Syllabus for CECS 329, Concepts of Computer Science Theory

Instructor: Dr. Todd Ebert

Fall 2026, Last Updated August 7th, 2026

General Course Information

Academic Unit Department of Computer Engineering and Computer Science, California State University, Long Beach

Prerequisite CECS 328 with a grade of "C" or better.

Catalog Description Seminar on fundamental topics in theoretical computer science. Topics include regular languages, finite automata, context-free languages, Turing machines, computability theory, computational complexity, and NP-completeness.

Course Meeting Times MW 9:30-10:45 am (Section 4, 7690), Friday 8:00-10:45 am (Section 3, 8871)

Instructor Dr. Todd Ebert (Todd.Ebert at csulb.edu)

Instructor Office Hours MW 10:45-11:15, TuTh 9:30 am - 11:00 am, ECS 548

Textbook M. Sipser, "Theory of Computation", Cengage Learning, 2012, 3rd Edition, 978-1133187790

Why Study Theory of Computation?

The purpose of having a theory of computation is to establish truths about computing with the help of mathematical and logical reasoning. These truths are often motivated by basic questions such as

- What is computing? What does it mean to be a computing problem?
- What are the different ways that computing can manifest itself?

- Given a computing problem L, is it possible to solve L via a "classical" computer program? A quantum-computer program? If the answer to either is yes, how much computer memory is needed? How fast can it be solved?

These questions get asked in all areas of computer science and applied computing, often with important ramifications that affect the entire global village. For example, in the area of computer networking Little's Law relates the average number of packets in a queue to the average waiting time and arrival rate, which is crucial for managing router traffic and buffers. This is a theoretical result that is very specific to a particular discipline. However, this course focuses entirely on theoretical issues that are important to all disciplines of computing: studying different computational models and their capabilities, the question of whether or not certain problems are computable, and, if they are computable, their intrinsic computational complexity in terms the amount of time and memory needed to solve them.

In terms of intellectual development, the study of theoretical computer science strengthens one's abstract reasoning abilities, the ability to work with and reason about rule-based systems, and overall general problem solving ability. On one hand, it improves one's ability to be detail-oriented towards solving technical problems, and on the other it develops an "executive perspecitive" towards computing, i.e. the ability to see the forest from the trees. Indeed, the tech leaders of the world are comfortable with sailing high above the computing landscape with a beautiful and inspiring view while later being able to swoop down to ground zero to tackle challenging fundamental technical problems. Theory helps develop the required mental flexibility.

In applied computing, today's amazing new app is tomorrow's brief mention. However, theoretical computer science attempts to grasp eternal knowledge that exists regardless of whether or not we ourselves exist. The goal is to open doors of understanding and hope that future generations will have the wisdom and capability to keep them open.

Course Topics

Finite Automata and Regular Languages Deterministic Finite Automata (DFA's), Nondeterministic Finite Automata (NFA's), DFA and NFA computations, designing DFA's and NFA's, converting an NFA to a DFA, regular languages and their closure properties, regular expressions, converting an NFA to a regular expression and vice versa (Kleene's Theorem), non-regular languages.

Context-Free Languages and Pushdown Automata Context-Free Grammars (CFG's), CFG derivations and derivation trees, ambiguity of grammars, inherent ambiguity of some CFL's, designing CFL's, Nondterministic Pushdown Automata (PDA's) and their equivalence with CFL's, designing PDA's.

Turing Machines Turing machine computations, designing Turing machines, nondeterministic turing machines, Turing decidability versus Turing recognizablility, recursive and recursively enumerable languages, other types of turing machines.

Decidability and Undecidability decidable problems for both DFA's and PDA's. Undecidable problems for PDA's and Turing machines.

Complexity Theory Mapping reducibilities, the complexity classes `P`, `NP`, and co-`NP`, `NP`-complete problems, Cook's Theorem, proving `NP`-completeness via mapping reducibility, other complexity classes.

Learning Outcomes

This course has twelve core **learning outcomes (LO's)** that will be regularly assessed throughout the semester. For each outcome, your understanding is assessed via a problem that is provided on an **assessment**, also commonly referred to as a *quiz* or *exam*. On your first attempt at demonstrating competency with respect to a learning outcome, you earn both points as well as a pass (P) or no pass (NP) grade. Receiving a pass grade means that you have successfully demonstrated competency with respect to that LO and are no longer required to demonstrate competency.

The following are some guidelines for success towards passing each of the learning outcomes.

1. When preparing for the learning-outcome problem, carefully read its official description in the syllabus so that you know what is expected for a passing mark.

2. Each LO has a set of core exercises (with solutions) to help you prepare. These exercises are located at the end of the lecture that introduces the LO.

