CSE 311 Lecture 17: Strong Induction

Emina Torlak and Sami Davies

Topics

Induction

A brief review of Lecture 16.

Strong induction

Induction with a stronger hypothesis.

Using strong induction

An example proof and when to use strong induction.

Recursively defined functions

Recursive function definitions and examples.

Induction

A brief review of Lecture 16.

A template for proofs by induction

① **Let $P(n)$ be** *[definition of $P(n)$]*.

We will show that $P(n)$ is true for every integer $n \geq 0$ by induction.

② **Base case ($n = 0$):**

[Proof of $P(0)$.]

③ **Inductive hypothesis:**

Suppose that $P(k)$ is true for an arbitrary integer $k \geq 0$.

④ **Inductive step:**

We want to prove that $P(k+1)$ is true.

*[Proof of $P(k+1)$. This proof **must** invoke the inductive hypothesis.]*

⑤ **The result follows for all $n \geq 0$ by induction.**

Induction $$\frac{P(0);\, \forall k.\, P(k) \rightarrow P(k+1)}{\therefore\ \forall n.\, P(n)}$$

A template for proofs starting at any integer $b \in \mathbb{Z}$

① **Let $P(n)$ be** *[definition of $P(n)$]*.

We will show that $P(n)$ is true for every integer $n \geq b$ by induction.

② **Base case ($n = b$):**

[Proof of $P(b)$.]

③ **Inductive hypothesis:**

Suppose that $P(k)$ is true for an arbitrary integer $k \geq b$.

④ **Inductive step:**

We want to prove that $P(k+1)$ is true.

[Proof of $P(k+1)$. This proof ***must*** *invoke the inductive hypothesis.]*

⑤ **The result follows for all $n \geq b$ by induction.**

Induction with a stricter hypothesis

If you can't prove $P(k) \rightarrow P(k+1)$ then you need a stricter hypothesis!

Find a $Q(n)$ such that $Q(n) \rightarrow P(n)$ and $Q(k) \rightarrow Q(k+1)$.

Prove $Q(n)$ by induction to conclude that $P(n)$ holds (by MP).

There is no recipe for finding $Q(n)$.

But looking at examples can help.

And also thinking about why the attempted proof of $P(k) \rightarrow P(k+1)$ fails.

For a proof like this, you must argue that $Q(n) \rightarrow P(n)$.

If it's not obvious that $P(n)$ follows from $Q(n)$, you have to prove it.

Otherwise, it's enough to state that you are using a stricter hypothesis.

There is no recipe for determining what is obvious :)

Strong induction

Induction with a stronger hypothesis.

Recall how induction works

$$\textbf{Induction}\ \frac{P(0);\ \forall k.\ P(k) \to P(k+1)}{\therefore \forall n.\ P(n)}$$

Domain: natural numbers (ℕ).

How do we get $P(5)$ from $P(0)$ and $\forall k.\ P(k) \to P(k+1)$?

1. First, we have $P(0)$.
2. Since $P(k) \to P(k+1)$ for all k, we have $P(0) \to P(1)$.
3. Applying Modus Ponens to 1 and 2, we get $P(1)$.
4. Since $P(k) \to P(k+1)$ for all k, we have $P(1) \to P(2)$.
5. Applying Modus Ponens to 3 and 4, we get $P(2)$.

⋮

11. Applying Modus Ponens to 9 and 10, we get $P(5)$.

$$\begin{array}{l} P(0) \\ \quad \Downarrow \scriptstyle P(0)\to P(1) \\ P(1) \\ \quad \Downarrow \scriptstyle P(1)\to P(2) \\ P(2) \\ \quad \Downarrow \scriptstyle P(k)\to P(k+1) \\ P(5) \end{array}$$

Recall how induction works

$$\textbf{Induction}\ \frac{P(0);\ \forall k.\ P(k) \rightarrow P(k+1)}{\therefore\ \forall n.\ P(n)}$$

Domain: natural numbers ($\mathbb{N}$).

How do we get $P(5)$ from $P(0)$ and $\forall k.\ P(k) \rightarrow P(k+1)$?

1. First, we have $P(0)$.
2. Since $P(k) \rightarrow P(k+1)$ for all k, we have $P(0) \rightarrow P(1)$.
3. Applying Modus Ponens to 1 and 2, we get $P(1)$.
4. Since $P(k) \rightarrow P(k+1)$ for all k, we have $P(1) \rightarrow P(2)$.
5. Applying Modus Ponens to 3 and 4, we get $P(2)$.

$\vdots$

11. Applying Modus Ponens to 9 and 10, we get $P(5)$.

$$\begin{tikzcd}
P(0) \arrow[d, Rightarrow, "P(0)\rightarrow P(1)"] \\
P(1) \arrow[d, Rightarrow, "P(1)\rightarrow P(2)"] \\
P(2) \arrow[d, Rightarrow, "P(k)\rightarrow P(k+1)"] \\
P(5)
\end{tikzcd}$$

Note that we have $P(0), \dots, P(k)$ when proving $k+1$.
So we can safely assume all of them, rather than just $P(k)$.

