INDUCTIVELY GENERATED DATA

Generative processes

- Virtually every infinite set considered in programming is ***generated by a process***.

  The fundamental example is the set $\mathbb{N}$ of natural numbers:

  - **Initial object** ("base"): The number 0 is in $\mathbb{N}$.

Generative processes

- Virtually every infinite set considered in programming is ***generated by a process***.

  The fundamental example is the set $\mathbb{N}$ of natural numbers:

  - **Initial object** (“base”): The number 0 is in $\mathbb{N}$.

    Generative step: If $n \in N$ then “next” of n, $\mathbf{s}\,n$, is $\in \mathbb{N}$

Generative processes

- Virtually every infinite set considered in programming is ***generated by a process***.

  The fundamental example is the set $\mathbb{N}$ of natural numbers:

  - **Initial object** ("base"): The number 0 is in $\mathbb{N}$.

    Generative step: If $n \in N$ then "next" of n, $\mathtt{s}\, n$, is $\in \mathbb{N}$

- Implicit assumptions:

  The meanings of 0 and "next" are known and given.

Generating $\{0,1\}^*$

- **Base.** The ***empty string*** is in $\{0,1\}^*$.
- **Generative step.**
  If $w \in \{0,1\}^*$ then $0\,w$ and $1\,w$ are $\in \{0,1\}^*$

Generating $\{0,1\}^*$

- **Base.** The ***empty string*** is in $\{0,1\}^*$.
- **Generative step.**
  If $w \in \{0,1\}^*$ then $0\,w$ and $1\,w$ are $\in \{0,1\}^*$

• Implicit assumptions:

The meanings of the empty string
and of juxtaposition are known.

Generating $\{0,1\}^*$

- **Base.** The ***empty string*** is in $\{0,1\}^*$.
- **Generative step.**
  If $w \in \{0,1\}^*$ then $0\,w$ and $1\,w$ are $\in \{0,1\}^*$

• Implicit assumptions:

  The meanings of the empty string
  and of juxtaposition are known.

• Note: We generate strings here "from the head";
  This conforms with the general use of constructors,
  and relflected in the functions *head* and *tail*.

Generating $\{0,1\}^*$

- **Base.** The ***empty string*** is in $\{0,1\}^*$.
- **Generative step.**
  If $w \in \{0,1\}^*$ then $0\,w$ and $1\,w$ are $\in \{0,1\}^*$

• Implicit assumptions:

The meanings of the empty string
and of juxtaposition are known.

• Note: We generate strings here "from the head";
This conforms with the general use of constructors,
and relflected in the functions *head* and *tail*.

• But numerals are in fact generated from the tail:

$$[\mathbf{7654321}]_{10} = 1 + 10 \cdot [\mathbf{765432}]_{10}$$

Format of generative definitions

- Two parts in a generative dfn of set S:
  - **Base:**
    Particular known objects are in S.

Format of generative definitions

- Two parts in a generative dfn of set S:
  - **Base:**
    Particular known objects are in S.
  - **Generative steps:**
    If certain objects are in S
    then so are vertain objects obtained from those.

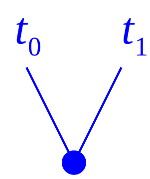
Another example: Binary trees

- ***Binary tree*** means here a *finite, ordered, unlabeled binary tree*

Base: The singleton tree • is in $\mathbf{BT}$.

Generative step:

If t_0, t_1 are binary trees then so is



Implicit assumptions:

We know what a singleton tree and juncture of trees mean.

Try this...

- Generate the set E of even natural numbers.

Try this...

- Generate the set E of even natural numbers.
  - Base: 0
  - Generative step: If $n \in E$ then $n{-}2 \in E$.

Boolean terms

- Two ways to define boolean terms (closed ones: no variables):

Boolean terms

- Two ways to define boolean terms (closed ones: no variables):
- **IBT**: Infix boolean terms:
  - `0` and `1` are in **IBT**
  - If $t, t' \in \mathbf{IBT}$ then $(t) \wedge (t') \in \mathbf{IBT}$ and $(t) \vee (t') \in \mathbf{IBT}$

Boolean terms

- Two ways to define boolean terms (closed ones: no variables):
- $\mathbf{IBT}$: Infix boolean terms:
  - `0` and `1` are in $\mathbf{IBT}$
  - If $t, t' \in \mathbf{IBT}$ then $(t) \wedge (t') \in \mathbf{IBT}$ and $(t) \vee (t') \in \mathbf{IBT}$
- $\mathbf{PBT}$: Prefix boolean terms:
  - `0` and `1` are in $\mathbf{PBT}$
  - If $t, t' \in \mathbf{PBT}$ then $\wedge\, t\, t' \in \mathbf{PBT}$ and $\vee\, t\, t' \in \mathbf{PBT}$

Boolean terms

- Two ways to define boolean terms (closed ones: no variables):
- **IBT**: Infix boolean terms:
  - `0` and `1` are in **IBT**
  - If $t, t' \in \mathbf{IBT}$ then $(t) \wedge (t') \in \mathbf{IBT}$ and $(t) \vee (t') \in \mathbf{IBT}$
- **PBT**: Prefix boolean terms:
  - `0` and `1` are in **PBT**
  - If $t, t' \in \mathbf{PBT}$ then $\wedge\, t\, t' \in \mathbf{PBT}$ and $\vee\, t\, t' \in \mathbf{PBT}$
- Main difference between IBT and PBT:
  No parentheses in PBT !

Lists of natural numbers

- Generate $\mathbb{L}(\mathbb{N})$ the ***lists of natural numbers***.
- Fix a textual coding of $\mathbb{N}$, say binary numerals.

Lists of natural numbers

- Generate $\mathbb{L}(\mathbb{N})$ the ***lists of natural numbers***.
- Fix a textual coding of $\mathbb{N}$, say binary numerals.
  - $\Box$ is a list of naturals.
  - If ℓ is a list and k a numeral then $k : \ell$ is a list.

Lists of natural numbers

- Generate $\mathbb{L}(\mathbb{N})$ the ***lists of natural numbers***.
- Fix a textual coding of $\mathbb{N}$, say binary numerals.
  - $\square$ is a list of naturals.
  - If ℓ is a list and k a numeral then $k : \ell$ is a list.
- Examples: $1 : \square$, $0 : 101 : 10011 : 10 : \square$.

REASONING ABOUT INDUCTIVE DATA

Infinite sets, finite minds

- $\mathbb{N}$ is infinite. $\{0,1\}^*$ is infinite.
  But our minds and our proofs are finite.

Infinite sets, finite minds

- $\mathbb{N}$ is infinite. $\{0, 1\}^*$ is infinite.

