6—Inductive Proofs

CS 3234: Logic and Formal Systems

Martin Henz and **Aquinas Hobor**

September 16, 2010

Generated on Thursday 14th October, 2010, 15:39

Inductive definitions

- Often one wishes to define a set with a collection of rules that determine the elements of that set. Simple examples:
  - Binary trees
  - Natural numbers
  - The syntax of a logic (*e.g.*, propositional logic)
- What does it mean to define a set by a collection of rules?

Example: Binary trees (w/o data at nodes)

- $\bullet$ is a binary tree;
- if l and r are binary trees, then so is $\widehat{l \quad r}$

Examples of binary trees:

- $\bullet$

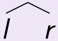
Example: Binary trees (w/o data at nodes)

- $\bullet$ is a binary tree;
- if l and r are binary trees, then so is $\widehat{l \quad r}$

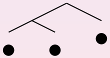
Examples of binary trees:

- $\bullet$
- $\widehat{\bullet \quad \bullet}$

Example: Binary trees (w/o data at nodes)

- • is a binary tree;
- if l and r are binary trees, then so is 

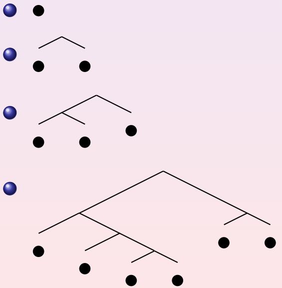
Examples of binary trees:

- •
- 
- 

Example: Binary trees (w/o data at nodes)

- $\bullet$ is a binary tree;
- if l and r are binary trees, then so is $l \quad r$

Examples of binary trees:



Example 2: Natural numbers in unary (base-1) notation

- Z is a natural;
- if n is a natural, then so is $S(n)$.

Example 2: Natural numbers in unary (base-1) notation

- Z is a natural;
- if n is a natural, then so is $S(n)$.

We pronouce Z as “zed” and “S” as successor. We can now define the natural numbers as follows:

$$\textbf{zero} \equiv Z$$

Example 2: Natural numbers in unary (base-1) notation

- Z is a natural;
- if n is a natural, then so is $S(n)$.

We pronouce Z as "zed" and "S" as successor. We can now define the natural numbers as follows:

$$\begin{array}{rcl} \textbf{zero} & \equiv & Z \\ \textbf{one} & \equiv & S(Z) \end{array}$$

Example 2: Natural numbers in unary (base-1) notation

- Z is a natural;
- if n is a natural, then so is $S(n)$.

We pronouce Z as "zed" and "S" as successor. We can now define the natural numbers as follows:

$$\begin{array}{rcl} \textbf{zero} & \equiv & Z \\ \textbf{one} & \equiv & S(Z) \\ \textbf{two} & \equiv & S(S(Z)) \end{array}$$

Example 2: Natural numbers in unary (base-1) notation

- Z is a natural;
- if n is a natural, then so is $S(n)$.

We pronouce Z as "zed" and "S" as successor. We can now define the natural numbers as follows:

$$
\begin{array}{rcl}
\textbf{zero} & \equiv & Z \\
\textbf{one} & \equiv & S(Z) \\
\textbf{two} & \equiv & S(S(Z)) \\
\ldots & &
\end{array}
$$

It's possible to view naturals as trees, too:

zero	$\equiv$	Z	Z
one	$\equiv$	$S(Z)$	$\begin{array}{c} \mathsf{S} \\ \vert \\ Z \end{array}$
two	$\equiv$	$S(S(Z))$	$\begin{array}{c} \mathsf{S} \\ \vert \\ \mathsf{S} \\ \vert \\ Z \end{array}$

$\dots$

Examples (more formally)

- Binary trees: The set *Tree* is defined by the rules

$$\frac{}{\bullet} \qquad\qquad \frac{t_l \qquad t_r}{\overset{\wedge}{t_l \quad t_r}}$$

- Naturals: The set *Nat* is defined by the rules

$$\frac{}{Z} \qquad\qquad \frac{n}{S(n)}$$

Given a collection of rules, what set does it define?

- What is the set of trees?
- What is the set of naturals?

Do the rules pick out a unique set?

