables and terminals

**Variables:** Symbols appearing on left-hand side of rule

**Terminals:** Symbols appearing only on right-hand side

**Start Variable:** Top left symbol

**Grammars generate strings**

1. Write down start variable
2. Replace any variable according to a rule
   - Repeat until only terminals remain
3. Result is the generated string
4. \( L(G) \) is the language of all generated strings.

\[ L(G_1) = \{0^k1^k \mid k \geq 0\} \]

In \( G_1 \):

- 3 rules
- R,S
- 0,1

Example of \( G_1 \) generating a string
Quick review of today

1. Conversion of DFAs to regular expressions
   Summary: DFAs, NFAs, regular expressions are all equivalent

2. Proving languages not regular by using the pumping lemma and closure properties

3. Context Free Grammars