  But our minds and our proofs are finite.

- ***So how can we prove anything about*** $\mathbb{N}$ ***?***

Infinite sets, finite minds

- $\mathbb{N}$ is infinite. $\{0,1\}^*$ is infinite.

  But our minds and our proofs are finite.

- ***So how can we prove anything about $\mathbb{N}$?***

- Trying many cases is never sufficient.

  Example: *"For all n at least one of $2^n + 1$ and $2^n - 1$ is prime."*

  $2, 3, 5, 7, 17, 31$ Hooray!

Infinite sets, finite minds

- $\mathbb{N}$ is infinite. $\{0,1\}^*$ is infinite.
  But our minds and our proofs are finite.
- ***So how can we prove anything about $\mathbb{N}$?***
- Trying many cases is never sufficient.
  Example: *"For all n at least one of 2^n+1 and 2^n-1 is prime."*
  $2, 3, 5, 7, 17, 31$ Hooray!
  Oops: Both 63 and 65 are composite.

Infinite sets, finite minds

- $\mathbb{N}$ is infinite. $\{0,1\}^*$ is infinite.

  But our minds and our proofs are finite.

- ***So how can we prove anything about $\mathbb{N}$?***

- Trying many cases is never sufficient.

  Example: *"For all n at least one of 2^n+1 and 2^n-1 is prime."*

  $2, 3, 5, 7, 17, 31$ Hooray!

  Oops: Both 63 and 65 are composite.

- Another try (Fermat): *All numbers $2^{2^n}+1$ are prime*

  $3, 5, 17, 257, 65537$. Yahoo!

Infinite sets, finite minds

- $\mathbb{N}$ is infinite. $\{0,1\}^*$ is infinite.

  But our minds and our proofs are finite.

- ***So how can we prove anything about $\mathbb{N}$?***

- Trying many cases is never sufficient.

  Example: *"For all n at least one of 2^n+1 and 2^n-1 is prime."*

  $2, 3, 5, 7, 17, 31$ Hooray!

  Oops: Both 63 and 65 are composite.

- Another try (Fermat): *All numbers $2^{2^n}+1$ are prime*

  $3, 5, 17, 257, 65537$. Yahoo!

  Oops (Euler): Next one is 4,294,967,297, which is divisible by 641.

Infinite sets, finite minds

- $\mathbb{N}$ is infinite. $\{0,1\}^*$ is infinite.

  But our minds and our proofs are finite.

- ***So how can we prove anything about $\mathbb{N}$?***

- Trying many cases is never sufficient.

  Example: *"For all n at least one of 2^n+1 and 2^n-1 is prime."*

  $2, 3, 5, 7, 17, 31$ Hooray!

  Oops: Both 63 and 65 are composite.

- Another try (Fermat): *All numbers $2^{2^n}+1$ are prime*

  $3, 5, 17, 257, 65537$. Yahoo!

  Oops (Euler): Next one is 4,294,967,297, which is divisible by 641.

- ***So how can we hope to prove***

  that all natural numbers are such-and-such ?

Finitely generated infinities!

- The secret is that inductive data is generated by ***finite rules.***
- Therefore we have a finite tool for proving that all generated objects satisfy certain properties.

Following the process

- Suppose we generate $\mathbb{N}$ using a green pen.

Following the process

- Suppose we generate $\mathbb{N}$ using a green pen.
  - 0 is a green natural.
  - If x is a green natural, then so is its successor.

Following the process

- Suppose we generate $\mathbb{N}$ using a green pen.
  - 0 is a green natural.
  - If x is a green natural, then so is its successor.

0

Following the process

- Suppose we generate $\mathbb{N}$ using a green pen.
  - 0 is a green natural.
  - If x is a green natural, then so is its successor.

0 1

Following the process

- Suppose we generate $\mathbb{N}$ using a green pen.
  - 0 is a green natural.
  - If x is a green natural, then so is its successor.

0 1 2

Following the process

- Suppose we generate $\mathbb{N}$ using a green pen.
  - 0 is a green natural.
  - If x is a green natural, then so is its successor.

0 1 2 3

Following the process

- Suppose we generate $\mathbb{N}$ using a green pen.
  - 0 is a green natural.
  - If x is a green natural, then so is its successor.

0 1 2 3 4

Following the process

- Suppose we generate $\mathbb{N}$ using a green pen.
  - 0 is a green natural.
  - If x is a green natural, then so is its successor.

0 1 2 3 4

- They all come out green:
  As we generate $\mathbb{N}$ we make sure that we start with green,
  and that each step maintains green-ness.

Following the process

- Suppose we generate $\mathbb{N}$ using a green pen.
  - 0 is a green natural.
  - If x is a green natural, then so is its successor.

0 1 2 3 4

- They all come out green:
  As we generate $\mathbb{N}$ we make sure that we start with green,
  and that each step maintains green-ness.
- Green-ness is here the process' ***invariant:***
  True at the outset, and preserved by the steps.

The principle of induction for $\mathbb{N}$

- Suppose $P(x)$ is a property of natural numbers x.
  $P(x)$ abbreviates here *"x has the property P"*

The principle of induction for $\mathbb{N}$

- Suppose $P(x)$ is a property of natural numbers x.
  $P(x)$ abbreviates here "x *has the property* P"
- Assume:
  - ***Base.*** $P(0)$ and
  - ***Step.*** For all $n \in \mathbb{N}$, $P(n)$ implies $P(n+1)$.

The principle of induction for $\mathbb{N}$

- Suppose $P(x)$ is a property of natural numbers x.
  $P(x)$ abbreviates here "*x has the property P*"
- Assume:
  - ***Base.*** $P(0)$ and
  - ***Step.*** For all $n \in \mathbb{N}$, $P(n)$ implies $P(n+1)$.
- Conclude: $P(x)$ for all $x \in \mathbb{N}$.

The principle of induction for $\mathbb{N}$

- Suppose $P(x)$ is a property of natural numbers x.
  $P(x)$ abbreviates here "x *has the property* P"
- Assume:
  - ***Base.*** $P(0)$ and
  - ***Step.*** For all $n \in \mathbb{N}$, $P(n)$ implies $P(n+1)$.
- Conclude: $P(x)$ for all $x \in \mathbb{N}$.
- As natural numbers are being generated,
  they all come out satisfying P.

The principle of induction for $\mathbb{N}$

- Suppose $P(x)$ is a property of natural numbers x.
  $P(x)$ abbreviates here "x *has the property* P"
- Assume:
  - ***Base.*** $P(0)$ and
  - ***Step.*** For all $n \in \mathbb{N}$, $P(n)$ implies $P(n+1)$.
- Conclude: $P(x)$ for all $x \in \mathbb{N}$.
- As natural numbers are being generated,
  they all come out satisfying P.
- *A property of natural numbers that holds for zero*
  and is invariant under successor
  is true of every natural number.