There can be many sets that satisfy a given collection of rules

- $IndNum = \{Z, S(Z), \ldots\}$
- $CoIndNum = \{Z, S(Z), S(S(Z)), \ldots, S(S(S(\ldots)))\}$
- $WeirdNum = MyNum \cup \{\infty, S(\infty), \ldots\}$, where ∞ is an arbitrary symbol.

All three of these different sets satisfy the rules defining numerals.

An inductively defined set is the **least set** for the given rules (*i.e.*, the extremal clause).

Example: $IndNum = \{Z, S(Z), S(S(Z)), \ldots\}$ is the least set that satisfies these rules:

- $Z \in Num$
- if $n \in Num$, then $S(n) \in Num$.

What do we mean by "least"?

Answer: The smallest with respect to the subset ordering on sets.

- Contains no "junk", only what is required by the rules.
- Since $\mathit{CoIndNum} \supsetneq \mathit{IndNum}$, *CoIndNum* is ruled out by the extremal clause.
- Since $\mathit{WeirdNum} \supsetneq \mathit{IndNum}$, *WeirdNum* is ruled out by the extremal clause.
- *IndNum* is "ruled in" because it has no "junk". That is, for any set S satisfying the rules, $S \supset \mathit{IndNum}$

We almost always want to define sets with inductive definitions, and so have some simple notation to do so quickly:

$$S = \text{Constructor}_1(\ldots) \;\mid\; \text{Constructor}_2(\ldots) \;\mid\; \ldots$$

where S can appear in the $\ldots$ on the right hand side (along with other things). The Constructor_i are the names of the different rules (sometimes text, sometimes symbols). This is called a *recursive definition*.

Examples:

- Binary trees: $\tau = \bullet \;\mid\; \widehat{\tau \quad \tau}$
- Naturals: $\mathbb{N} = Z \;\mid\; S(\mathbb{N})$

There is a close connection between a recursive definition and a definition by rules:

- Binary trees: $\tau = \bullet \mid \overset{\frown}{\tau \quad \tau}$

$$\frac{}{\bullet} \qquad\qquad \frac{t_l \quad t_r}{\overset{\frown}{t_l \quad t_r}}$$

- Naturals: $\mathbb{N} = Z \mid S(\mathbb{N})$

$$\frac{}{Z} \qquad\qquad \frac{n}{S(n)}$$

A definition written in "recursive definition style" is assumed to be the least set satisfying the rules; that is, the notation means that

CoInductive sets

What about the other two choices? Is there any value in them?

- $CoIndNum = \{Z, S(Z), S(S(Z)), \ldots, S(S(S(\ldots)))\}$
- $WeirdNum = MyNum \cup \{\infty, S(\infty), \ldots\}$, where ∞ is an arbitrary symbol.

As a rule, there is no point at all to *WeirdNum*: it is just a set that we don't want—and if we do, we can define it inductively by $WeirdNum = Z \mid \infty \mid S(WeirdNum)$.

CoInductive sets

What about the other two choices? Is there any value in them?

- $CoIndNum = \{Z, S(Z), S(S(Z)), \ldots, S(S(S(\ldots)))\}$
- $WeirdNum = MyNum \cup \{\infty, S(\infty), \ldots\}$, where ∞ is an arbitrary symbol.

As a rule, there is no point at all to *WeirdNum*: it is just a set that we don't want—and if we do, we can define it inductively by $WeirdNum = Z \mid \infty \mid S(WeirdNum)$.

But there is value to the set CoIndNum. This is the *greatest* set that can be defined using a set of rules without adding junk like ∞. Such a set is called *co-inductively* defined, and is useful for reasoning about infinitely-long objects such as streams.

What's the Big Deal with inductively defined sets?

Inductively defined sets "come with" an *induction principle*.
Suppose I is inductively defined by rules R.

- To show that every $x \in I$ has property P, it is enough to show that regardless of which rule is used to "build" x, P holds; this is called *taking cases* or *inversion*.
- Note that one can take cases also on co-inductively defined sets like *CoIndNum*—but not on sets like *WeirdNum*.
- Sometimes, taking cases is not enough; in that case we can attempt a more complicated proof where we show that P is preserved by each of the rules of R; this is called *structural induction* or *rule induction*. We need to have an inductively defined set; *we cannot do induction over coinductive sets*.

