COSE212: Programming Languages

Lecture 2 — Inductive Definitions (2)

Hakjoo Oh
2019 Fall

Contents

- More examples of inductive definitions
  - natural numbers, strings, booleans
  - lists, binary trees
  - arithmetic expressions, propositional logic
- Structural induction
  - three example proofs

Natural Numbers

The set of natural numbers:

$$\mathbb{N} = \{0, 1, 2, 3, \ldots\}$$

is inductively defined:

$$\frac{}{0} \qquad \frac{n}{n+1}$$

The inference rules can be expressed by a grammar:

$$n \rightarrow 0 \mid n + 1$$

Interpretation:

- 0 is a natural number.
- If n is a natural number then so is $n + 1$.

Strings

The set of strings over alphabet $\{a, \ldots, z\}$, e.g., ϵ, a, b, $\ldots$, z, aa, ab, $\ldots$, az, ba, $\ldots$ az, aaa, $\ldots$, zzz, and so on. Inference rules:

$$\frac{}{\epsilon} \qquad \frac{\alpha}{a\alpha} \qquad \frac{\alpha}{b\alpha} \qquad \cdots \qquad \frac{\alpha}{z\alpha}$$

or simply,

$$\frac{}{\epsilon} \qquad \frac{\alpha}{x\alpha} \; x \in \{a, \ldots, z\}$$

In grammar:

$$
\begin{aligned}
\alpha \;\; \rightarrow \;\; & \epsilon \\
| \;\; & x\alpha \quad (x \in \{a, \ldots, z\})
\end{aligned}
$$

Boolean Values

The set of boolean values:

$$\mathbb{B} = \{true, false\}.$$

If a set is finite, just enumerate all of its elements by axioms:

$$\overline{true} \qquad \overline{false}$$

In grammar:

$$b \rightarrow true \mid false$$

Lists

Examples of lists of integers:

1. **nil**
2. $14 \cdot$ **nil**
3. $3 \cdot 14 \cdot$ **nil**
4. $-7 \cdot 3 \cdot 14 \cdot$ **nil**

Inference rules:

$$\overline{\textbf{nil}} \qquad \frac{l}{n \cdot l} \; n \in \mathbb{Z}$$

In grammar:

$$
\begin{aligned}
l \;\; \rightarrow \;\; & \textbf{nil} \\
| \;\; & n \cdot l \quad (n \in \mathbb{Z})
\end{aligned}
$$

Lists

A proof that $-7 \cdot 3 \cdot 14 \cdot \mathbf{nil}$ is a list of integers:

$$\cfrac{\cfrac{\cfrac{\overline{\mathbf{nil}}}{14 \cdot \mathbf{nil}} \; 14 \in \mathbb{Z}}{3 \cdot 14 \cdot \mathbf{nil}} \; 3 \in \mathbb{Z}}{-7 \cdot 3 \cdot 14 \cdot \mathbf{nil}} \; -7 \in \mathbb{Z}$$

The proof tree is also called *derivation tree* or *deduction tree*.

Binary Trees

Examples of binary trees:

1. **leaf**
2. $(2, \textbf{leaf}, \textbf{leaf})$
3. $(1, (2, \textbf{leaf}, \textbf{leaf}), \textbf{leaf})$
4. $(1, (2, \textbf{leaf}, \textbf{leaf}), (3, (4, \textbf{leaf}, \textbf{leaf}), \textbf{leaf}))$

Inference rules:

$$\frac{}{\textbf{leaf}} \qquad \frac{t_1 \quad t_2}{(n, t_1, t_2)} \; n \in \mathbb{Z}$$

In grammar:

$$\begin{aligned} t \;\; \rightarrow \;\; & \textbf{leaf} \\ | \;\; & (n, t, t) \quad (n \in \mathbb{Z}) \end{aligned}$$

Binary Trees

A proof that

$$(1, (2, \textbf{leaf}, \textbf{leaf}), (3, (4, \textbf{leaf}, \textbf{leaf}), \textbf{leaf}))$$

is a binary tree:

$$\cfrac{\cfrac{\overline{\textbf{leaf}}}{(2, \textbf{leaf}, \textbf{leaf})} \; 2 \in \mathbb{Z} \quad \cfrac{\cfrac{\overline{\textbf{leaf}}}{(4, \textbf{leaf}, \textbf{leaf})} \; 4 \in \mathbb{Z}}{(3, (4, \textbf{leaf}, \textbf{leaf}), \textbf{leaf})} \; 3 \in \mathbb{Z}}{(1, (2, \textbf{leaf}, \textbf{leaf}), (3, (4, \textbf{leaf}, \textbf{leaf}), \textbf{leaf}))} \; 1 \in \mathbb{Z}$$