The principle of induction for $\mathbb{N}$

- Suppose $P(x)$ is a property of natural numbers x.
  $P(x)$ abbreviates here *“x has the property P”*
- Assume:
  - ***Base.*** $P(0)$ and
  - ***Step.*** For all $n \in \mathbb{N}$, $P(n)$ implies $P(n+1)$.
- Conclude: $P(x)$ for all $x \in \mathbb{N}$.
- As natural numbers are being generated,
  they all come out satisfying P.
- *A property of natural numbers that holds for zero*
  and is invariant under successor
  is true of every natural number.
- The premise of the STEP is often called the “induction assumption”
  or the Induction Hypothesis (IH).

Example

- Show that $2^x < 2^{x+1}$ for all $x \in \mathbb{N}$. What is the property?

Example

- Show that $2^x < 2^{x+1}$ for all $x \in \mathbb{N}$.
- $P(x)$ is $2^x < 2^{x+1}$

Example

- Show that $2^x < 2^{x+1}$ for all $x \in \mathbb{N}$.
- If we know that
  - $2^x < 2^{x+1}$ is true for $x = 0$; and
  - $2^x < 2^{x+1}$ for $x = n$
    implies that $2^x < 2^{x+1}$ for $x = n + 1$
- then $2^x < 2^{x+1}$ for all $x \in \mathbb{N}$.

Example

- Show that $2^x < 2^{x+1}$ for all $x \in \mathbb{N}$.
- If we know that
  - $2^x < 2^{x+1}$ is true for $x = 0$; and
  - $2^x < 2^{x+1}$ for $x = n$
    implies that $2^x < 2^{x+1}$ for $x = n + 1$
- then $2^x < 2^{x+1}$ for all $x \in \mathbb{N}$.
- But we do have
  - Base: $2^0 = 1 < 2 = 2^{0+1}$

Example

- Show that $2^x < 2^{x+1}$ for all $x \in \mathbb{N}$.
- If we know that
  - $2^x < 2^{x+1}$ is true for $x = 0$; and
  - $2^x < 2^{x+1}$ for $x = n$
    implies that $2^x < 2^{x+1}$ for $x = n + 1$
- then $2^x < 2^{x+1}$ for all $x \in \mathbb{N}$.
- But we do have
  - Base: $2^0 = 1 < 2 = 2^{0+1}$
  - Step: If $2^n < 2^{n+1}$ ($P(x)$ for $x = n$) then
  $2^{n+1} = 2^n + 2^n < 2^{n+1} + 2^{n+1} = 2^{n+2}$ nextt ($P(x)$ for $x = n+1$)

Example

- Show that $2^x < 2^{x+1}$ for all $x \in \mathbb{N}$.
- If we know that
  - $2^x < 2^{x+1}$ is true for $x = 0$; and
  - $2^x < 2^{x+1}$ for $x = n$
    implies that $2^x < 2^{x+1}$ for $x = n + 1$
- then $2^x < 2^{x+1}$ for all $x \in \mathbb{N}$.
- But we do have
  - Base: $2^0 = 1 < 2 = 2^{0+1}$
  - Step: If $2^n < 2^{n+1}$ ($P(x)$ for $x = n$) then
  $2^{n+1} = 2^n + 2^n < 2^{n+1} + 2^{n+1} = 2^{n+2}$ nextt ($P(x)$ for $x = n{+}1$)
- By Induction, $2^x < 2^{x+1}$ for all $x \in \mathbb{N}$.

Try this...

- Prove by induction on $\mathbb{N}$ that $x \leqslant 2^x$ for all $x \in \mathbb{N}$.
  We are given that exponentiation is an increasing function.
- By Induction $x \leqslant 2^x$ for all $x \in \mathbb{N}$.

Try this...

- Prove by induction on $\mathbb{N}$ that $x \leqslant 2^x$ for all $x \in \mathbb{N}$.
  We are given that exponentiation is an increasing function.
  - Base: For $x = 0$ we have $x^2 = 0 < 1 = 2^x$.
  - Step: Assume $n \leqslant 2^n$. Then

$$
\begin{aligned}
n + 1 &\leqslant 2^n + 1 && \text{(IH)} \\
&= 2^n + 2^0 \\
&\leqslant 2^n + 2^n && \text{(exponentiation is increasing)} \\
&= 2^{n+1}
\end{aligned}
$$

- By Induction $x \leqslant 2^x$ for all $x \in \mathbb{N}$.

Example: Divisibility

- $P(x): \quad x^3 + 2x$ is divisible by 3.
- By Induction:

Example: Divisibility

- $P(x): \quad x^3 + 2x$ is divisible by 3.
- By Induction:
  - **Base.** For $x = 0$
    $x^3 + 2x = 0^3 + 2 \cdot 0 = 0$ which is divisible by 3.

Example: Divisibility

- $P(x): \quad x^3 + 2x$ is divisible by 3.
- By Induction:
  - **Base.** For $x = 0$
    $x^3 + 2x = 0^3 + 2 \cdot 0 = 0$ which is divisible by 3.
  - **Step.** Assume $P(n)$ (IH). Then for $x = n+1$

$$\begin{aligned} x^3 + 2x &= (n+1)^3 + (2n+2) \\ &= (n^3 + 3n^2 + 3n + 1) + (2n+2) \\ &= (n^3 + 2n) + 3(n^2 + n + 1) \end{aligned}$$

$x^3 + 2x$ is the sum of numbers divisible by 3,
and is therefore divisible by 3.

Example: Divisibility

- $P(x): \quad x^3 + 2x$ is divisible by 3.
- By Induction:
  - **Base.** For $x = 0$
    $x^3 + 2x = 0^3 + 2 \cdot 0 = 0$ which is divisible by 3.
  - **Step.** Assume $P(n)$ (IH). Then for $x = n{+}1$

$$
\begin{aligned}
x^3 + 2x &= (n+1)^3 + (2n+2) \\
&= (n^3 + 3n^2 + 3n + 1) + (2n+2) \\
&= (n^3 + 2n) + 3(n^2 + n + 1)
\end{aligned}
$$

  $x^3 + 2x$ is the sum of numbers divisible by 3,
  and is therefore divisible by 3.

- By Induction $x^3 + 2x$ is divisible by 3, for all $x \in \mathbb{N}$.

