Learning Requires Doing

A goal for this course is for you to gain experience using models of computation, formal languages, and proofs to solve problems. The homework assignments are essential practice for you to think about the material outside of class and identify what you don’t understand.

In many courses, homework turns into a painful cycle: If you make mistakes on an assignment, turn it in, get your grade back a week and a half later, throw it in a notebook because we’ve moved on to other material, and don’t think about those problems again (except maybe the night before an exam), you won’t learn from your mistakes, and the time and effort you spent on the homework will have been wasted.

This semester we’re making an effort to do better. Instead of submitting assignments (often late) and then waiting for me to grade and comment on your work after all assignments are in, I will be releasing example solutions as soon as the assignment is due and then asking you to correct your own work. This ensures that you will review the exercises while you still remember what you were thinking, and you will get credit for learning from what you did wrong, rather than be punished for not knowing the material perfectly from the start.

*It’s entirely reasonable to make mistakes when you’re first learning material.* As such, I want the homework assignments to be low stress. This is an experiment, so we may adjust how homework is handled as the semester goes on, but this is my promise: If you make a serious effort to solve the problems, turn them in by the deadline, and carefully review your work to understand how your solutions could be improved, then your homework grade will be very high, even if you initially make mistakes on every problem.

Assignment Expectations

1. **Assignments will only be accepted until the due date.**

   This is for everyone’s benefit. Unfortunately, it’s impractical to allow late assignments without punishing the rest of the class by delaying releasing solutions. (And, in practice, late submissions are usually half-hearted bids for partial credit rather than attempts to learn.)

   You should try every assignment even if you’re unable to submit it on time. Remember that the assignments are meant to give you practice; completing them will help you to learn the material (and thus to succeed on future assignments and exams).

   If you know you won’t be able to submit an assignment, talk to me. If you’re experiencing a major problem – medical, psychological, family,
etc. – that is interfering with your ability to complete your class work, you should talk with the Dean of Studies, your class advisor, or Health Services, who will recommend appropriate accommodations to all of your professors, who honestly want to help you.

2 Late corrections will accepted with a penalty.

If you’re unable to do your corrections by the day they’re due, they can be turned in up to the start of the next class, for 75% of the original value.

3 You must make a serious attempt to answer every problem for an assignment to count.

This doesn’t mean you need to get them all right, but it’s unacceptable not to attempt a problem, e.g., because it looks hard or you didn’t leave yourself enough time. To count toward a complete submission, any problem for which you can’t give a full solution needs to be accompanied by a clear explanation of your thinking, e.g.,

“Here’s an NFA I could design for the language, but I couldn’t make an equivalent regular expression because everything I tried would include these strings that aren’t in the language, for instance, …”

Or,

“I could prove that A is a subset of B, but proving that B is also a subset of A seems to require that B exhibit the following properties that I don’t know how to prove…”

4 You need to submit two copies of your assignment.

For assignments to be fair for everyone, I need to know what work you did originally as well as your corrections. Each assignment will list the due date for your original solutions and for your corrections. The time for completing the original work and the corrections may vary based on the length of the assignment and to accommodate the semester schedule.

5 You can submit electronically or on paper.

You have your choice of submitting both your original assignment and your corrected assignment either by email as a PDF or on paper. If you submit on paper, please bring your assignment to class. If you’re unable to come to class, you can put your assignment in my mailbox in the Computer Science Department office or slip it under the door of my office, but these are less preferable.

6 You can complete assignments individually or work with a partner.

Be sure you understand the collaboration policy given in the syllabus.
Assignments must be neat and clear.

Your assignment must be easy to read and must clearly indicate both your original solutions and your corrections. This will be easiest to do if you type your solutions.

You are encouraged to prepare assignments using the typesetting language \LaTeX, which is a standard tool for publishing research in computer science and mathematics. You may find it easiest to use Overleaf (overleaf.com) to edit \LaTeX in a web browser. \LaTeX templates will be available for each assignment on the course website.

Be sure your name and the assignment number are on the first page. Printed work should be stapled in the top left corner. Do not submit work on paper with frayed edges torn from a notebook.

How to Correct Your Assignments

The second part of completing an assignment is to go through your work and the posted example solutions and to correct your solution. The point is not to agonize over assigning yourself points, so we’ll mark solutions coarsely as 0–4 with no partial points. Instead, you should focus on understanding the example solution and what you could improve in your answer, even if your answer was “right”.

Grading with fine-grained points is an exercise in frustration.

0 No serious attempt.

If you didn’t give a serious answer to a problem, it will be marked as 0.

1 Incomplete or mostly incorrect.

These answers require significant correction. If your solution is close enough that it can be edited to be correct, e.g., by filling in the part after you got stuck, do so, but most answers in this category are so far off – for instance, giving a proof in the wrong direction – that you should instead rewrite the example solution in your own words and write a comment on what you misunderstood.
Why did you prove the wrong direction, misunderstand the question, construct a malformed Turing machine, etc.? If you can identify for yourself what you were doing wrong, then you'll know what to do right the next time!

2 On the right track.

If your answer is more or less incorrect, but your answer is on the right track or if it's correct but muddled in the argument, then you've written a solution that should be edited to be correct. Don't copy the example solution. Instead, take the time to understand what the significant differences are between your answer and the example solution and fix your answer to correct its shortcomings.

Write a description of what you changed and why the original answer was incorrect or unclear.

3 Mostly correct.

If your answer has correct logic and is written coherently, it nonetheless might have a missing step or two. Write an explanation of what could be improved. E.g., “I didn’t realize the description of the language included \( \epsilon \). To accept the empty string, just change the start state to an accepting state and everything else stays the same.” Or, “I missed a step in the proof. If \( A \) is an infinite language, then…”

4 Turing’s ghost weeps with pride.

If, having read the example solution, you think that your solution is entirely correct, you should mark it as a 4. If the answer is simple, e.g., a list of states that a \( \text{DFA} \) goes through, you don’t need to write anything more.

For a problem with a non-trivial answer, e.g., a state diagram of a \( \text{DFA} \) or a proof, you should briefly comment on the differences between the example solution and your own. E.g., “My solution is functionally equivalent but doesn’t reuse state \( q_2 \) when inputs begin with a since I found it clearer to have separate branches of computation.”

Problems with significant parts, e.g., designing multiple finite automata, should have each part marked separately. Problems with only small parts, e.g., giving a few short answers about an example I provide, should be marked as a single problem. If you’re not sure, ask.

How I’ll Grade Your Assignments

You’re marking and correcting your own assignments in order to clarify your understanding of the course material. I’ll still assign the final grades, as follows:

The assignments will be graded on the same 0–4 range as the problems. I will go over your answers and corrections, adjust the integer scores, and average the results.
If your marks are accurate and your assessments and corrections of your mistakes are careful, I will add up to 1 point (i.e., a quarter of the total possible points) to the assignment score. So, if you have mistakes on every problem, you can still have a perfect final score, but only if you go over your work carefully!

Conversely, if you don’t complete corrections or your corrections show that you haven’t carefully considered the solutions, then you may lose up to $\frac{1}{2}$ point, so an assignment where every problem was answered perfectly could become a $3\frac{1}{2}$ out of 4. I will also remove $\frac{1}{2}$ point from the final score for assignments that don’t meet the formatting expectations, e.g., missing your name or messy work.