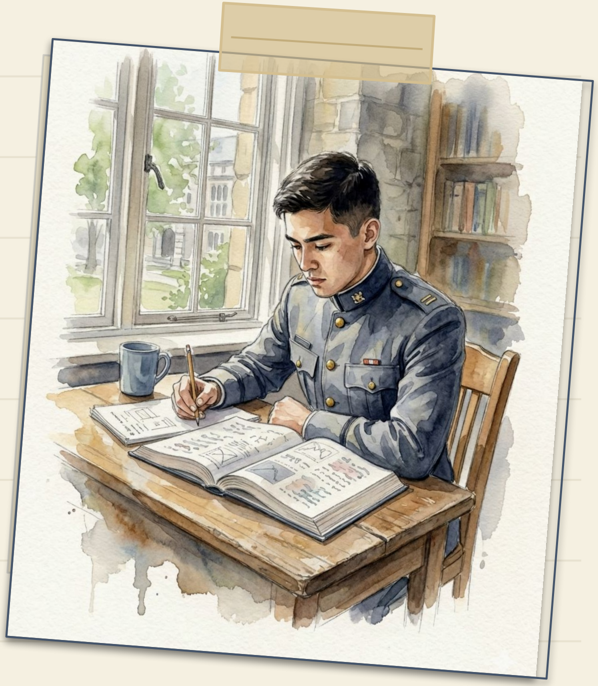
Teaching CS Theory
at a U.S. Service Academy

What is theory <u>for</u>? And <u>how</u> to teach it to cadets who are pulled in three directions at once?



Ryan E. Dougherty

Department of Electrical Engineering & Computer Science · United States Military Academy at West Point

How I was taught, and what I changed

How I was taught

- Theory as a spectator sport: watch the expert prove things at the board.
- Great lectures, but wouldn't be able to reproduce them.
- Performed well but didn't *trust in myself* that I could meaningfully contribute.

How I teach now

- *Theory is something that cadets proactively do.*
- *Class time spent reasoning together.*
- *Goal: cultivate intrinsic and extrinsic trust.*

What is theory *for*?

The pragmatic answer

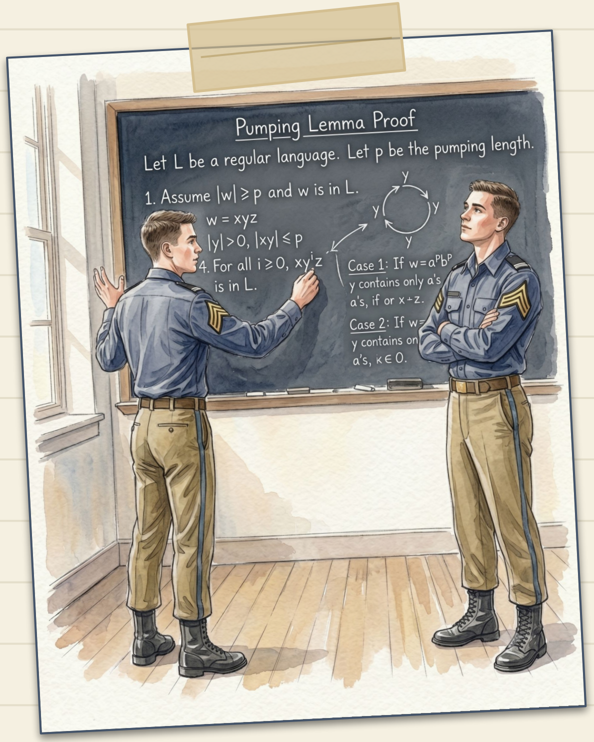
Concrete, durable skills: precise definitions, rigorous reasoning, knowing what is and isn't solvable, and judging an argument's soundness, including a machine's.

The *civic* answer

A space where cadets from any background start from the same definitions and practice reaching conclusions everyone can verify.

Personal belief: theory is for everyone, even Army cadets, and we should make theory innately incorporate raw pragmatism and be civically engaging.

