Lecture 12: Invariant Principle

Date: September 25, 2019.

Euclid's Algorithm

To compute $\gcd(a, b)$, we can assume WLOG a, b are positive, and $a \geq b$.

```
gcd(a,b)
    x = a; y = b;
    while (y > 0)
        r = rem(x,y)
        x = y
        y = r
    return x
```

State Machines. Binary relation, called *transitions*, on a set, called *states*.

Execution. A possible sequence of steps the machine might take, i.e., sequence of states beginning with the start state, and successive states in the sequence are related by the transition relation.

Reachable States. A state that appears in some execution.

Preserved Invariant. A predicate P on states such that whenever $P(q)$ holds and $q \longrightarrow r$ then $P(r)$ holds.

Theorem 1 (Invariant Principle, Floyd)**.** *If a preserved invariant holds for the start state then it is true for all reachable states.*

Proposition 2. *Preserved invariant of GCD algorithm starting from state (a, b) is that $P(x, y) :\ \gcd(x, y) = \gcd(a, b)$.*

Theorem 3. *When Euclid's algorithm halts, it correctly outputs the GCD of its inputs.*

Fast Exponentiation

For $a \in \mathbb{R}$ and $b \in \mathbb{N}$, the goal is to compute a^b.

```
FastExp(a,b)
    x = a; y = 1; z = b;
    while (z ≠ 0)
        r = rem(z,2)
        z = qcnt(z,2)
        if (r = 1) then y = xy
        x = x*x
    return y
```

Proposition 4. *Preserved invariant for fast exponentiation is $P(x, y, z)$: $z \in \mathbb{N}$ AND $yx^z = a^b$.*

Theorem 5 (Partial Correctness)**.** *When the algorithm halts, the value returned is a^b.*

Extended GCD Algorithm

Theorem 6 (Bézout)**.** *For any integers a, b, $\gcd(a, b)$ is a linear combination of a and b, i.e., there are integers s, t such that $\gcd(a, b) = sa + tb$.*

To compute $\gcd(a, b)$, we can assume WLOG a, b are positive, and $a \geq b$.

```
gcd(a,b)
    x = a; y = b;


    while (y > 0)

        r = rem(x,y)
        x = y
        y = r



    return x
```