03b—Inductive Definitions

CS 5209: Foundation in Logic and AI

Martin Henz and Aqinas Hobor

January 28, 2010

Generated on Monday 1st February, 2010, 16:38
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

What does it mean to define a set by a collection of rules?
Example 1: Binary trees (w/o data at nodes)

- is a binary tree;
- if $l$ and $r$ are binary trees, then so is $l \xrightarrow{\text{}} r$

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 $S(n)$.

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

- $\text{zero} \equiv Z$
- $\text{one} \equiv S(Z)$
- $\text{two} \equiv S(S(Z))$
- $\ldots$

CS 5209: Foundation in Logic and AI

03b—Inductive Definitions
It's possible to view naturals as trees, too:

\[
\begin{align*}
\text{zero} & \equiv Z \\
\text{one} & \equiv S(Z) \\
\text{two} & \equiv S(S(Z)) \\
\ldots
\end{align*}
\]
Examples (more formally)

- Binary trees: The set $\text{Tree}$ is defined by the rules

\[
\begin{array}{c}
\text{null} \\
\hline
\end{array}
\quad
\begin{array}{c}
t_l \\
\hline
\end{array}
\quad
\begin{array}{c}
t_r \\
\hline
\end{array}
\]

- Naturals: The set $\text{Nat}$ is defined by the rules

\[
\begin{array}{c}
\mathbb{Z} \\
\hline
\end{array}
\quad
\begin{array}{c}
n \\
\hline
\end{array}
\quad
\begin{array}{c}
S(n) \\
\hline
\end{array}
\]
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

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

Both $MyNum$ and $YourNum$ satisfy the rules defining numerals (i.e., the rules are true for these sets).

Really?
MyNum Satisfies the Rules

MyNum = \{Z, Succ(Z), S(S(Z)), \ldots\}

Does MyNum satisfy the rules?

- Z ∈ MyNum. √
- If n ∈ MyNum, then S(n) ∈ MyNum. √
YourNum Satisfies the Rules

YourNum = \{Z, S(Z), S(S(Z)), \ldots\} \cup \{\infty, S(\infty), \ldots\}

Does YourNum satisfy the rules?

- Z \in YourNum. √
- If n \in YourNum, then S(n) \in YourNum. √
Both MyNum and YourNum satisfy all rules.

It is not enough that a set satisfies all rules.

Something more is needed: an extremal clause.

- “and nothing else”
- “the least set that satisfies these rules”
An inductively defined set is the **least set** for the given rules.

Example: $MyNum = \{Z, S(Z), S(S(Z)), \ldots\}$ 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 YourNum $\supseteq$ MyNum, YourNum is ruled out by the extremal clause.
- MyNum is “ruled in” because it has no “junk”. That is, for any set S satisfying the rules, S $\supset$ MyNum
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 \( \text{Constructor}_i \) are the names of the different rules (sometimes text, sometimes symbols). This is called a recursive definition.

Examples:

- Binary trees: \( \tau = \bullet \mid \tau \trie \tau \)
- Naturals: \( \mathbb{N} = \mathbb{Z} \mid S(\mathbb{N}) \)
There is a close connection between a recursive definition and a definition by rules:

- Binary trees: \( \tau = \bullet \mid \tau \cdot \tau \)

  \[
  \begin{array}{c}
  \_ \\
  \_ \\
  \bullet \\
  \end{array}
  \quad \begin{array}{c}
  t_l \\
  \_ \\
  t_r \\
  \end{array}
  \]

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

  \[
  \begin{array}{c}
  \_ \\
  \_ \\
  \mathbb{Z} \\
  \end{array}
  \quad \begin{array}{c}
  n \\
  \_ \\
  S(n) \\
  \end{array}
  \]

“recursive definition style” means that the extremal clause holds.
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.
- 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.
Consider the following definition:

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

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

Does $P$ satisfy the rules $Z$ and $S(n)$?
How to take cases

To show that every \( n \in \text{Nat} \) 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 \( 0 \).
- For any natural \( n \), the natural \( S(n) \) has sign \( 1 \).

Let \( P = \text{“Every natural has sign } 0 \text{ or } 1 \text{.”} \). Does \( P \) hold for all \( \mathbb{N} \)?

Proof. We take cases on the structure of \( n \) as follows:

- \( Z \) has sign \( 0 \), so \( P \) holds for \( Z \). \( \checkmark \)
- For any \( n \), \( S(n) \) has sign \( 1 \), so \( P \) holds for any \( S(n) \). \( \checkmark \)

Thus, \( P \) holds for all naturals. 

CS 5209: Foundation in Logic and AI 03b—Inductive Definitions 18
Example: Even and Odd Naturals

- 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 $0$.

Let $P$ be: Every natural has parity $0$ or parity $1$.

Can we prove this by taking cases?
Taking cases

We need to show $P = \text{“Every natural has parity 0 or parity 1.”}$,
- $Z$ 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 $Z$. √
- 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...
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!
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. √
- 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. √
- 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). √

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

- has property $P$.
- for all $\tau_1$ and $\tau_2$, $\tau_1 \rightarrow \tau_2$ has property $P$.

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

- has property $P$.
- if $\tau_1$ and $\tau_2$ have property $P$, then $\tau_1 \rightarrow \tau_2$ has property $P$.

Note that we do not require that $\tau_1$ and $\tau_2$ be the same height!
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!
To show: Every tree has a height, defined as follows:

- The height of is 0.
- If the tree has height and the tree has height , then the tree 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!
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:** $\langle l, r \rangle$ 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.