Example: Sign of a Natural

Consider the following definition:

- The natural Z has sign $\mathbf{0}$.
- For any natural n, the natural $S(n)$ has sign $\mathbf{1}$.

Let P be the following property: Every natural has sign $\mathbf{0}$ or $\mathbf{1}$.

Example: Sign of a Natural

Consider the following definition:

- The natural Z has sign $\mathbf{0}$.
- For any natural n, the natural $S(n)$ has sign $\mathbf{1}$.

Let P be the following property: Every natural has sign $\mathbf{0}$ or $\mathbf{1}$.

Does P satisfy the rules $\dfrac{}{Z}$ $\dfrac{n}{S(n)}$?

How to take cases

To show that every $n \in Nat$ has property P, it is enough to show:

- Z has property P.
- For any n, $S(n)$ has property P.

How to take cases

To show that every $n \in Nat$ has property P, it is enough to show:

- Z has property P.
- For any n, $S(n)$ has property P.

Recall:

- The natural Z has sign $\mathbf{0}$.
- For any natural n, the natural $S(n)$ has sign $\mathbf{1}$.

Let $P =$ "Every natural has sign $\mathbf{0}$ or $\mathbf{1}$.". Does P hold for all $\mathbb{N}$?

How to take cases

To show that every $n \in Nat$ has property P, it is enough to show:

- Z has property P.
- For any n, $S(n)$ has property P.

Recall:

- The natural Z has sign $\mathbf{0}$.
- For any natural n, the natural $S(n)$ has sign $\mathbf{1}$.

Let $P =$ "Every natural has sign $\mathbf{0}$ or $\mathbf{1}$.". Does P hold for all $\mathbb{N}$?

Proof. We take cases **on the structure of n** as follows:

- Z has sign $\mathbf{0}$, so P holds for Z. $\surd$

How to take cases

To show that every $n \in Nat$ has property P, it is enough to show:

- Z has property P.
- For any n, $S(n)$ has property P.

Recall:

- The natural Z has sign $\mathbf{0}$.
- For any natural n, the natural $S(n)$ has sign $\mathbf{1}$.

Let $P =$ "Every natural has sign $\mathbf{0}$ or $\mathbf{1}$.". Does P hold for all $\mathbb{N}$?

Proof. We take cases **on the structure of n** as follows:

- Z has sign $\mathbf{0}$, so P holds for Z. $\surd$
- For any n, $S(n)$ has sign $\mathbf{1}$, so P holds for any $S(n)$. $\surd$

Thus, P holds for all naturals.

Example: Even and Odd Naturals

- The natural Z has parity $\mathbf{0}$.
- If n is a natural with parity $\mathbf{0}$, then $S(n)$ has parity $\mathbf{1}$.
- If n is a natural with parity $\mathbf{1}$, then $S(n)$ has parity $\mathbf{0}$.

Let P be: Every natural has parity $\mathbf{0}$ or parity $\mathbf{1}$.

Example: Even and Odd Naturals

- The natural Z has parity $\mathbf{0}$.
- If n is a natural with parity $\mathbf{0}$, then $S(n)$ has parity $\mathbf{1}$.
- If n is a natural with parity $\mathbf{1}$, then $S(n)$ has parity $\mathbf{0}$.

Let P be: Every natural has parity $\mathbf{0}$ or parity $\mathbf{1}$.

Can we prove this by taking cases?

Taking cases

We need to show $P =$ "Every natural has parity $\mathbf{0}$ or parity $\mathbf{1}$.",

- Z has property P.
- For any n, $S(n)$ has property P.

Where parity is defined by

- The natural Z has parity $\mathbf{0}$.
- If n is a natural with parity $\mathbf{0}$, then $S(n)$ has parity $\mathbf{1}$.
- If n is a natural with parity $\mathbf{1}$, then $S(n)$ has parity $\mathbf{0}$.

Taking cases

We need to show $P =$ "Every natural has parity $\mathbf{0}$ or parity $\mathbf{1}$.",

- Z has property P.
- For any n, $S(n)$ has property P.

