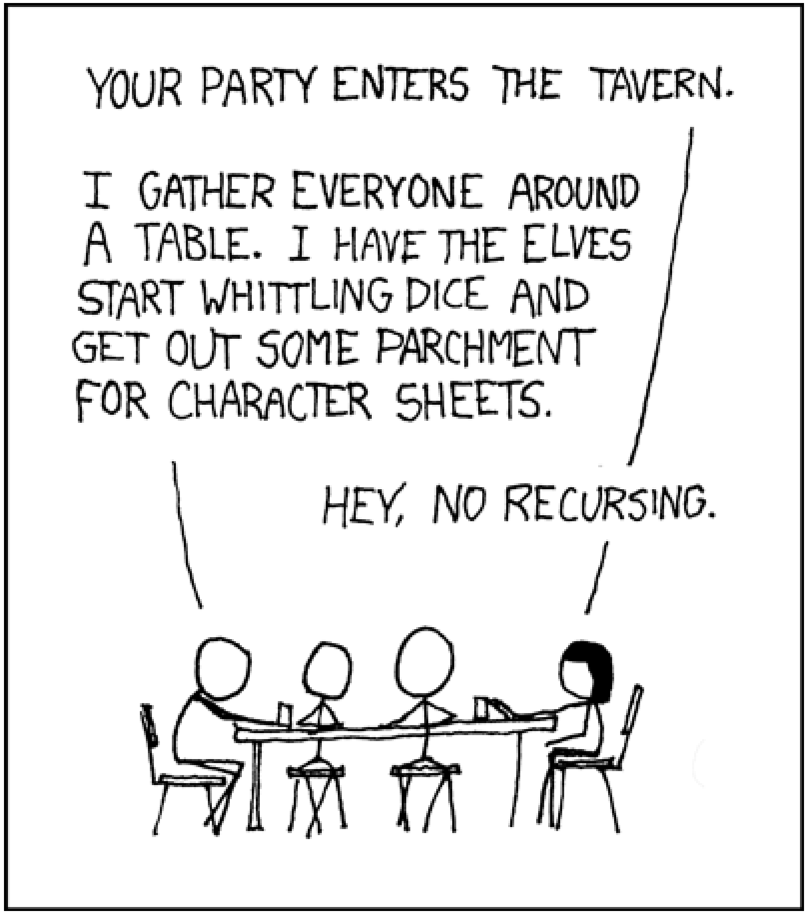
Fall 2015

Lecture 17: Strong induction & Recursive definitions



administrative

Midterm review session Sunday @ 1:00 pm (EEB 105)

MIDTERM MONDAY (IN THIS ROOM, USUAL TIME)

No office hours on Monday/Wednesday

Closed book.
One page (front and back) of notes allowed.

Exam includes induction!
Homework #5 is due on Friday, Nov 13th.

$$P(0)$$

$$\forall k\ \Big(\big(P(0) \land P(1) \land P(2) \land \cdots \land P(k)\big) \rightarrow P(k+1)\Big)$$

