18.404 Review Session
8–8:30 — Language classes (Jason)
8:30–9 — Mapping reduction, computation history (Jocelin)
9–9:30 — Pumping lemma, decidability (Jett)

How to prove things are regular, aren’t they regular, CFLs, decidable, recognizable?

Proving things are regular:
- Construct DFA/NFA
- Write a regular expression
- Closure properties
- Split into easier problems

Closure example: Show \( L = \{ a^m b^n \mid m, n \geq 0 \text{ and even length} \} \) is regular.

We can write \( L = \underbrace{\{(a^n b^n)\}}_{\text{reg}} \cap \underbrace{\{(\text{even length})\}}_{\text{reg}} \)

Since both parts are regular, \( L \) is regular.

Proving things aren’t regular:
- Pumping lemma
  - Proof by contradiction
  Assume language is regular
  Show some string violates pumping lemma

  \[ L = \{ a^m b^n \mid n \geq 0 \} \]
  Assume \( L \) is regular
  \[ s = a^p b^q = a a a \ldots a b b b \ldots b \]
  Assume \( L \) is regular
  Then \( L \cap \{(a^n b^m)^3\} \) is also regular
  \[ \underbrace{\{(a^n b^m)^3\}}_{\text{reg}} \]
  \[ \Rightarrow \underbrace{\{(a^n b^m)^3\}}_{\text{reg}} \] is also regular
  \[ \text{Contradiction!} \]
Proving things are **CFLs**
- Construct PDA (description is fine)
  \[ \{a^n b^n \mid n \geq 0\} \]
  - on \(w\),
    - read 'a's and push them onto stack until none left
    - read 'b's and pop 'a's off stack
    - accept if stack is empty
    - reject otherwise
- Write grammar
  \[ \{a^n b^n \mid n \geq 0\} \quad S \rightarrow aSb \mid \epsilon \]
  - \(\text{CFL} \cap \text{REG} = \text{CFL}\)
- Closure properties
  - \(\text{CFL} \cap \text{CFL} = \text{not a CFL}\)

Proving things aren't **CFL**
- Context-free pumping lemma (example later)
- Closure properties (via contradiction)

Proving things are **decidable**
- Construct a decider TM, prove it always halts. (example later)
- We can use problems that we know are decidable
  - \(A_{DFA}, E_{DFA}, EQ_{DFA}, ACFG, ECFG\)

Proving things aren't **decidable**
- General reduction (example later)
  - Proof by contradiction
    - Assume decider exists
    - Use it to construct a decider for \(A_{TM}\)
      - (or another undecidable language)
  - Computation history (example later)

Proving things are **recognizable**
- Construct TM recognizing the language
  - Prove it halts for \(x \in L\)
    - (no requirement for \(x \notin L\))

Proving things aren't **recognizable**
- Mapping reduction \(A_{TM} \leq L\)
  - (equivalently, \(A_{TM} \leq \overline{L}\))
Mapping Reducibility

Show that if $A$ is T-recog & $A \leq_m \overline{A}$, then $A$ is decidable.

**Prf:** $A \leq_m \overline{A} \implies \overline{A} \leq_m A$

Since $A$ is T-recog, then $\overline{A}$ must also be T-recog.

If $A$ and $\overline{A}$ are both T-recog, then $A$ is decidable.

**Prf:** Construct a decider $D$ for $A$. Say $M_1$ recog A & $M_2$ recog $\overline{A}$.

$D$: On $\langle x \rangle$:

- Run $M_1$ and $M_2$ on $x$ in parallel.
- If $x \in A$:
  - Then $M_1$ will eventually acc. $x$.
  - $\implies D$ accs.
- If $x \notin A$:
  - Then $M_2$ will eventually acc. $x$.
  - $\implies D$ rej.
Define an erase \( TM \) to be a \( TM \) that can erase and read on input tape, but can't write other symbols on tape, replacing with blank.

Show \( E_{\text{erased}TM} \) is undecidable.

\[
E_{\text{erased}TM} = \{ M \mid M \text{ is erase } TM, L(M) = \emptyset \}
\]

Proof: Reduce from \( \text{ATM} \).

