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.

Answer:

Let

$$X = \{x_0, x_1, \dots\}$$
$$Y = \{y_0, y_1, \dots\}$$

According to the problem setting, we have

$$\mathcal{P}(Y) = \{\emptyset, \{y_0\}, \{y_1\}, \dots, \{y_0, y_1\}, \dots\}$$
$$X \times Y = \{(x_0, y_0), (x_0, y_1), \dots, (x_1, y_0), \dots\}$$
$$A = \{F | F : X \to \mathcal{P}(Y)\}$$
$$B = \mathcal{P}(X \times Y) = \{\emptyset, \{(x_0, y_0)\}, \{(x_0, y_1)\}, \dots, \{(x_0, y_0), (x_0, y_1)\}, \dots\}$$

We define a function

$$f : B \to A$$

such that

$$\forall b \in B, f(b) = F_b$$

where

$$\forall x \in X, F_b(x) = \{y | (x, y) \in b\} \in \mathcal{P}(Y)$$

We now need to show f is an injection. To show it, we need to show

$$\forall b_1, b_2 \in B, b_1 \neq b_2 \implies f(b_1) \neq f(b_2)$$

We know

$$b_1 \neq b_2 \implies \exists (x_i, y_j) \in b_1 \cup b_2, (x_i, y_j) \notin b_1 \cap b_2$$

without loss of generality, let

$$(x_i, y_j) \in b_1, (x_i, y_j) \notin b_2$$

We have

$$y_j \in f(b_1)(x_i)$$

and

$$y_j \notin f(b_2)(x_i)$$

which gives

$$f(b_1) \neq f(b_2)$$

Thus, f is injective.

We next need to show f is a surjection. To show it, we need to show

$$\forall F : X \to \mathcal{P}(Y) \in A, \exists b \in B, f(B) = F$$

For some $F \in A$, let

$$b = \bigcup_{x \in X} (\{x\} \times F(x))$$

It is obvious that

$$b \in B \quad \text{and} \quad f(b) = F$$

Hence, f is surjective.

Since f is both injective and surjective, f is bijective.

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 three 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: CPAChecker 2.0 should find a violation for `Property1a`, verify that `Property1b` is safe, and find a violation for `Property2b`. If your output does not match that and you are using version 2.0 then you may not 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.

Answer:

The following screenshots show the results of runing the CPAChecker on the specs `Property[1a,1b,2a]`. We can see that `property1b` is validated while the other two are violated by line 1963 and line 1997 respectively.

`Property1a` means that if encountered with a label "PROPERTY1A" (case insensitive) in the execution flow, the program is buggy. I think `tcas.i` is a reasonable test suite because it has complex paths and we can easily define a path that leads to an error to figure out whether the CPAChecker does what it is supposed to do. It proved that the CPAChecker did a good job in finding buggy paths.

I believe the CPAChecker is quite usable because I can clearly see the erroneous path that leads to the violation of my spec in the report and it is a concrete path which helps to debug the program.

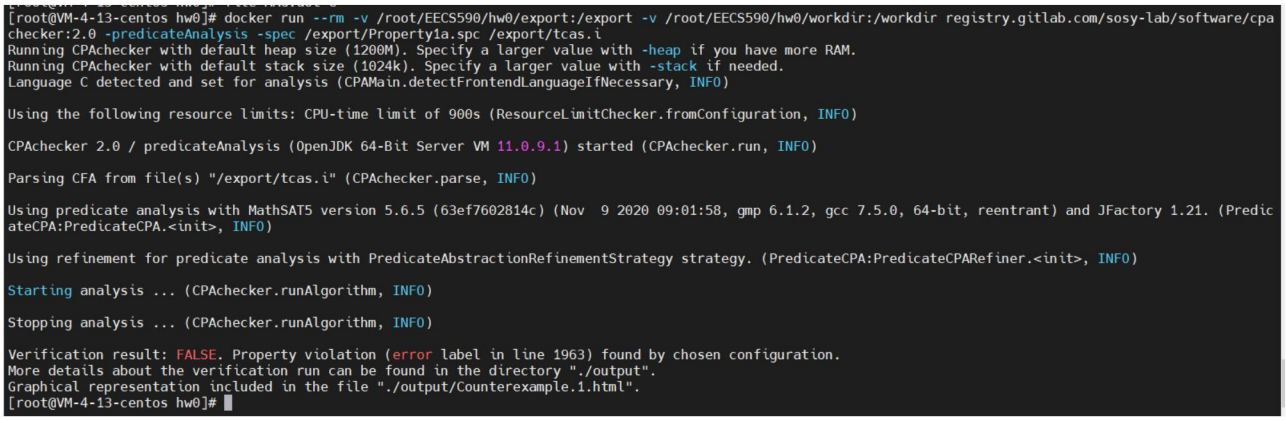

Figure 1: Result of `Property1a`: SPEC violated.