Example: Divisibility

- $P(x):\quad x^3+2x$ is divisible by 3.
- By Induction:
  - **Base.** For $x=0$
    $x^3+2x = 0^3+2\cdot 0 = 0$ which is divisible by 3.
  - **Step.** Assume $P(n)$ (IH). Then for $x=n+1$

$$\begin{aligned} x^3+2x &= (n+1)^3+(2n+2) \\ &= (n^3+3n^2+3n+1)+(2n+2) \\ &= (n^3+2n)+3(n^2+n+1) \end{aligned}$$

x^3+2x is the sum of numbers divisible by 3,
and is therefore divisible by 3.

- By Induction x^3+2x is divisible by 3, for all $x\in\mathbb{N}$.

Iterated summation

$$(\star) \qquad 0 + 1 + 2 + \cdots + x \ = \ x(x+1)/2$$

- By Induction :

Iterated summation

$$(\star) \qquad 0 + 1 + 2 + \cdots + x \;=\; x(x+1)/2$$

- By Induction :
  - **Base.** $(\star)$ is true for $x = 0$: $\quad 0 = 0 \cdot (0+1)/2$.

Iterated summation

$$(\star) \qquad 0 + 1 + 2 + \cdots + x \;=\; x(x+1)/2$$

- By Induction :
  - **Base.** $(\star)$ is true for $x = 0$: $0 = 0 \cdot (0+1)/2$.
  - **Step.** Assume $(\star)$ for $x = n$. Then for $x = n+1$

$$\begin{aligned}
0 + 1 + \cdots + x &= 0 + 1 + \cdots + n + (n+1) \\
&= \tfrac{n(n+1)}{2} + (n+1) \qquad \text{(IH)} \\
&= (n+1)(\tfrac{1}{2}n + 1) \\
&= \tfrac{1}{2}(n+1)(n+2) \\
&= \tfrac{1}{2}x(x+1)
\end{aligned}$$

That is, $(\star)$ for $x = n+1$.

Iterated summation

$$(\star) \qquad 0+1+2+\cdots+x \;=\; x(x+1)/2$$

- By Induction :
  - **Base.** $(\star)$ is true for $x=0$: $0 = 0\cdot(0+1)/2$.
  - **Step.** Assume $(\star)$ for $x=n$. Then for $x=n+1$

$$\begin{aligned} 0+1+\cdots+x &= 0+1+\cdots+n+(n+1) \\ &= \tfrac{n(n+1)}{2}+(n+1) \qquad \text{(IH)} \\ &= (n+1)(\tfrac{1}{2}n+1) \\ &= \tfrac{1}{2}(n+1)(n+2) \\ &= \tfrac{1}{2}x(x+1) \end{aligned}$$

  That is, $(\star)$ for $x=n+1$.

- Conclude: $(\star)$ holds for every $x\in\mathbb{N}$.

Involving other data

- A property of natural numbers may refer to non-numeric data!

  ($\star$) *Every set with* x *elements has* 2^x *subsets*

- By Induction.

Involving other data

- A property of natural numbers may refer to non-numeric data!

  ($\star$) *Every set with x elements has 2^x subsets*

- By Induction.

  - **Base.** $x = 0$. The only set with 0 elements is $\emptyset$, which has just $2^0 = 1$ subset, namely $\emptyset$ itself.

Involving other data

- A property of natural numbers may refer to non-numeric data!

  ($\star$) *Every set with x elements has 2^x subsets*

- By Induction.

  - **Base.** $x = 0$. The only set with 0 elements is $\emptyset$, which has just $2^0 = 1$ subset, namely $\emptyset$ itself.
  - **Step.** Assume $P(n)$ (IH).

  For $x = n+1$ let S be a set with $n+1$ elements.
  Choose $a \in S$ (S can't be empty!) and let $S^- =_{\mathrm{df}} S - \{a\}$.

Involving other data

- A property of natural numbers may refer to non-numeric data!

  $(\star)$ *Every set with* x *elements has* 2^x *subsets*

- By Induction.

  - **Base.** $x = 0$. The only set with 0 elements is $\emptyset$, which has just $2^0 = 1$ subset, namely $\emptyset$ itself.

  - **Step.** Assume $P(n)$ (IH).

  For $x = n+1$ let S be a set with $n+1$ elements.
  Choose $a \in S$ (S can't be empty!) and let $S^- =_{\mathrm{df}} S - \{a\}$.
  By IH S^- has 2^n subsets $A_1, \ldots, A_{2^n}$.
  Subsets of S: $A_1, \ldots, A_{2^n}$, $A_1 \cup \{a\}, \ \ldots, \ A_{2^n} \cup \{a\}$
  which are all different. So S has $2^n + 2^n = 2^{n+1}$ subsets.

Involving other data

- A property of natural numbers may refer to non-numeric data!

  ($\star$) *Every set with x elements has 2^x subsets*

- By Induction.
  - **Base.** $x = 0$. The only set with 0 elements is $\emptyset$, which has just $2^0 = 1$ subset, namely $\emptyset$ itself.
  - **Step.** Assume $P(n)$ (IH).

    For $x = n{+}1$ let S be a set with $n{+}1$ elements.

    Choose $a \in S$ (S can't be empty!) and let $S^- =_{\rm df} S - \{a\}$.

    By IH S^- has 2^n subsets $A_1, \ldots, A_{2^n}$.

    Subsets of S: $A_1, \ldots, A_{2^n}$, $A_1 \cup \{a\}, \ \ldots, \ A_{2^n} \cup \{a\}$

    which are all different. So S has $2^n + 2^n = 2^{n+1}$ subsets.

- By Induction ($\star$) for all $x \in \mathbb{N}$.

Starting Induction elsewhere

- ***Show $x^2 > x$ for all $x > 1$.***
- We wish to start induction from 2.

Starting Induction elsewhere

- ***Show*** $x^2 > x$ ***for all*** $x > 1$**.**
- We wish to start induction from 2.

  But that's the same as Induction for the property

$$(x + 2)^2 > (x + 2) \quad !$$

Starting Induction elsewhere

- ***Show*** $x^2 > x$ ***for all*** $x > 1$.
- We wish to start induction from 2.

  But that's the same as Induction for the property

$$(x+2)^2 > (x+2) \quad !$$

- We refer to this as Shifted Induction:
  - **Base.** $2^2 = 4 > 2$
  - **Step.** $n^2 > n$ implies

$$\begin{aligned} (n+1)^2 &= n^2 + 2n + 1 \\ &> n + 2n + 1 \quad \text{(IH)} \\ &> n + 1 \quad \text{since } n > 0) \end{aligned}$$

Starting Induction elsewhere

- ***Show*** $x^2 > x$ ***for all*** $x > 1$.
- We wish to start induction from 2.