The strong induction rule of inference

$$\textbf{Strong Induction}\ \frac{P(0);\ \forall k.\ (P(0) \land P(1) \land \ldots \land P(k)) \to P(k+1)}{\therefore\ \forall n.\ P(n)}$$

Domain: ℕ.

The strong induction rule of inference

Strong Induction $$\frac{P(0);\ \forall k.\ (P(0) \land P(1) \land \ldots \land P(k)) \rightarrow P(k+1)}{\therefore\ \forall n.\ P(n)}$$

Domain: ℕ.

Strong induction for P follows from ordinary induction for Q where

$$Q(k) = P(0) \land P(1) \land P(2) \land \ldots \land P(k)$$

The strong induction rule of inference

Strong Induction $$\frac{P(0);\ \forall k.\ (P(0) \land P(1) \land \ldots \land P(k)) \to P(k+1)}{\therefore\ \forall n.\ P(n)}$$

Domain: $\mathbb{N}$.

Strong induction for P follows from ordinary induction for Q where

$$Q(k) = P(0) \land P(1) \land P(2) \land \ldots \land P(k)$$

To see why, note the following:

$$Q(0) \equiv P(0)$$
$$Q(k+1) \equiv Q(k) \land P(k+1)$$
$$(\forall n.\ Q(n)) \equiv (\forall n.\ P(n))$$

Strong inductive proofs for any base case $b \in \mathbb{Z}$

① **Let $P(n)$ be *[definition of $P(n)$]*.**

We will show that $P(n)$ is true for every integer $n \geq b$ by strong induction.

② **Base case ($n = b$):**

[Proof of $P(b)$.]

③ **Inductive hypothesis:**

Suppose that for some arbitrary integer $k \geq b$, $P(j)$ is true for every integer $b \leq j \leq k$.

④ **Inductive step:**

We want to prove that $P(k + 1)$ is true.

*[Proof of $P(k + 1)$. The proof **must** invoke the strong inductive hypothesis.]*

⑤ **The result follows for all $n \geq b$ by strong induction.**

Using strong induction

An example proof and when to use strong induction.

Example: the fundamental theorem of arithmetic

Fundamental theorem of arithmetic

Every positive integer greater than 1 has a unique prime factorization.

Examples

$$48 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3$$
$$591 = 3 \cdot 197$$
$$45,523 = 45,523$$
$$321,950 = 2 \cdot 5 \cdot 5 \cdot 47 \cdot 137$$
$$1,234,567,890 = 2 \cdot 3 \cdot 3 \cdot 5 \cdot 3,607 \cdot 3,803$$

Example: the fundamental theorem of arithmetic

Fundamental theorem of arithmetic

Every positive integer greater than 1 has a unique prime factorization.

Examples

$$48 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3$$
$$591 = 3 \cdot 197$$
$$45,523 = 45,523$$
$$321,950 = 2 \cdot 5 \cdot 5 \cdot 47 \cdot 137$$
$$1,234,567,890 = 2 \cdot 3 \cdot 3 \cdot 5 \cdot 3,607 \cdot 3,803$$

We use strong induction to prove that a factorization into primes exists (but not that it is unique).

Prove that every integer ≥ 2 is a product of primes

Prove that every integer ≥ 2 is a product of primes

① **Let $P(n)$ be "n is a product of one or more primes".**

We will show that $P(n)$ is true for every integer $n \geq 2$ by strong induction.

Prove that every integer ≥ 2 is a product of primes

① **Let $P(n)$ be "n is a product of one or more primes".**

We will show that $P(n)$ is true for every integer $n \geq 2$ by strong induction.

② **Base case ($n = 2$):**

2 is prime, so it is a product of primes. Therefore $P(2)$ is true.

Prove that every integer ≥ 2 is a product of primes

① **Let $P(n)$ be "n is a product of one or more primes".**

We will show that $P(n)$ is true for every integer $n \geq 2$ by strong induction.

② **Base case** ($n = 2$)**:**

2 is prime, so it is a product of primes. Therefore $P(2)$ is true.

③ **Inductive hypothesis:**

Suppose that for some arbitrary integer $k \geq 2$, $P(j)$ is true for every integer $2 \leq j \leq k$.

Prove that every integer ≥ 2 is a product of primes

① **Let $P(n)$ be "n is a product of one or more primes".**

We will show that $P(n)$ is true for every integer $n \geq 2$ by strong induction.

② **Base case ($n = 2$):**

2 is prime, so it is a product of primes. Therefore $P(2)$ is true.

③ **Inductive hypothesis:**

Suppose that for some arbitrary integer $k \geq 2$, $P(j)$ is true for every integer $2 \leq j \leq k$.

④ **Inductive step:**

We want to prove that $P(k + 1)$ is true, i.e., $k + 1$ is a product of primes.

Prove that every integer ≥ 2 is a product of primes

① **Let $P(n)$ be "n is a product of one or more primes".**

We will show that $P(n)$ is true for every integer $n \geq 2$ by strong induction.

② **Base case ($n = 2$):**

2 is prime, so it is a product of primes. Therefore $P(2)$ is true.

