1. TTL Consistency Checking among timing constraints.

In order to have a meaningful monitoring approach, it is needed to make sure the system's timing constraints are correct and there is no inconsistency among all. The TTL deduction system (TTL Natural Deduction System) has been proposed to verify the consistency among TTL statements, and simplify the monitoring logic. The TTL reasoning system is able to combine and accumulate errors in the combined and cascaded statements and thus, support a rigorous and correct monitoring process. We have developed a tool, using Python and Z3 theorem prover, to verify the consistency of a set of timing constraints. The monitoring technique also considers the error tolerance in expressing CPS temporal requirements. The error tolerance is intended to capture the aggregate error of the system, including the measurement error of the monitoring device.

A. What is the reasoning system?

The reasoning system is useful to detect the inconsistency among timing specifications of a target system. In fact, it receives a set of timing constraints written in TTL and determines whether they are consistent. In order to do that it is needed to:

i. Run TMA for each timing constraint
ii. Receive the mathematical equations for each timing constraint
iii. Put the mathematical formula into a python code
iv. Install Z3 python prover on Ubuntu
v. Run Z3 python prover and see the result.

B. How the reasoning system works?

Since Z3 python tool is a SMT solver for mathematical logic and it does not understand TTL syntax, it is required we convert all TTL timing constraint into mathematical logic. For this, TMA has a part to do the conversion. After running TMA for a timing constraint, a file is created, "*convetredConstraints.txt*". In this file, each row contains a mathematical logic expression to check the consistency. By converting this text file into a python code, we are able to see the satisfaction or un-satisfaction for the set of timing constraints.

In order to have Z3 SMT solver, it should be installed on Ubuntu. The link to install is here:

https://pypi.org/project/z3-solver/

C. An example for the TTL deduction system.

We can run TMA for two different timing constrains on two signals:

```
((L({s32,15.7,/},{t3,0,/},0.001))<15);
((L({s32,15.7,/},{t3,0,/},0.001))>2);
```

Based on these timing constraints, the content of "*convetredConstraints.txt*" is changed to:

```
1  T(t3>0)-T(s32>15.7)<749951
2  T(t3>0)-T(s32>15.7)>100049
```

Now, it is required to convert these mathematical logic expressions into a python code like below:

```
1  from z3 import *
2  A = Int('t3>0')
3  B = Int('s32>15.7')
4
5  consistency = Solver()
6
7  consistency.add(A - B < 749951,
8                  A - B > 100049 )
9
10 print consistency.check()
11
```

After running this code on Ubuntu, it shows "*sat*" or *"unsat"* as the result.