

CS3350 Automata, Computability and Formal Languages
Spring 2026. Homework Assignment 1
Due: 03/01/2026 11:59PM MST
Individual assignment

For this assignment, you are supposed to hand in:

- code implementing the functionality missing in the provided boilerplate code, written in Python 3, or a reimplementation of the complete DFA class, written in any programming language,
- a short write-up explaining your code, your testing strategy and the design choices you made.

Submit your code and your write-up (in the form of a PDF) by email to

`utep-spring-2026-automata-hw1@christoph-lauter.org`

The provided boilerplate code has a complete `__init__` constructor for the DFA and full support for parsing of regular expressions and their transformation into deterministic finite automata.

It is your job to

- understand the provided DFA class code sufficiently into detail in order to be able to write yourself code for this class, to
- implement the methods
 - `__remove_epsilon`,
 - `complement`,
 - `intersection`,
 - `union` and
 - `star`,

which are currently stubbed out but left unimplemented,

- and to test your code and its integration into the existing code with various examples.

You should start by implementing and testing the `__remove_epsilon` method. Once this method works well, you can implement the other methods. For those methods, you can get help and ideas in particular from the analysis of the method `concat`, which is provided.

The code you turn in must be sufficiently readable for the grader to give a correct assessment. It must not contain syntax errors and should be hardened with respect to incorrect usage as much as possible.

The write-up must describe your design decisions, your testing strategy and should contain drawings of some automata the code produces for a small number of regular expressions of your choice. You may produce these drawings with the help of scripting and external tools; in this case you need to turn in these scripts as well.

In the case when you do not know how to program in Python, in the case when you are allergic to Python or just because you love the challenge, you are allowed to translate the existing boilerplate into any¹ other programming language and enhance it with the missing methods. All functionality and underlying algorithmic principles must be kept, though.

¹Use discretion to choose this programming language wisely, though. Java, C++, OCaml or even C are wise choices. Assembly, Visual Basic, Fortran etc. are less wise choices. For exotic programming languages like F#, Haskell, Lisp, Prolog, Cobol, Brainf**ck, Whitespace etc. check with your instructor **before** starting the work.