Question 2

Proposition 0.1. *Let X and Y be sets. There is a 1-to-1 correspondence between the sets A and B, where $A := X \to 2^Y$, and $B := 2^{X \times Y}$.*

Proof. Define the following function $f : B \to A$ as follows: For every $S = \{(x, y)\} \subseteq X \times Y$, we have

$$S \longmapsto f(S),$$
$$f(S)(x) := \{y \mid (x, y) \in S\},\ \forall x \in X.$$

It's easy to verify that $f(S)$ is indeed a function from X to 2^Y.

- **Injective**

First we show the injectivity. Suppose there are two sets S_1, S_2 such that $f(S_1) = f(S_2)$, i.e.,

$$f(S_1)(x) = f(S_2)(x),\ \forall x \in X.$$

By definition, this means

$$\{y \mid (x, y) \in S_1\} = \{y \mid (x, y) \in S_2\},\ \forall x \in X.$$

Therefore, for every $(x, y) \in (X, Y)$ we have $(x, y) \in S_1$ if and only if $(x, y) \in S_2$, i.e., $S_1 = S_2$.

- **Surjective**

Then we show that $S \mapsto f(S)$ is onto $X \to 2^Y$, i.e., for any $g \in X \to 2^Y$, there exists an $S \subseteq X \times Y$ such that $f(S) = g$.

Consider the following construction: Let

$$S_g := X \times g(x) = \{(x, y) \mid y \in g(x), x \in X\},\ \forall g \in X \to 2^Y,$$

we can verify that

$$f(S_g)(x) = \{y \mid (x, y) \in S_g\} = \{y \mid y \in g(x)\} = g(x),\ \forall x \in X.$$

Since $S \mapsto f(S)$ is both injective and surjective, it's thus a bijection $B \longleftrightarrow A$. □

Peer Review ID: 300528319 — enter this when you fill out your peer evaluation via gradescope

Question 3

3.1 Results

Property 1a

```
Using the following resource limits: CPU-time limit of 900s
(ResourceLimitChecker.fromConfiguration, INFO)

CPAchecker 4.0 / predicateAnalysis (OpenJDK 64-Bit Server VM 17.0.13) started
(CPAchecker.run, INFO)

Parsing CFA from file(s) "tcas.i" (CPAchecker.parse, INFO)

Using predicate analysis with MathSAT5 version 5.6.10 (9293adc746be)
(May 31 2023 12:38:06, gmp 6.2.0, 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".
```

Property 1b

```
Using the following resource limits: CPU-time limit of 900s
(ResourceLimitChecker.fromConfiguration, INFO)

CPAchecker 4.0 / predicateAnalysis (OpenJDK 64-Bit Server VM 17.0.13) started
(CPAchecker.run, INFO)

Parsing CFA from file(s) "tcas.i" (CPAchecker.parse, INFO)

Using predicate analysis with MathSAT5 version 5.6.10 (9293adc746be)
(May 31 2023 12:38:06, gmp 6.2.0, 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"..
```

Property 2b

```
Using the following resource limits: CPU-time limit of 900s
(ResourceLimitChecker.fromConfiguration, INFO)

CPAchecker 4.0 / predicateAnalysis (OpenJDK 64-Bit Server VM 17.0.13)
started (CPAchecker.run, INFO)

Parsing CFA from file(s) "tcas.i" (CPAchecker.parse, INFO)

Using predicate analysis with MathSAT5 version 5.6.10 (9293adc746be)
(May 31 2023 12:38:06, gmp 6.2.0, 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".
```

3.2 Analysis

According to the documentation for Version 3.0 (`https://arxiv.org/pdf/2409.02094`), the option `-predicateAnalysis` can be used for larger programs and is stronger than regular value analysis and symbolic execution (Page 14 and Section 4.5). The tool will use CEGAR and Craig interpolation to learn predicates to reduce the state space.

I searched for the name "tcas" and it suggested for "Traffic Collision Avoidance System", which seems to be relevant, and reasonable as a test case. Property 1a wants to make sure that `Up_separation` is less than `thresh`, or `Down_separation` is greater than or equal to `thresh` (Line 1958);Property 1b wants to make sure that `Up_separation` is greater than or equal to `thresh`, or `Down_separation` is less than `thresh` (Line 1970); Property 2b wants to make sure that `Up_separation` is greater than or equal to `thresh`, or `Down_separation` is greater than or equal to `thresh`, or `Up_separation` is less than or equal to `Down_separation` (Line 1993). In the code, the `thresh` could be different values like $400, 500, 640, \ldots$. A-series properties are checked with an upward RA (Resolution Advisory), while B-series are checked with a downward RA. The result for Property 1a suggests that it's possible to have `Up_separation` $\geq$ `thresh` while `Down_separation` $<$ `thresh` during the operation of the system.

Regarding the usability, I think the CPA checker would be more effective for those who are already very familiar with the concept of model checking. It still seems very likely for general users to make mistakes in some steps of this verification process, which would lead to incorrect results. While constructing the input (property specifications as regular expressions) does not seem to be trivial, the tool does provide a lot of information in the output. The fancy html page rendered the control flow automations with comparison to the source code. The help page is useful. Given the the diagram is very large, it's not convenient to locate the error even the the search function. Moreover, the runtime was indeed shorter than I expected.