Exercise 0F-2. Set Theory

Let's define a function $f : A \to B$ as
$f(a) = \{(x, y) \mid y \in a(x)\}$

Let's define another function $g : B \to A$ as
$(g(b))(x) = \{y \mid (x, y) \in b\}$

g is an inverse of f because

$$\begin{aligned}(g \circ f(a))(x) &= \{y \mid (x, y) \in b, b \in \{(x, y) \mid y \in a\}\} \\ &= \{y \mid y \in a(x)\} \\ &= a(x)\end{aligned}$$

which means $g \circ f(a) = a$. Since f is invertible, it is bijective.

We have shown that there is a correspondance between the set of functions $A = (X \to Pow(Y))$ and the set of relations $B = Pow(X \times Y)$, by showing that there is a bijective function between A and B.

Peer Review ID: 62066018 — enter this when you fill out your peer evaluation via gradescope

Exercise 0F-3. Model Checking

```
# output for 1a
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".
```

```
# output for 1b
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: 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".
```

Peer Review ID: 62066018 — enter this when you fill out your peer evaluation via gradescope

```
# output for 2b
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 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".
```

The CPAChecker checks that the given c program (in this case `tcas.i`) satisfies the specifications (in this case `Propertyxx.spc`). When we run the CPAChecker, it first parses the c program into a CFG (Control Flow Graph), and then it walks down the CFG to analyse the predicates. Take `Property1a.spc` as an example, it specifies that it's illegal to reach the line labled with PROPERTY1A, and CPAChecker proves that the contrain is broken by showing an example path from init to line PROPERTY1A.

From validity perspective, `tcas.i` is a reasonable test, for it does proves that the CPAChecker works for the specific case, where the global variables in the key predicates is never modified after initialization, and there is only one thread in the program. However, it is far from a good "test suite" for it checks only the very basic case.

Also, the test should not only include c program files, but also include specification files. In our example, `tcas.i` does not work as a test case on its own; it has to be paired with `Propertyxx.spc`s to be a test case. We could also make the test suite more complete by adding more specificaitons, even though this would not solve the issues mentioned previously.

1 HW0

- **0 pts** Correct