③ **Inductive hypothesis:**

Suppose that for some arbitrary integer $k \geq 2, P(j)$ is true for every integer $2 \leq j \leq k$.

④ **Inductive step:**

We want to prove that $P(k+1)$ is true, i.e., $k+1$ is a product of primes.

Case: $k+1$ is prime. Then by definition, $k+1$ is a product of primes.

Prove that every integer ≥ 2 is a product of primes

① **Let $P(n)$ be "n is a product of one or more primes".**

We will show that $P(n)$ is true for every integer $n \geq 2$ by strong induction.

② **Base case ($n = 2$):**

2 is prime, so it is a product of primes. Therefore $P(2)$ is true.

③ **Inductive hypothesis:**

Suppose that for some arbitrary integer $k \geq 2$, $P(j)$ is true for every integer $2 \leq j \leq k$.

④ **Inductive step:**

We want to prove that $P(k+1)$ is true, i.e., $k+1$ is a product of primes.

Case: $k+1$ is prime. Then by definition, $k+1$ is a product of primes.

Case: $k+1$ is composite. Then by $k+1 = ab$ for some integers ab where $2 \leq a, b \leq k$.

By inductive hypothesis, we have $P(a) = p_1 p_2 \ldots p_r$ and $P(b) = q_1 q_2 \ldots q_s$, where $p_1, p_2, \ldots, p_r, q_1, q_2, \ldots, q_s$ are prime. Thus, $k+1 = ab = p_1 p_2 \ldots p_r q_1 q_2 \ldots q_s$, which is a product of primes.

Prove that every integer ≥ 2 is a product of primes

① **Let $P(n)$ be "n is a product of one or more primes".**

We will show that $P(n)$ is true for every integer $n \geq 2$ by strong induction.

② **Base case ($n = 2$):**

2 is prime, so it is a product of primes. Therefore $P(2)$ is true.

③ **Inductive hypothesis:**

Suppose that for some arbitrary integer $k \geq 2$, $P(j)$ is true for every integer $2 \leq j \leq k$.

④ **Inductive step:**

We want to prove that $P(k+1)$ is true, i.e., $k+1$ is a product of primes.

Case: $k+1$ is prime. Then by definition, $k+1$ is a product of primes.

Case: $k+1$ is composite. Then by $k+1 = ab$ for some integers ab where $2 \leq a, b \leq k$.

By inductive hypothesis, we have $P(a) = p_1 p_2 \ldots p_r$ and $P(b) = q_1 q_2 \ldots q_s$, where $p_1, p_2, \ldots, p_r, q_1, q_2, \ldots, q_s$ are prime. Thus, $k+1 = ab = p_1 p_2 \ldots p_r q_1 q_2 \ldots q_s$, which is a product of primes.

⑤ **The result follows for all $n \geq 2$ by strong induction.**

Strong induction is particularly useful when …

We need to reason about procedures that given an input k invoke themselves recursively on an input different from $k - 1$.

Example:

Euclidean algorithm for computing $\mathrm{GCD}(a, b)$.

```
// Assumes a >= b >= 0.
public static int gcd(int a, int b) {
  if (b == 0)
    return a;             // GCD(a, 0) = a
  else
    return gcd(b, a % b); // GCD(a, b) = GCD(b, a mod b)
}
```

We will use strong induction to reason about this algorithm and other *functions with recursive definitions*.

Recursively defined functions

Recursive function definitions and examples.

Giving a recursive definition for a function

To define a recursive function f over $\mathbb{N}$, give its output in two cases:

Base case: the value of $f(0)$.

Recursive case: the value of $f(n+1)$, given in terms of $f(n)$.

Giving a recursive definition for a function

To define a recursive function f over $\mathbb{N}$, give its output in two cases:

- **Base case**: the value of $f(0)$.
- **Recursive case**: the value of $f(n+1)$, given in terms of $f(n)$.

Examples:

$F(0) = 1, F(n+1) = F(n) + 1$

$G(0) = 1, G(n+1) = 2 \cdot G(n)$

$K(0) = 1, K(n+1) = (n+1) \cdot K(n)$

Giving a recursive definition for a function

To define a recursive function f over $\mathbb{N}$, give its output in two cases:

Base case: the value of $f(0)$.

Recursive case: the value of $f(n+1)$, given in terms of $f(n)$.

Examples:

$F(0) = 1, F(n+1) = F(n) + 1$ $n + 1$ for $n \in \mathbb{N}$

$G(0) = 1, G(n+1) = 2 \cdot G(n)$

$K(0) = 1, K(n+1) = (n+1) \cdot K(n)$

Giving a recursive definition for a function

To define a recursive function f over $\mathbb{N}$, give its output in two cases:

Base case: the value of $f(0)$.

Recursive case: the value of $f(n+1)$, given in terms of $f(n)$.

Examples:

$F(0) = 1, F(n+1) = F(n) + 1$ $n+1$ for $n \in \mathbb{N}$

$G(0) = 1, G(n+1) = 2 \cdot G(n)$ 2^n for $n \in \mathbb{N}$

$K(0) = 1, K(n+1) = (n+1) \cdot K(n)$

Giving a recursive definition for a function

To define a recursive function f over $\mathbb{N}$, give its output in two cases:

Base case: the value of $f(0)$.

