Lecture 2 — Inductive Definitions (2)

CSE307: Programming Languages

Minseok Jeon

2026 Spring

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:

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

The inference rules can be expressed by a grammar:

$$\boldsymbol{n} \to 0 \mid \boldsymbol{n}+1$$

Interpretation:

- 0 is a natural number.
- If $\boldsymbol{n}$ is a natural number then so is $\boldsymbol{n}+1$.

Strings

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

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

or simply,

$$\frac{}{\boldsymbol{\epsilon}} \qquad \frac{\boldsymbol{\alpha}}{\boldsymbol{x}\boldsymbol{\alpha}}\ \boldsymbol{x} \in \{\texttt{a}, \dots, \texttt{z}\}$$

In grammar:

$$\begin{array}{rcll} \boldsymbol{\alpha} & \rightarrow & \boldsymbol{\epsilon} & \\ & \mid & \boldsymbol{x}\boldsymbol{\alpha} & (\boldsymbol{x} \in \{\texttt{a}, \dots, \texttt{z}\}) \end{array}$$

Boolean Values

The set of boolean values:

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

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

$$\frac{}{\mathit{true}} \qquad \frac{}{\mathit{false}}$$

In grammar:

$$\boldsymbol{b} \rightarrow \mathit{true} \mid \mathit{false}$$

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:

In grammar:

Prove that $-7 \cdot 3 \cdot 14 \cdot$ **nil** is a list of integers:

Lists

Examples of lists of integers:

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

Inference rules:

$$\frac{}{\mathbf{nil}} \qquad \frac{\boldsymbol{l}}{\boldsymbol{n} \cdot \boldsymbol{l}}\ \boldsymbol{n} \in \mathbb{Z}$$

In grammar:

$$\begin{array}{rcll} \boldsymbol{l} & \rightarrow & \mathbf{nil} & \\ & \mid & \boldsymbol{n} \cdot \boldsymbol{l} & (\boldsymbol{n} \in \mathbb{Z}) \end{array}$$

Lists

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

$$\dfrac{\dfrac{\dfrac{\overline{\textbf{nil}} \quad 14 \in \mathbb{Z}}{14 \cdot \textbf{nil}} \quad 3 \in \mathbb{Z}}{3 \cdot 14 \cdot \textbf{nil}} \quad -7 \in \mathbb{Z}}{-7 \cdot 3 \cdot 14 \cdot \textbf{nil}}$$

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

Binary Trees

Examples of binary trees:

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

Inference rules:

In grammar:

Prove that $(1, (2, \mathbf{leaf}, \mathbf{leaf}), (3, (4, \mathbf{leaf}, \mathbf{leaf}), \mathbf{leaf}))$ is a binary 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:

$$\begin{array}{lcl} t & \rightarrow & \textbf{leaf} \\ & \mid & (n, t, t) \quad (n \in \mathbb{Z}) \end{array}$$

Binary Trees

A proof that

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

is a binary tree:

$$\dfrac{\dfrac{\dfrac{}{\textbf{leaf}} \quad 2 \in \mathbb{Z}}{(2, \textbf{leaf}, \textbf{leaf})} \qquad \dfrac{\dfrac{\dfrac{}{\textbf{leaf}} \quad 4 \in \mathbb{Z}}{(4, \textbf{leaf}, \textbf{leaf})} \quad 3 \in \mathbb{Z}}{(3, (4, \textbf{leaf}, \textbf{leaf}), \textbf{leaf})} \quad 1 \in \mathbb{Z}}{(1, (2, \textbf{leaf}, \textbf{leaf}), (3, (4, \textbf{leaf}, \textbf{leaf}), \textbf{leaf}))}$$

Binary Trees: a different version

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

Inference rules:

In grammar:

Prove that $((1, 2), (3, \textbf{nil}))$ is a binary tree:

Binary Trees: a different version

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

Inference rules:

