Mathematics for Computer Science

MIT 6.042J/18.062J

State Machines, I: Invariants

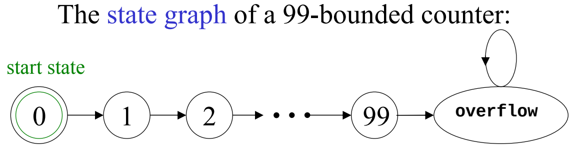
State machines

State machine:

Step by step procedure, possibly responding to input.

State machines

The state graph of a 99-bounded counter:



States: {0,1,…,99, `overflow`}

Transitions: i → i+1 $0 \le i < 99$

99 → `overflow`

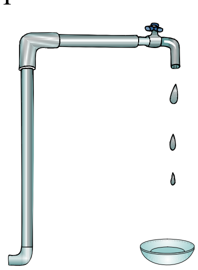
`overflow` → `overflow`

Die Hard

Picture source: http://movieweb.com/movie/diehard3/

Die Hard Once & For All

Supplies:



Water

3 Gallon Jug

9 Gallon Jug

State machines

Die hard state machine

State = amount of water in the jug: (b,l)

where $0 \le b \le 9$ and $0 \le l \le 3$.

Start State = (0,0)

State machines

Die Hard Transitions:

1. Fill the little jug: $(b,l) \to (b,3)$ *for* $l < 3$
2. Fill the big jug: $(b,l) \to (9,l)$ *for* $b < 9$
3. Empty the little jug: $(b,l) \to (b,0)$ *for* $l > 0$
4. Empty the big jug: $(b,l) \to (0,l)$ *for* $b > 0$

 October 21, 2005 lec 7f.7

State machines

5. Pour from big jug into little jug (for $b > 0$):
   (i) If no overflow, then $(b,l) \to (0,\ b+l)$,
   $\underbrace{\text{no overflow}}_{b\,+\,l\,\leq\,3}$
   (ii) otherwise $(b,l) \to (b-(3-l),\ 3)$.

6. Pour from little jug into big jug.
   Likewise.

 October 21, 2005 lec 7f.8

State Invariants

Die hard once and for all

Invariant:

P(state) ::= "3 divides the number of gallons in each jug."

$$P((b,l)) ::= (3 \mid b \wedge 3 \mid l)$$

 October 21, 2005 lec 7f.9

State Invariants

Floyd's Invariant Method

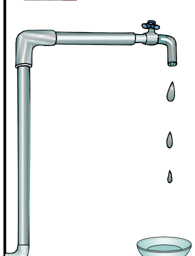
(just like induction)

1) Base case: Show $P(start)$.
2) Invariant case: Show

   if $P(q)$ and $q \longrightarrow r$, then $P(r)$.
3) Conclusion: P holds for *all reachable states*, including final state (if any).

 October 21, 2005 lec 7f.10

Die Hard Once & For All

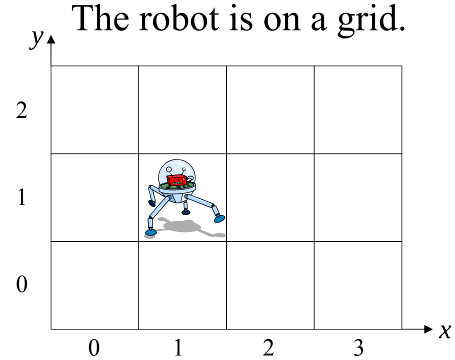


Corollary: No state (4,x) is reachable, so Bruce Dies!

 October 21, 2005 lec 7f.11

A Robot on the Diagonal

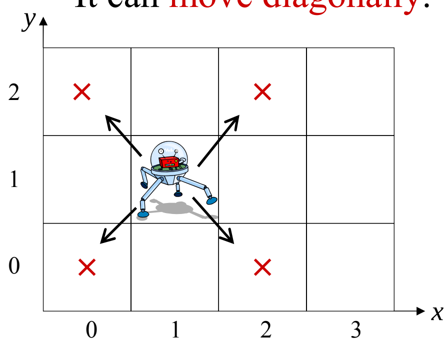
The robot is on a grid.



 October 21, 2005 lec 7f.12

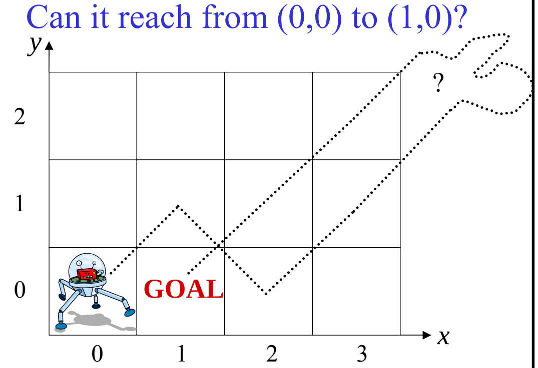
A Robot on the Diagonal

It can move diagonally.



A Robot on the Diagonal

Can it reach from (0,0) to (1,0)?



Robot Invariant

NO!

$P((x, y)) ::= x + y$ is even
$P((0, 0))$ is true.

Transition adds ±1 to **both** x and y

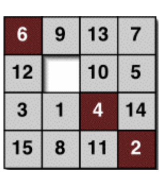
Robot Invariant

So all positions (x, y) reachable by robot have $x + y$ even, but $1 + 0 = 1$ is odd.

Therefore (1,0) is not reachable.

Team Problem

Problem 1



The Fifteen Puzzle Explained!

GCD correctness

The Euclidean Algorithm:

Computing GCD(a, b)

1. Set $x := a$, $y := b$.
2. If $y = 0$, return x & terminate;
3. else set $(x, y) := (y, \text{rem}(x,y))$ *simultaneously*;
4. Go to step 2.

GCD correctness

Example: GCD(414,662)

= GCD(662, 414) since rem(414,662) = 414

= GCD(414, 248) since rem(662,414) = 248

= GCD(248, 166) since rem(414,248) = 166

= GCD(166, 82) since rem(248,166) = 82

= GCD(82, 2) since rem(166,82) = 2

= GCD(2, 0) since rem(82,2) = 0

Return value: 2.

GCD correctness

Euclid Algorithm as State Machine:

- States ::= $\mathbb{N}\times\mathbb{N}$,
- start ::= (a,b),
- state transitions defined by the rule

  $(x,y) \to (y, \text{rem}(x,y))$ for $y \neq 0$.

GCD correctness

The Invariant is

$P((x,y))$ ::= $[\gcd(a,b) = \gcd(x,y)]$.

P(*start*): at start $x = a$, $y = b$, so

$P(start) \equiv [\gcd(a,b) = \gcd(a,b)]$

which holds trivially.

GCD correctness

Transitions: $(x, y) \to (y, \text{rem}(x, y))$

Invariant holds by

Lemma: $\gcd(x, y) = \gcd(y, \text{rem}(x,y))$,

for $y \neq 0$.

GCD correctness

Conclusion: on termination

$x = \gcd(a,b)$.

Proof: On termination, $y = 0$, so

$x = \gcd(x, 0) = \underbrace{\gcd(x, y) = \gcd(a, b)}_{\text{invariant}}$

GCD Termination

y decreases at each step &

$y \in \mathbb{N}$

(another invariant).

Well Ordering implies reaches minimum & stops.

Robert W Floyd (1934–2001)

Eulogy by Knuth: http://www.acm.org/pubs/membernet/stories/floyd.pdf
Picture source: http://www.stanford.edu/dept/news/report/news/november7/floydobit-117.html

Team Problem

Problem 2