3. During assessments, you are allowed to bring pens/pencils, eraser, and a non-programmable scientfic calculator **All other computing devices (cell phones, laptops, smart watches, etc.), notes, and books must be put away before starting. Before receiving the assessment you should have your space cleared and allow me to witness you putting your phone and watch away in a closed compartment. Failure to abide by these rules is grounds for receiving a final course grade of F without the possibility of having the grade forgiven**.

4. To help prevent cheating during in-class assessments, whenever possible, please select a seat that leaves one or more empty seats between you and your nearest neighbor. **When there is sufficient evidence of student plagiarism during an assessment, all involved students are subject to receiving a final course grade of F without the possibility of having the grade forgiven**.

5. Carefully read the directions for each problem and include all the pertinent steps of your solution. The problem statement is like a contract in that, so long as you provide exactly what is being asked of you and show sufficient work, then the outcome should be a passing mark. Please do not hesitate to raise your hand if you have any doubts about the contract is requiring of you.

6. Solve each LO assessment problem on a **SINGLE and SEPARATE** sheet of paper that includes your first name and surname (the one that determines your alphabetical order).

7. During assessments, both a five and two-minute warning will be announced before solutions are to be turned in. Please have the courtesy to turn in your work before the final call. A student who is still working after the final call will receive a warning and any subsequent infractions will result in the student's work not being graded.

8. At the end of the semester each student will receive grade points in accordance with the following table and based on the number of LO's passed (LO's 1-10 only).

Number Passed	**Grade Points**
0-3	0
4	0.5
5	1.0
6	1.5
7	2.0
8	3.0
9	3.5
10	4.0

The following are the LO's for which competency must be demonstrated.

LO1. The ability to design and draw the state diagram of a DFA that accepts a given regular language. The ability to provide the computation of the DFA on one or more input words.

LO2. The ability to i) provide the state diagram of an NFA that accepts a given language and only uses a prescribed number of states, ii) provide the δ-transition table for an NFA N that is defined via a state diagram, and iii) convert the table to an equivalent DFA M (i.e. M and N accept the same language).

LO3. Given the description of a regular language, the ability to provide a regular expression that correctly describes the language. An understanding of the basic language operations: including set-theoretic operations (union, intersection, complement, symmetric difference, and difference) as well as the concatenation and star operations.

LO4. The ability to demonstrate the equivalence between NFA's and regular expressions: convert a regular expression to an NFA that recognizes the language described by the expression, and convert an NFA to a regular expression that describes the language recognized by the NFA.

LO5. Given the description of a context-free language L, the ability to provide a context-free grammar G for which $L = L(G)$. Also, given a context free grammar G and a word $w \in L(G)$, the ability to derive w using the rules of G.

LO6. Given the description of some context-free language, the ability to design a pushdown automaton that accepts the language.

LO7. Given a Turing machine M, the ability to demonstrate a computation with respect to M and some input to M. An understanding of the deterministic Turing machine model of computation, including the ability to design a Turing machine for the purpose of having it behave in a specified way.

LO8. The ability to provide the definition of what it means to be a mapping reduction f from some problem A to another problem B, demonstrate one of the following reductions for some instance $x \in A$, and verify that the solution to $x \in A$ is the same as the solution to $f(x) \in B$.

(a) `Even` $\leq_m$ `Odd` See Exercise 1 of the Mapping Reducibility lecture for the kind of problem to expect.

(b) `Max Independent Set` $\leq_m$ `Max Clique`

(c) `Set Partition` $\leq_m^p$ `Subset Sum`

(d) `Hamilton Path` $\leq_m^p$ `LPath`

(e) `Subset Sum` $\leq_m^p$ `Set Partition`

(f) `Vertex Cover` $\leq_m^p$ `Half Vertex Cover`

(g) `Clique` $\leq_m^p$ `Half Clique`

LO9. An understanding of the complexity classes `P`, `NP`, and `co-NP`, including the ability to identify the appropriate complexity class for some decision problem, identify an appropriate certificate for an `NP` problem, and the ability to show that an `NP` verififer requires a polynomial number of steps.

LO10. The ability to demonstrate and/or answer questions about the following polynomial-time mapping reductions that establish the NP-completeness of some decision problem: 3SAT to Clique, 3SAT to Subset Sum, SAT to 3SAT, 3SAT to DHP, DHP to UHP, and Hamilton Cycle to Traveling Salesperson.

LO11. **Working General Knowledge.** A working knowledge of all the basic definitions and concepts introduced in the class. In other words, being able to recite the definitions and results and use them within the context of basic logic and problem solving.

LO12. **Advanced Ability.** The ability to apply and extend the core skills acquired from previous learning outcomes to solve more advanced problems.

Weekly Exercises

Associated with each learning outcome presented in lecture will be a set of assigned practice exercises, as well as some additional exercises that pertain to a related peripheral learning outcome and/or represent a more advanced exercise that may form the basis for an Advanced Ability assessment problem. Working General Knowledge problems are based on definitions and results taken directly from the lecture notes.

