18.404 midterm review sessions

HALLO! — We'll offer problem-solving practice sessions,° in-person. These midterm review sessions are **completely optional**. It's easy to imagine a non-attendee earning an excellent midterm grade.

← Coordinates:
Mon 7:30-9pm in room 2-190 with Zed, Sarah
Tue 7:30-9pm in room 2-190 with Leo, Nathan

If you choose to work through these practice problems, you may want to play a kind of **Concept Bingo**. Notice whether and in what context you use the following concepts:°

← Your solutions to these practice problems will probably miss a few of these concepts. That's okay. These problems support multiple nice solutions; different solutions will use different concepts in this list.

AUTOMATA

- ☐ closure properties for regular languages and context-free languages
- ☐ pigeonhole principle on set of states (e.g. pumping lemma)
- ☐ equivalence of FAs and regular expressions
- ☐ equivalence of PDAs and CFGs

COMPUTABILITY, ABSTRACT PRINCIPLES

- ☐ closure properties for decidable languages and recognizable languages
- ☐ properties of mapping reductions (e.g. transitivity, complement)
- ☐ simulation of one TM variant by another

COMPUTABILITY, CONCRETE EXAMPLES

- ☐ decidability of $A_{LBA}, A_{PDA}, E_{PDA}$
- ☐ undecidability of $A_{TM}, ALL_{TM}, E_{TM}, ALL_{LBA}, E_{LBA}, ALL_{PDA}$
- ☐ recognizability of $A_{TM}, \overline{E_{TM}}, \overline{ALL_{LBA}}, \overline{E_{LBA}}, \overline{ALL_{PDA}}$

COMPUTABILITY, REDUCTION TECHNIQUES

- ☐ simulation for creating mapping reductions
- ☐ equivalence of enumerators and TM recognizers
- ☐ computation history method

What's above is an INcomplete list of concepts. Everything from class so far (excluding time complexity) is fair game.

regular languages

showing regularity

Let $L = \{w$ containing exactly one `a` or exactly one `b`$\}$.
Show that L is regular by explicitly giving:

- ☐ a regular expression;
- ☐ an NFA or DFA.

pumping lemma question

Let $L = \{w$ contains only `0`'s and $|w| = k^2$ for some $k \in \mathbb{N}\}$. Show that L is not regular.

regular languages with lots of 1s

Call a language **major** if it includes some string s with with strictly more `1`s than `0`s. ∘ For example, the language $\texttt{0}^\star\texttt{1}^\star$ is major; the language $(\texttt{0} \cup \texttt{10})^\star$ is not major.

← This concept seems to involve (unbounded) *counting*. Which machines that we discussed can count? For instance, can DFAs count?

- ☐ Is $L = \{\langle M \rangle \mid M$ is a DFA with a major language$\}$ decidable?

context-free languages

showing context-freeness

- ☐ Is $L = \{w\#x \text{ such that } w^R \text{ is a substring of } x\}$ context-free?

closure properties

Suppose A is context free, B is not context free. Which of the following are possible:

- ☐ $B = A \cap C$ for C regular?

- ☐ $B = A \cap C$ for C context free?

- ☐ $B = A \cup C$ for C regular?

- ☐ $B = A \cup C$ for C context free?

mapping reductions

the butterfly's head

Remember: A_{TM} and $\overline{A_{TM}}$ lie on opposite wings° of our butterfly diagram.

← This is a poetic way to say that A_{TM} is recognizable but its complement $\overline{A_{TM}}$ is not.

☐ True or false? $\overline{A_{TM}} \leq_m A_{TM}$.

Can we invent some problem (i.e., language) that's harder than both A_{TM} and $\overline{A_{TM}}$? Well, we could make the problem's solutions encode solutions both to A_{TM} and to $\overline{A_{TM}}$. So let $J = 0A_{TM} \cup 1\overline{A_{TM}}$. That's shorthand for:

$$J = \{b(\langle M \rangle, w) \text{ such that TM } M \text{ accepts } w \text{ if and only if bit } b \text{ equals } \texttt{0}\}$$

Then J sits atop both A_{TM} and its complement in the butterfly:° $A_{TM} \leq_m J$ and $\overline{A_{TM}} \leq_m J$. Moreover, J is on the axis of symmetry: $J \leq_m \bar{J}$.°

← ☐ [A] Prove this!

← ☐ [B] And prove this, too!

☐ Finally, is it true that $\overline{E_{LBA}} \leq_m J$?

ahh! shh!

Consider these two languages:°

$$AH = \{\langle M \rangle \text{ such that } M \text{ is a TM that halts on } \textit{all} \text{ inputs}\}$$

$$SH = \{\langle M \rangle \text{ such that } M \text{ is a TM that halts on } \textit{some} \text{ input}\}$$

← You'll notice that AH and SH seem dual. But we've learned to tread with care when making such comparisons.
☐ What goes wrong with the following attempt to reduce $AH \leq_m \overline{SH}$?
Reduction attempt: *map* $\langle M \rangle$ *to* $\langle \tilde{M} \rangle$ *where* $\tilde{M}$ *simulates* M *and loops forever if* $\tilde{M}$ *halts, else halt. Then* $\langle M \rangle \in AH$ *if and only if* $\langle \tilde{M} \rangle \notin SH$.

☐ Is AH recognizable? How about its complement?

☐ Is SH recognizable? How about its complement?

a mapping reduction edge case

☐ True or false? Whenever A is decidable and B is regular: $A \leq_m B$.

all together now

double the tapes is double the fun

For this problem, we define a *2-tape DFA*. A 2-tape DFA has a pair of inputs (x, y), presented on two (finite) tapes, with a read-only, left-to-right head on each tape. The transition function is a mapping $Q \times (\Sigma \times \Sigma) \to Q \times \{\texttt{Head 1}, \texttt{Head 2}\}$, to indicate the new state and which head to step right. The machine accepts if the machine is in an accepting state when one of the heads moves off the end of its tape.

We'll be interested in showing that the following language is decidable:

$$ALL_{\text{2-tape DFA}} = \{\langle M\rangle \mid M \text{ is a 2-tape DFA that accepts every input pair } (x, y)\}.$$

☐ Given a 2-tape DFA M, show how to construct a PDA A with alphabet $\Sigma \cup \{\#\}$, such that $L(A)$ is empty if and only if M accepts every (x, y).

☐ Using the above, conclude that $ALL_{\text{2-tape DFA}}$ is dedidable.

once more, with nondeterminism!

We'll now make a slight modification to the setup: we instead consider 2-tape *NFA*s. The picture is the same as before, except now the transition function is nondeterministic — i.e. it is of the form $Q \times (\Sigma \times \Sigma) \to \mathcal{P}(Q \times \{\texttt{Head 1}, \texttt{Head 2}\})$. As before, we define the ALL problem:

$$ALL_{\text{2-tape NFA}} = \{\langle M\rangle \mid M \text{ is a 2-tape NFA that accepts every input pair } (x, y)\}.$$

Interestingly, this modification of the model actually changes the story entirely.

☐ Show that $ALL_{\text{2-tape NFA}}$ is undecidable.