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.

I constrcut a function $f : B \to A$ and prove that f is an injection and a surjection.
Assume Z is a subset of $X \times Y$, $a : X \times \mathcal{P}(Y) \in A$ is a function and $f(Z) = a$.
For any $x \in X$, $a(x)$ is a subset of $\mathcal{P}(Y)$, which defined by relations in Z. $a(x) = \{y | y \in Y \bigwedge (x, y) \in Z\}$.

An example: $X = \{1, 2\}$ and $Y = \{3\}$
$A = \{a_1, a_2, a_3, a_4\}$
$B = \{\emptyset, \{(1, 3)\}, \{(2, 3)\}, \{(1, 3), (2, 3)\}\}$
$f(\emptyset) = a_1$, $f(\{(1, 3)\}) = a_2$, $f(\{(2, 3)\}) = a_3$, $f(\{(1, 3), (2, 3)\}) = a_4$
$a_1(1) = \emptyset, a_1(2) = \emptyset$,
$a_2(1) = \{3\}, a_2(2) = \emptyset$,
$a_3(1) = \emptyset, a_3(2) = \{3\}$,
$a_4(1) = \{3\}, a_4(2) = \{3\}$

f is an injection: $f(Z_1) = f(Z_2)$ implies $Z_1 = Z_2$. Assume $f(Z_1) = f(Z_2)$ and $Z_1 \neq Z_2$. $Z_1 = \bigcup\{(x_i, y_1) | y_1 \in f(Z_1)\}, (x_i \in X)$, $Z_2 = \bigcup\{(x_i, y_2) | y_2 \in f(Z_2)\}, (x_i \in X)$. For each $x \in X$, we have $\{(x, y_1) | y_1 \in f(Z_1)\} = \{(x, y_2) | y_2 \in f(Z_2)\}$, so there's a contradiction.

f is a surjection: Each element $a \in A$ has non-empty preimage in B. $\forall a \in A, \exists Z \in B$, $f(Z) = a$. We can find a Z in a's pre-image. $Z = \bigcup\{(x, y) | y \in f(Z)\}$, (foreach $x \in X$). Each element of Z is a relationship between an element in X and an element in Y. Z must be a subset of $X \times Y$, i.e. Z is a subset of 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 two 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: if your output when checking `Property1a` does not indicate something like "No property violation found by chosen configuration" then you 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.

CPAChecker v1.6.1 is used since I failed to install Java 11 on the CAEN computer. The result for property 1a different from the hint. Figure 1 and Figure 1 show the output of terminal. CPAChecker finds a violation for Property2b and verifies Property1a and Property1b are safe.

Property1a means that if (need_downward_RA=1) $\bigwedge$ (Up_Separation >= thresh) $\bigwedge$ (Down_Separation < thresh) are satisfied, there is an error.

One program file and several properties are not enough to test CPAChecker. CAPChecker is a usable tool. It requires specification and program files as input and outputs correctness report or counter-example. The experiment indicates that CAPChecker is efficient and cross platform. In addition,, CAPAChecker shows the ability to find counter-example with graphs effectively, which provides useful information. There are some threats to validity. CPAChecker only works for C and Java (experimental). Also it is laborious to install it on Windows platform.

```
[[23:19] [redacted]@caen-vnc-vm17:~/Documents/eecs590/hw0 $ ../CPAchecker-1.6.1-unix/scripts/cpa.sh -predicateA
nalysis -spec Property1a.spc tcas.i
Running CPAchecker with default heap size (1200M). Specify a larger value with -heap if you have more RAM.
Running CPAchecker with default stack size (1024k). Specify a larger value with -stack if needed.
Using the following resource limits: CPU-time limit of 900s (ResourceLimitChecker.fromConfiguration, INFO)

CPAchecker 1.6.1 (OpenJDK 64-Bit Server VM 1.8.0_282) started (CPAchecker.run, INFO)

Using predicate analysis with SMTInterpol 2.1-238-g1f06d6a-comp and JFactory 1.21. (PredicateCPA:Predicate
CPA.<init>, INFO)

No invariants are computed (PredicateCPA:InvariantsManager.<init>, INFO)

Using refinement for predicate analysis with PredicateAbstractionRefinementStrategy strategy. (PredicateCP
A: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".
Run /afs/umich.edu/user/y/o/[redacted]/Documents/eecs590/CPAchecker-1.6.1-unix/scripts/report-generator.py to
show graphical report.
[[23:19] [redacted]@caen-vnc-vm17:~/Documents/eecs590/hw0 $ ../CPAchecker-1.6.1-unix/scripts/cpa.sh -predicateA
nalysis -spec Property1b.spc tcas.i
Running CPAchecker with default heap size (1200M). Specify a larger value with -heap if you have more RAM.
Running CPAchecker with default stack size (1024k). Specify a larger value with -stack if needed.
Using the following resource limits: CPU-time limit of 900s (ResourceLimitChecker.fromConfiguration, INFO)

CPAchecker 1.6.1 (OpenJDK 64-Bit Server VM 1.8.0_282) started (CPAchecker.run, INFO)

Using predicate analysis with SMTInterpol 2.1-238-g1f06d6a-comp and JFactory 1.21. (PredicateCPA:Predicate
CPA.<init>, INFO)

No invariants are computed (PredicateCPA:InvariantsManager.<init>, INFO)

Using refinement for predicate analysis with PredicateAbstractionRefinementStrategy strategy. (PredicateCP
A: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".
Run /afs/umich.edu/user/y/o/[redacted]/Documents/eecs590/CPAchecker-1.6.1-unix/scripts/report-generator.py to
show graphical report.
```

Figure 1: Output for property 1a & 1b

```
CPAchecker 1.6.1 (OpenJDK 64-Bit Server VM 1.8.0_282) started (CPAchecker.run, INFO)

Using predicate analysis with SMTInterpol 2.1-238-g1f06d6a-comp and JFactory 1.21. (PredicateCPA:Predicate
CPA.<init>, INFO)

No invariants are computed (PredicateCPA:InvariantsManager.<init>, INFO)

Using refinement for predicate analysis with PredicateAbstractionRefinementStrategy strategy. (PredicateCP
A:PredicateCPARefiner.<init>, INFO)

Starting analysis ... (CPAchecker.runAlgorithm, INFO)

Error path found, starting counterexample check with CPACHECKER. (CounterexampleCheckAlgorithm.checkCounte
rexample, INFO)

Using the following resource limits: CPU-time limit of 900s (CounterexampleCheck:ResourceLimitChecker.from
Configuration, INFO)

Repeated loading of Eclipse source parser (CounterexampleCheck:EclipseParsers.getClassLoader, INFO)

Error path found and confirmed by counterexample check with CPACHECKER. (CounterexampleCheckAlgorithm.chec
kCounterexample, INFO)

Stopping analysis ... (CPAchecker.runAlgorithm, INFO)

Verification result: FALSE. Property violation (error label in tcas.i, line 1997) found by chosen configur
ation.
More details about the verification run can be found in the directory "./output".
Run /afs/umich.edu/user/y/o/[redacted]/Documents/eecs590/CPAchecker-1.6.1-unix/scripts/report-generator.py to
show graphical report.
```

Figure 2: Output for property 2b

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