Recursive case: the value of $f(n+1)$, given in terms of $f(n)$.

Examples:

$F(0) = 1, F(n+1) = F(n) + 1$	$n+1$ for $n \in \mathbb{N}$
$G(0) = 1, G(n+1) = 2 \cdot G(n)$	2^n for $n \in \mathbb{N}$
$K(0) = 1, K(n+1) = (n+1) \cdot K(n)$	$n!$ for $n \in \mathbb{N}$

Giving a recursive definition for a function

To define a recursive function f over $\mathbb{N}$, give its output in two cases:

Base case: the value of $f(0)$.

Recursive case: the value of $f(n+1)$, given in terms of $f(n)$.

Examples:

$F(0) = 1, F(n+1) = F(n) + 1$	$n + 1$ for $n \in \mathbb{N}$
$G(0) = 1, G(n+1) = 2 \cdot G(n)$	2^n for $n \in \mathbb{N}$
$K(0) = 1, K(n+1) = (n+1) \cdot K(n)$	$n!$ for $n \in \mathbb{N}$

When the recursive case refers only to $f(n)$, as in these examples, we can prove properties of $f(n)$ easily using ordinary induction.

Example: prove $n! \leq n^n$ for all $n \geq 1$

Example: prove $n! \leq n^n$ for all $n \geq 1$

① **Let $P(n)$ be $n! \leq n^n$.**

We will show that $P(n)$ is true for every integer $n \geq 1$ by induction.

Example: prove $n! \leq n^n$ for all $n \geq 1$

① **Let $P(n)$ be $n! \leq n^n$.**

We will show that $P(n)$ is true for every integer $n \geq 1$ by induction.

② **Base case ($n = 1$):**

$1! = 1 \cdot 0! = 1 \cdot 1 = 1 = 1^1$ so $P(1)$ is true.

Example: prove $n! \leq n^n$ for all $n \geq 1$

① **Let $P(n)$ be $n! \leq n^n$.**

We will show that $P(n)$ is true for every integer $n \geq 1$ by induction.

② **Base case ($n = 1$):**

$1! = 1 \cdot 0! = 1 \cdot 1 = 1 = 1^1$ so $P(1)$ is true.

③ **Inductive hypothesis:**

Suppose that $P(k)$ is true for an arbitrary integer $k \geq 1$.

Example: prove $n! \leq n^n$ for all $n \geq 1$

① **Let $P(n)$ be $n! \leq n^n$.**

We will show that $P(n)$ is true for every integer $n \geq 1$ by induction.

② **Base case ($n = 1$):**

$1! = 1 \cdot 0! = 1 \cdot 1 = 1 = 1^1$ so $P(1)$ is true.

③ **Inductive hypothesis:**

Suppose that $P(k)$ is true for an arbitrary integer $k \geq 1$.

④ **Inductive step:**

We want to prove that $P(k+1)$ is true, i.e., $(k+1)! \leq (k+1)^{(k+1)}$.

$$
\begin{aligned}
(k+1)! &= (k+1) \cdot k! && \text{by definition of } ! \\
&\leq (k+1) \cdot k^k && \text{by the inductive hypothesis} \\
&\leq (k+1) \cdot (k+1)^k && \text{since } k \geq 0 \\
&= (k+1)^{(k+1)} && \text{which is exactly } P(k+1).
\end{aligned}
$$

Example: prove $n! \leq n^n$ for all $n \geq 1$

① **Let $P(n)$ be $n! \leq n^n$.**

We will show that $P(n)$ is true for every integer $n \geq 1$ by induction.

② **Base case ($n = 1$):**

$1! = 1 \cdot 0! = 1 \cdot 1 = 1 = 1^1$ so $P(1)$ is true.

③ **Inductive hypothesis:**

Suppose that $P(k)$ is true for an arbitrary integer $k \geq 1$.

④ **Inductive step:**

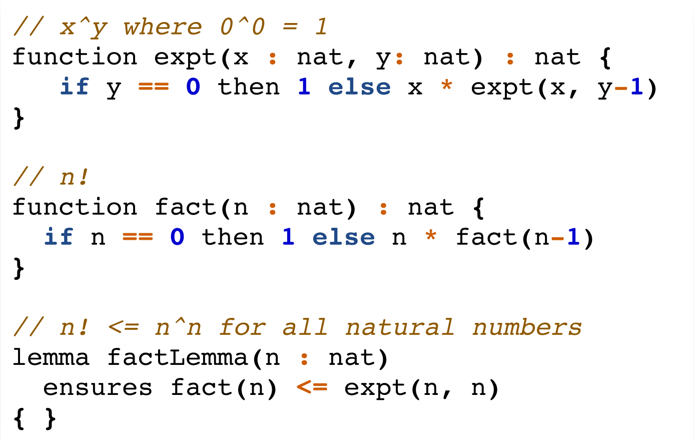
We want to prove that $P(k+1)$ is true, i.e., $(k+1)! \leq (k+1)^{(k+1)}$.

$$\begin{aligned}
(k+1)! &= (k+1) \cdot k! && \text{by definition of !} \\
&\leq (k+1) \cdot k^k && \text{by the inductive hypothesis} \\
&\leq (k+1) \cdot (k+1)^k && \text{since } k \geq 0 \\
&= (k+1)^{(k+1)} && \text{which is exactly } P(k+1).
\end{aligned}$$

⑤ **The result follows for all $n \geq 1$ by induction.**

Fun: can we verify $n! \leq n^n$ for all natural numbers?