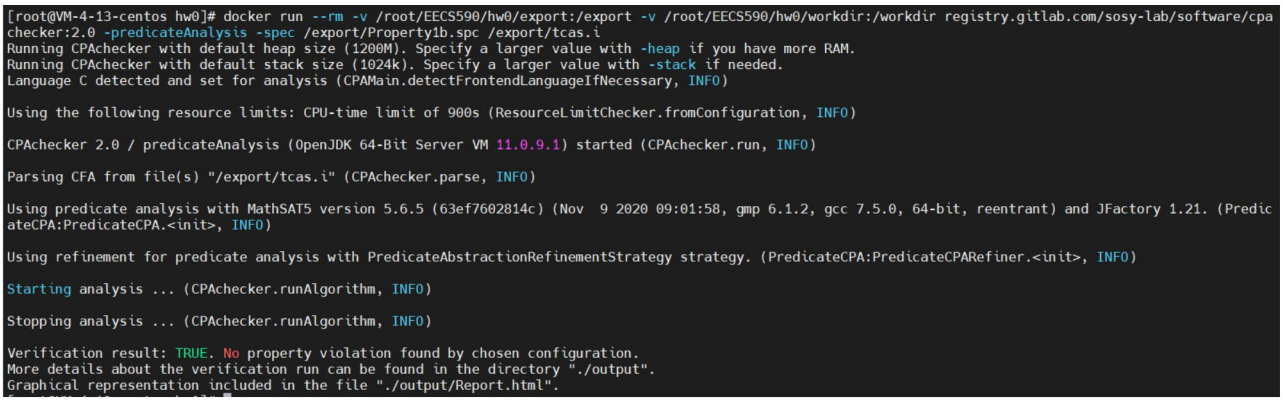

Figure 2: Result of `Property1b`: SPEC validated.

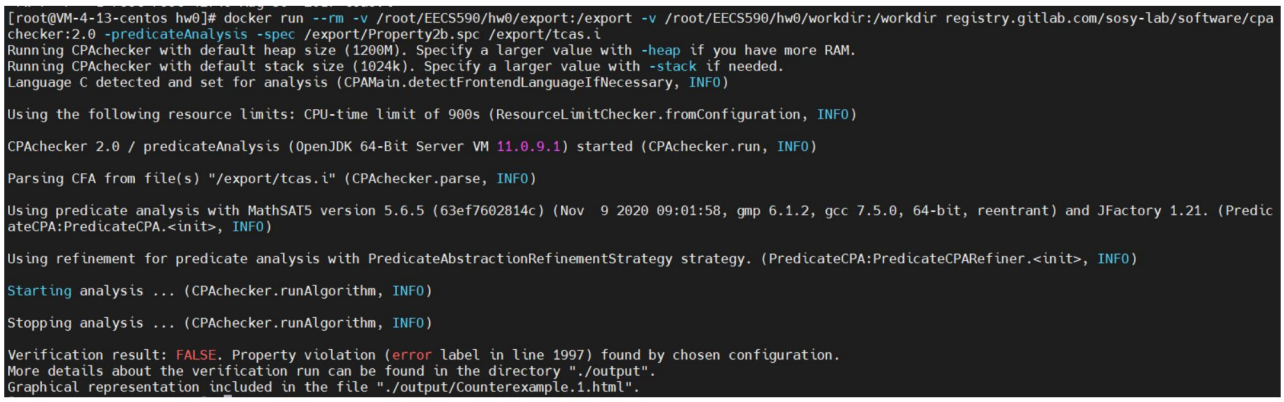

Figure 3: Result of `Property2b`: SPEC violated.

1 HW0

- **0 pts** Correct