Where parity is defined by

- The natural Z has parity $\mathbf{0}$.
- If n is a natural with parity $\mathbf{0}$, then $S(n)$ has parity $\mathbf{1}$.
- If n is a natural with parity $\mathbf{1}$, then $S(n)$ has parity $\mathbf{0}$.

Proof. We take cases **on the structure of n** as follows:

- Z has parity $\mathbf{0}$, so P holds for Z. $\surd$

Taking cases

We need to show $P =$ "Every natural has parity $\mathbf{0}$ or parity $\mathbf{1}$.",

- Z has property P.
- For any n, $S(n)$ has property P.

Where parity is defined by

- The natural Z has parity $\mathbf{0}$.
- If n is a natural with parity $\mathbf{0}$, then $S(n)$ has parity $\mathbf{1}$.
- If n is a natural with parity $\mathbf{1}$, then $S(n)$ has parity $\mathbf{0}$.

Proof. We take cases **on the structure of n** as follows:

- Z has parity $\mathbf{0}$, so P holds for Z. $\surd$
- For any n, $S(n)$ has parity

Taking cases

We need to show $P =$ "Every natural has parity $\mathbf{0}$ or parity $\mathbf{1}$.",

- Z has property P.
- For any n, $S(n)$ has property P.

Where parity is defined by

- The natural Z has parity $\mathbf{0}$.
- If n is a natural with parity $\mathbf{0}$, then $S(n)$ has parity $\mathbf{1}$.
- If n is a natural with parity $\mathbf{1}$, then $S(n)$ has parity $\mathbf{0}$.

Proof. We take cases **on the structure of n** as follows:

- Z has parity $\mathbf{0}$, so P holds for Z. $\surd$
- For any n, $S(n)$ has parity well...

Taking cases

We need to show $P =$ "Every natural has parity **0** or parity **1**.",

- Z has property P.
- For any n, $S(n)$ has property P.

Where parity is defined by

- The natural Z has parity **0**.
- If n is a natural with parity **0**, then $S(n)$ has parity **1**.
- If n is a natural with parity **1**, then $S(n)$ has parity **0**.

Proof. We take cases **on the structure of n** as follows:

- Z has parity **0**, so P holds for Z. $\checkmark$
- For any n, $S(n)$ has parity well... hmmm...

Taking cases

We need to show $P =$ "Every natural has parity $\mathbf{0}$ or parity $\mathbf{1}$.",

- Z has property P.
- For any n, $S(n)$ has property P.

Where parity is defined by

- The natural Z has parity $\mathbf{0}$.
- If n is a natural with parity $\mathbf{0}$, then $S(n)$ has parity $\mathbf{1}$.
- If n is a natural with parity $\mathbf{1}$, then $S(n)$ has parity $\mathbf{0}$.

Proof. We take cases **on the structure of n** as follows:

- Z has parity $\mathbf{0}$, so P holds for Z. $\surd$
- For any n, $S(n)$ has parity well... hmmm... it is unclear;

Taking cases

We need to show $P =$ "Every natural has parity $\mathbf{0}$ or parity $\mathbf{1}$.",

- Z has property P.
- For any n, $S(n)$ has property P.

Where parity is defined by

- The natural Z has parity $\mathbf{0}$.
- If n is a natural with parity $\mathbf{0}$, then $S(n)$ has parity $\mathbf{1}$.
- If n is a natural with parity $\mathbf{1}$, then $S(n)$ has parity $\mathbf{0}$.

Proof. We take cases **on the structure of n** as follows:

- Z has parity $\mathbf{0}$, so P holds for Z. $\surd$
- For any n, $S(n)$ has parity well... hmmm... it is unclear; it depends on the parity of n.

Taking cases

We need to show $P =$ "Every natural has parity $\mathbf{0}$ or parity $\mathbf{1}$.",

- Z has property P.
- For any n, $S(n)$ has property P.

Where parity is defined by

- The natural Z has parity $\mathbf{0}$.
- If n is a natural with parity $\mathbf{0}$, then $S(n)$ has parity $\mathbf{1}$.
- If n is a natural with parity $\mathbf{1}$, then $S(n)$ has parity $\mathbf{0}$.

