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) bewteen 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 Construct a function $f : (X \to \mathcal{P})) \to \mathcal{P}(X \times Y)$

$$f(g) = \{(x, y)|y \in g(x)\}$$

First, to prove f is injective,
we have to show that for all $g_1 \in A$ and $g_2 \in A$ if $f(g_1) = f(g_2)$, then $g_1 = g_2$. Assume $f(g_1) = f(g_2)$, and by the definition of f, we know that $f(g_1) = \{(x, y)|y \in g_1(x)\} = f(g_2) = \{(x, y)|y \in g_2(x)\}$. Therefore, for any pair (x, y), we know that if $y \in g_1(x)$, then $y \in g_2(x)$. Therefore, for any x, $g_1(x)$ and $g_2(x)$ must be equal sets. Thus, f is injective.
Then prove that f is surjective.
We have to show that for every $b \in B$, there exists a function $g \in A$ such that $f(g) = b$. Let $b \in B$ be an arbitrary element. By the definition of B, we know that $b \in \mathcal{P}(X \times Y)$. Therefore, b is a set with element (x, y), in which $x \in X$ and $y \in Y$. We can construct a function $g(x)$, such that $g(x) = \{y|(x, y) \in b\}$. Then we know that

$$f(g) = \{(x, y)|y \in g(x)\} = \{(x, y)|(x, y) \in b\} = b$$

Hence, f is surjective.
Since we have proved that f is both injective and surjective, f is 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.

It is your responsibility to find a machine on which CPAChecker works properly (but feel free to check the class forum if you are getting stuck).

Hint: CPAChecker 2.0 should find a violation for `Property1a`, verify that `Property1b` is safe, and find a violation for `Property2b`. If your output does not match that and you are using version 2.0 then you may not have not set things up correctly.

What is going on when you run CPAChecker using the commands listed? In at most three paragraphs, summarize your experience with the CPAChecker tool. What does `Property1a` mean? Is `tcas.i` a reasonable test suite? What has been proved? Did you find CPAChecker to be a usable tool? You may find the graphical reporting option of CPAChecker to be helpful here. For full credit, do not restate my lecture on counter-example guided abstraction refinement; instead, discuss your thoughts and experience using this tool. 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. A reader should be able to identify your main claim, the arguments you are making, and your conclusion.

Results for Property1a

Using the following resource limits:CPU-time limit of 900s(ResourceLimitChecker.fromConfiguration, INFO)
CPAchecker 2.0 / predicateAnalysis (OpenJDK 64-Bit Server VM 11.0.9.1) started (CPAchecker.run, INFO)
Parsing CFA from file(s) "tcas.i" (CPAchecker.parse, INFO)
Using predicate analysis with MathSAT5 version 5.6.5 (63ef7602814c) (Nov 9 2020 09:01:58, gmp 6.1.2, gcc 7.5.0, 64-bit, reentrant) and JFactory 1.21. (PredicateCPA:PredicateCPA.¡init¿, INFO)
Using refinement for predicate analysis with PredicateAbstractionRefinementStrategy strategy. (PredicateCPA:PredicateCPARefiner.¡init¿, INFO)
Starting analysis ... (CPAchecker.runAlgorithm, INFO)
Stopping analysis ... (CPAchecker.runAlgorithm, INFO)
Verification result: FALSE. Property violation (error label in line 1963) found by chosen configuration.
More details about the verification run can be found in the directory "./output".
Graphical representation included in the file "./output/Counterexample.1.html".

Results for Property1b

Using the following resource limits:CPU-time limit of 900s(ResourceLimitChecker.fromConfiguration, INFO)
CPAchecker 2.0 / predicateAnalysis (OpenJDK 64-Bit Server VM 11.0.9.1) started (CPAchecker.run, INFO)
Parsing CFA from file(s) "tcas.i" (CPAchecker.parse, INFO)
Using predicate analysis with MathSAT5 version 5.6.5 (63ef7602814c) (Nov 9 2020 09:01:58, gmp 6.1.2, gcc 7.5.0, 64-bit, reentrant) and JFactory 1.21. (PredicateCPA:PredicateCPA.¡init¿, INFO)
Using refinement for predicate analysis with PredicateAbstractionRefinementStrategy strategy. (PredicateCPA:PredicateCPARefiner.¡init¿, INFO)
Starting analysis ... (CPAchecker.runAlgorithm, INFO)
Stopping analysis ... (CPAchecker.runAlgorithm, INFO)
Verification result: TRUE. No property violation found by chosen configuration.
More details about the verification run can be found in the directory "./output".
Graphical representation included in the file
"./output/Report.html".

Results for Property2b

Using the following resource limits:CPU-time limit of 900s (ResourceLimitChecker.fromConfiguration,INFO)
CPAchecker 2.0 / predicateAnalysis (OpenJDK 64-Bit Server VM 11.0.9.1) started (CPAchecker.run, INFO)
Parsing CFA from file(s) "tcas.i" (CPAchecker.parse, INFO)
Using predicate analysis with MathSAT5 version 5.6.5 (63ef7602814c) (Nov 9 2020 09:01:58, gmp 6.1.2, gcc 7.5.0, 64-bit, reentrant) and JFactory 1.21. (PredicateCPA:PredicateCPA.¡init¿, INFO)
Using refinement for predicate analysis with PredicateAbstractionRefinementStrategy strategy. (PredicateCPA:PredicateCPARefiner.¡init¿, INFO)
Starting analysis ... (CPAchecker.runAlgorithm, INFO)
Stopping analysis ... (CPAchecker.runAlgorithm, INFO)

Verification result: FALSE. Property violation (error label in line 1997) found by chosen configuration.
More details about the verification run can be found in the directory "./output".
Graphical representation included in the file "./output/Counterexample.1.html".

`Property1a` means if there is a label "PROPERTY1A" in the execution path, then it will goes to the ERROR state. I think CPAChecker is a very powerful tool for configurable software verification. The graphical representation is very helpful for visualizing the content flow of the program. The abstract reachability graph clearly shows the abstract state. The statistics presents some interesting data for example PredicateCPARefiner statistics, which allows us to check the refinement procedure.

Overall, I think `tcas.i` is a good test suite for CPA checker. Firstly, it can test the reachability of a certain error location under certain variable and provide the error path when the error is detected. Secondly, the content flow automation of `tcas.i` is relatively complex, which involves many variables.

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.

1 HW0

- **0 pts** Correct