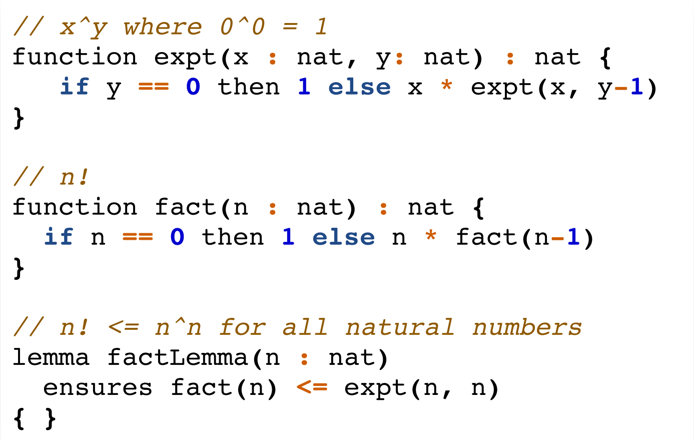
Prove $n! \leq n^n$ for $n \in \mathbb{N}$ with Dafny:



```
// x^y where 0^0 = 1
function expt(x : nat, y: nat) : nat {
   if y == 0 then 1 else x * expt(x, y-1)
}

// n!
function fact(n : nat) : nat {
  if n == 0 then 1 else n * fact(n-1)
}

// n! <= n^n for all natural numbers
lemma factLemma(n : nat)
  ensures fact(n) <= expt(n, n)
{ }
```

Fun: can we verify $n! \leq n^n$ for all natural numbers?

Prove $n! \leq n^n$ for $n \in \mathbb{N}$ with Dafny:

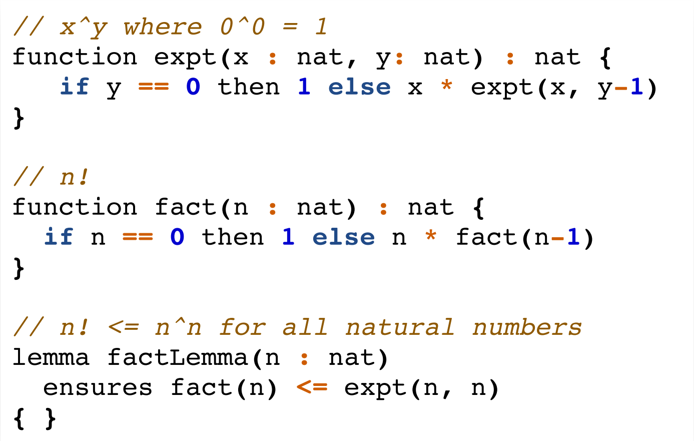


```
// x^y where 0^0 = 1
function expt(x : nat, y: nat) : nat {
   if y == 0 then 1 else x * expt(x, y-1)
}

// n!
function fact(n : nat) : nat {
  if n == 0 then 1 else n * fact(n-1)
}

// n! <= n^n for all natural numbers
lemma factLemma(n : nat)
  ensures fact(n) <= expt(n, n)
{ }
```

Dafny can't prove this theorem because the proof involves several steps that are too difficult for Dafny to discover on its own.

Fun: can we verify $n! \leq n^n$ for all natural numbers?

