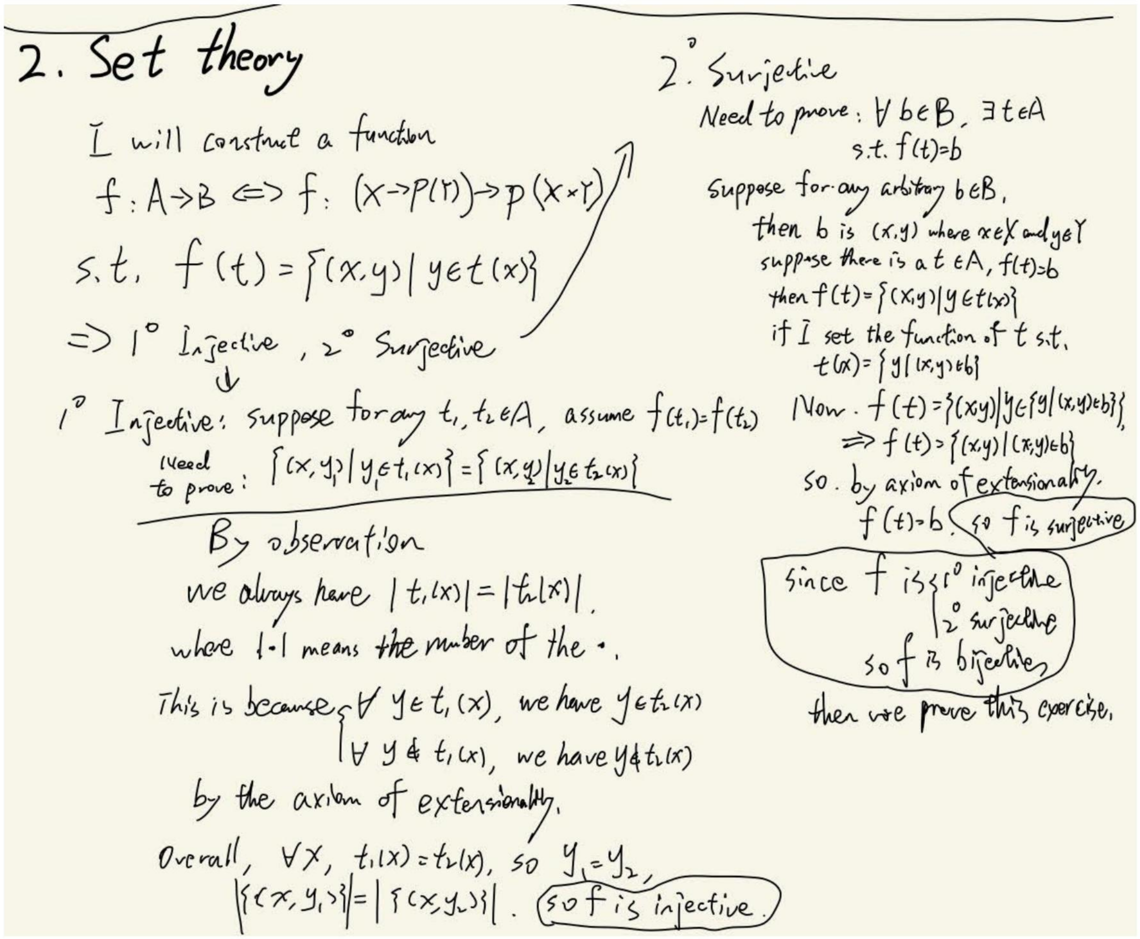


2. Set theory

I will construct a function

$f: A \to B \Leftrightarrow f: (X \to P(Y)) \to P(X \times Y)$

s.t. $f(t) = \{(x,y) \mid y \in t(x)\}$

$\Rightarrow$ 1° Injective, 2° Surjective

1° Injective: suppose for any $t_1, t_2 \in A$, assume $f(t_1) = f(t_2)$

Need to prove: $\{(x,y_1) \mid y_1 \in t_1(x)\} = \{(x,y_2) \mid y_2 \in t_2(x)\}$

By observation

we always have $|t_1(x)| = |t_2(x)|$.

where $|\cdot|$ means the muber of the $\cdot$.

This is because $\begin{cases} \forall y \in t_1(x), \text{ we have } y \in t_2(x) \\ \forall y \notin t_1(x), \text{ we have } y \notin t_2(x) \end{cases}$

by the axiom of extensionality.

Overall, $\forall x$, $t_1(x) = t_2(x)$, so $y_1 = y_2$,

$|\{(x,y_1)\}| = |\{(x,y_2)\}|$. so f is injective.

2° Surjective

Need to prove: $\forall b \in B$, $\exists t \in A$ s.t. $f(t) = b$

Suppose for any arbitrary $b \in B$,

then b is (x,y) where $x \in X$ and $y \in Y$

suppose there is a $t \in A$, $f(t) = b$

then $f(t) = \{(x,y) \mid y \in t(x)\}$

if I set the function of t s.t.

$t(x) = \{y \mid (x,y) \in b\}$

Now. $f(t) = \{(x,y) \mid y \in \{y \mid (x,y) \in b\}\}$,

$\Rightarrow f(t) = \{(x,y) \mid (x,y) \in b\}$

so. by axiom of extensionality.