Proof. We take cases **on the structure of n** as follows:

- Z has parity $\mathbf{0}$, so P holds for Z. $\surd$
- For any n, $S(n)$ has parity well... hmmm... it is unclear; it depends on the parity of n. **X**

Taking cases

We need to show P = "Every natural has parity **0** or parity **1**.",

- Z has property P.
- For any n, $S(n)$ has property P.

Where parity is defined by

- The natural Z has parity **0**.
- If n is a natural with parity **0**, then $S(n)$ has parity **1**.
- If n is a natural with parity **1**, then $S(n)$ has parity **0**.

Proof. We take cases **on the structure of n** as follows:

- Z has parity **0**, so P holds for Z. $\surd$
- For any n, $S(n)$ has parity well... hmmm... it is unclear; it depends on the parity of n. **X**

We are stuck! We need an extra fact about n's parity...

Induction hypothesis

This fact is called an *induction hypothesis*. To get such an induction hypothesis we do *induction*, which is a more powerful way to take cases. To show that every $n \in Num$ has property P, we must show that every rule preserves P; that is:

- Z has property P.
- if n has property P, then $S(n)$ has property P.

The new part is "if n has property P, then ..."; this is the induction hypothesis.

Induction hypothesis

This fact is called an *induction hypothesis*. To get such an induction hypothesis we do *induction*, which is a more powerful way to take cases. To show that every $n \in Num$ has property P, we must show that every rule preserves P; that is:

- Z has property P.
- if n has property P, then $S(n)$ has property P.

The new part is "if n has property P, then ..."; this is the induction hypothesis.

Note that for the naturals, structural induction is just ordinary mathematical induction!

Using induction to fix our proof

Every natural has parity **0** or parity **1**.

Proof. We take cases **on the structure of n** as follows:

- Z has parity **0**, so P holds for Z. $\surd$

Using induction to fix our proof

Every natural has parity **0** or parity **1**.

Proof. We take cases **on the structure of n** as follows:

- Z has parity **0**, so P holds for Z. $\checkmark$
- For any n, we can't determine the parity of $S(n)$ until we know something about the parity of n. **X**

Using induction to fix our proof

Every natural has parity **0** or parity **1**.

Proof. We take cases **on the structure of n** as follows:

- Z has parity **0**, so P holds for Z. $\surd$
- For any n, we can't determine the parity of $S(n)$ until we know something about the parity of n. **X**

Proof. We **do induction on the structure of n** as follows:

- Z has parity **0**, so P holds for Z. $\surd$

Using induction to fix our proof

Every natural has parity **0** or parity **1**.

Proof. We take cases **on the structure of n** as follows:

- Z has parity **0**, so P holds for Z. $\surd$
- For any n, we can't determine the parity of $S(n)$ until we know something about the parity of n. **X**

Proof. We **do induction on the structure of n** as follows:

- Z has parity **0**, so P holds for Z. $\surd$
- Given an n such that P holds on n, show that P holds on $S(n)$. Since P holds on n, the parity of n is **0** or **1**. If the parity of n is **0**, then the parity of $S(n)$ is **1**. If the parity of n is **1**, then the parity of $S(n)$ is **0**. In either case, the parity of $S(n)$ is **0** or **1**, so if P holds on n then P holds on $S(n)$. $\surd$

Thus, P holds for an natural n.

Extending case analysis and structural induction to trees

Case analysis: to show that every tree has property P, prove that

- $\bullet$ has property P.
- for all τ_1 and τ_2, $\begin{array}{c} \diagup\!\diagdown \\ \tau_1 \quad \tau_2 \end{array}$ has property P.

Structural induction: to show that every tree has property P, prove

- $\bullet$ has property P.
- if τ_1 and τ_2 have property P, then $\begin{array}{c} \diagup\!\diagdown \\ \tau_1 \quad \tau_2 \end{array}$ has property P.

Extending case analysis and structural induction to trees

Case analysis: to show that every tree has property P, prove that

- $\bullet$ has property P.
- for all τ_1 and τ_2, $\overset{\frown}{\tau_1 \quad \tau_2}$ has property P.

Structural induction: to show that every tree has property P, prove

- $\bullet$ has property P.
- if τ_1 and τ_2 have property P, then $\overset{\frown}{\tau_1 \quad \tau_2}$ has property P.