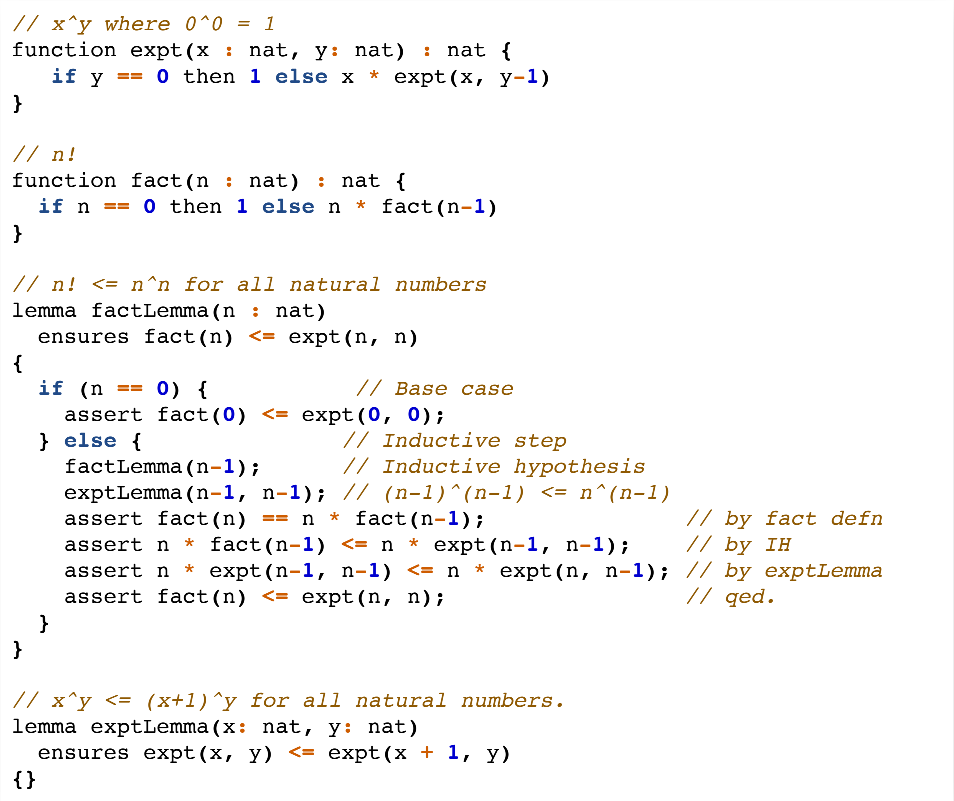
Prove $n! \leq n^n$ for $n \in \mathbb{N}$ with Dafny:



```
// x^y where 0^0 = 1
function expt(x : nat, y: nat) : nat {
   if y == 0 then 1 else x * expt(x, y-1)
}

// n!
function fact(n : nat) : nat {
  if n == 0 then 1 else n * fact(n-1)
}

// n! <= n^n for all natural numbers
lemma factLemma(n : nat)
  ensures fact(n) <= expt(n, n)
{ }
```

Dafny can’t prove this theorem because the proof involves several steps that are too difficult for Dafny to discover on its own.

Really prove $n! \leq n^n$ for $n \in \mathbb{N}$ with Dafny:



```
// x^y where 0^0 = 1
function expt(x : nat, y: nat) : nat {
   if y == 0 then 1 else x * expt(x, y-1)
}

// n!
function fact(n : nat) : nat {
  if n == 0 then 1 else n * fact(n-1)
}

// n! <= n^n for all natural numbers
lemma factLemma(n : nat)
  ensures fact(n) <= expt(n, n)
{
  if (n == 0) {            // Base case
    assert fact(0) <= expt(0, 0);
  } else {                // Inductive step
    factLemma(n-1);       // Inductive hypothesis
    exptLemma(n-1, n-1); // (n-1)^(n-1) <= n^(n-1)
    assert fact(n) == n * fact(n-1);                // by fact defn
    assert n * fact(n-1) <= n * expt(n-1, n-1);     // by IH
    assert n * expt(n-1, n-1) <= n * expt(n, n-1); // by exptLemma
    assert fact(n) <= expt(n, n);                   // qed.
  }
}

// x^y <= (x+1)^y for all natural numbers.
lemma exptLemma(x: nat, y: nat)
  ensures expt(x, y) <= expt(x + 1, y)
{}
```

Defining a recursive function with multiple base cases

A recursive function can have more than one base case.

- **Base cases** give the value of $f(0), \ldots, f(m)$ where $m \geq 0$.
- **Recursive case** defines $f(n+1)$ in terms of $f(n-m), \ldots, f(n-1), f(n)$ for all $n \geq m+1$.

Defining a recursive function with multiple base cases

A recursive function can have more than one base case.

Base cases give the value of $f(0), \ldots, f(m)$ where $m \geq 0$.

Recursive case defines $f(n+1)$ in terms of $f(n-m), \ldots, f(n-1), f(n)$ for all $n \geq m+1$.

Or it defines $f(n)$ in terms of $f(n-1-m), \ldots, f(n-1)$.

Defining a recursive function with multiple base cases

A recursive function can have more than one base case.

Base cases give the value of $f(0), \ldots, f(m)$ where $m \geq 0$.

Recursive case defines $f(n+1)$ in terms of $f(n-m), \ldots, f(n-1), f(n)$ for all $n \geq m+1$.

Or it defines $f(n)$ in terms of $f(n-1-m), \ldots, f(n-1)$.

Example: Fibonacci numbers

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

Defining a recursive function with multiple base cases

A recursive function can have more than one base case.

Base cases give the value of $f(0), \ldots, f(m)$ where $m \geq 0$.

Recursive case defines $f(n+1)$ in terms of $f(n-m), \ldots, f(n-1), f(n)$ for all $n \geq m+1$.

Or it defines $f(n)$ in terms of $f(n-1-m), \ldots, f(n-1)$.

Example: Fibonacci numbers

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

When the recursive function has multiple base cases, we use strong induction to prove its properties. And we also extend the strong induction proof template to account for the additional base cases.

Strong inductive proofs with base cases $b, \ldots, b+m$

① **Let $P(n)$ be** *[definition of $P(n)$]*.

We will show that $P(n)$ is true for every integer $n \geq b$ by strong induction.

② **Base cases ($n = b, \ldots, n = b + m$):**

[Proof of $P(b), \ldots, P(b+m)$.]

