CSC3310 Algorithms

Proof by Induction

William Retert

Milwaukee School of Engineering

Induction

Induction Proof technique for showing that some predicate is true for all positive integers.

Induction

Induction Proof technique for showing that some predicate is true for all positive integers.

Possible predicates $P(n)$:

- $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$
- $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$
- Peasant multiplication generates the correct result when multiplying an n-bit integer x by an integer y
- etc.

Basic steps:

1. Show $P(n)$ true for $n = 0$
2. Show that if $P(n)$ is true, then $P(n+1)$ is also true.

Induction

Induction Proof technique for showing that some predicate is true for all positive integers.

Possible predicates $P(n)$:

- $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$
- $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$
- Peasant multiplication generates the correct result when multiplying an n-bit integer x by an integer y
- etc.

Basic steps:

1. Show $P(n)$ true for $n = 0$
2. Show that if $P(n-1)$ is true, then $P(n)$ is also true.

Induction

Induction Proof technique for showing that some predicate is true for all positive integers.

Possible predicates $P(n)$:

- $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$
- $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$
- Peasant multiplication generates the correct result when multiplying an n-bit integer x by an integer y
- etc.

Basic steps:

1. Show $P(n)$ true for $n = 0$
2. Show that if $P(n-1)$ is true, then $P(n)$ is also true. Alternately, (strong induction), show that if $P(i)$ is true $\forall 1 \leq i < n$, then $P(n)$ is true

$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

$n = 0$

$$\sum_{i=0}^{0} i = 0 = \frac{0(1)}{2}$$

$P(n-1) \vdash P(n)$

$$\begin{aligned}
\sum_{i=0}^{n} i &= n + \sum_{i=0}^{n-1} i \\
&= n + \frac{(n-1)((n-1)+1)}{2} \qquad \text{(Induction!)} \\
&= n + \frac{(n-1)n}{2} \\
&= \frac{2n}{2} + \frac{n^2-n}{2} \\
&= \frac{n^2+n}{2} \\
&= \frac{n(n+1)}{2}
\end{aligned}$$

$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$

$n = 0$

$$\sum_{i=0}^{0} 2^i = 2^0 = 1 = 2^1 - 1$$

$P(n-1) \vdash P(n)$

$$\begin{aligned}\sum_{i=0}^{n} 2^i &= 2^n + \sum_{i=0}^{n-1} 2^i \\ &= 2^n + (2^n - 1) && \text{(Induction!)} \\ &= 2 \cdot (2^n) - 1 \\ &= 2^{n+1} - 1\end{aligned}$$

"Peasant" Multiplication

Theorem

The "peasant" multiplication algorithm is correct. That is, for any n-bit integer x, $\textsc{Peasant}(x, y) = x \cdot y \ \ \forall y \in \mathbb{Z}$

procedure PEASANT(x,y)
  $res \leftarrow 0$
  while $x > 0$ **do**
    if x is odd **then**
      $res \leftarrow res + y$
    end if
    $x \leftarrow \lfloor x \div 2 \rfloor$
    $y \leftarrow y + y$
  end while
end procedure

We will rewrite the algorithm to be recursive in order to simplify the proof.

"Peasant" Multiplication

Theorem

The "peasant" multiplication algorithm is correct. That is, for any n-bit integer x,
$\textsc{Peasant}(x, y) = x \cdot y \ \ \forall y \in \mathbb{Z}$

procedure $\textsc{Peasant}$(x,y)
  $res \leftarrow 0$
  if $x > 0$ **then**
    $res \leftarrow \textsc{Peasant}(\lfloor x \div 2 \rfloor, y + y)$
    if x is odd **then**
      $res \leftarrow res + y$
    end if
  end if
  return *res*
end procedure

“Peasant” Multiplication

Theorem

The “peasant” multiplication algorithm is correct. That is, for any n-bit integer x, $\textsc{Peasant}(x, y) = x \cdot y \;\; \forall y \in \mathbb{Z}$

procedure $\textsc{Peasant}$(x,y)
    $res \leftarrow 0$
    if $x > 0$ **then**
        $res \leftarrow \textsc{Peasant}(\lfloor x \div 2 \rfloor, y + y)$
        if x is odd **then**
            $res \leftarrow res + y$
        end if
    end if
    return *res*
end procedure

$n = 0$

If x has 0 bits, then $x = 0$, and the algorithm correctly computes $0 \times y = 0$

"Peasant" Multiplication

Theorem

The "peasant" multiplication algorithm is correct. That is, for any n-bit integer x,
$\textsc{Peasant}(x, y) = x \cdot y \ \ \forall y \in \mathbb{Z}$

```
procedure PEASANT(x,y)
    res ← 0
    if x > 0 then
        res ← PEASANT(⌊x÷2⌋, y+y)
        if x is odd then
            res ← res + y
        end if
    end if
    return res
end procedure
```

$P(n-1) \vdash P(n)$

If x is a n-bit number, then $\lfloor x \div 2 \rfloor$ is an $(n-1)$ bit number. Therefore, by induction,
$\textsc{Peasant}(\lfloor x \div 2 \rfloor, y+y) = (\lfloor x \div 2 \rfloor) \times (y+y)$
If x is even, our return value is then

$$\frac{x}{2} \times (2y) = x \cdot y$$

. If x is odd, the return value is

$$\begin{aligned} \frac{x-1}{2} \times (2y) + y &= (x-1)y + y \\ &= x \cdot y - y + y \\ &= x \cdot y \end{aligned}$$