$$\therefore \forall n\ P(n)$$

Follows from ordinary induction applied to

$$Q(n) = P(0) \land P(1) \land P(2) \land \cdots \land P(n)$$

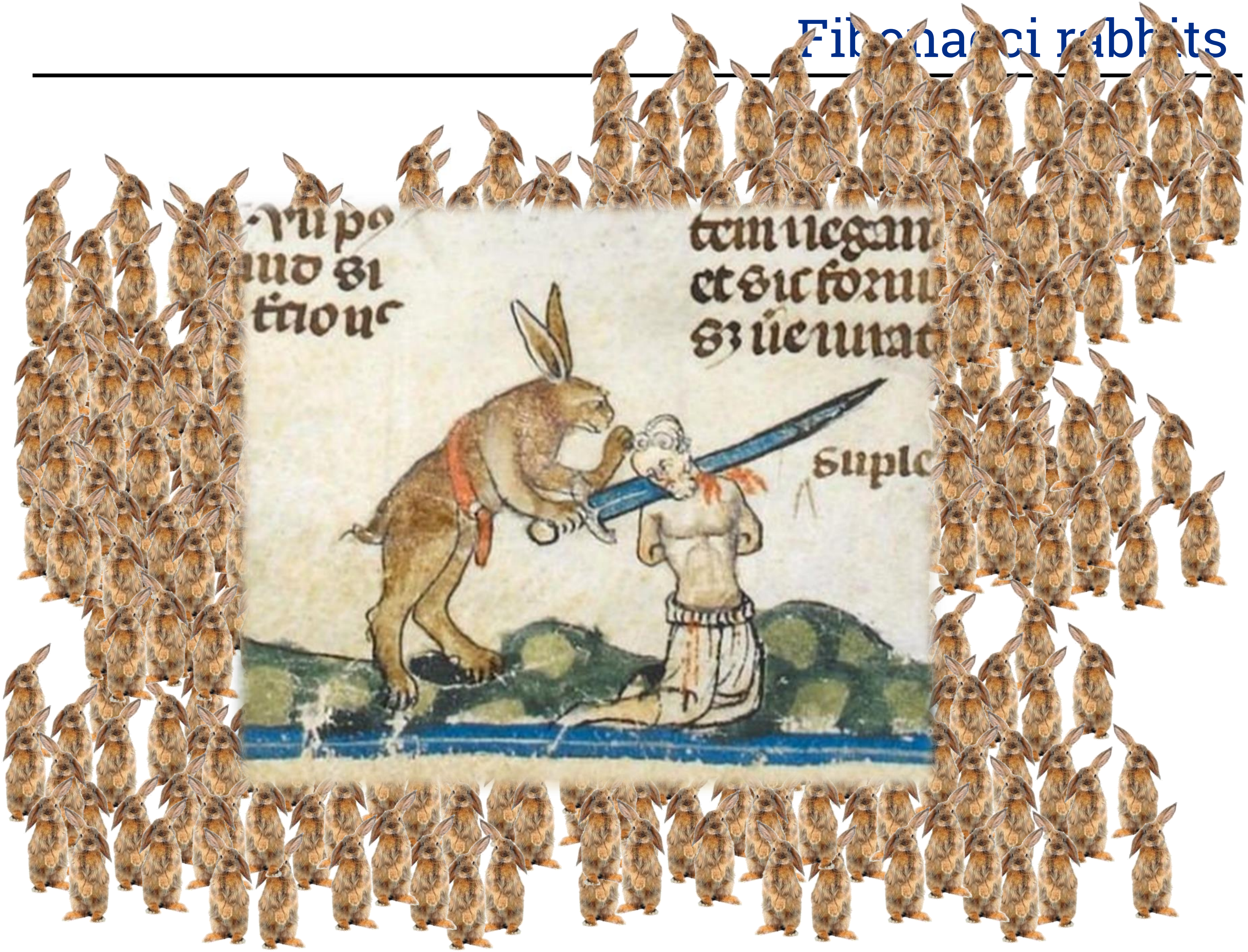
1. By induction we will show that $P(n)$ is true for every $n \geq 0$
2. Base Case: Prove $P(0)$
3. Inductive Hypothesis: $P(0) \wedge P(1) \wedge \dots \wedge P(k)$
   Assume that for some arbitrary integer $k \geq 0$, $P(j)$ is true for every j from 0 to k
4. Inductive Step:
   Prove that $P(k+1)$ is true using the Inductive Hypothesis (that $P(j)$ is true for all values $\leq k$)
5. Conclusion: Result follows by induction

0, 1, 1, 2, 3, 5, 8, ...

Fibonacci numbers

$$f_0 = 0$$
$$f_1 = 1$$
$$f_n = f_{n-1} + f_{n-2} \text{ for all } n \geq 2$$

Fibonacci rabbits



$f_0 = 0,\ f_1 = 1$

bounding the Fibonacci numbers

Theorem: $f_n < 2^n$ for all $n \geq 2$

By strong ind: $P(n)$ is true $\forall n \geq 0$

$P(n) =$ "$f_n < 2^n$"

Base case: $P(0)$: $0 = f_0 < 2^0 = 1$ ✓

$P(1)$: $1 = f_1 < 2^1 = 2$ ✓

IH: Assume for some $k \geq 1$
$P(j)$ holds for $0 \leq j \leq k$

IS: $f_{k+1} = f_k + f_{k-1}$ by def'n $k+1 \geq 2$

$< 2^k + 2^{k-1}$ by IH, $P(k)$, $P(k-1)$

$= 2^{k-1}(2+1) < 4 \cdot 2^{k-1} = 2^{k+1}$

0 1 1 2

bounding the Fibonacci numbers

Theorem: $\underbrace{2^{\frac{n}{2}-1} \le f_n}$ $< 2^n$ for all $n \ge 2$

$1+\sqrt{2} \ge 2$

$P(n) =$ " $f_n \ge 2^{\frac{n}{2}-1}$ "

Base case: $P(2) =$ " $1 \ge 2^{\frac{2}{2}-1} = 1$ " ✓

IH: Assume for some $k \ge 2$ that $P(j)$ holds for all $2 \le j \le k$.

