Last time:
- Finite automata, regular languages
- Regular operations $\cup, \circ, \ast$
- Regular expressions
- Closure under $\cup$

Today:
- Nondeterminism
- Closure under $\circ$ and $\ast$
- Regular expressions $\rightarrow$ finite automata

Goal: Show finite automata equivalent to regular expressions

- This week’s check-ins will not be counted
- TA office hours will be posted tomorrow
- Chat is restricted to TAs only.
Problem Sets

- 35% of overall grade
- Problems are hard! Leave time to think about them.
- Writeups need to be clear and understandable, handwritten ok.
  Level of detail in proofs comparable to lecture: focus on main ideas.
  Don’t need to include minor details.
- Submit via gradescope (see Canvas) by 2:30pm Cambridge time.
  Late submission accepted (on gradescope) until 11:59pm following day:
    1 point (out of 10 points) per late problem penalty.
  After that solutions are posted so not accepted without S3 excuse.
- Optional problems:
  Don’t count towards grade except for A+.
  Value to you (besides the challenge):
    Recommendations, employment (future grading, TA, UROP)
- Problem Set 1 is due in one week.
Closure Properties for Regular Languages

**Theorem:** If $A_1$, $A_2$ are regular languages, so is $A_1A_2$ (closure under $\circ$)

Recall proof attempt: Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize $A_1$

$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize $A_2$

Construct $M = (Q, \Sigma, \delta, q_0, F)$ recognizing $A_1A_2$

$M$ should accept input $w$ if $w = xy$ where $M_1$ accepts $x$ and $M_2$ accepts $y$.

Doesn’t work: Where to split $w$?

Hold off. Need new concept.
Nondeterministic Finite Automata

Nondeterminism doesn't correspond to a physical machine we can build. However, it is useful mathematically.

New features of nondeterminism:
- multiple paths possible (0, 1 or many at each step)
- ε-transition is a “free” move without reading input
- Accept input if some path leads to accept

Example inputs:
- ab
- aa
- aba
- abb

Check-in 2.1
What does $N_1$ do on input $aab$?
(a) Accept
(b) Reject
(c) Both Accept and Reject
Defn: A nondeterministic finite automaton (NFA) $N$ is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$

- all same as before except $\delta$
- $\delta: Q \times \Sigma \rightarrow \mathcal{P}(Q) = \{R | R \subseteq Q\}$
  - power set $\Sigma \cup \{\varepsilon\}$
- In the $N_1$ example: $\delta(q_1, a) = \{q_1, q_2\}$
  $\delta(q_1, b) = \emptyset$

Ways to think about nondeterminism:

Computational: Fork new parallel thread and accept if any thread leads to an accept state.

Mathematical: Tree with branches. Accept if any branch leads to an accept state.

Magical: Guess at each nondeterministic step which way to go. Machine always makes the right guess that leads to accepting, if possible.
Converting NFAs to DFAs

**Theorem:** If an NFA recognizes $A$ then $A$ is regular

**Proof:** Let NFA $M = (Q, \Sigma, \delta, q_0, F)$ recognize $A$

  Construct DFA $M' = (Q', \Sigma, \delta', q'_0, F')$ recognizing $A$

  (Ignore the $\epsilon$-transitions, can easily modify to handle them)

**IDEA:** DFA $M'$ keeps track of the subset of possible states in NFA $M$.

Check-in 2.2

If $M$ has $n$ states, how many states does $M'$ have by this construction?

(a) $2n$
(b) $n^2$
(c) $2^n$

Construction of $M'$:

$Q' = \mathcal{P}(Q)$

$\delta'(R, a) = \frac{R}{R \in Q'}$

$q'_0 = \{q_0\}$

$F' = \{R \in Q' \mid R \text{ intersects } F\}$
Recall Theorem: If $A_1, A_2$ are regular languages, so is $A_1 \cup A_2$.
(The class of regular languages is closed under union)

New Proof (sketch): Given DFAs $M_1$ and $M_2$ recognizing $A_1$ and $A_2$
Construct NFA $M$ recognizing $A_1 \cup A_2$.
Closure under $\circ$ (concatenation)

**Theorem:** If $A_1, A_2$ are regular languages, so is $A_1A_2$

**Proof sketch:** Given DFAs $M_1$ and $M_1$ recognizing $A_1$ and $A_2$,

Construct NFA $M$ recognizing $A_1A_2$

$M$ should accept input $w$ if $w = xy$ where $M_1$ accepts $x$ and $M_2$ accepts $y$.

$w = \underline{x} \underline{y}$

Nondeterministic $M'$ has the option to jump to $M_2$ when $M_1$ accepts.
**Theorem:** If $A$ is a regular language, so is $A^*$

**Proof sketch:** Given DFA $M$ recognizing $A$,
Construct NFA $M'$ recognizing $A^*$

Check-in 2.3
If $M$ has $n$ states, how many states does $M'$ have by this construction?
(a) $n$
(b) $n + 1$
(c) $2n$
Regular Expressions $\rightarrow$ NFA

**Theorem:** If $R$ is a regular expr and $A = L(R)$ then $A$ is regular

**Proof:** Convert $R$ to equivalent NFA $M$:

If $R$ is **atomic**:
- $R = a$ for $a \in \Sigma$
- $R = \varepsilon$
- $R = \emptyset$

Equivalent $M$ is:
- $\begin{align*}
R = a & \quad \xrightarrow{a} \circ \\
R = \varepsilon & \quad \xrightarrow{} \\
R = \emptyset & \quad \xrightarrow{}
\end{align*}$

If $R$ is **composite**:
- $R = R_1 \cup R_2$
- $R = R_1 \circ R_2$
- $R = R_1^*$

Use closure constructions

**Example:**
Convert $(a \cup ab)^*$ to equivalent NFA
- $a$: $\begin{align*}
\xrightarrow{a} \circ
\end{align*}$
- $b$: $\begin{align*}
\xrightarrow{b} \circ
\end{align*}$
- $ab$: $\begin{align*}
\xrightarrow{a} \xrightarrow{\varepsilon} \xrightarrow{b} \circ
\end{align*}$
- $a \cup ab$: $\begin{align*}
\xrightarrow{\varepsilon} \xrightarrow{a} \xrightarrow{\varepsilon} \xrightarrow{b} \circ
\end{align*}$
- $(a \cup ab)^*$:
Quick review of today

1. Nondeterministic finite automata (NFA)
2. Proved: NFA and DFA are equivalent in power
3. Proved: Class of regular languages is closed under $\circ,\ast$
4. Conversion of regular expressions to NFA

Check-in 2.4

Recitations start tomorrow online (same link as for lectures). They are optional, unless you need more help. You may attend any recitation(s).
Which do you think you’ll attend? (you may check several)
(a) 10:00  (b) 11:00  (c) 12:00
(d) 1:00    (e) 2:00    (f) I prefer a different time (please post on piazza, but no promises)