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.

My Answer:

We construct a function $f : (X \to \mathcal{P}(Y)) \to \mathcal{P}(X \times Y)$ (which means $f : A \to B$):

$$f(a) := \{(x, y) | y \in a(x), x \in X\}$$

Then we prove f is bijective.

First, we prove f is injective. We prove this by contradiction. We assume $\exists a_1, a_2 \in A, a_1 \neq a_2, f(a_1) = f(a_2)$, *i.e.* f is not injective. By the definition of set equivalent, we have $\forall x \in X, \forall y \in Y, (x, y) \in f(a_1) \Leftrightarrow (x, y) \in f(a_2), (x, y) \notin f(a_1) \Leftrightarrow (x, y) \notin f(a_2)$. Thus $\forall x \in X, \forall y \in Y, y \in a_1(x) \Leftrightarrow y \in a_2(x), y \notin a_1(x) \Leftrightarrow y \notin a_2(x), a_1(x) \in \mathcal{P}Y, a_2(x) \in \mathcal{P}Y$. So we have $\forall x \in X, a_1(x) = a_2(x)$, which is equivalent to $a_1 = a_2$. This is contradict to our assumption. So $\forall a_1, a_2 \in A, f(a_1) = f(a_2)$ iff $a_1 = a_2$.

Second, we prove f is surjective. We prove this by contradiction. We assume $\exists b \in B, \forall a \in A, f(a) \neq b$, *i.e.* f is not surjective. We define a new function g. g is a function with input $x \in X$ and output as a set s. We initial g as, $\forall x \in X, g(x) = \emptyset$ (empty set). Then, we iterate all $(x, y) \in b$, we add y to set $g(x)$. After the iteration, we can see that $\forall x \in X$, $g(x)$ is empty set or a set of elements from Y, *i.e.* $g(x) \in \mathcal{P}Y$. Thus $g : X \to \mathcal{P}Y$, *i.e.* $g \in A$. We also notice $\forall x \in X, \forall y \in Y, (x, y) \in b \Leftrightarrow (x, y) \in f(g), (x, y) \notin b \Leftrightarrow (x, y) \notin f(g)$. Thus, $f(g) = b$, which is contradict to our assumption. So f is surjective.

Thus, f is both injective and surjective, *i.e.* bijective. So there is a 1-1 correspondence between the set 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.

My Answer:
When we run the CPAChecker as listed in homework page on specs: `Property[1a, 1b, 2b]`, the `Property1a` is violated in line 1963, the `Property2b` is violated in line 1970, and the `Property1b` is validated.

`Property1a` means it will report error if it reach label "`Property1a`" (case nonsensitive). The `tcas.i` is a reasonable test suite, because it has quite complex paths and `Property[1a, 1b, 2a, 2b, 3a, 3b, ...]`, which are good test cases of hard reachability problem for CPAChecker. This proves that the CPAChecker does well in checking counterexample feasibility and refining abstraction (if it has a similar structure with SLAM).

I think the CPAChecker is usable. It clearly shows the line it finds the error, and its graphical report shows the control flow and reachability graph no only for `[1a, 1b, 2b]` but also with other variables like `[4a, 4b]`. I think it can help programmer quickly debug the program.

Here are the last ten non-empty output lines of `Property[1a, 1b, 2b]`

1a.

```
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) "hw0/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".
```

1b.

```
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) "hw0/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".
```

2b.

```
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) "hw0/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".
```

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

Peer Review ID: 61839672 — enter this when you fill out your peer evaluation via gradescope