ECS 122B: Conceptual Homework #5

Instructor: Aaron Kaloti

Summer Session #2 2020

1 Changelog

You should always refer to the latest version of this document.

- v.1: Initial version.

2 Grading

- **Due date: 5:00 PM on Wednesday, 09/09. There is no "grace period."**
- A subset of the problems will be graded for correctness. The rest will be graded on completion.

3 Submission Requirements

- Your submission must be created with LaTeX. **Handwritten/scanned solutions, or solutions not typed with LaTeX, will earn no credit.**
  - You should avoid doing this assignment at the last minute, because it will take a bit of time to get used to LaTeX and to type your answers up into a LaTeX document.
- Your submission must consist of one file, the PDF generated by your LaTeX file (`hw5_answers.pdf`). Since Gradescope does not let you submit files that are neither PDFs nor images, I will not have you submit the `.tex` file. However, if it is clear that your submission was not made with LaTeX, we may email you to ask that you send us your `.tex` file, and if you cannot (i.e. if it turns out that your PDF was in fact not created with LaTeX), you will get a zero on this assignment.
- You may be penalized if, when submitting on Gradescope, you mark the wrong page for a given homework problem, because dealing with this slows down grading. At the beginning of the 08/06 lecture, I talk about how to mark the pages of your submission on Gradescope, starting at around 3:20 in the video.
- When using LaTeX, you should make use of the math notation/mode where sensible. (The math mode refers to when you put mathematical equations, etc. between dollar signs so that LaTex makes them look nice.) Repeated failures to do this, to the point that the purpose of using LaTeX is defeated or the readability of your answers is impeded, may result in a penalty on this assignment.

4 Regarding Collaboration

- You may not copy answers from any sources, including online sources such as Chegg, StackOverflow, or any solutions manual of any textbook.
- You may partner with ***at most one*** other student on this assignment. In other words, you can work in *pairs*. You do not have to partner with anyone. (In fact, I think it is better that you do not.) If you partner with someone, it must be a committed partnership; that is, you two will have the same submission, and you must mark on Gradescope that you have partnered for this assignment by following the directions here.
- If students that were not in the same pair seem to have excessively similar answers, they will be reported to the OSSJA for suspicion of academic misconduct. Do not copy answers from (or share answers with) any student who you are not partnered with.

5 Identification

Enter the members of your pair. (You can partner with at most one other student.) If you are not partnered with anyone, then leave the second box empty. You can remove the use of `\vspace` in the `.tex` file.

Pair member #1:

Julio Beas

Pair member #2:

Han Nguyen

6 Problems

Place your answer into the answer boxes; you can remove the use of `\vspace` in the answer boxes in the `.tex` file. If you are using your own LaTeX template, you may find the `setcounter` directive useful; it lets you set the number of the next section, and you can read more here.

6.1 SAT Satisfying Assignment

Find a satisfying assignment (of the $n = 4$ variables) for this SAT instance. If there is none, then say so.
$(x_1 \vee \bar{x_2} \vee x_3) \wedge (x_4 \vee x_2 \vee \bar{x_1}) \wedge (\bar{x_3} \vee \bar{x_4}) \wedge (x_3 \vee x_2 \vee x_4) \wedge (x_3 \vee \bar{x_4})$

Letting $x_1 = false, x_2 = false, x_3 = true, x_4 = false$ will satisfy the SAT assignment as it makes the statement result in $true$.

6.2 Using the 2-SAT Randomization Algorithm

Suppose that we have a formula with the below clauses (each of them are ANDed together, of course).

1. $(x_1 \vee x_2)$
2. $(x_1 \vee \bar{x_4})$
3. $(x_3 \vee x_4)$
4. $(\bar{x_1} \vee \bar{x_3})$
5. $(\bar{x_4} \vee \bar{x_1})$
6. $(x_2 \vee \bar{x_1})$.

Suppose that we are using the randomization algorithm to find a satisfying assignment. Our RNG (random number generator) generates the following stream of integers: $2, 1, 2, 1, 1, 2, 1$. A 1 indicates that the left term of the chosen clause should be flipped, and a 2 indicates that the right term should be flipped. When arbitrarily choosing an unsatisfied clause, you should choose the unsatisfied one with the lowest number (using the numbers in the list above, e.g. the first clause has #1). **Provide the current truth assignment after each iteration of the algorithm.** It is possible that not all numbers in the given stream of integers will be used. **If a truth assignment is not found by the time the stream of seven randomly generated integers has been exhausted, then stop after the final step and say what the final unsatisfying assignment is**. The initial truth assignment is shown below.

