Guide to Assignments

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 the class has moved on to other material, and don’t think about those problems again until exam time, 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 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.¹
If you know you won’t be able to submit an assignment, talk to me, as early as possible.

2 Late corrections will be 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. After that, corrections won’t be accepted.

3 You must make a serious attempt to answer every exercise 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 and just wait for the solutions, e.g., because it looks hard or you didn’t leave yourself enough time.²
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 and what you tried, 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 also match 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 see 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.

5 You can collaborate!
For assignments, you have the option of working with a partner in the class – or of talking to a generative AI tool (like ChatGPT). Both of these options are meant to help you learn by working through the exercises with help.

Whether your collaborator is a classmate or a machine, you must acknowledge that assistance, and you must be careful that you are an active, engaged participant. It’s never acceptable to submit as your work something you didn’t (co-)create.

If you’re interested in collaboration or generative AI, review the relevant policies in the syllabus.

6 Assignments must be neat and clear.

Your assignment must be easy to read and must clearly indicate both your original solutions and your corrections.

In this course, we care about your ability to communicate mathematical arguments precisely. Small errors in wording or in notation can make an otherwise correct proof completely incorrect. As such, we ask that you submit clear, typeset solutions (as a PDF) rather than handwritten answers.

You can type your answers using any software you like, but I recommend using \TeX, 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 \TeX in a web browser. \TeX templates will be available for each assignment on the course website.

If you find it easier to draw diagrams by hand and scan them to include in your assignment, you can do so. (See this guide to “scanning” using your phone.)
How to correct your assignments

The second part of completing an assignment is to go through your initial answers and the provided example solutions and to correct your work. The goal is not to agonize over assigning yourself points, so we’ll just evaluate answers using the categories 0, X, ✓−, ✓, and ✓+. Rather than focus on your “score” for an exercise, 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 should be marked as 0.

- **X**: 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!

- **✓−**: Borderline.
  
  If your answer is more or less incorrect but 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.

• ✓: Mostly correct.

An answer that demonstrates understanding of the problem and is generally clear and correct. Nonetheless, there are some minor flaws in the logic or presentation.

Note what needs to be changed and make the appropriate corrections, e.g.,

“I didn’t realize the description of the language included $\varepsilon$. To accept the empty string, just change the start state to an accept state and everything else stays the same.”

or,

“I missed a step in the proof. If $A$ is an infinite language, then...”

• ✓+: Turing’s ghost weeps with pride.

The solution is clear and correct. There are at most one or two trivial flaws that can be easily corrected.

For a problem with a non-trivial answer, e.g., a state diagram of a 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.”
Examples

Below are two example exercises with a student’s initial answer and good corrections.

Exercise 1

Prove or disprove: If \( A \subseteq B \) and \( A \subseteq C \), then \( A \subseteq B \cap C \).

Answer:

\[
\text{PROOF } \quad \text{Since } A \subseteq B, \text{ it means that some group of the elements of } B \text{ is the set } A. \text{ Since } A \subseteq C, \text{ it means that some group of the elements of } C \text{ is the set } A. \text{ Therefore, some group of the elements of } B \cap C \text{ is the set } A, \text{ so } A \subseteq B \cap C. \quad \blacksquare
\]

Evaluation: ✓ –

Corrections and comments:

The basic idea of this proof is right, but the writing is not clear and convincing. It doesn’t introduce the variables \((A, B, \text{ and } C)\) properly, and the argument is too high-level; it should be about individual elements.

\[
\text{PROOF } \quad \text{Let } A \text{ and } B \text{ be sets, where } A \subseteq B \text{ and } A \subseteq C. \text{ We will show that } A \subseteq B \cap C. \text{ Consider any } x \in A. \text{ Since } x \in A \text{ and } A \subseteq B, \text{ we know that } x \in B. \text{ Since } x \in A \text{ and } A \subseteq C, \text{ we know that } x \in C. \text{ This means that every } x \in A \text{ satisfies } x \in B \cap C, \text{ so } A \subseteq B \cap C, \text{ which is what we needed to show. } \quad \blacksquare
\]

Exercise 2

Prove or disprove: If \( A \subset B \) and \( A \subset C \), then \( A \subset B \cap C \).

Answer:

\[
\text{PROOF } \quad \text{Since } A \subset B, \text{ it means that some group of the elements of } B \text{ is the set } A, \text{ and there are some other elements of } B. \text{ Since } A \subset C, \text{ it means that some group of}
\]
the elements of \( C \) is the set \( A \), and there are some other elements of \( C \). Therefore, some group of the elements of \( B \cap C \) is the set \( A \), and there are some other elements of \( B \cap C \), so \( A \subset B \cap C \). ■

Evaluation: ×

Corrections and comments:

This proof has similar stylistic problems to the previous exercise, but most importantly the informal writing hides the fact that it’s wrong! My answer misses an important requirement, which is that the claim is about proper subsets (\( \subset \) not \( \subseteq \)).

To disprove the claim, consider this counterexample:

\[
A = \{1\} \\
B = \{1, 2\} \\
C = \{1, 3\}
\]

You can see that this satisfies the antecedent: \( A \subset B \) and \( A \subset C \). However, \( B \cap C = \{1\} \). While it’s true that \( A \subseteq B \cap C \), there’s no element in \( B \cap C \) that isn’t in \( A \), so it can’t be a proper subset.

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:

I will go over your answers and corrections, assigning a point value to each answer in the range 0–4, typically corresponding to your own evaluation of your work using the categories \( \circ \), \( X \), \( \checkmark - \), \( \checkmark \), and \( \checkmark + \), but raising or lowering at my discretion. I will average these points to get an initial score for the assignment as a whole.

If your assessments and corrections of your answers 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.

Submission instructions

This semester, we will be using Gradescope to handle assignment submissions. You'll receive an email when I add you to the course Gradescope.

When submitting on Gradescope, if you're working with a partner, please list both of your names: Have one person submit and then, after the submission completes, add the other student’s name to the submission on Gradescope.