Proving algorithms correct

math, simpler algorithm, ...

"Correct" means implementation matches specification

Proving GCDRec is correct, ie

$$GCDRec(a,b) = gcd(a,b) \quad \text{for all } a,b \in \mathbb{N}.$$

↑ implementation, recursive algorithm

↑ specification, math defn.

Proof: By (strong) induction on b.

- Base case: when $b=0$,
  $GCDRec(a,0) = a = gcd(a,0)$ ✓.
- Now, let $b > 0$ and suppose
  $GCDRec(a,b') = gcd(a,b')$
  for any a and any $b' < b$.

Then

$GCDRec(a,b)$

$=$ { definition of GCDRec }

$GCDRec(b, a \bmod b)$

$=$ { Ind. hyp — GCDRec is correct for $b' < b$ } + $a \bmod b < b$.

$gcd(b, a \bmod b)$

$=$ { math }

$gcd(a,b)$.

∎

Prove: GCDIter correct.

key idea: loop invariant. True after each loop iteration.

In this case: $gcd(m,n) = gcd(a,b)$.

- This invariant is true before the loop runs the first time because $a=m$ and $b=n$.
- If the invariant is true, then it will still be true after the loop body executes once.

  Because:
  - $gcd(a,b) = gcd(a \bmod b, b)$
  - we know a, b will not be 0.
- If the invariant is true when the loop finishes, then it implies correctness of the algorithm.
  - When loop stops, a or b is 0.
    if $a=0$, $gcd(0,b)=b$ + vice versa.
    Since $gcd(a,b) = gcd(m,n)$, this is correct.

- we must also show the while loop always stops after a finite # of iterations.

  Each loop, one stays same + other gets smaller.