- x_1: True
- x_2: False
- x_3: False
- x_4: False

- Iteration 1: $(x_3 \vee x_4)$ is unsatisfied. By the random integer 2, we will set $x_4 =$ True.
- Iteration 2: $(\bar{x_4} \vee \bar{x_1})$ is unsatisfied. By the random integer 1, we will set $x_4 =$ False.
- Iteration 3: $(x_3 \vee x_4)$ is unsatisfied. By the random integer 2, we will set $x_4 =$ True.
- Iteration 4: $(\bar{x_4} \vee \bar{x_1})$ is unsatisfied. By the random integer 1, we will set $x_4 =$ False.
- Iteration 5: $(x_3 \vee x_4)$ is unsatisfied. By the random integer 1, we will set $x_3 =$ True.
- Iteration 6: $(\bar{x_1} \vee \bar{x_3})$ is unsatisfied. By the random integer 2, we will set $x_3 =$ False.
- Iteration 7: $(x_3 \vee x_4)$ is unsatisfied. By the random integer 1, we will set $x_3 =$ True.

The final truth assignment is $(x_1 \vee x_2) \wedge (x_1 \vee \bar{x_4}) \wedge (x_3 \vee x_4) \wedge (\bar{x_1} \vee \bar{x_3}) \wedge (\bar{x_4} \vee \bar{x_1}) \wedge (x_2 \vee \bar{x_1})$ with $x_1 =$ True, $x_2 =$ False, $x_3 =$ True, $x_4 =$ False. This is an unsatisfied assignment, since $(x_2 \vee \bar{x_1})$ returns False.

6.3 Understanding the 2-SAT Randomization Algorithm

Whenever a slide number is mentioned, it is talking about the slides on probability and randomized algorithms.

6.3.1 Transition Probabilities

Explain why, on slide 36, it is that $\Pr(X_{i+1} = j+1 | X_i = j) \geq \frac{1}{2}$ and $\Pr(X_{i+1} = j-1 | X_i = j) \leq \frac{1}{2}$. Specifically, why do they both not *equal* $\frac{1}{2}$?

The reason is because $\Pr(X_{i+1} = j+1 | X_i = j)$ and $\Pr(X_{i+1} = j-1 | X_i = j)$ is not a constant. They may return 1/2 and other times return different values like 0 or 1 depending on past values.

6.3.2 h_0

In slide #38, why does $h_0 = h_1 + 1$?

Can inductively show that $\forall 0 \leq j \leq n-1, h_j = h_{j+1} + 2j + 1$. If we have h_0, it will be $h_0 = h_1 + 2 * 0 + 1 = h_1 + 1$.

6.4 Verifying Polynomial Identities

In the problem of verifying polynomial identities, we are given two polynomials, $F(x)$ and $G(x)$, and we wish to verify that these polynomials are equal. $F(x)$ is written as a product of d monomials $\prod_{i=1}^{d}(x - a_i)$, where d is the degree of $F(x)$ and $G(x)$. $G(x)$ is written in its canonical form $\sum_{i=0}^{d} c_i x^i$. For example, we may have the following:

$$F(x) \stackrel{?}{=} G(x)$$

$$(x+1)(x-2)(x+3)(x-4)(x+5)(x-6) \stackrel{?}{=} x^6 - 7x^3 + 25$$

6.4.1 Deterministic Algorithm

One obvious approach for verifying the equality of $F(x)$ and $G(x)$ is to convert $F(x)$ to its canonical form and then see if the result matches $G(x)$. That is, we consecutively multiply the ith monomial with the product of the first $i-1$ monomials. In between multiplications of monomials, we add terms that can be added together, e.g. $x^2 + x - 2x - 2$ would become $x^2 - x - 2$ before we multiply by the next monomial. In big-Θ notation and in terms of d, how many multiplications are done in this approach in the worst case[1]? Explain.

[1]In this context, "worst case" would merely imply that 0 is not a root of the polynomial, meaning that there is no monomial $x - a_i$ such that $a_i = 0$.

This approach will result in $\Theta(d^2)$ multiplications in the worst case. We say that as $F(x)$ is the multiplication of ith monomial with the product of the first $i-1$ monomials, we have the total of $(i)(i-1)$ where i goes to d. Hence, there are (d^2-d) multiplications of coefficients; therefore it is $\Theta(d^2)$.

6.4.2 Randomized Algorithm

We can use a randomized algorithm to solve this problem in $\Theta(d)$ time. Below are the steps of the algorithm:

- Choose a random integer in the range $[1, 100d]$.
- Compute $F(r)$ and $G(r)$.
- If $F(r) \neq G(r)$, report that the two polynomials are not equivalent.
- If $F(r) = G(r)$, report that the two polynomials are equivalent.

This algorithm has a chance of erroneously reporting the wrong answer when $F(x)$ and $G(x)$ are not equivalent. Show that the probability of this occurring (when $F(x)$ and $G(x)$ are not equivalent) is at most $\frac{1}{100}$.

Hint: You need to figure out the maximum number of numbers that could cause $F(x)$ and $G(x)$ to be equal when they are not equivalent polynomials. Rearranging the equation $F(x) = G(x)$ may help. You may also need to review the "fundamental theorem of algebra", although it is easy to stumble into overly complex definitions of this theorem.

By the Fundamental Theorem of Algebra, we say that r is a root of the polynomial $F(x) - G(x)$ with its degree of at most d. Hence, since there are at most d choices of r so that $F(r) = G(r)$ whereas $F(x) \neq G(x)$, and d ranges from 1 to 100, we say that the probability of choosing such r is $\frac{1}{100}$.