Exercise 0F-2. Set Theory [5 points].

If we define f as.

$f : B \to A$ s.t. $(f(b))(x) = y$ $\forall (x,y) \in b$, $b \in B$.

then if $f(b_1) = f(b_2) = g$, $b_1, b_2 \in B$,

$(f(b_1))(x) = (f(b_2))(x) = g(x)$ $\forall x \in X$

Therefore $b_1 = b_2 = \{(x,y) \mid g(x) = y\}$.

$\therefore$ injection

For all ~~For a given~~ $a \in A$, you can construct a set

$$Q = \{(x,y) \mid a(x) = y,\ x \in X, y \in Y\}$$

by definition, $Q \in B$.

~~And using the f that we defined earlier,~~

Then, $(f(Q))(x) = y = a(x)$

$f(Q) = a$

~~Therefore $Q = f^{-1}(a)$~~

Therefore, $\forall a \in A\ \exists Q \in B$ s.t. $f(Q) = a$

Subjection.

$\therefore$ A and B is one-to-one.

Peer Review ID: 61923331 — enter this when you fill out your peer evaluation via gradescope

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

`Property1a.spc`

```
Using the following resource limits: CPU-time limit of 900s (ResourceLimitChecker.fromConfi
guration, 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.<ini
t>, INFO)

Using refinement for predicate analysis with PredicateAbstractionRefinementStrategy strateg
y. (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 c
onfiguration.
More details about the verification run can be found in the directory "./output".
Graphical representation included in the file "./output/Counterexample.1.html".
```

`Property1b.spc`

```
Using the following resource limits: CPU-time limit of 900s (ResourceLimitChecker.fromConfi
guration, 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.<ini
t>, INFO)

Using refinement for predicate analysis with PredicateAbstractionRefinementStrategy strateg
y. (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".
```

Property2b.spc

```
Using the following resource limits: CPU-time limit of 900s (ResourceLimitChecker.fromConfi
guration, 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.<ini
t>, INFO)

Using refinement for predicate analysis with PredicateAbstractionRefinementStrategy strateg
y. (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 c
onfiguration.
More details about the verification run can be found in the directory "./output".
Graphical representation included in the file "./output/Counterexample.1.html".
```

I believe the checker is searching and looking for violations of properties that are given by the config file as regex expressions. Hence the name of each file corresponds to the property each config is trying to find.

tcas.i is a good benchmark in that we can understand that some checks fail and some check pass easily, leading us to believe our config is doing something right. However, I believe a code that's over 700 lines is not fit to be the most intuitive benchmark to test against, as it would take too much time to understand the code and guess the right output for the given configs. Therefore, I believe a more succinct benchmark would be more useful to quickly test the validity of the configs and CFAchecker, as it would require less time.

Finally, for the CFAchecker, as a novice of the tool, I think a verbose option would be useful to see what's happening under the hood. So that I can understand the tool better.

1 HW0

- 0 pts Correct