Regular Expression
Working with Regular Expressions

Let $R$ be a regular expression. Then,

$$R \cup \emptyset = R$$

$$R \circ \emptyset = \emptyset$$

$$R \cup \varepsilon = R \cup \varepsilon$$

$$R \circ \varepsilon = R$$
Given a regular expression $R$, we showed how to construct an NFA $N$ such that $L(R) = L(N)$.
Regular Expressions $\Rightarrow$ Regular Languages

**Theorem.** A language is regular if and only if a regular expression describes it.

**Proof.** By induction on the definition of regular expressions.

**Base case:**
1. If $a \in \Sigma$, then $a$ is regular.
2. $\varepsilon$ is regular.
3. $\emptyset$ is regular.

**Inductive step:**
4. If $R$ and $S$ are regular, then $(R \cup S)$ is regular.
5. If $R$ and $S$ are regular, then $(R \circ S)$ is regular.
6. If $R$ is regular, then $(R)^*$ is regular.
Proof in Action

Build an NFA that recognizes the regular expression \((ab \cup a)^*\)
Proof in Action

Build an NFA that recognizes the regular expression \( a(a \cup b)^* a \)
Given a NFA $N$, we will show how to construct a regular expression $R$ such that $L(N) = L(R)$.
Proof Idea

If we can convert any NFA into something that looks like this, we can just read off the regular expression.

Need a machine where transitions can be labelled by regular expressions.
Building intuition

Self loops:

Alternate paths:

Adjacent paths:
From NFAs to Regular Expressions
From NFAs to Regular Expressions

- Add a new start state with \( \varepsilon \) arrows to old start state
From NFAs to Regular Expressions

- Add new accept state with \( \varepsilon \) arrows from old accept states. Remove old accept states.
From NFAs to Regular Expressions

- Eliminate each state (except \(s\) and \(f\)); replacing them with regular expressions

Can we eliminate this state?
From NFAs to Regular Expressions

- Eliminate each state (except $s$ and $f$); replacing them with regular expressions
From NFAs to Regular Expressions

- Eliminate each state (except \(s\) and \(f\)); replacing them with regular expressions
From NFAs to Regular Expressions

- Eliminate each state (except $s$ and $f$); replacing them with regular expressions
From NFAs to Regular Expressions

- Eliminate each state (except $s$ and $f$); replacing them with regular expressions

```
Can we eliminate this state?
```
From NFAs to Regular Expressions

- Eliminate each state (except $s$ and $f$); replacing them with regular expressions
From NFAs to Regular Expressions

- Eliminate each state (except $s$ and $f$); replacing them with regular expressions

![Diagram](image-url)
General Construction: NFA to Regular Expressions

First change the NFA to the following special form:

- Add new start state with $\varepsilon$ arrow to old start state.
- Add new accept state with $\varepsilon$ arrows from old accept states.
- Swap old accept states to non accept.
- Multiple labels on arrow are replaced by union of labels.
- Add arrows labeled $\emptyset$ between states that had no arrows.
General Construction: NFA to Regular Expressions

Select a state $q_{rip}$ (different from $s$ and $f$) and for any $q_i \notin \{q_{rip}, f\}$ and $q_j \notin \{q_{rip}, s\}$ and eliminate $q_{rip}$ as follows. Repeat until only $s$ and $f$ remain.

Before:

![Diagram of NFA]

After:

$$q_i \xrightarrow{(R_1)(R_2)^* (R_3) \cup (R_4)} q_j$$
Regular Expression?
A Regular Expression for an Old Friend