  But that's the same as Induction for the property

$$(x+2)^2 > (x+2) \quad !$$

- We refer to this as Shifted Induction:
  - **Base.** $2^2 = 4 > 2$
  - **Step.** $n^2 > n$ implies

$$\begin{aligned} (n+1)^2 &= n^2 + 2n + 1 \\ &> n + 2n + 1 \quad \text{(IH)} \\ &> n + 1 \quad \text{since } n > 0) \end{aligned}$$

- Conclusion: $x^2 > x$ for all integers $x > 1$.

Shifted Induction

- The template for such reasoning is ***Shifted Induction***
- Given a property $P(x)$ of natural numbers, and $b \in \mathbb{N}$,

- Assume:
  - **Shifted Base.** P true of b; and
  - **Shifted Step.** For all $n \geqslant b$, $P(n)$ implies $P(n+1)$
- Conclude: $P(x)$ for all $x \geqslant b$.
- Induction is a special case, with $b = 0$.

Another example

- $3^n > 5 \cdot 2^n$ for all $n \geqslant 4$.
- By Shifted Induction with initial value 4.

Another example

- $3^n > 5 \cdot 2^n$ for all $n \geqslant 4$.
- By Shifted Induction with initial value 4.
  - Basis. $3^4 = 81 > 80 = 5 \cdot 2^4$

Another example

- $3^n > 5 \cdot 2^n$ for all $n \geqslant 4$.
- By Shifted Induction with initial value 4.
  - Basis. $3^4 = 81 > 80 = 5 \cdot 2^4$
  - Step. If $3^n > 5 \cdot 2^n$ then

$$\begin{aligned} 3^{n+1} &= 3 \cdot 3^n \\ &> 3 \cdot (5 \cdot 2^n) \quad \text{(IH)} \\ &> 2 \cdot 5 \cdot 2^n \\ &= 5 \cdot 2^{n+1} \end{aligned}$$

Inductive reasoning in general

- The principle of inductive reasoning applies
  to any inductively generated set S, not just $\mathbb{N}$.
- If $P(x)$ makes sense for $x \in S$,
  is true for every base element of S
  and remains true under the generative steps for S,
  then $P(x)$ is true for all $x \in S$.

Inductive reasoning in general

- The principle of inductive reasoning applies
  to any inductively generated set S, not just $\mathbb{N}$.
- If $P(x)$ makes sense for $x \in S$,
  is true for every base element of S
  and remains true under the generative steps for S,
  then $P(x)$ is true for all $x \in S$.
- The underlying reason is the same as for $\mathbb{N}$:
  as the elements of S are generated,
  the property P invariantly holds.

Involving non-numeric data

- $P(x)$ may mention also non-numeric data.
- Example: Take "Every set with x elements has 2^x subsets"

Involving non-numeric data

- $P(x)$ may mention also non-numeric data.
- Example: Take "Every set with x elements has 2^x subsets"
- We prove by Induction on $\mathbb{N}$ that $P(x)$ for all $x \in \mathbb{N}$.

Involving non-numeric data

- $P(x)$ may mention also non-numeric data.
- Example: Take "Every set with x elements has 2^x subsets"
- We prove by Induction on $\mathbb{N}$ that $P(x)$ for all $x \in \mathbb{N}$.
  - **Base.** $P(0)$: "Every set with 0 elements has $2^0 = 1$ subsets". Indeed $\emptyset$ has one subset.

Involving non-numeric data

- $P(x)$ may mention also non-numeric data.
- Example: Take "Every set with x elements has 2^x subsets"
- We prove by Induction on $\mathbb{N}$ that $P(x)$ for all $x \in \mathbb{N}$.
  - **Base.** $P(0)$: "Every set with 0 elements has $2^0 = 1$ subsets". Indeed $\emptyset$ has one subset.
  - **Step.** Assume $P(n)$.
    To prove $P(n+1)$ let S be a set with $n+1$ elements.
    Choose $a \in S$ and let $S^- = S - \{a\}$.
    By IH S^- has 2^n subsets $A_1, \ldots, A_{2^n}$.
    The subsets of S are $A_1, \ldots, A_{2^n}, A_1 \cup \{a\}, \ldots, A_{2^n} \cup \{a\}$, which are all different.
    So S has $2^n + 2^n = 2^{n+1}$ subsets.

Involving non-numeric data

- $P(x)$ may mention also non-numeric data.
- Example: Take "Every set with x elements has 2^x subsets"
- We prove by Induction on $\mathbb{N}$ that $P(x)$ for all $x \in \mathbb{N}$.
  - **Base.** $P(0)$: "Every set with 0 elements has $2^0 = 1$ subsets". Indeed $\emptyset$ has one subset.
  - **Step.** Assume $P(n)$.
    To prove $P(n+1)$ let S be a set with $n+1$ elements.
    Choose $a \in S$ and let $S^- = S - \{a\}$.
    By IH S^- has 2^n subsets $A_1, \ldots, A_{2^n}$.
    The subsets of S are $A_1, \ldots, A_{2^n}, A_1 \cup \{a\}, \ldots, A_{2^n} \cup \{a\}$, which are all different.
    So S has $2^n + 2^n = 2^{n+1}$ subsets.
  - By Induction on $\mathbb{N}$ $P(x)$ holds for all $x \in \mathbb{N}$.

Induction on strings

- Let $P(x)$ be a property of Σ-strings.

Induction on strings

- Let $P(x)$ be a property of Σ-strings.
- ***Assume:***
  - **Base.** $P(\varepsilon)$
  - **Steps.** For each $\sigma \in \Sigma$ and $w \in \Sigma^*$
    $P(w)$ implies $P(\sigma w)$

Induction on strings

- Let $P(x)$ be a property of Σ-strings.
- ***Assume:***
  - **Base.** $P(\varepsilon)$
  - **Steps.** For each $\sigma \in \Sigma$ and $w \in \Sigma^*$
    $P(w)$ implies $P(\sigma w)$
- ***Conclude:*** $P(w)$ for all $w \in \Sigma^*$.

Example: Swapping

- For $w \in \{0,1\}^*$ let $\backsim(w)$ ("swap w") be w with 0 and 1 interchanged: $\backsim 001 = 110$.

  We show (⋆) $\backsim(\backsim(w)) = w$

Example: Swapping

- For $w \in \{\mathtt{0}, \mathtt{1}\}^*$ let $\backsim(w)$ ("swap w") be w with $\mathtt{0}$ and $\mathtt{1}$ interchanged: $\backsim\mathtt{001} = \mathtt{110}$.