③ **Inductive hypothesis:**

Suppose that for some arbitrary integer $k \geq b + m$, $P(j)$ is true for every integer $b \leq j \leq k$.

④ **Inductive step:**

We want to prove that $P(k+1)$ is true.

[Proof of $P(k+1)$. The proof ***must*** *invoke the strong inductive hypothesis.]*

⑤ **The result follows for all $n \geq b$ by strong induction.**

Example: prove $f_n < 2^n$ for all $n \geq 0$

$$\begin{aligned} f_0 &= 0 \\ f_1 &= 1 \\ f_n &= f_{n-1} + f_{n-2} \\ &\quad \text{for all } n \geq 2 \end{aligned}$$

Example: prove $f_n < 2^n$ for all $n \geq 0$

① **Let $P(n)$ be $f_n < 2^n$ where f is the Fibonacci function.**

We will show that $P(n)$ is true for every integer $n \geq 0$ by strong induction.

$$\begin{aligned} f_0 &= 0 \\ f_1 &= 1 \\ f_n &= f_{n-1} + f_{n-2} \\ &\quad \text{for all } n \geq 2 \end{aligned}$$

Example: prove $f_n < 2^n$ for all $n \geq 0$

① **Let $P(n)$ be $f_n < 2^n$ where f is the Fibonacci function.**

We will show that $P(n)$ is true for every integer $n \geq 0$ by strong induction.

② **Base cases ($n = 0, n = 1$):**

$f_0 = 0 < 1 = 2^0$ so $P(0)$ is true.
$f_1 = 1 < 2 = 2^1$ so $P(1)$ is true.

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

Example: prove $f_n < 2^n$ for all $n \geq 0$

① **Let $P(n)$ be $f_n < 2^n$ where f is the Fibonacci function.**

We will show that $P(n)$ is true for every integer $n \geq 0$ by strong induction.

② **Base cases ($n = 0, n = 1$):**

$f_0 = 0 < 1 = 2^0$ so $P(0)$ is true.

$f_1 = 1 < 2 = 2^1$ so $P(1)$ is true.

③ **Inductive hypothesis:**

Suppose that for some arbitrary integer $k \geq 1$, $P(j)$ is true for every integer $0 \leq j \leq k$.

$$\begin{aligned} f_0 &= 0 \\ f_1 &= 1 \\ f_n &= f_{n-1} + f_{n-2} \\ &\quad \text{for all } n \geq 2 \end{aligned}$$

Example: prove $f_n < 2^n$ for all $n \geq 0$

① **Let $P(n)$ be $f_n < 2^n$ where f is the Fibonacci function.**

We will show that $P(n)$ is true for every integer $n \geq 0$ by strong induction.

② **Base cases ($n = 0, n = 1$):**

$f_0 = 0 < 1 = 2^0$ so $P(0)$ is true.
$f_1 = 1 < 2 = 2^1$ so $P(1)$ is true.

③ **Inductive hypothesis:**

Suppose that for some arbitrary integer $k \geq 1$, $P(j)$ is true for every integer $0 \leq j \leq k$.

④ **Inductive step:**

We want to prove that $P(k+1)$ is true, i.e., $f_{k+1} < 2^{k+1}$ for $k + 1 \geq 2$.

$$\begin{aligned} f_{k+1} &= f_k + f_{k-1} && \text{by definition of } f \\ &< 2^k + 2^{k-1} && \text{by the inductive hypothesis} \\ &< 2^k + 2^k = 2 \cdot 2^k = 2^{k+1} && \text{which is exactly } P(k+1). \end{aligned}$$

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

Example: prove $f_n < 2^n$ for all $n \geq 0$

① **Let $P(n)$ be $f_n < 2^n$ where f is the Fibonacci function.**

We will show that $P(n)$ is true for every integer $n \geq 0$ by strong induction.

② **Base cases ($n = 0, n = 1$):**

$f_0 = 0 < 1 = 2^0$ so $P(0)$ is true.
$f_1 = 1 < 2 = 2^1$ so $P(1)$ is true.

③ **Inductive hypothesis:**

Suppose that for some arbitrary integer $k \geq 1$, $P(j)$ is true for every integer $0 \leq j \leq k$.

④ **Inductive step:**

We want to prove that $P(k+1)$ is true, i.e., $f_{k+1} < 2^{k+1}$ for $k+1 \geq 2$.

