

CECS 329, Homework Assignment 2, Fall 2025, Dr. Ebert

Directions: Please review the Homework section on pages 5 and 6 of the syllabus including a list of all rules and guidelines for writing and submitting solutions.

Due Date: Saturday September 13th as a PDF file upload to the HW2 Canvas dropbox.

Problems

1. Automated reasoning tools have found good use in the area of planning and scheduling, especially in support of mission-critical operations. For example, NASA uses automated planning and scheduling to operate spacecraft, manage robotic rovers, and run the Deep Space Network. The 2SAT algorithm is an example of an automated-reasoning algorithm because it automates the basic logical inferences that humans sometimes make when making decisions. For example, six employees, Amy, Bao, Carlos, Daniel, Elias, and Francis are part of the workforce for a large-scale project. The project supervisor must know which of them will attend work on Saturday to help the project stay on schedule. The following are the constraints that she is faced with.
 - If Carlos does not attend, then Elias will also not attend since they both had planned to spend the day together.
 - If Daniel attends, then Amy will not attend because in that case she will look after their children.
 - Bao and Carlos will either both attend or both not attend because their project positions are highly dependent on each other.
 - Either Amy or Bao must attend but not both, because their skill sets are almost identical and there is only need for one of them.
 - If Carlos attends then Daniel will not attend because Daniel's job has only a small amount of work left, and Carlos will be able to cover for him.
 - Either Daniel or Elias (or both) will attend.
 - Either Francis or Elias will attend, but not both.
 - If Amy attends then Francis will not attend.
- a. Using the variables $A, B, \dots, F$, Translate each of the above eight statements to a Boolean formula. For example, A stands for "Amy will attend work on Saturday". Next to each formula, write the implication formulas that are implied by the formula. Hint 1: use $\leftrightarrow$ and $\oplus$ for equivalence and exclusive-or. Hint 2: the above statements imply 22 different implication formulas. (10 pts)
- b. Use the implications from the previous part to construct the implication graph associated with the logic problem. (5 pts)

- c. Perform the **Improved 2SAT** algorithm by computing the necessary reachability sets. Use alphabetical order and positive literal before negative literal when choosing the reachability set to compute next. Draw the resulting reduced **2SAT** instance whenever a consistent reachability set is computed. Either provide a final satisfying assignment for $\mathcal{C}$ or indicate why $\mathcal{C}$ is unsatisfiable. Remember that there is no need to compute a reachability set for some literal l if l has already been assigned a truth value. (15 pts)
2. This problem is inspired by my supervisor at Arcadia Design Systems who told me “linear and log-linear [algorithms]: good. Anything else, be very careful how you use it”. Suppose that a computer’s cpu is capable of executing a single instruction in 5×10^{-10} seconds. The plan is to run a time-intensive program on the computer for one full week. For each of the following scenarios determine the largest problem size n that can be solved during this time. Show all steps for full credit.
- a. The program solves instances of **3SAT** and implements the best known **3SAT** algorithm which requires about $50 \left(\frac{4}{3}\right)^n$ instructions to solve a **3SAT** instance that depends on n variables. (7 pts)
 - b. The program solves instances of **Matrix Multiplication** and requires about $75n^3$ instructions to multiply two $n \times n$ matrices. **Side note:** although n is a natural size parameter for **Matrix Multiplication**, an instance of the problem is two $n \times n$ matrices, and hence n^2 better representsts the actual instance size. Thus, **Matrix Multiplication** officially requires $O(m^{\frac{3}{2}})$ steps, where $m = n^2$. In any case, the traditional algorithm for multiplying matrices still grows cubically with respect to the dimension of the input matrices. (7 pts)
 - c. The program solves instances of **Matrix Addition** and requires about $50n^2$ instructions to multiply two $n \times n$ matrices. (6 pts)