Lecture Notes for CSCI 341: Theory of Computation

Set 1-Intro

Professor Talmage

August 28, 2024

---

1 Motivating Exercises

Exercise: Discuss with a neighbor: What is a problem that a computer cannot solve?

- Is your problem something that would appear to be a computing problem? (This question is easier if it is not...)
- Why is it impossible? Is it something in the problem or in the computer that makes it impossible?
- Can you name a well-defined computing problem a computer cannot solve? (What is a "computing problem"?)
- Is there an entire domain of problems computers cannot solve?

This course is about building the definitions and tools to be able to answer these questions. We will formally define "problem" and "computer" to discuss (and prove!) what a computer can and cannot do.

Exercise: Consider the logical flow of determining whether a number is divisible by 11. There is a trick for this: Alternately add and subtract digits. If the result is divisible by 11, then the original number was divisible by 11. For example, consider 1850378549032. It is not at all clear whether this is divisible by 11, but the procedure is fairly straightforward:

$$1 - 8 + 5 - 0 + 3 - 7 + 8 - 5 + 4 - 9 + 0 - 3 + 2 = -9$$

-9 is not divisible by 11, so we know the original number is not, either.
Create a flow chart to demonstrate this procedure for determining divisibility by 11. You should not have many more than 5 states.
What happens if the number is huge, and the result of the procedure is not obviously divisible by 11? (e.g. 909090909090909090909) Can you add just one or two edges to your flow chart to handle this case?

The mathematical models of computers we will look at this semester closely resemble the flowchart we just developed for applying the divisibility algorithm. With a few tweaks, this is actually all we need to represent the power of a computer, and then we will be able to properly discuss what we can and cannot compute.

2 Administrivia

Some notes on the syllabus and the expectations in this course:

- Office Hours. Listed in the syllabus, check there for changes. Should also be publicly visible on my google calendar. Reach out (email me) if you need to meet at another time.
- Communication:
  - Email is primary. Check yours! If sending me a class-related message, make sure to include "CSCI 341" in the subject line so that it is properly sorted. Otherwise, I am more likely to forget to respond when it gets buried.
  - I'm happy to answer small questions over Discord, though larger questions are generally better in person or over email.
  - I will generally respond fairly promptly during working hours, though sometimes I am in class, a meeting, office hours, etc. and will get to it when that is over. Do not expect a response in the evening or over the weekend. If I am working during those times, I may respond, but I will most likely leave it until the next working day.
- Accommodations: OAR, religious, athletic, etc. please inform me ASAP of needs or conflicts.
- Attendance: I expect you to be in every class, and to interact both with me and with each other. Your engagement is a significant portion of your grade. The exception is that you should not come to class if you are sick. Rest up, get well, and reach out to me to catch up on any missed material.
- Assignments:
  - Homework: Weekly, due Wednesday night at 11:59PM.
    - This is where you cement knowledge, by working hard to use it.
    - You can work alone or in a group. If you work in a group, submit only one copy, and be sure that everyone's name is both on the assignment and is added to the gradescope submission. If you work in a group on only some problems, credit the entire group on that problem in your writeup.
    - 10% per day late penalty, max two days.
    - Typeset your solutions with LaTeX. Recitation this week will get you up and running. Also, ask me and others in the class for help.
    - Grading: I will choose one problem per week on which I will provide feedback. The other problems will be graded on a 0/1/2 completion/completeness basis.
  - Quizzes: Weekly, posted Friday afternoon, due Monday night at 11:59PM. 30 minutes on gradescope. *Individual-only!*
    - High-level checks that you are keeping up with material from class that week.
    - Must be entirely solo!
    - Each question is one point, no partial credit. For each question you get wrong, do the same-number problem on the retake available during the next week.
  - Three exams. Planned for Sep. 30, Nov. 4, Dec. 19 (11:45-2:45), midterms may move.
  - Recitation: Interactive, group-based problem solving sessions. You will not submit anything, your grade is based on participation.

3 Course Outline

We will start with a review of some of the mathematical language we need to speak about computing. Then, we start our exploration by building a model of a computer. To do this, we will start with a simple computing machine for classifying strings. We will find that machine's limits, then strengthen it and repeat until we have a computer. Such simple models serve two purposes.

1. If I unloaded a full-blown model of computation on you today, we would all be completely lost and bewildered. By building up, the final model will seem natural.

2. Simple models of computation are sufficient for many tasks. Once you are familiar with them, you can use simpler tools for applicable problems, which is more efficient, more elegant, and easier to prove correct.

Once we have a complete model of computation, we will explore the limits of what it *can* do, then turn our attention to efficiency and discuss what it can do *efficiently*. Formal models are useful again to allow us to make general claims of what problems can and cannot be solved and how efficiently.