Note that we do not require that τ_1 and τ_2 be the same height!

Structural induction vs. induction on naturals

You are probably familiar with regular mathematical induction: to prove something for any natural n, first prove it is true about 0 and then show that if it is true about n then it is true about $n + 1$.

Structural induction vs. induction on naturals

You are probably familiar with regular mathematical induction: to prove something for any natural n, first prove it is true about 0 and then show that if it is true about n then it is true about $n+1$.

How does structural induction compare to regular mathematical induction on, say, the height of trees?

Structural induction vs. induction on naturals

You are probably familiar with regular mathematical induction: to prove something for any natural n, first prove it is true about 0 and then show that if it is true about n then it is true about $n + 1$.

How does structural induction compare to regular mathematical induction on, say, the height of trees?

- For both types of induction, the base case is the same:
  - $\bullet$ has property P.

Structural induction vs. induction on naturals

You are probably familiar with regular mathematical induction: to prove something for any natural n, first prove it is true about 0 and then show that if it is true about n then it is true about $n + 1$.

How does structural induction compare to regular mathematical induction on, say, the height of trees?

- For both types of induction, the base case is the same:
  $\bullet$ has property P.
- For structural induction:
  if τ_1 and τ_2 have property P, then $\overset{\frown}{\tau_1 \quad \tau_2}$ has property P.

Structural induction vs. induction on naturals

You are probably familiar with regular mathematical induction: to prove something for any natural n, first prove it is true about 0 and then show that if it is true about n then it is true about $n+1$.

How does structural induction compare to regular mathematical induction on, say, the height of trees?

- For both types of induction, the base case is the same:
  $\bullet$ has property P.
- For structural induction:
  if τ_1 and τ_2 have property P, then $\underset{\tau_1 \quad \tau_2}{\bigwedge}$ has property P.
- For regular mathematical induction on the height of trees:
  if τ_1 and τ_2 **are trees of height n** and have property P, then $\underset{\tau_1 \quad \tau_2}{\bigwedge}$ **is a tree of height $\mathbf{n+1}$** and has property P.

Structural induction vs. induction on naturals

You are probably familiar with regular mathematical induction: to prove something for any natural n, first prove it is true about 0 and then show that if it is true about n then it is true about $n + 1$.

How does structural induction compare to regular mathematical induction on, say, the height of trees?

- For both types of induction, the base case is the same:
  $\bullet$ has property P.
- For structural induction:
  if τ_1 and τ_2 have property P, then $\widehat{\tau_1 \quad \tau_2}$ has property P.
- For regular mathematical induction on the height of trees:
  if τ_1 and τ_2 **are trees of height n** and have property P, then $\widehat{\tau_1 \quad \tau_2}$ **is a tree of height** $\mathbf{n + 1}$ and has property P.

How can we justify case analysis and induction?

Let I be a set inductively defined by rules R.

How can we justify case analysis and induction?

Let I be a set inductively defined by rules R.

- Case analysis is really a lightweight "special case" of structural induction where we do not use the induction hypothesis. If structural induction is sound, then case analysis will be as well.

How can we justify case analysis and induction?

Let I be a set inductively defined by rules R.

- Case analysis is really a lightweight "special case" of structural induction where we do not use the induction hypothesis. If structural induction is sound, then case analysis will be as well.
- One way to think of a property P is that it is exactly the set of items that have property P. We would like to show that if you are in the set I then you have property P, that is, $P \supseteq I$.

How can we justify case analysis and induction?

Let I be a set inductively defined by rules R.

- Case analysis is really a lightweight "special case" of structural induction where we do not use the induction hypothesis. If structural induction is sound, then case analysis will be as well.
- One way to think of a property P is that it is exactly the set of items that have property P. We would like to show that if you are in the set I then you have property P, that is, $P \supseteq I$.
- Remember that I is (by definition) the smallest set satisfying the rules in R.

How can we justify case analysis and induction?

Let I be a set inductively defined by rules R.