Cadets must be <u>their own</u> truth authority



"Become the computer."

Unique Challenges at West Point

- "When will I use this?"
- Sleep deficit
- Decaying engagement

Unique Advantages of West Point

The cadet ethos

- High baseline motivation
- An “embrace the hard thing” culture
- Bright cadets with good attitudes for learning.

The structure

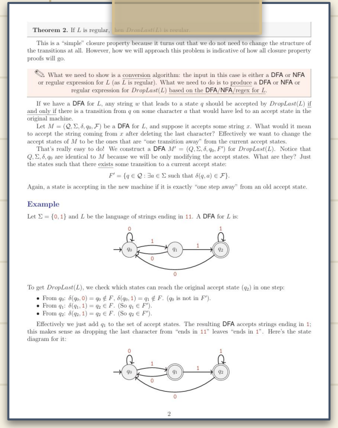
- Small sections (~18) with mandatory attendance
- High institutional trust and encouragement for bold course design.
- Shared groups: cadets already know how to operate as a team.

The combination of small classrooms and motivated cadets who embrace difficulty cultivates <u>trust</u> with meaningful <u>risk</u>.

Inverting the Classroom

Before

Guided course notes +
some pre-recorded videos



Theorem 2. If L is regular, then $DropLast(L)$ is regular.

This is a "simple" closure property because it turns out that we do not need to change the structure of the transitions at all. However, how we will approach this problem is indicative of how all closure property proofs will go.

What we need to show is a conversion algorithm: the input in this case is either a DFA or NFA or regular expression for L (as L is regular). What we need to do is to produce a DFA or NFA or regular expression for $DropLast(L)$ based on the DFA/NFA/regex for L.

If we have a DFA for L, any string w that leads to a state q should be accepted by $DropLast(L)$ if and only if there is a transition from q on some character a that would have led to an accept state in the original machine.

Let $M = (\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{F})$ be a DFA for L, and suppose it accepts some string x. What would it mean to accept the string coming from x after deleting the last character? Effectively we want to change the accept states of M to be the ones that are "one transition away" from the current accept states.

That's really easy to do! We construct a DFA $M' = (Q, \Sigma, \delta, q_0, F')$ for $DropLast(L)$. Notice that Q, Σ, δ, q_0 are identical to M because we will be only modifying the accept states. What are they? Just the states such that there exists some transition to a current accept state:

$$F' = \{q \in \mathcal{Q} : \exists a \in \Sigma \text{ such that } \delta(q, a) \in \mathcal{F}\}.$$

Again, a state is accepting in the new machine if it is exactly "one step away" from an old accept state.

Example

Let $\Sigma = \{0, 1\}$ and L be the language of strings ending in 11. A DFA for L is:

To get $DropLast(L)$, we check which states can reach the original accept state (q_2) in one step:

- From q_0: $\delta(q_0, 0) = q_0 \notin F$, $\delta(q_0, 1) = q_1 \notin F$. ($q_0$ is not in F').
- From q_1: $\delta(q_1, 1) = q_2 \in F$. (So $q_1 \in F'$).
- From q_2: $\delta(q_2, 1) = q_2 \in F$. (So $q_2 \in F'$).

Effectively we just add q_1 to the set of accept states. The resulting DFA accepts strings ending in 1; this makes sense as dropping the last character from "ends in 11" leaves "ends in 1". Here's the state diagram for it:

2

In class

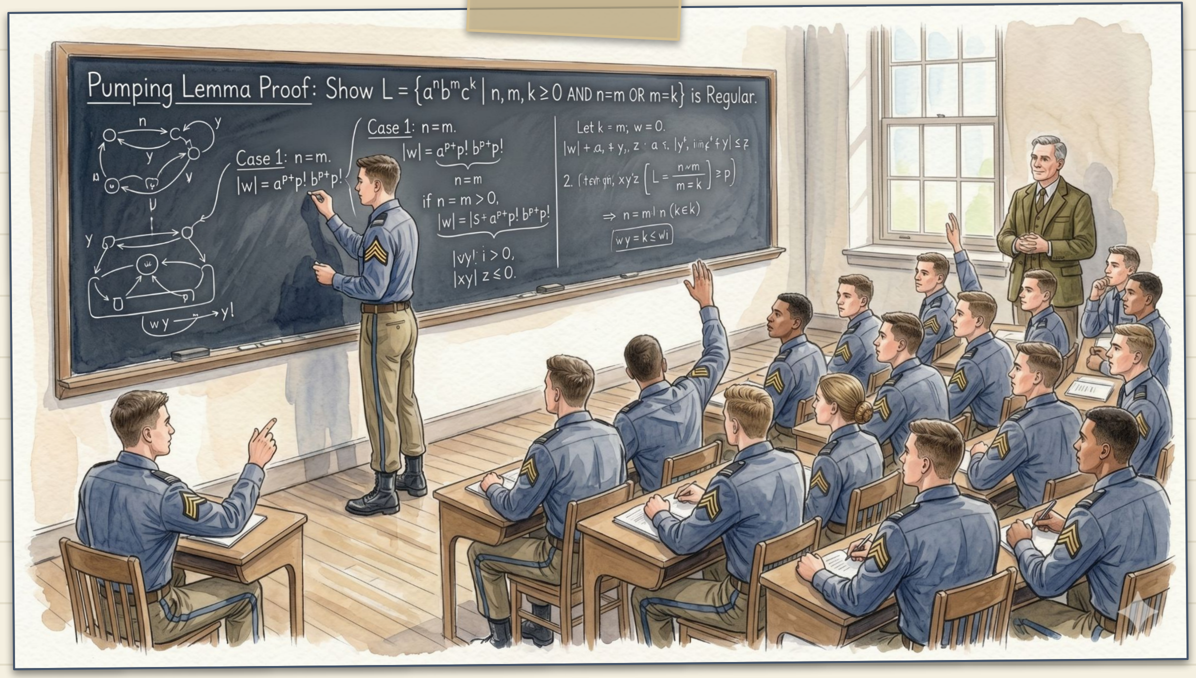
Cadets work through and defend reasoning arguments together

After

Targeted practice tuned to where they actually struggled

Contextualizing the Problem

Cadets in Charge

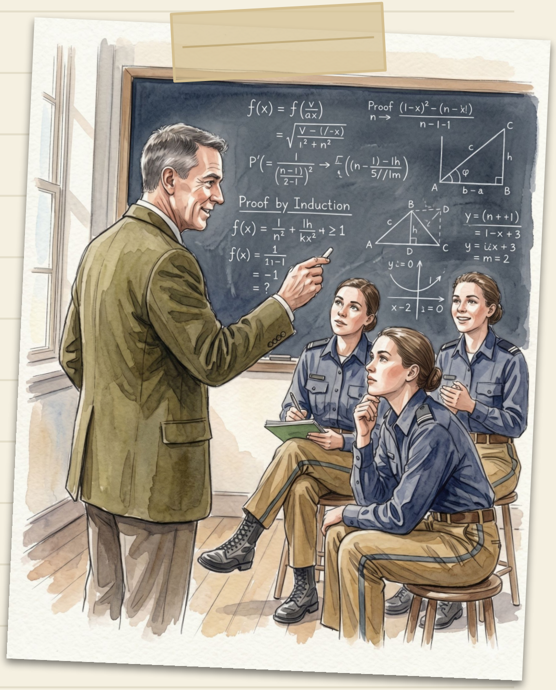


Why critical thinking matters <u>now</u>

"Does this conclusion actually follow?"

My Recommendations

- *Emphasize* the pragmatic skill and the civic why.
- Stop lecturing proofs at cadets, actively *engage* with their reasoning.
- Treat self-trust as something you *build*, not something they should already have.
- *Engineer* early experiences of verification.
- Internalize your context's advantages and boldly *explore* within limits.



Questions?

What is your theory course <u>for</u>? And does how you teach it reflect that answer?

Ryan E. Dougherty

ryan.dougherty@westpoint.edu