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) between 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.

Solution: Consider the function $f : B \to A$ as follows. Given $b \in B$, let $a = f(b)$, where $\forall x \in X$, $a(x) \coloneqq \{y \in Y : (x, y) \in b\}$.

The function f is injective, as shown in the following. Consider $b_1, b_2 \in B$ with $b_1 \neq b_2$. Assume without loss of generality that there is some $(x, y) \in b_1$ such that $(x, y) \notin b_2$. Let $a_1 = f(b_1)$ and $a_2 = f(b_2)$. We have that $y \in a_1(x)$ and $y \notin a_2(x)$ by our construction for f. Therefore $a_1 \neq a_2$.

The function f is surjective, as shown in the following. Consider any $a \in A$. Let $b \in B$ be defined as $b \coloneqq \{(x, y) : x \in X, y \in a(x)\}$. We have that $f(b) = a$, given by the following. Given $x \in X$,

$$\begin{aligned} f(b)(x) &\coloneqq \{y \in Y : (x, y) \in b\} && \text{By our construction for } f. \\ &= \{y \in Y : y \in a(x)\} && \text{By definition for } b. \\ &= a(x) \end{aligned}$$

Exercise 0F-3. Model Checking [10 points]. Command line outputs:

```
(base) yun-rongluo@Yun-Rongs-MacBook-Air-2 CPAchecker-4.0-unix % docker run -v $(pwd):/workdir -u $UID:$GID sosylab/cpachecker --predicateAnalysis --spec Property1a.spc tcas.i
WARNING: The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested
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.
Language C detected and set for analysis (CPAMain.detectFrontendLanguageIfNecessary, INFO)

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 1965) 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".
```

Figure 1: 1a.spc

- What is going on when you run CPAChecker using the commands listed? What does `Property1a` mean? Is `tcas.i` a reasonable test suite? What has been proved? (This is the heart of the question. You may have to read ugly code, understand a legacy

No questions assigned to the following page.

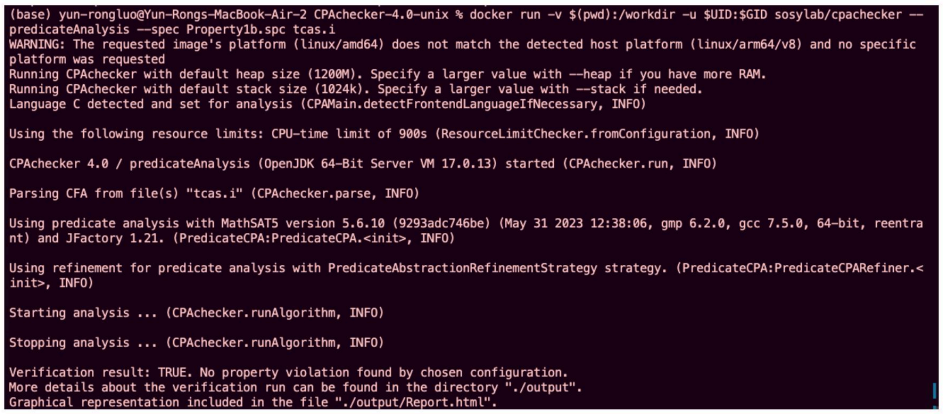

Figure 2: 1b.spc

tool, and apply concepts from class. It is expected that answering this well will require time.)

CPAChecker either reports "Verification result": FALSE and provides an execution trace that eventually violates the checked property, or reports "Verification result: TRUE", indicating that the implementation does not violate the checked property.

Property1a is a "bad condition" that the designer does not want the program to reach, which is Up_Separation $\geqslant$ thresh && Down_Separation $<$ thresh. Property1a is checked when "need_downward_RA= 1". I think the designer want to ensure that when the program sets "need_downward_RA= 1", it will not satisfy Property1a. I think tcas.i is a reasonable test suite since it is not too complicated for the model checker to analyze, and it contains a few properties that are interesting but not trivial. It seems to me that tcas.i is still a high-level abstraction for a real-world design. Coming up with a suitable implementation abstraction and properties to prove could still be challenging for designers.

- Did you find CPAChecker to be a usable tool? How easy is it to provide the inputs to CPAChecker? What information is present in the graphical (HTML) output?

  Solution: I think this is a usable tool with rather easy setup and user-friendly output displays. The graphical output visualizes the counter-example trace. Specifically, nodes are commands in the function or a call to other functions, and edges are predicate values that control the execution flow.

No questions assigned to the following page.

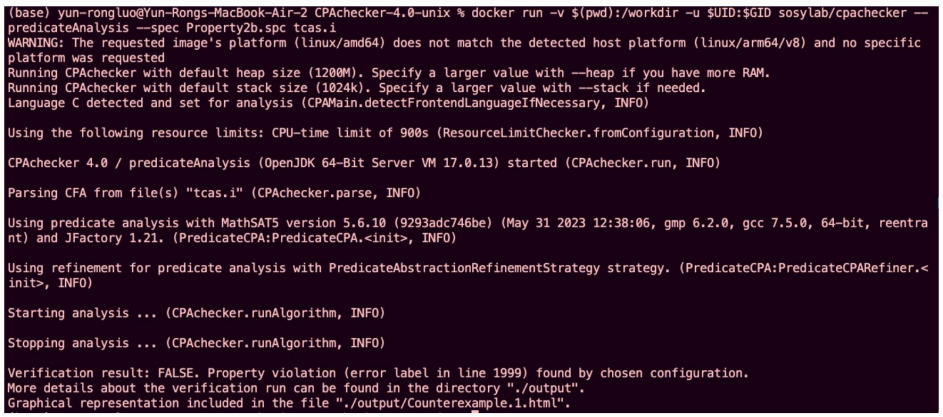

Figure 3: 2b.spc