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.

Listing 1: Checking 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".
```

Exercise 0F-3. Model Checking [10 points].

Answer. The first 10 non-empty lines output for each rules is as follows, 1 for Property1a, 2 for Property1b and 3 for Property2b. These were obtained on an Ubuntu 20.04 VM.

The tool is performing static analysis over the data-flow amongst variables to evaluate the predicates at compile time. `Property1a` results in a error if it's possible for the control flow to lead to a `goto` statement jumping to the "PROPERTY1A" (Where any of the characters can be small or capital). This is the same case with `Property2b`. Whereas for `Property 1b`, the predicate leading to the goto statement evaluates to false statically. The checker has proved statically that the traffic collision avoidance system can end up in an error state, and therefore will fail to operate safely on deployment.

Although tcas.i showcases the abilty of the checker, the resolved macros and header-file inclusions can reduce the readability by adding more lines and exposing complex patterns that were wrapped within macros. The original source file could have been a better choice. The tool seemed easy to understand and use in this case, but I believe that more complex properties would be harder to specify due to the bare-bones string matching approach to identifying an error path. It might be better if the tool automatically detects a control-path leading to an exception or error being thrown.

Listing 2: Checking 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".
```

Listing 3: Checking 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".
```

Q2

Given,

$B = P(X \times Y)$ $A = X \to P(Y)$

Let $b \in B$,

b is of the form,

$\{ (x_i, y_k), (x_j, y_l), \dots \}$

$x_i, x_j \in X$ $y_k, y_l \in Y$

For each relation in b that is drawn from $X \times Y$,

We can construct a unique function $f : X \to Y$ such that $f(x_i) = \{ y_i : (x_i, y_i) \in b \}$

But, $f \in A$. So there is a unique element f in A for each $b \in B$

$\Rightarrow$ Surjective

Assuming there are 2 functions f_1 & f_2 in A that are equal,

~~[illegible]~~

For each $x \in X$, we can construct $b_2, b_1 \in B$ such that $b_1 = \{(x, y_i) : y_i \in f_1(x)\}$

& $b_2 = \{(x, y_i) : y_i \in f_2(x)\}$

Since $f_1(x) = f_2(x) = b_1 = b_2 \in B$

$\Rightarrow$ Injective

Peer Review ID: 62081936 — enter this when you fill out your peer evaluation via gradescope

1 HW0

- **- 0 pts** Correct

Peer Review ID: 62081936 — enter this when you fill out your peer evaluation via gradescope