  We show (⋆) $\backsim(\backsim(w)) = w$

- The proof is by induction on $\{\mathtt{0}, \mathtt{1}\}^*$.

Example: Swapping

- For $w \in \{0,1\}^*$ let $\backsim(w)$ ("swap w") be w with 0 and 1 interchanged: $\backsim 001 = 110$.

  We show (⋆) $\backsim(\backsim(w)) = w$

- The proof is by induction on $\{0,1\}^*$.

  - **Basis.** $\backsim(\backsim(\varepsilon)) = \backsim(\varepsilon) = \varepsilon$

Example: Swapping

- For $w \in \{0,1\}^*$ let $\backsim(w)$ ("swap w") be w with 0 and 1 interchanged: $\backsim 001 = 110$.

  We show (⋆) $\backsim(\backsim(w)) = w$

- The proof is by induction on $\{0,1\}^*$.

  - **Basis.** $\backsim(\backsim(\varepsilon)) = \backsim(\varepsilon) = \varepsilon$
  - **Step for 0.** If $\backsim(\backsim(x)) = x$

  then
  $$\begin{aligned} \backsim(\backsim(0x)) &= \backsim(1\backsim(x)) \\ &= 0\backsim(\backsim(x)) \\ &= 0x \qquad \text{(IH)} \end{aligned}$$

  Step for 1 is similar.

Example: Swapping

- For $w \in \{0,1\}^*$ let $\backsim(w)$ ("swap w") be w with 0 and 1 interchanged: $\backsim 001 = 110$.

  We show $(\star)$ $\backsim(\backsim(w)) = w$

- The proof is by induction on $\{0,1\}^*$.

  - **Basis.** $\backsim(\backsim(\varepsilon)) = \backsim(\varepsilon) = \varepsilon$
  - **Step for 0.** If $\backsim(\backsim(x)) = x$

  then
  $$\begin{aligned} \backsim(\backsim(0x)) &= \backsim(1\backsim(x)) \\ &= 0\backsim(\backsim(x)) \\ &= 0x \qquad \text{(IH)} \end{aligned}$$

  Step for 1 is similar.

- By induction on $\{0,1\}^*$ $(\star)$ for all $w \in \{0,1\}^*$.

Dealing with several inputs

- Prove $|x \cdot u| = |x| + |u|$ ($x, u \in \Sigma^*$).

Dealing with several inputs

- Prove $|x \cdot u| = |x| + |u|$ ($x, u \in \Sigma^*$).
- ***Problem: This is a property of a pair of strings!***

Dealing with several inputs

- Prove $|x \cdot u| = |x| + |u|$ ($x, u \in \Sigma^*$).
- Solution: Read it as a property of one x:

$$|x \cdot u| = |x| + |u| \quad \text{for all } u \in \Sigma^* \qquad (\star)$$

Dealing with several inputs

- Prove $|x \cdot u| = |x| + |u|$ ($x, u \in \Sigma^*$).
- Solution: Read it as a property of one x:

$|x \cdot u| = |x| + |u|$ for all $u \in \Sigma^*$ $(\star)$

- **Basis:** $x = \varepsilon$.

$|\varepsilon \cdot u| = |u|$ since $\varepsilon \cdot u = u$

$|\varepsilon| + |u| = 0 + |u| = |u|$

Dealing with several inputs

- Prove $|x \cdot u| = |x| + |u|$ ($x, u \in \Sigma^*$).
- Solution: Read it as a property of one x:

$$|x \cdot u| = |x| + |u| \quad \text{for all } u \in \Sigma^* \tag{$\star$}$$

- **Basis:** $x = \varepsilon$.

$$\begin{aligned} |\varepsilon \cdot u| &= |u| && \text{since } \varepsilon \cdot u = u \\ |\varepsilon| + |u| &= 0 + |u| = |u| \end{aligned}$$

- **Step:** Assume $(\star)$ for $x = w$.

  For $x = \sigma w$ we have for all $u \in \Sigma^*$

$$\begin{aligned} |\sigma w \cdot u| &= |\sigma(w \cdot u)| \\ &= 1 + |w \cdot u| \\ &= 1 + |w| + |u| \quad \text{(IH)} \\ &= (|\sigma w|) + |u| \end{aligned}$$

Dealing with several inputs

- Prove $|x \cdot u| = |x| + |u|$ ($x, u \in \Sigma^*$).
- Solution: Read it as a property of one x:

$|x \cdot u| = |x| + |u|$ for all $u \in \Sigma^*$ $(\star)$

  - **Basis:** $x = \varepsilon$.

    $|\varepsilon \cdot u| = |u|$ since $\varepsilon \cdot u = u$

    $|\varepsilon| + |u| = 0 + |u| = |u|$

  - **Step:** Assume $(\star)$ for $x = w$.

    For $x = \sigma w$ we have for all $u \in \Sigma^*$ $|\sigma w \cdot u| = |\sigma w| + |u|$

- By induction on Σ^* conclude $(\star)$ for all $x \in \Sigma^*$.

Unambiguous PBT's

- A PBT t is **unambiguous** if it is exactly one of:
  - `0` or `1`
  - $\wedge\, t_0\, t_1$ or $\vee\, t_0\, t_1$ for some unique terms t_0 and t_1 .

Unambiguous PBT's

- A PBT t is **unambiguous** if it is exactly one of:
  - $\mathtt{0}$ or $\mathtt{1}$
  - $\wedge\, t_0\, t_1$ or $\vee\, t_0\, t_1$ for some unique terms t_0 and t_1 .
- That is: t can't be both $\wedge\, t_0\, t_1$ and $\wedge\, t'_0\, t'_1$ unless $t'_0 = t_0$ and $t'_1 = t_1$.

Unambiguous PBT's

- A PBT t is **unambiguous** if it is exactly one of:
  - $\mathtt{0}$ or $\mathtt{1}$
  - $\wedge\, t_0\, t_1$ or $\vee\, t_0\, t_1$ for some unique terms t_0 and t_1 .
- That is: t can't be both $\wedge\, t_0\, t_1$ and $\wedge\, t'_0\, t'_1$ unless $t'_0 = t_0$ and $t'_1 = t_1$.
- Theorem: *Every PBT is unambiguous*

Unambiguous PBT's

- A PBT t is **unambiguous** if it is exactly one of:
  - $\mathbf{0}$ or $\mathbf{1}$
  - $\wedge t_0 t_1$ or $\vee t_0 t_1$ for some unique terms t_0 and t_1 .
- That is: t can't be both $\wedge t_0 t_1$ and $\wedge t'_0 t'_1$ unless $t'_0 = t_0$ and $t'_1 = t_1$.
- Theorem: *Every PBT is unambiguous*
- ***How to prove this?***

Unambiguous PBT's

- A PBT t is **unambiguous** if it is exactly one of:
  - `0` or `1`
  - $\wedge\, t_0\, t_1$ or $\vee\, t_0\, t_1$ for some unique terms t_0 and t_1 .
- That is: t can't be both $\wedge\, t_0\, t_1$ and $\wedge\, t_0'\, t_1'$ unless $t_0' = t_0$ and $t_1' = t_1$.
- Theorem: *Every PBT is unambiguous*
- ***How to prove this?***
- Induction on terms does not work:
  If $t = \wedge\, t_0\, t_1$ what can we possibly conclude from assuming that t_0 and t_1 are unambiguous?