Binary Trees: a different version

Binary tree examples: 1, $(1, \mathbf{nil})$, $(1, 2)$, $((1, 2), \mathbf{nil})$, $((1, 2), (3, 4))$.
Inference rules:

$$\overline{n} \; n \in \mathbb{Z} \qquad \frac{t}{(t, \mathbf{nil})} \qquad \frac{t}{(\mathbf{nil}, t)} \qquad \frac{t_1 \quad t_2}{(t_1, t_2)}$$

In grammar:

$$
\begin{aligned}
t \;\; \rightarrow \;\; & n \quad (n \in \mathbb{Z}) \\
| \;\; & (t, \mathbf{nil}) \\
| \;\; & (\mathbf{nil}, t) \\
| \;\; & (t, t)
\end{aligned}
$$

A proof that $((1, 2), (3, \mathbf{nil}))$ is a binary tree:

$$\frac{\dfrac{\overline{1} \quad \overline{2}}{(1, 2)} \quad \dfrac{\overline{3}}{(3, \mathbf{nil})}}{((1, 2), (3, \mathbf{nil}))}$$

Expressions

Expression examples: 2, -2, $1 + 2$, $1 + (2 * (-3))$, etc.

Inference rules:

$$\overline{n} \; n \in \mathbb{Z} \qquad \frac{e}{-e} \qquad \frac{e_1 \quad e_2}{e_1 + e_2} \qquad \frac{e_1 \quad e_2}{e_1 * e_2} \qquad \frac{e}{(e)}$$

In grammar:

$$
\begin{aligned}
e \quad \rightarrow \quad & n \quad (n \in \mathbb{Z}) \\
| \quad & -e \\
| \quad & e + e \\
| \quad & e * e \\
| \quad & (e)
\end{aligned}
$$

Example:

$$\cfrac{\overline{1} \quad \cfrac{\overline{2} \quad \cfrac{\cfrac{\overline{3}}{-3}}{(-3)}}{2 * (-3)}}{1 + (2 * (-3))} \cfrac{}{(2 * (-3))}$$

Propositional Logic

Examples:

- T, F
- $T \land F$
- $T \lor F$
- $(T \land F) \land (T \lor F)$
- $T \Rightarrow (F \Rightarrow T)$

Propositional Logic

Syntax:

$$
\begin{aligned}
f \quad \rightarrow \quad & T \mid F \\
\mid \quad & \neg f \\
\mid \quad & f \wedge f \\
\mid \quad & f \vee f \\
\mid \quad & f \Rightarrow f
\end{aligned}
$$

Semantics ($[\![f]\!]$):

$$
\begin{aligned}
[\![T]\!] \quad &= \quad true \\
[\![F]\!] \quad &= \quad false \\
[\![\neg f]\!] \quad &= \quad \text{not } [\![f]\!] \\
[\![f_1 \wedge f_2]\!] \quad &= \quad [\![f_1]\!] \text{ andalso } [\![f_2]\!] \\
[\![f_1 \vee f_2]\!] \quad &= \quad [\![f_1]\!] \text{ orelse } [\![f_2]\!] \\
[\![f_1 \Rightarrow f_2]\!] \quad &= \quad [\![f_1]\!] \text{ implies } [\![f_2]\!]
\end{aligned}
$$

Propositional Logic

$$\begin{aligned}
[\![(T \wedge (T \vee F)) \Rightarrow F]\!] &= [\![T \wedge (T \vee F)]\!] \text{ implies } [\![F]\!] \\
&= ([\![T]\!] \text{ andalso } [\![T \vee F]\!]) \text{ implies } false \\
&= (true \text{ andalso } ([\![T]\!] \text{ orelse } [\![F]\!])) \text{ implies } false \\
&= (true \text{ andalso } (true \text{ orelse } false)) \text{ implies } false \\
&= false
\end{aligned}$$

Structural Induction

A technique for proving properties about inductively defined sets.

> To prove that a proposition $P(s)$ is true for all structures s, prove the following:
>
> 1. (Base case) P is true on simple structures (those without substructures)
>
> 2. (Inductive case) If P is true on the substructures of s, then it is true on s itself. The assumption is called *induction hypothesis (I.H.)*.