- Case analysis is really a lightweight "special case" of structural induction where we do not use the induction hypothesis. If structural induction is sound, then case analysis will be as well.
- One way to think of a property P is that it is exactly the set of items that have property P. We would like to show that if you are in the set I then you have property P, that is, $P \supseteq I$.
- Remember that I is (by definition) the smallest set satisfying the rules in R.
- Hence if P satisfies (is preserved by) the rules of R, then $P \supseteq I$.

How can we justify case analysis and induction?

Let I be a set inductively defined by rules R.

- Case analysis is really a lightweight "special case" of structural induction where we do not use the induction hypothesis. If structural induction is sound, then case analysis will be as well.
- One way to think of a property P is that it is exactly the set of items that have property P. We would like to show that if you are in the set I then you have property P, that is, $P \supseteq I$.
- Remember that I is (by definition) the smallest set satisfying the rules in R.
- Hence if P satisfies (is preserved by) the rules of R, then $P \supseteq I$.
- This is why the extremal clause matters so much!

Example: Height of a Tree

- To show: Every tree has a height, defined as follows:
  - The height of $\bullet$ is 0.
  - If the tree l has height h_l and the tree r has height h_r, then the tree $\overset{\frown}{l \quad r}$ has height $1 + max(h_l, h_r)$.
- Clearly, every tree has at most one height, but does it have any height at all?

Example: Height of a Tree

- To show: Every tree has a height, defined as follows:
  - The height of $\bullet$ is 0.
  - If the tree l has height h_l and the tree r has height h_r, then the tree $\widehat{l \quad r}$ has height $1 + max(h_l, h_r)$.
- Clearly, every tree has at most one height, but does it have any height at all?
- It may seem obvious that every tree has a height, but notice that the justification relies on structural induction!
  - An "infinite tree" does not have a height!
  - But the extremal clause rules out the infinite tree!

Example: height

- Formally, we prove that for every tree t, there exists a number h satisfying the specification of height.
- Proceed by induction **on the structure of trees**, showing that the property "there exists a height h for t" satisfies (is preserved by) these rules.

Example: height

- Rule 1: $\bullet$ is a tree.
  Does there exist h such that h is the height of *Empty*?
  Yes! Take h=0.
- Rule 2: $\overset{\frown}{l \quad r}$ is a tree if l and r are trees.
  Suppose that there exists h_l and h_r, the heights of l and r, respectively (*the induction hypothesis*).
  Does there exist h such that h is the height of $Node(l, r)$?
  Yes! Take $h = 1 + max(h_l, h_r)$.

Thus, we have proved that all trees have a height.

Please see the Coq script.

Extension: the syntax of propositional logic

We have already seen a major example of a recursive definition in class: the syntax of propositional logic!

$$F = \text{Atom}(\alpha) \;\mid\; \neg F \;\mid\; F \vee F \;\mid\; F \wedge F \;\mid\; F \rightarrow F$$

It is perfectly reasonable to do case analysis and structural induction on the syntax of a formula ϕ. In fact, we will see an example of this shortly!

Extension: the structure of a natural deduction proof

We have seen another important kind of tree-like structure in class already: natural deduction proofs! In homework 1, you did proofs using a "3 column" style; in homework 2, you will do a few proofs using the graphical tree-style, such as this proof of $p \wedge q \vdash q \wedge p$:

$$\dfrac{\dfrac{p \wedge q}{q}[\wedge e2] \qquad \dfrac{p \wedge q}{p}[\wedge e1]}{q \wedge p}[\wedge i]$$

It is *also* reasonable to do structural induction on the structure of a formal proof. We will see an example of this shortly, too!

- An inductively defined set is the least set closed under a collection of rules.
- Rules have the form:
  "If $x_1 \in X$ and $\ldots$ and $x_n \in X$, then $x \in X$."
- Notation:
  $$\frac{x_1 \qquad \cdots \qquad x_n}{x}$$
- Notation: sometimes we can define the entire set easily with a recursive definition: $S = \mathrm{C}_1(\ldots) \mid \mathrm{C}_2(\ldots) \mid \ldots$

- Inductively defined sets admit proofs by rule induction.
- For each rule

$$\frac{x_1 \qquad \cdots \qquad x_n}{x}$$

  assume that $x_1 \in P$, ..., $x_n \in P$, and show that $x \in P$.
- Conclude that every element of the set is in P.