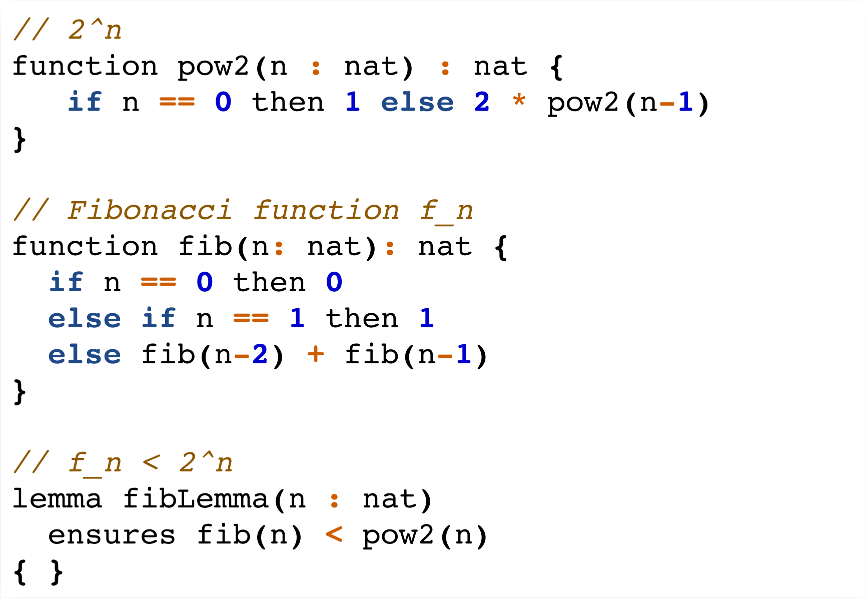
$$\begin{aligned} f_{k+1} &= f_k + f_{k-1} && \text{by definition of } f \\ &< 2^k + 2^{k-1} && \text{by the inductive hypothesis} \\ &< 2^k + 2^k = 2 \cdot 2^k = 2^{k+1} && \text{which is exactly } P(k+1). \end{aligned}$$

⑤ **The result follows for all $n \geq 0$ by induction.**

$$\begin{aligned} f_0 &= 0 \\ f_1 &= 1 \\ f_n &= f_{n-1} + f_{n-2} \\ &\quad \text{for all } n \geq 2 \end{aligned}$$

Fun: can we verify $f_n < 2^n$ for all $n \geq 0$?

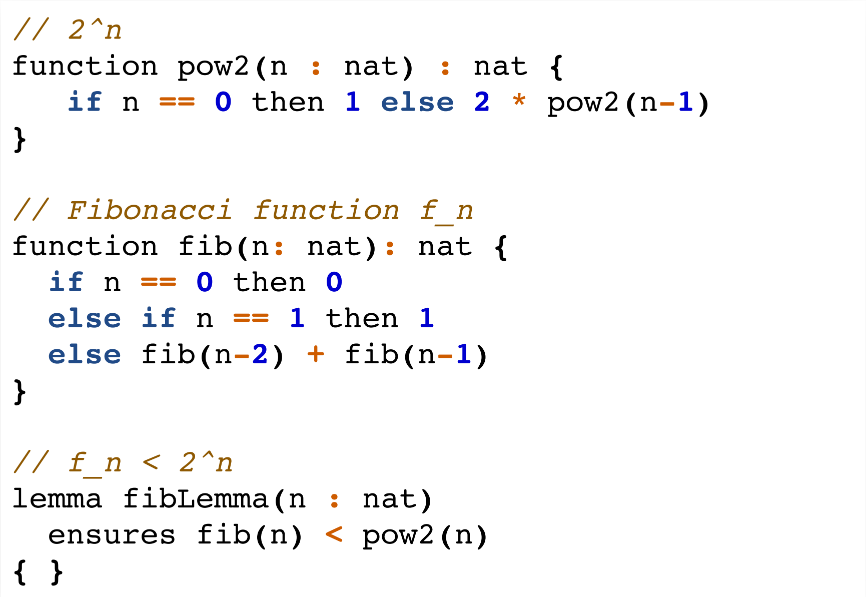
Prove $f_n < 2^n$ for $n \geq 0$ with Dafny:



```
// 2^n
function pow2(n : nat) : nat {
   if n == 0 then 1 else 2 * pow2(n-1)
}

// Fibonacci function f_n
function fib(n: nat): nat {
  if n == 0 then 0
  else if n == 1 then 1
  else fib(n-2) + fib(n-1)
}

// f_n < 2^n
lemma fibLemma(n : nat)
  ensures fib(n) < pow2(n)
{ }
```

Fun: can we verify $f_n < 2^n$ for all $n \geq 0$?

Prove $f_n < 2^n$ for $n \geq 0$ with Dafny:



```
// 2^n
function pow2(n : nat) : nat {
   if n == 0 then 1 else 2 * pow2(n-1)
}

// Fibonacci function f_n
function fib(n: nat): nat {
  if n == 0 then 0
  else if n == 1 then 1
  else fib(n-2) + fib(n-1)
}

// f_n < 2^n
lemma fibLemma(n : nat)
  ensures fib(n) < pow2(n)
{ }
```

Yes, Dafny can prove this theorem automatically!

Summary

Induction lets us prove statements about all natural numbers.

A proof by induction must show that $P(0)$ is true (*base case*).

And it must use the *inductive hypothesis* $P(k)$ to show that $P(k+1)$ is true (*inductive step*).

Induction also lets us prove theorems about integers $n \geq b$ for $b \in \mathbb{Z}$.

Adjust all parts of the proof to use $n \geq b$ instead of $n \geq 0$.

Strong induction lets us assume a stronger inductive hypothesis.

This makes some proofs easier.

But every proof by strong induction can be transformed into a proof by ordinary induction and vice versa.