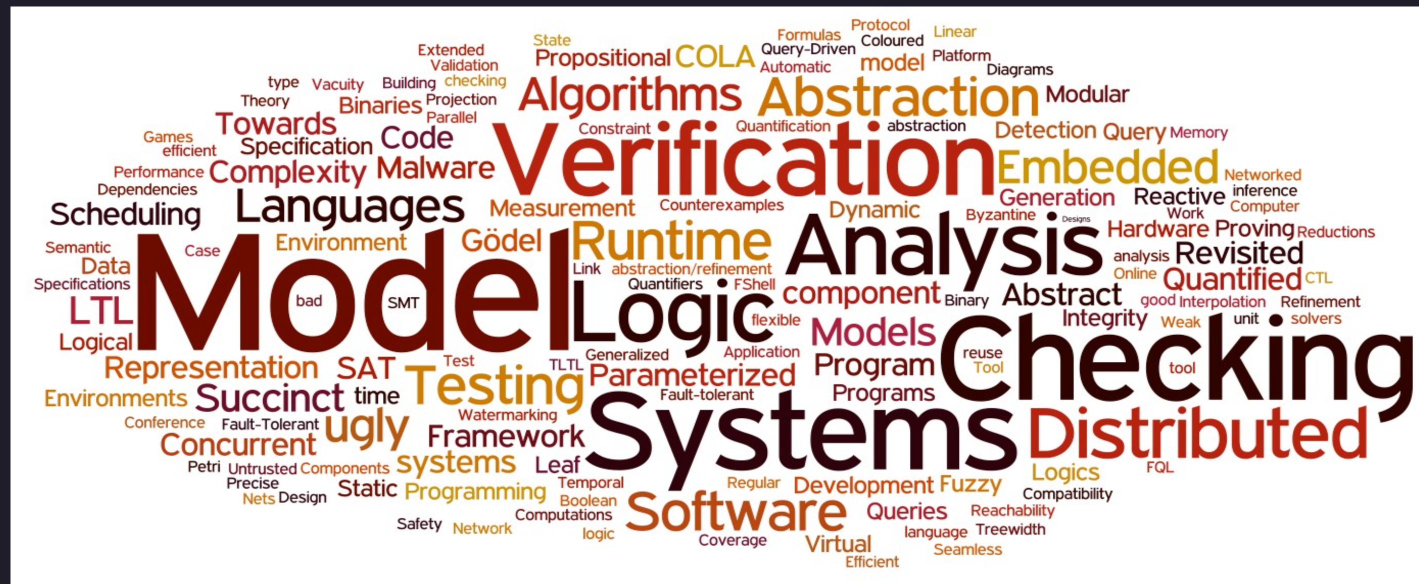
Tutorial 4 : Program Verification

CS60030 Formal Systems

PALLAB DASGUPTA,
FNAE, FASc,
A K Singh Distinguished Professor in AI,
Dept of Computer Science & Engineering
Indian Institute of Technology Kharagpur
Email: pallab@cse.iitkgp.ac.in
Web: http://cse.iitkgp.ac.in/~pallab



INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

FMSAFE
FORMAL METHODS FOR SAFETY CRITICAL SYSTEMS

Weakest Pre-condition

Starting with the postcondition and statements, fill in the intermediate assertions and weakest precondition:

```
x = 2*x;
z = x+1;
{z != 0 }
```

```
x = y - 2;
z = x + y;
{z > 0 }
```

```
w = 2 * w;
z = v - 2 ;
y = -w;
x = min(y,z);
{x < 0}
```

```
y = 2 * x;
y = y + 2;
z = y/2;
{z > x}
```

Compute *wp(A, x < y)* in as simple form as you can, where A is the following code fragment

```
if (x > y)
tmp = x ;
x = y;
y = tmp;
```

Program Abstraction

Check if the following code computes the maximum of two no.s for any given set of input

```
{ true }
if ( x > y) : r = x;
else : r = y;
```

Analyze the program using the domain Parity and then Sign.

```
y = 5;
x = -2 * y;
if (x > 0){
x = x - (y%2 - 1);
y = x * (y - 1);
else
y = -1;
```

Program Verification

```
L1 :    a = b = i = 0;
L2 :    while (a <= 10) {
L3 :        a = b + i;
L4 :        b = a + 1;
L5 :        i = i + 1;
L6 :    }
L7 :    if (b > 20) {
L8 :    error: exit(-1);
L9 :    }
```

1. Construct a Boolean program corresponding to this program P, using only the predicates $(a \leq 10)$, $(b \geq 0)$, $(i = 0)$, and $(b \leq 20)$
2. Show that the error location is reachable in the Boolean program you constructed
3. Explain whether the error identified in this Boolean program is a spurious counter-example.

Program Verification

Use abstract interpretation on the following program to determine whether the assertion is true. Use the sign domain as your abstract domain, which has only three elements, [+, −, 0], representing +ve, −ve, and zero respectively. Show the values collected in each location. Also indicate the outcome:

```
L1 :    x = 1;
L2 :    if (y < = 10){
L3 :        y = 10;
    }
L3 :else{
L5 :    while (x < y){
L6 :        x = 2 * x;
L7 :        y = y - 1;
        }
    }
L8 :    x = y + 1;
L9 :    assert (x > 0);
```