$f(t) = b$. so f is surjective

since f is $\begin{cases} 1° \text{ injective} \\ 2° \text{ surjective} \end{cases}$

so f is bijective

then we prove this exercise.

Peer Review ID: 62071685 — enter this when you fill out your peer evaluation via gradescope

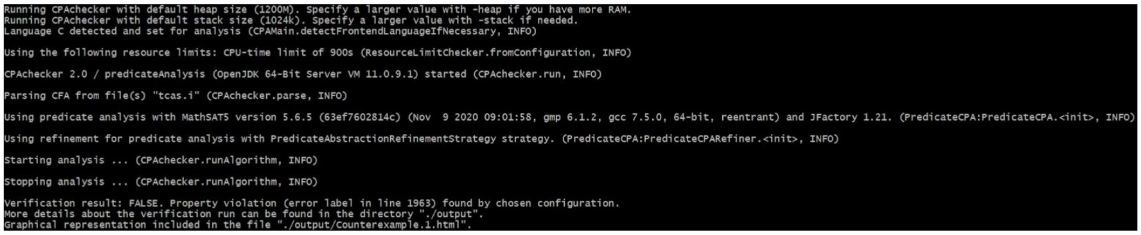
3. I used to run CPAchecker on WindowOS.
However, it throws some errors that I can't figure out.
Then by suggestions from GSI in Piazza post, I used the Linux server for EECS583, which I took last semester. Then, anything is fine, and I successfully run all the commands.

Tcas.i: I think it is a correct suite, since by observation of the output, it passes when using the spec file Property1b, means that tcas.i sucessfully verify it.
For Property1a and Property2b, the CPA checker got false. I think CPA checker is a very useful tool since when it got false, it will provide with me a counterexample.

Furthermore, for command like, "-predicateAnalysis" will let CPA checker to enable the predicate Analysis attributes, "-spec" will notify CPA check that there will be a speificartion file, and that file will verify the source code by some specifications.

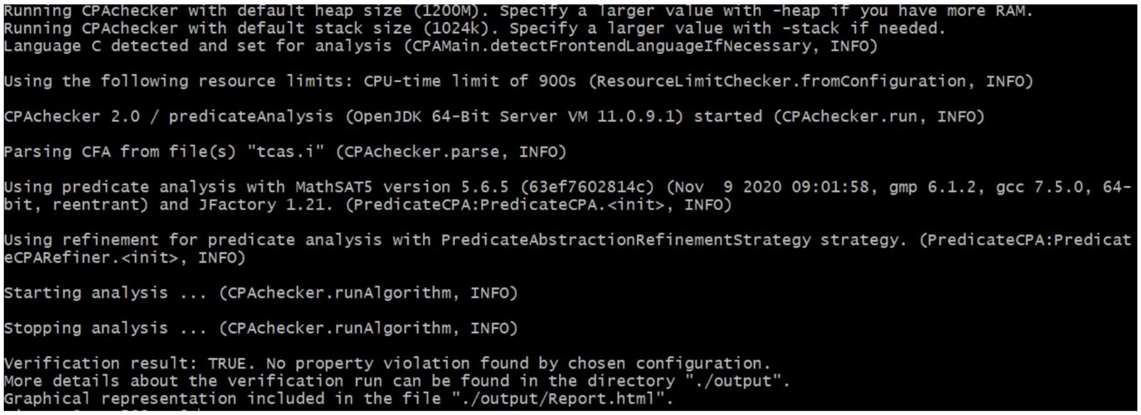
Experience: Overall, I think CPA checker let my interest for this class heats up. it is easy to operate, reasonable to understand. I would like to investigate and understand more about it. I wish I could get familiar with this tool after this semester.

1a



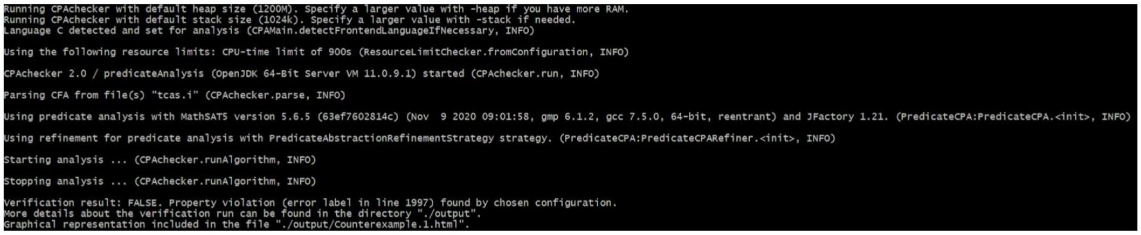
```
Running CPAchecker with default heap size (1200M). Specify a larger value with -heap if you have more RAM.
Running CPAchecker with default stack size (1024k). Specify a larger value with -stack if needed.
Language C detected and set for analysis (CPAMain.detectFrontendLanguageIfNecessary, INFO)

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

CPAchecker 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.<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".
```

1b



```
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 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.<init>, INFO)

Using refinement for predicate analysis with PredicateAbstractionRefinementStrategy strategy. (PredicateCPA:Predicat
eCPARefiner.<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".
```

2b



```
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 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.<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".
```

1 HW0

- **0 pts** Correct

Peer Review ID: 62071685 — enter this when you fill out your peer evaluation via gradescope