$P(2)$
$P(3)$
$\vdots$
$P(k)$
IH

[If $k=2$ then $f_3 \ge 2^{\frac{3}{2}-1}$ b/c $2 \ge \sqrt{2}$]

IS: $f_{k+1} = f_k + f_{k-1}$ by defn and $k+1 \ge 2$

Otherwise for $k \ge 3$

$P(k+1)$

$\overset{(IH)}{\ge} 2^{\frac{k}{2}-1} + 2^{\frac{k-1}{2}-1} = \frac{1}{2}\left(2^{\frac{k}{2}} + 2^{\frac{k-1}{2}}\right)$ $2^{\frac{k+1}{2}}$ ''

$= \frac{1}{2} 2^{\frac{k-1}{2}}\left(2^{\frac{1}{2}} + 1\right) \ge \frac{1}{2} 2^{\frac{k-1}{2}} \cdot 2$

By str ind, $P(n)\ \forall n \ge 2$.

$f_0 = 0$; $f_1 = 1$; $f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Theorem: $2^{n/2-1} \leq f_n < 2^n$ **for all** $n \geq 2$

Proof:

1. Let P(n) be "$2^{n/2-1} \leq f_n < 2^n$. By (strong) induction we prove P(n) for all n ≥ 2.
2. **Base Case:** P(2) is true: $f_2=1$, $2^{2/2-1}=2^0=1 \leq f_2$, $2^2=4>f_2$
3. **Ind.Hyp:** Assume $2^{j/2-1} \leq f_j < 2^j$ for all integers j with $2 \leq j \leq k$ for for some arbitrary integer k ≥ 2.
4. **Ind. Step:** $\boxed{\text{Goal: Show } 2^{(k+1)/2-1} \leq f_{k+1} < 2^{k+1}}$

   <u>Case k=2</u>: P(3) is true: $f_3=f_2+f_1=1+1=2$, $2^{3/2-1}=2^{1/2} \leq 2 = f_3$, $2^3=8 > f_3$

   <u>Case k≥3:</u>

$$f_{k+1} = f_k + f_{k-1} \geq 2^{k/2-1} + 2^{(k-1)/2-1} \quad \text{by I.H. since k-1} \geq 2$$

$$> 2^{(k-1)/2-1} + 2^{(k-1)/2-1} = 2 \cdot 2^{(k-1)/2-1} = 2^{(k+1)/2-1}$$

$$f_{k+1} = f_k + f_{k-1} < 2^k + 2^{(k-1)} \quad \text{by I.H. since k-1} \geq 2$$

$$< 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}$$

For all $d > 0$

Theorem: For any integers n ≥ 0 and $d \geq 0$, there are integers q and r such that $n = dq + r$ and $0 \leq r \leq d - 1$.

$P(n) =$

Base case: $P(0)$: For any $d > 0$, $0 = d \cdot 0 + 0$

So $P(0)$ holds.

IH: For some $k \geq 0$, $P(j)$ holds for all $0 \leq j \leq k$.

IS: Let $d \geq 0$ be arbitrary

[Goal: $k+1 = dq + r$ for some q, r $0 \leq r \leq d-1$]

If $k+1 < d$ $k+1 = d \cdot 0 + \underbrace{k+1}$ $\Rightarrow P(k+1)$

$r = k+1 \leq d-1$

If $k+1 \geq d$ then $k+1 = a + d$ for some $0 \leq a \leq k$

By $P(a)$: $a = d \cdot q' + r$ $k+1 = d(q'+1) + r$ $\Rightarrow P(k+1)$

running time of Euclid's algorithm

running time of Euclid's algorithm

Theorem: Suppose that Euclid's algorithm takes n steps for $\gcd(a, b)$ with $a > b$, then $a \geq f_{n+1}$.

Proof:

Set $r_{n+1} = a, r_n = b$ then Euclid's algorithm computes

$$
\begin{aligned}
r_{n+1} &= q_n r_n + r_{n-1} \\
r_n &= q_{n-1} r_{n-1} + r_{n-2} \\
&\vdots \\
r_3 &= q_2 r_2 + r_1 \\
r_2 &= q_1 r_1
\end{aligned}
$$

each quotient $q_i \geq 1$

$r_1 \geq 1$