Exercise 0F-2. Set Theory [5 points]. This answer should appear after the first page of your submission and may be shared during class peer review.

This exercise is meant to help you refresh your knowledge of set theory and functions. Let X and Y be sets. Let $\mathcal{P}(X)$ denote the powerset of X (the set of all subsets of X). There is a 1-1 correspondence (i.e., a bijection) between the sets A and B, where $A = X \to \mathcal{P}(Y)$ and $B = \mathcal{P}(X \times Y)$. Note that A is a set of functions and B is a (or can be viewed as a) set of relations. This correspondence will allow us to use functional notation for certain sets in class. This is Exercise 1.4 from page 8 of the Winskel textbook.

Demonstrate the correspondence between A and B by presenting an appropriate function and proving that it is a bijection. For example, you might construct a function $f : B \to A$ and prove that f is an injection and a surjection.

Answer: Let's construct a function $f : A \to B$, $f(a) = \{(x, y)|y \in a(x)\}$.
If $f(a_1) = f(a_2)$, by definition $\{(x, y)|y \in a_1(x)\} = \{(x, y)|y \in a_2(x)\}$. By the axiom of extensionality in Set Theory, two sets are equal if they have exactly the same elements. So for any x, $a_1(x) = a_2(x)$. Thus f is injective.
Let $b \in \mathcal{P}(X \times Y)$, so every element of b is of the form (x, y) with $x \in X$ and $y \in Y$. We construct an a such that $f(a) = b$. By definition of f, $f(a) = \{(x, y)|y \in a(x)\}$. Let $a(x) = \{y|(x, y) \in b\}$, so $f(a) = \{(x, y)|y \in \{y|(x, y) \in b\}\}$, which simplifies to $f(a) = \{(x, y)|(x, y) \in b\}$. So $f(a) = b$, thus f is surjective.
Since f is injective and surjective, it is also bijective.

Exercise 0F-3. Model Checking [10 points]. This answer should appear after the first page of your submission and may be shared during class peer review.

Download the CPAChecker software model-checking tool using the instructions on the homework webpage. Read through enough of the manual to run the tool on the `tcas.i` testcase provided on the homework webpage. Check the three properties given. For each command, copy or screenshot the last ten non-empty lines of output from CPAChecker and include them as part of your answer to this question.

In this graduate-level class, it is your responsibility to get CPAChecker up and running properly. This may require you to set up a virtual machine. This level of systems programming experience is a prerequisite for the class and is intentionally part of an early assignment (i.e., before the drop deadline) to help students determine if they have the right incoming preparation.

Hint: if your output when checking `Property1a` does not indicate something like "Verification result: FALSE. Property violation (error label in line 1963) found by chosen configuration." then you may not have set things up correctly.

In at most three paragraphs, summarize your experience with the CPAChecker tool.

- What is going on when you run CPAChecker using the commands listed? What does `Property1a` mean? Is `tcas.i` a reasonable test suite? What has been proved? (This is the heart of the question. You may have to read ugly code, understand a legacy

Question assigned to the following page: 3

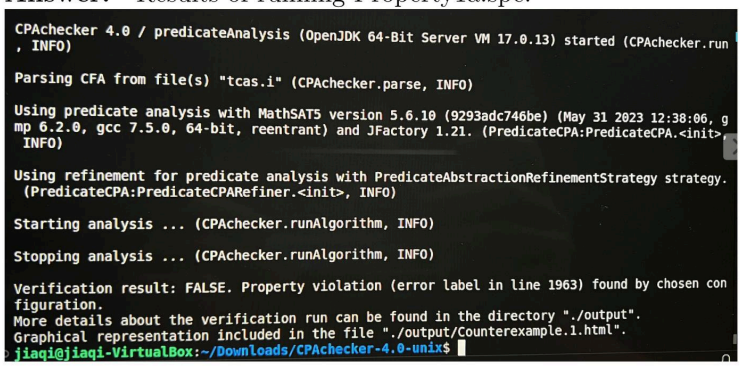
tool, and apply concepts from class. It is expected that answering this well will require time.)

- Did you find CPAChecker to be a usable tool? How easy is it to provide the inputs to CPAChecker? What information is present in the graphical (HTML) output?

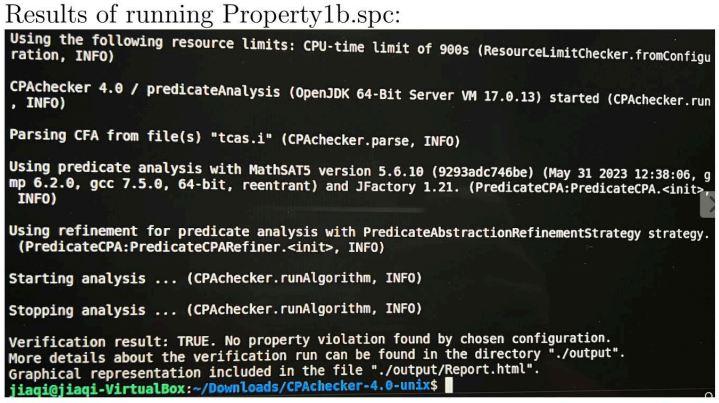
For full credit, do not restate the lecture on counter-example guided abstraction refinement; instead, discuss your thoughts and experience using this tool (including its input requirements, output guarantees, and context). Focus on threats to validity (e.g., imagine that you were writing a paper and using this as an experiment) over usability.

Both your ideas and also the clarity with which they are expressed (i.e., your English prose) matter. You can use a tool like ChatGPT to refine your prose, but be wary of false claims. A reader should be able to identify your main points, the arguments you are making, and your conclusion.

Answer: Results of running Property1a.spc:

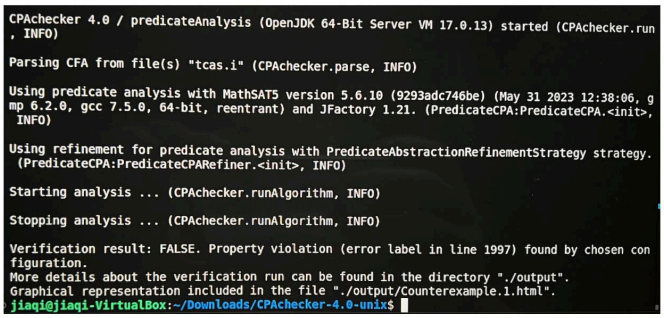


Results of running Property1b.spc:



Results of running Property2b.spc:

Question assigned to the following page: 3



When I ran CPAChecker using the commands listed, it parsed CFA from the files and did the analysis. Then, it produced html files for the graphical representation. `Property1a` means this location's upper separation is bigger than the threshold and its down separation is smaller than the threshold. Although tcas.i seems ugly, the main function of tcas.i is just a bunch of if statements. So, I think it is a reasonable test suite. It proved that CPAChecker detects error locations that are specified by the label "PROPERTY1a".
I think CPAChecker is a usable tool. It is very easy to provide the inputs to CPAChecker using the command lines if you know where are the files you want to test. In the graphical output, we can find the control-flow automata (CFA) of the input program, the abstract reachability graph (ARG) that was constructed by the chosen configuration, statistics, and an error path that violates the specification if the verdict is FALSE.

Submission. Turn in your assignment as a single PDF document via the `gradescope` website. Your name and Michigan email address must appear on the first page of your PDF submission but may not appear anywhere else.

Peer Review ID: 300135193 — enter this when you fill out your peer evaluation via gradescope