Pushing induction through

- Maybe induction on strings over $\Sigma = \{\mathtt{0}, \mathtt{1}, \wedge, \vee\}$

Pushing induction through

- Maybe induction on strings over $\Sigma = \{\mathtt{0}, \mathtt{1}, \wedge, \vee\}$
- But we can't conclude that $\wedge t_1 t_2$ is unambiguous from the assumption that t_1, t_2 are unambiguous!

Pushing induction through

- Maybe induction on strings over $\Sigma = \{\mathtt{0}, \mathtt{1}, \wedge, \vee\}$
- But we can't conclude that $\wedge t_1 t_2$ is unambiguous from the assumption that t_1, t_2 are unambiguous!
- And the string $t_1 t_2$ is not a term!

Pushing induction through

- Maybe induction on strings over $\Sigma = \{\mathtt{0}, \mathtt{1}, \wedge, \vee\}$
- But we can't conclude that $\wedge t_1 t_2$ is unambiguous from the assumption that t_1, t_2 are unambiguous!
- And the string $t_1 t_2$ is not a term!
- Solution: A broader notion of "non-ambiguity"!

  $w \in \Sigma^*$ is ***unambiguous*** if it can'e be read as a ***concatenation*** of terms in more than one way:

Pushing induction through

- Maybe induction on strings over $\Sigma = \{\texttt{0}, \texttt{1}, \wedge, \vee\}$
- But we can't conclude that $\wedge t_1 t_2$ is unambiguous
  from the assumption that t_1, t_2 are unambiguous!
- And the string $t_1 t_2$ is not a term!
- Solution: A broader notion of "non-ambiguity"!
  $w \in \Sigma^*$ is ***unambiguous*** if it can'e be read
  as a ***concatenation*** of terms in more than one way:
- If $w = t_1 \cdots t_k = t'_1 \cdots t'_m$
  then $m = k$ and $t_i = t'_i$ for $i \in [1..k]$.

A generalized non-ambiguity theorem

- **Non-ambiguity Theorem.** *Every* $w \in \Sigma^*$ *is unambiguous.*
- The proof is by induction on Σ^*.

A generalized non-ambiguity theorem

- **Non-ambiguity Theorem.** *Every* $w \in \Sigma^*$ *is unambiguous.*
- The proof is by induction on Σ^*.
  - **Basis.** If w is ε then it is unambiguous vacuously.

A generalized non-ambiguity theorem

- **Non-ambiguity Theorem.** *Every* $w \in \Sigma^*$ *is unambiguous.*
- The proof is by induction on Σ^*.
  - **Basis.** If w is ε then it is unambiguous vacuously.
  - **Step:** Assume that w is unambiguous and

$$(\star) \qquad \sigma w = t_1 \cdots t_k = t'_1 \cdots t'_m$$

- **Step:** Assume that w is unambiguous and

$$(\star) \qquad \sigma w = t_1 \cdots t_k = t'_1 \cdots t'_m$$

- **Step:** Assume that w is unambiguous and

$$(\star) \quad \sigma w = t_1 \cdots t_k = t'_1 \cdots t'_m$$

  - Case $\sigma = \mathtt{0}$. ($\sigma = \mathtt{1}$ is similar.)

    $\mathtt{0}$ is the only term starting with $\mathtt{0}$.

    So $t_1 = t'_1 = \mathtt{0}$ and $w = t_2 \cdots t_k = t'_2 \cdots t'_m$.

- **Step:** Assume that w is unambiguous and

$$(\star) \qquad \sigma w = t_1 \cdots t_k = t'_1 \cdots t'_m$$

  - Case $\sigma = \mathtt{0}$. ($\sigma = \mathtt{1}$ is similar.)

    $\mathtt{0}$ is the only term starting with $\mathtt{0}$.

    So $t_1 = t'_1 = \mathtt{0}$ and $w = t_2 \cdots t_k = t'_2 \cdots t'_m$.

    By IH $k-1 = m-1$ and $t_i = t'_i$ for $i = 2..k$.

    So $k = m$ and $t_i = q_i$ for $i = 1..k$.

- **Step:** Assume that w is unambiguous and

$$(\star) \qquad \sigma w = t_1 \cdots t_k = t'_1 \cdots t'_m$$

  - Case $\sigma = \mathtt{0}$. ($\sigma = \mathtt{1}$ is similar.)

    $\mathtt{0}$ is the only term starting with $\mathtt{0}$.

    So $t_1 = t'_1 = \mathtt{0}$ and $w = t_2 \cdots t_k = t'_2 \cdots t'_m$.

    By IH $k-1 = m-1$ and $t_i = t'_i$ for $i = 2..k$.

    So $k = m$ and $t_i = q_i$ for $i = 1..k$.

  - Case $\sigma = \wedge$. ($\sigma = \vee$ is similar.)

    $t_1 = \wedge\, q\, r$ and $t'_1 = \wedge\, q'\, r'$ for some q, r, q', r'.

- **Step:** Assume that w is unambiguous and

$$(\star) \qquad \sigma w = t_1 \cdots t_k = t'_1 \cdots t'_m$$

  - Case $\sigma = \mathtt{0}$. ($\sigma = \mathtt{1}$ is similar.)

    $\mathtt{0}$ is the only term starting with $\mathtt{0}$.

    So $t_1 = t'_1 = \mathtt{0}$ and $w = t_2 \cdots t_k = t'_2 \cdots t'_m$.

    By IH $k-1 = m-1$ and $t_i = t'_i$ for $i = 2..k$.

    So $k = m$ and $t_i = q_i$ for $i = 1..k$.

  - Case $\sigma = \wedge$. ($\sigma = \vee$ is similar.)

    $t_1 = \wedge\, q\, r$ and $t'_1 = \wedge\, q'\, r'$ for some q, r, q', r'.