Class Meetings

Our class meetings will be devoted to the following three activities.

Course lecture notes These notes have several examples. Some have provided solutions while the solutions to others will be demonstrated in class. The notes also contain all the needed definitions, formulas, theorems, exercises, and exercise solutions that are needed to perform well in the class.

In-Class Assignments (ICA's) For some class meetings time will be alloted for students to work in groups of three to solve a collective problem that reinforces understanding of one of the course learning outcomes. Work on ICA's counts towards 10% of a student's final grade. A student must be present in class to receive credit for an ICA and all ICA assignment papers must be turned in by the end of the class meeting for which it was assigned. For the MW section, students should expect to have an ICA at every class meeting, except for the first week of class and the meeting of, the meeting before, and the meeting after a learning outcome assessment. For the Friday section, students should expect to have an ICA at every class meeting, except for the first meeting and the meetings for Learning Outcome Assessments 2-5. Note: there are **no makeups for ICA's**.

Learning Outcome Assessment (LOA) About once every three weeks (plus the final exam) there will be a learning outcome assessment. The assessment will have one problem for each of the learning outcomes that have been covered up to and including the previous week. Each assessment will feature one 25-point problem for assessing each of the following: i) working general knowledge (LO11), ii) advanced ability (LO12), and iii) each of the first ten learning outcomes that has been covered in lecture but *not previously assessed*. There will also be one "make up" problem for each of the previously assessed learning outcomes that have yet to be passed. For example, suppose LO's 1 and 2 were assessed on the first assessment, and LO's 3 and 4 are first assessed on the second assessment. Then the second assessment will have four 25-point problems (LO's 3 and 4, General Working Knoweldge, Advanced Ability) and two makeup problems (LO's 1 and 2). MW students will have the entire class period to complete each LOA, while Friday students will have the entire class period for LOA's 2-5. Important: both Advanced Ability and Working General Knowledge problems are based on knowledge obtained *after* the previous learning outcome assessment. The final exam will have one extra cumulative General Working Knowlege problem worth 50 points. The fifth assessment will not have any Advanced Ability or General Working Knowledge problems.

LOA Dates (MW Class): September 16th, October 7th, October 28th, November 18th, December 9th, December 18th (8:00-10:00 am)

LOA Dates (Friday Class): September 18th, October 9th, October 30th, November 20th, December 11th, December 18th (10:15-12:15 am)

Learning Outcome Assessment Makeups

Since there are a total of six (including final exam) assessments, **there are no makeups for LOA's**. However, if a student has to miss a single LOA (excluding the final exam) due to a documented accident, emergency, or illness, any missed points will be added at the end of the semester, and the zero-point totals will be replaced with values that reflect the student's overall performance during the semester. For example, if during the semester the student attained one half of the total Advanced Ability points, then the zero will be replaced with 12.5 (i.e. one half of 25 points). Acceptable documentation includes a note or letter from a medical establishment that explains the emergency or illness, a police report, and/or insurance documentation. Please note that **this is one-time forgiveness and no such accomodation will be made for missing subsequent assessments**.

Supplemental Lectures

Because there will be in-class activities throughout the semester, in order to keep pace with the course objectives, a supplemental lecture recording may be provided at the end of the week. The lecture will last anywhere between 30 to 60 minutes. It's important to review the lecture before the next class meeting, as the following in-class assignment may depend on it.

Reading Assignments

A reading assignment will be provided on most weeks of the semester. Reading the textbook will offer a somewhat alternative and more comprehensive viewpoint of the subject matter. Please check the "Reading Assignments" link at the course website for the current and past assignments. The reading assignment topics pertain to those topics that will be covered in the class meetings for the following week.

Final Grade Determination

At the end of the semester the number of LO's 1-10 that a student passed will be converted to grade points (see section on learning outcomes). Also, each of the total points earned for ICA's, General Working Knowledge, Advanced Ability, and LO's 1-10 will be converted to separate grade points. These five grade points are then weighted as follows in order to determine your final course grade points.

Category	Percentage Weight
Learning Outcomes Passed	30%
Learning Outcome Total Points	25%
General Working Knowledge	20%
Advanced Ability	15%
In-Class Assignments	10%

The grade points earned for each category are weighted according to the percentage weight to obtain the overall course grade point average (OGPA)

$$\mathrm{OGPA} = 0.3 \times \mathrm{LOP} + 0.25 \times \mathrm{LOTP} + 0.2 \times \mathrm{GWK} + 0.15 \times \mathrm{AA} + 0.1 \times \mathrm{ICA}$$

which is rounded to the nearest tenth. The grade point average is then converted to a letter grade: A (3.5 and above), B (2.5 to 3.4), C (1.8 to 2.4), D (0.5 to 1.7), and F (0-0.5).

Registration Deadlines

September 7th Last day to add or drop classes without approval

September 21st Deadline to add a course

November 20th Withdrawal deadline