Idea: Construct erase \( TM \) \( E \) that accepts accepting comp. hists. for \( M \) on \( w \).

Problem: Can only erase, not write w/ special chars.

Take in comp. history \( x_2 \) as input.
Tape:

```
# q_0 w_1 ... w_k # # ... # q_{acc} #
```

```
# q_0' w_1' ... w_k' # # ... # q_{acc}'#
```

1. Check first config $c_0$ is a valid start w/ $q_0$, $w$.
2. Check final config $c_n'$ valid w/ $q_{acc}$.
3. Check valid transitions from $c_i$ to $c_{i+1}$.
   a. Erase chars in $c_0'$ (go after $\ast$, erase until next #)
   b. Check $c_0$ to $c_i'$ (first nonblanks after $\ast$ until next hash)
   c. As you check character-by-character zigzagging from 1st copy to corresponding char on 2nd copy, delete the characters you just checked.

→ so you can find next corresponding chars in configs.

*note: technically details around checking
around head location $q_k$, such as checking its adj. symbols, to see if valid state transition

d. Continue: check $c_i$ to $c_{i+1}'$ character-by-character erasing the character you just checked.

e. Done when everything after $\#$ is erased.

---

Decider for $A_{TM}$
On $\langle M, w \rangle$:
1. Construct ERASE TM $E_{M,w}$ as detailed above.
2. Use decider to see if $L(E_{M,w}) = \emptyset$.
3. If decider accs, we rej. rej we acc.
* Remark: This is also valid mapping reduction from \( \overline{A_{TM}} \) to \( E_{eraseTM} \):

\[
\begin{align*}
\langle M, w \rangle & \quad f \quad \text{eraseTM} \\
M \text{ doesn't acc. } w & \quad \emptyset \\
\langle M, w \rangle & \quad f \quad \text{eraseTM} \\
M \text{ acc. } w & \quad \text{lang. nonempty}
\end{align*}
\]

So \( \overline{A_{TM}} \leq_m E_{eraseTM} \Rightarrow E_{eraseTM} \text{ T-unrecog.} \)
2.48 (book) Let $C_1$ be the language of all strings that contain a 1 in their middle third.

$\Sigma = \{0, 1\}$

$C_1 = \{abc \mid a, c \in \Sigma^*; b \in \Sigma^*1\Sigma^*; |d| = |c| \geq |b|\}$

a. Show $C_1$ is a CFL.

\[
S \rightarrow ATA \\
T \rightarrow ATAA \mid AATA \mid ATA \mid 1 \\
A \rightarrow \Sigma
\]
\[ C_2 = \{abc \mid a, c \in \Sigma^* \mid b \in \Sigma^1 \Sigma^{1*1} \Sigma^* \mid \text{il} = 1c \geq 1b \} \]

b. Show \( C_2 \) is not a CFL.

Assume \( C_2 \) is a CFL.
Then it has a pumping length \( p \).
Consider the string \( S = 0^p10^{p-2}10^p \).

Regardless of how \( u \), \( v \), \( x \), \( y \), and \( z \) is chosen,
the pumping lemma conditions cannot be satisfied.

2 cases:
- \( v \) and \( y \) contain only zeros.
\[ S = 0000000000100000000000000 \]
  - no matter where \( u \) and \( y \) are,
pumping them will move the 1s out of the middle.
- \( u \) or \( y \) contains a 1.
  - pump down, getting rid of the 1.
4.27 (book:) Let $E = \{ \langle M \rangle \mid M$ is a DFA which accepts some string $w$ with more 1s than 0s $\}$.

Show $E$ is decidable.

Key pts: - Construct a TM
  - $\{ \langle M \rangle \mid w \text{ contains more 1s than 0s} \}$ is a CFL.
  - $\text{CFL} \cap \text{reg} = \text{CFL}$

"on input $\langle M \rangle$:
  - Let $A$ be $\{ w \mid w \text{ contains more 1s than 0s} \}$.
  - Construct a CFG generating $L(M) \cap A$, $B$.
  - Feed $B$ into $E_{\text{CFG}}$ decider.
  - If it accepts, reject. If it rejects, accept."