    So $w = q\, r\, t_2 \, \cdots \, t_k = q'\, r'\, t'_2 \, \cdots \, t'_m$. By IH $k{+}1 = m{+}1$, $q = q'$, $r = r'$ and $t_i = q_i$ for $i = 2..k$.

- **Step:** Assume that w is unambiguous and

$$(\star) \quad \sigma w = t_1 \cdots t_k = t'_1 \cdots t'_m$$

  – Case $\sigma = \mathtt{0}$. ($\sigma = \mathtt{1}$ is similar.)

    $\mathtt{0}$ is the only term starting with $\mathtt{0}$.

    So $t_1 = t'_1 = \mathtt{0}$ and $w = t_2 \cdots t_k = t'_2 \cdots t'_m$.

    By IH $k-1 = m-1$ and $t_i = t'_i$ for $i = 2..k$.

    So $k = m$ and $t_i = q_i$ for $i = 1..k$.

  – Case $\sigma = \wedge$. ($\sigma = \vee$ is similar.)

    $t_1 = \wedge\, q\, r$ and $t'_1 = \wedge\, q'\, r'$ for some q, r, q', r'.

    So $w = q\, r\, t_2 \cdots t_k = q'\, r'\, t'_2 \cdots t'_m$. By IH $k+1 = m+1$, $q = q'$, $r = r'$ and $t_i = q_i$ for $i = 2..k$.

    So $k = m$ and $t_i = t'_i$ for $i = 1..k$.

A generalized non-ambiguity theorem

- **Non-ambiguity Theorem.** *Every $w \in \Sigma^*$ is unambiguous.*
- The proof is by induction on Σ^*.
  - **Basis.** If w is ε then it is unambiguous vacuously.
  - **Step:** Assume that w is unambiguous and $\sigma w = t_1 \cdots t_k = t'_1 \cdots t'_m$

A generalized non-ambiguity theorem

- **Non-ambiguity Theorem.** *Every $w \in \Sigma^*$ is unambiguous.*
- The proof is by induction on Σ^*.
  - **Basis.** If w is ε then it is unambiguous vacuously.
  - **Step:** Assume that w is unambiguous and

    $\sigma w = t_1 \cdots t_k = t'_1 \cdots t'_m$

  We concluded that $k = m$ and $t_i = t'_i$ for $i = 1..k$.

A generalized non-ambiguity theorem

- **Non-ambiguity Theorem.** *Every $w \in \Sigma^*$ is unambiguous.*
- The proof is by induction on Σ^*.
  - **Basis.** If w is ε then it is unambiguous vacuously.
  - **Step:** Assume that w is unambiguous and
    $\sigma w = t_1 \cdots t_k = t'_1 \cdots t'_m$
- By induction on Σ^* we conclude that
  every $w \in \Sigma^*$ is unambiguous.

A generalized non-ambiguity theorem

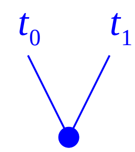
- **Non-ambiguity Theorem.** *Every $w \in \Sigma^*$ is unambiguous.*
- The proof is by induction on Σ^*.
  - **Basis.** If w is ε then it is unambiguous vacuously.
  - **Step:** Assume that w is unambiguous and $\sigma w = t_1 \cdots t_k = t'_1 \cdots t'_m$
- By induction on Σ^* we conclude that every $w \in \Sigma^*$ is unambiguous.
- In particular, every PBT t is a concatenation of 1 string, and therefore must be unambiguous as a term.

Induction over binary trees

- Recall that the set of binary trees is generated

  from a base tree • by juncture:

  if t_0, t_1 are binary trees then so is



- Let $P(x)$ be a property that makes sense for any binary tree t.

- If we can show that

  - **Base:** $P(\bullet)$; and
  - **Step:** If both $P(t_0)$ and $P(t_1)$
    then $P(t)$ for the juncture t above of t_0 and t_1

  then $P(t)$ is true for all binary trees t.

Example: Odd size of binary trees

- ***Can a binary tree have an even number of nodes?***

Example: Odd size of binary trees

- Every binary tree has an odd number of nodes.

Example: Odd size of binary trees

- Every binary tree has an odd number of nodes.
- Let $P(t)$ be the property

  "t has an odd number of nodes"

Example: Odd size of binary trees

- Every binary tree has an odd number of nodes.
- Let $P(t)$ be the property

  "t has an odd number of nodes"

  Induction on trees:

  - **Basis:** $P(\bullet)$ (since 1 is odd)

Example: Odd size of binary trees

- Every binary tree has an odd number of nodes.
- Let $P(t)$ be the property

  "t has an odd number of nodes"

  Induction on trees:

  - **Basis:** $P(\bullet)$ (since 1 is odd)
  - **Step:** Suppose t_0, t_1 are trees of odd sizes n_0 and n_1.
  Let t be obtained from t_0 and t_1. then the size of t is $n_0 + n_1 + 1$ which is again odd.

Example: Odd size of binary trees

- Every binary tree has an odd number of nodes.
- Let $P(t)$ be the property

  "t has an odd number of nodes"

  Induction on trees:

  - **Basis:** $P(\bullet)$ (since 1 is odd)
  - **Step:** Suppose t_0, t_1 are trees of odd sizes n_0 and n_1.

  Let t be obtained from t_0 and t_1. then the size of t is $n_0 + n_1 + 1$ which is again odd.
- By induction on binary tree we conclude that

  $P(t)$ for all binary trees t.

The mother of all inductions

- The Induction principle applies to all inductively generated sets.
- But induction on $\mathbb{N}$ is the one usually invoked. ***Why?***

The mother of all inductions

- The Induction principle applies to all inductively generated sets.
- But induction on $\mathbb{N}$ is the one usually invoked. ***Why?***
- Take generated set $\mathbb{G}$, $P(x)$ a property of $x \in \mathbb{G}$.
- Obtain induction over $x \in \mathbb{G}$ for property $P(x)$
  as induction over $n \in \mathbb{N}$ for the property:

  $P(n)$ *is true for all* $x \in \mathbb{G}$ *generated in* $\leqslant n$ *steps*

The mother of all inductions

- The Induction principle applies to all inductively generated sets.
- But induction on $\mathbb{N}$ is the one usually invoked. ***Why?***
- Take generated set $\mathbb{G}$, $P(x)$ a property of $x \in \mathbb{G}$.
- Obtain induction over $x \in \mathbb{G}$ for property $P(x)$
  as induction over $n \in \mathbb{N}$ for the property:

  $P(n)$ *is true for all* $x \in \mathbb{G}$ *generated in* $\leqslant n$ *steps*

- Note that $\mathbb{N}$ is the ***simplest*** infinite generated set:
  one initial object, ***one*** generative rule, involving ***one*** premise!