Example 1

Let S be the set defined by the following inference rules:

$$\overline{3} \qquad \frac{x \quad y}{x+y}$$

Prove that for all $x \in S$, x is divisible by 3.
Proof. By structural induction.

- (Base case) The base case is when x is 3. Obviously, x is divisible by 3.

- (Inductive case) The induction hypothesis (I.H.) is

  $$x \text{ is divisible by } 3, \qquad y \text{ is divisible by } 3.$$

  Let $x = 3k_1$ and $y = 3k_2$. Using I.H., we derive

  $$x + y \text{ is divisible by } 3$$

  as follows:
  $$\begin{aligned} x + y &= 3k_1 + 3k_2 \quad \cdots \text{ by I.H.} \\ &= 3(k_1 + k_2) \end{aligned}$$

$\square$

Example 2

Let S be the set defined by the following inference rules:

$$\frac{}{()} \qquad \frac{x}{(x)} \qquad \frac{x \quad y}{xy}$$

Prove that every element of the set has the same number of ('s and)'s.
Proof Restate the claim formally:

$$\text{If } x \in S \text{ then } l(x) = r(x)$$

where $l(x)$ and $r(x)$ denote the number of ('s and)'s, respectively:

$$
\begin{array}{llll}
l(()) & = & 1 & \qquad r(()) & = & 1 \\
l((x)) & = & l(x) + 1 & \qquad r((x)) & = & r(x) + 1 \\
l(xy) & = & l(x) + l(y) & \qquad r(xy) & = & r(x) + r(y)
\end{array}
$$

We prove it by structural induction:

- (Base case): The base case is when $x = ()$. Then
  $l(x) = 1 = r(x)$.

Example 2

- (Inductive case): There are two inductive cases:

$$\frac{x}{(x)} \qquad \frac{x \quad y}{xy}$$

Induction hypotheses (I.H.):

$$l(x) = r(x), \qquad l(y) = r(y).$$

- ▶ The first case. We prove $l((x)) = r((x))$:

$$
\begin{aligned}
l((x)) &= l(x) + 1 & \cdots \text{by definition of } l((x)) \\
&= r(x) + 1 & \cdots \text{by I.H.} \\
&= r((x)) & \cdots \text{by definition of } r((x))
\end{aligned}
$$

- ▶ The second case. We prove $l(xy) = r(xy)$:

$$
\begin{aligned}
l(xy) &= l(x) + l(y) & \cdots \text{by definition of } l(xy) \\
&= r(x) + r(y) & \cdots \text{by I.H.} \\
&= r(xy) & \cdots \text{by definition of } r(xy)
\end{aligned}
$$

□

Example 3

Let T be the set of binary trees:

$$\frac{}{\textbf{leaf}} \qquad \frac{t_1 \quad t_2}{(n, t_1, t_2)} \; n \in \mathbb{Z}$$

Prove that for all such trees, the number of leaves is always one more than the number of internal nodes.

Proof. Restate the claim more formally:

$$\text{If } t \in T \text{ then } l(t) = i(t) + 1$$

where $l(t)$ and $i(t)$ denote the number of leaves and internal nodes, respectively:

$$
\begin{array}{llll}
l(\textbf{leaf}) &=& 1 & i(\textbf{leaf}) &=& 0 \\
l(n, t_1, t_2) &=& l(t_1) + l(t_2) & i(n, t_1, t_2) &=& i(t_1) + i(t_2) + 1
\end{array}
$$

We prove it by structural induction:

- (Base case): The base case is when $t = \textbf{leaf}$, where $l(t) = 1$ and $i(t) = 0$.

- (Inductive case): The induction hypothesis:

$$l(t_1) = i(t_1) + 1, \qquad l(t_2) = i(t_2) + 1$$

Using I.H., we prove $l((n, t_1, t_2)) = i((n, t_1, t_2)) + 1$:

$$
\begin{array}{llll}
l((n, t_1, t_2)) &=& l(t_1) + l(t_2) & \text{definition of of } l \\
&=& i(t_1) + 1 + i(t_2) + 1 & \text{by induction hypothesis} \\
&=& i(n, t_1, t_2) + 1 & \text{definition of } i
\end{array}
$$

Summary

- Computer science is full of inductive definitions.
  - primitive values: booleans, characters, integers, strings, etc
  - compound values: lists, trees, graphs, etc
  - language syntax and semantics
- Structural induction
  - a general technique for reasoning about inductively defined sets