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:

Injection 1:

Let us propose the following function $F : A \to B$. Let $a \in A$. Each $a_i \in a$ has the following form:

$$x \to p, \text{ where } x \in X, \text{ and } p \in P(Y)$$

We therefore define

$$F = \{(x, p_i) \mid p_i \in p, \text{ and } (a_j = x \to p) \in a, \text{ and } a \in A\}$$

We must show $m \neq n \implies F(m) \neq F(n)$ where $m, n \in A$.

Let $a, a' \in A \ \wedge a \neq a'$. We can write $a_i \in a$ as $a_i = x_i \to p_i$ with $x_i \in X$ and $p_i \in P(Y)$. Similarly, we can write $a'_i \in a'$ as $a'_i = x'_i \to p'_i$ with $x'_i \in X$ and $p'_i \in P(Y)$.

We know from the definition of A that $\cup_i \, x_i = \cup_i \, x'_i$. Therefore, $a \neq a' \implies \cup_i \, p_i \neq \cup_i \, p'_i$. This means that

$$\{(x_i, p_{i_j}) \mid p_{i_j} \in p_i, (a_i = x_i \to p_i) \in a\}$$
$$\neq$$
$$\{(x'_i, p'_{i_j}) \mid p'_{i_j} \in p'_i, (a'_i = x'_i \to p'_i) \in a'\}$$

We thus have that $a \neq a' \implies F(a) \neq F(a')$ where $a, a' \in A$. F is therefore injective.

Injection 2:

Let us now propose a second function $G : B \to A$.

Let us define

$$G = \{x \to \emptyset \mid x \in (X - \{m \mid (m, n) \in b\}) \} \cup \{x \to \{n \mid (m, n) \in b, m = x\} \mid (x, y) \in b\}$$

Now let us consider $b, b' \in B$ where $b \neq b'$. This means that b and b' differ by at least one pair $(x, y), x \in X, y \in Y$. Without loss of generality, let us say that b' contains this differing pair.

If $x \notin \{m \mid (m, n) \in b\}$, then $(x \to \emptyset) \in F(b')$ but $(x \to \emptyset) \notin F(b)$.

Otherwise, if $x \in \{m \mid (m,n) \in b\}$, the function in $F(b')$ that begins with $x \rightarrow$ will point to a set that contains one more element (the element y) than the function in $F(b)$ that begins with $x \rightarrow$.
Therefore, we have that $b' \neq b \implies F(b') \neq F(b)$. G is therefore injective.

Conclusion:
We have that $F : A \rightarrow B$ in injective and $G : B \rightarrow A$ is injective. Thus, by the Schröder–Bernstein theorem, we know that there exists a bijection between A and B.

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.

Answer:
CPAChecker is a straightforward and usable tool that analyzes programs and determines whether they can reach an error state based on user-provided conditions. In my usage of the tool, I tested whether `Property1a` could be violated in the `tcas.i` program. `Property1a` ensures that the variable `thresh` respects the condition `Down_Separation` $<$ `thresh` $\leq$ `Up_Separation`.

`tcas.i` is a reasonable test suite as it emulates a real-world use case effectively: indeed, is checks to see if an important property maybe be violated during the execution of a program. This test suite has proved that there exists a path in this program where `Property1a` may

be violated, causing the program to enter an error state.

I found CPAChecker to be a usable tool. The main pain-point was the installation: there was no simple `apt-get` command that could be run to install it simply. However, they did make Docker versions available for ease of use. The graphical reporting option was of great help and gave detailed insights into the causes of the potential errors. However, this tool is thorough and contains a lot of information and is probably best suited for individuals quite familiar with the tool. Overall, CPAChecker is a powerful tool that gives insight into potential error states in a program in an accessible manner.

Appendix:

Property1a

```
   Using the following resource limits:  CPU-time limit of 900s (ResourceLimitChecker.fr
INFO)
   CPAchecker 2.0 / predicateAnalysis (OpenJDK 64-Bit Server VM 11.0.5) 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.  (PredicateCP
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".
```

Property1b

```
Using the following resource limits:  CPU-time limit of 900s (ResourceLimitChecker.fromC
INFO)
   CPAchecker 2.0 / predicateAnalysis (OpenJDK 64-Bit Server VM 11.0.5) 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.  (PredicateCP
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".

Property2b
Using the following resource limits:  CPU-time limit of 900s (ResourceLimitChecker.fromC
INFO)
   CPAchecker 2.0 / predicateAnalysis (OpenJDK 64-Bit Server VM 11.0.5) 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.  (PredicateCP
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".
```

---

1 HW0

- **0 pts** Correct