Specification and Verification of Embedded and Cyber-Physical Systems
Workshop on Trustworthy AI and CPS (TAC 2025)

1) **(Bounded Model Checking)** Describe the main steps involved in checking programs using the BMC technique, from reading the program to generating the SMT formulas. Consider the following example to describe each step in the technique. The **`assume`** directive can define constraints over (non-deterministic) variables, and the **`assert`** directive is used to check the system's correctness w.r.t. a given property.

```
#include <assert.h>
#define a 2
int main(int argc, char **argv) {
  long long int i=1, sn=0;
  unsigned int n=5;
  assume (n>=1);
  while (i<=n) {
    sn = sn + a;
    i++;
  }
  assert(sn==n*a);
}
```

```
def collatz (n: int) -> int:
  while n != 1:
    print(n, end = " -> ")
    if n % 2 == 0:
      n = n // 2
    else:
      n = 3 * n + 1
  print (1)

collatz (6) # This terminates
collatz (-1) # Non - terminating for invalid input (negative number)
```

2) **(Race Condition)** Specify the three properties described below for the following mutual exclusion algorithm using linear-time temporal logic (LTL).

int flag[2], turn, x, i;

void *t1(void *arg) {
flag[0] = 1;
turn = 1;
while (flag[1] == 1 && turn == 1) {};
//critical section
if (i==1) x=1;
//end of critical section
flag[0] = 0;
return NULL;
}
	void *t2(void *arg) {
	flag[1] = 1;
	turn = 0;
	while (flag[0] == 1 && turn == 0) {};
	//critical section
	if (i==2) x=2;
	//end of critical section
	flag[1] = 0;
	return NULL;
	}

i. At most, one process is in the critical region at any time.
ii. Whenever a process tries to enter its critical region, it will eventually succeed.
iii. A process can eventually ask to enter its critical region.

3) **(Cyber-threats)** A typical embedded system consists of a *human-machine interface* (keyboard and display), *processing unit* (real-time computer system), and *instrumentation interface* (sensor, network, and actuator) that can be connected to some physical plant. For example, Figure 1 illustrates a simple embedded system employed in a chemical process, which can be operated remotely via TCP/IP protocol. This embedded system contains two Analog-to-Digital Converters (ADC) and Digital-to-Analog Converters (DAC). The first ADC reads the "thermocouples" temperature and converts its analog to digital value. The second ADC reads the "pressure transducer" and converts its analog value to a digital value. The DAC converts the desired position of the valve/pump to an analog device, while a switch exists to turn on or turn off the heater. The overall objective of this simple embedded system is to keep the temperature and pressure of some chemical processes within well-defined limits by a remote operator.

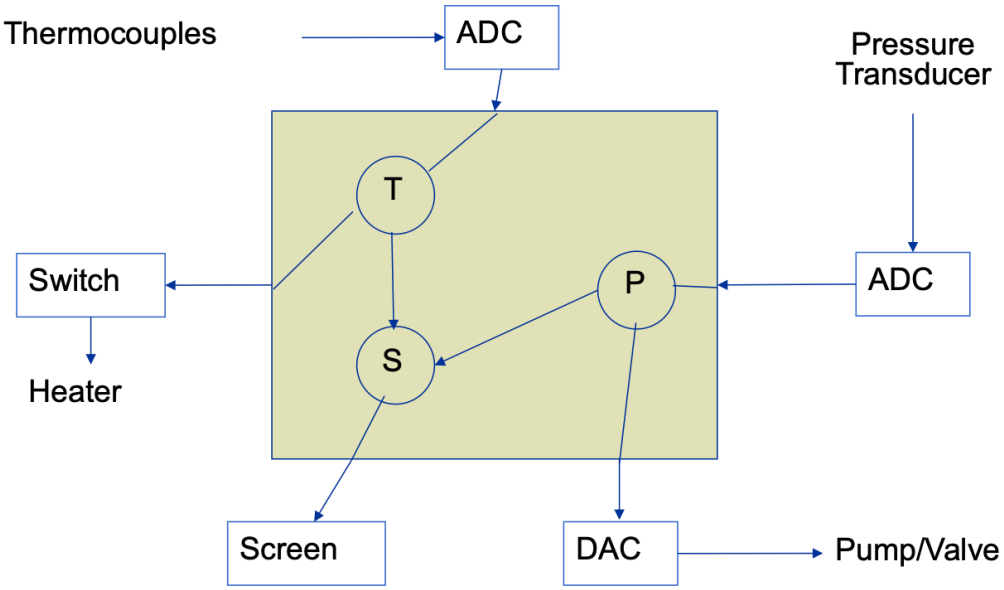

Figure 1: Simple Embedded System [1].

i. Specify the following properties of this simple embedded system using LTL:

a) A **T** process reads the measured values from a temperature sensor, turns the heating system on if the temperature is below 20°C, and turns off the heating system if the temperature is above 300°C.
b) Process **P** regulates pressure with a sensor opening the pump/valve when the pressure value is above 500bar and closing the pump/valve when the pressure value is below 100bar.
c) The measured values will be shown on a liquid crystal display (LCD) when the T and P processes transfer data to an S process.

ii. What are the **security objectives** of this simple embedded system?

iii. What are the sources of **security problems** that can arise when operating this simple embedded system remotely?

References

[1] Alan Burns, Andrew J. Wellings: *Real-Time Systems and Programming Languages - Ada, Real-Time Java and C / Real-Time POSIX*, Fourth Edition. International computer science series, Addison-Wesley 2009, ISBN 978-0-321-41745-9, pp. I-XVIII, 1-602.

[2] Lucas C. Cordeiro, Bernd Fischer, João Marques-Silva: *SMT-Based Bounded Model Checking for Embedded ANSI-C Software.* IEEE Trans. Software Eng. 38(4): 957-974 (2012).

[3] Christel Baier, Joost-Pieter Katoen: Principles of model checking. MIT Press 2008, ISBN 978-0-262-02649-9, pp. I-XVII, 1-975.