Exercise 0F-2. Set Theory

Let X and Y be sets, and let $A = X \to \mathcal{P}(Y)$ and $B = \mathcal{P}(X \times Y)$.

To show that there is a 1-1 correspondence between A and B, we will construct a bijective function $f : A \to B$. $\forall a \in A$, let

$$f(a) = \bigcup_{x \in X} \{(x, y) \mid y \in a(x)\}$$

Note that $f(a) \in B$, since $\forall (x, y) \in f(a)$, $x \in X$ and $y \in a(x) \subseteq Y$, so $(x, y) \in X \times Y$, so $f(a) \subseteq X \times Y$, so $f(a) \in \mathcal{P}(X \times Y)$.

To show that f is bijective, we will construct a function $f^{-1} : B \to A$ and show that f^{-1} is the inverse of f.

To construct f^{-1}, $\forall b \in B$, let $f^{-1}(b) = g$, where g is a function $g : X \to \mathcal{P}(Y)$ such that $\forall x \in X$, $g(x) = \{y \mid (x, y) \in b\}$. Note that $g \in A$, since $\forall x \in X$, $\forall y \in g(x)$, $(x, y) \in b \subseteq X \times Y$, so $y \in Y$, so $g(x) \subseteq Y$, so $g(x) \in \mathcal{P}(Y)$.

Now, we will show that f^{-1} is the inverse of f. First, we will show that $\forall a \in A$, $\forall x \in X$, $a(x) = f^{-1}(f(a))(x)$. By construction,

$$\begin{aligned} f^{-1}(f(a))(x) &= \{y \mid (x, y) \in f(a)\} \\ &= \{y \mid (x, y) \in \bigcup_{x' \in X} \{(x', y') \mid y' \in a(x')\}\} \\ &= \{y \mid y \in a(x)\} \\ &= a(x) \end{aligned}$$

Now, we will show that $\forall b \in B$, $b = f(f^{-1}(b))$. By construction,

$$\begin{aligned} f(f^{-1}(b)) &= \bigcup_{x \in X} \{(x, y) \mid y \in f^{-1}(b)(x)\} \\ &= \bigcup_{x \in X} \{(x, y) \mid y \in \{y' \mid (x, y') \in b\}\} \\ &= \bigcup_{x \in X} \{(x, y) \mid (x, y) \in b\} \\ &= b \end{aligned}$$

Thus, f^{-1} is the inverse of f, so f is invertible, meaning f is bijective, so a 1-1 correspondence exists between A and B.

Exercise 0F-3. Model 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".
```

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)
```

```
14
15 Verification result: TRUE. No property violation found by chosen
   ↪ configuration.
16 More details about the verification run can be found in the directory
   ↪ "./output".
17 Graphical representation included in the file "./output/Report.html".
```

Property2b

```
1 Using the following resource limits: CPU-time limit of 900s
   ↪ (ResourceLimitChecker.fromConfiguration, INFO)
2
3 CPAchecker 2.0 / predicateAnalysis (OpenJDK 64-Bit Server VM 11.0.9.1) started
   ↪ (CPAchecker.run, INFO)
4
5 Parsing CFA from file(s) "tcas.i" (CPAchecker.parse, INFO)
6
7 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)
8
9 Using refinement for predicate analysis with
   ↪ PredicateAbstractionRefinementStrategy strategy.
   ↪ (PredicateCPA:PredicateCPARefiner.<init>, INFO)
10
11 Starting analysis ... (CPAchecker.runAlgorithm, INFO)
12
13 Stopping analysis ... (CPAchecker.runAlgorithm, INFO)
14
15 Verification result: FALSE. Property violation (error label in line 1997)
   ↪ found by chosen configuration.
16 More details about the verification run can be found in the directory
   ↪ "./output".
17 Graphical representation included in the file
   ↪ "./output/Counterexample.1.html".
```

Experience with CPAChecker

First, I will go over my understanding of `tcas.i` and the three properties checked by CPAChecker. Properties 1a, 1b, and 2b check the values of two variables `Up_Separation` and `Down_Separation` relative to a a variable `thresh`. Without an understanding of the purpose of the traffic collision avoidance system code, it is hard to know why these properties are important, but at the very least by using CPAChecker we know which properties are violated and which are not. Also, using CPAChecker is fairly straightforward, and the counterexample execution trace is fairly clear when viewed in the graphical report.

Now, I will give my thoughts on how well `tcas.i` and the three properties demonstrate the functionality of CPAChecker. `tcas.i` does not seem to be a very complicated test case for CPAChecker. Looking at the statistics page of the graphical reports for each of the properties, properties 1a and 2b required 1 abstraction and 0 predicates to be discovered, while property 2a required 2 abstractions and 1 predicate to be discovered.

It does not seem like `tcas.i` and the three properties test the limits CPAChecker's abstraction refinement. Furthermore, in `tcas.i`, the properties are all checked at the end of the `main` function after all computation has completed. Not having the property checking interleaved with computation seems like a simple case. Overall, `tcas.i` does not seem like an effective test case for demonstrating the full range of CPAChecker.

1 HW0

- **0 pts** Correct