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

In grammar:

$$\begin{array}{rcll} t & \rightarrow & n & (n \in \mathbb{Z}) \\ & \mid & (t, \textbf{nil}) & \\ & \mid & (\textbf{nil}, t) & \\ & \mid & (t, t) & \end{array}$$

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

$$\dfrac{\dfrac{\dfrac{}{1} \quad \dfrac{}{2}}{(1, 2)} \qquad \dfrac{\dfrac{}{3}}{(3, \textbf{nil})}}{((1, 2), (3, \textbf{nil}))}$$

Expressions

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

Inference rules:

In grammar:

A proof that $1+(2*(-3))$ is an expression:

Expressions

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

Inference rules:

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

In grammar:

$$\begin{array}{rcll} e & \rightarrow & n & (n \in \mathbb{Z}) \\ & \mid & -e & \\ & \mid & e + e & \\ & \mid & e * e & \\ & \mid & (e) & \end{array}$$

A proof that $1+(2*(-3))$ is an expression:

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

Propositional Logic

Examples:

- $\boldsymbol{T}$, $\boldsymbol{F}$
- $\boldsymbol{T} \wedge \boldsymbol{F}$
- $\boldsymbol{T} \vee \boldsymbol{F}$
- $(\boldsymbol{T} \wedge \boldsymbol{F}) \wedge (\boldsymbol{T} \vee \boldsymbol{F})$
- $\boldsymbol{T} \Rightarrow (\boldsymbol{F} \Rightarrow \boldsymbol{T})$

Propositional Logic

Syntax:

$$f \quad \rightarrow \quad \boldsymbol{T} \mid \boldsymbol{F} \mid \neg f \mid f \wedge f \mid f \vee f \mid f \Rightarrow f$$

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

$$\begin{aligned}
[\![\boldsymbol{T}]\!] &= \mathit{true} \\
[\![\boldsymbol{F}]\!] &= \mathit{false} \\
[\![\neg f]\!] &= \begin{cases} \mathit{true} & \text{if } [\![f]\!] = \mathit{false} \\ \mathit{false} & \text{if } [\![f]\!] = \mathit{true} \end{cases} \\
[\![f_1 \wedge f_2]\!] &= \begin{cases} \mathit{true} & \text{if } [\![f_1]\!] = \mathit{true} \text{ and } [\![f_2]\!] = \mathit{true} \\ \mathit{false} & \text{otherwise} \end{cases} \\
[\![f_1 \vee f_2]\!] &= \begin{cases} \mathit{true} & \text{if } [\![f_1]\!] = \mathit{true} \text{ or } [\![f_2]\!] = \mathit{true} \\ \mathit{false} & \text{otherwise} \end{cases} \\
[\![f_1 \Rightarrow f_2]\!] &= \begin{cases} \mathit{false} & \text{if } [\![f_1]\!] = \mathit{true} \text{ and } [\![f_2]\!] = \mathit{false} \\ \mathit{true} & \text{otherwise} \end{cases}
\end{aligned}$$

Propositional Logic

What is the value of $[\![(\boldsymbol{T} \land (\boldsymbol{T} \lor \boldsymbol{F})) \Rightarrow \boldsymbol{F}]\!]$?

$$\begin{aligned}
[\![(\boldsymbol{T} \land (\boldsymbol{T} \lor \boldsymbol{F})) \Rightarrow \boldsymbol{F}]\!] &= \mathit{false} \quad \because [\![\boldsymbol{T} \land (\boldsymbol{T} \lor \boldsymbol{F})]\!] = \mathit{true} \text{ and } [\![\boldsymbol{F}]\!] = \mathit{false} \\
[\![\boldsymbol{T} \land (\boldsymbol{T} \lor \boldsymbol{F})]\!] &= \mathit{true} \quad \because [\![\boldsymbol{T}]\!] = \mathit{true} \text{ and } [\![\boldsymbol{T} \lor \boldsymbol{F}]\!] = \mathit{true} \\
[\![\boldsymbol{T} \lor \boldsymbol{F}]\!] &= \mathit{true} \quad \because [\![\boldsymbol{T}]\!] = \mathit{true}
\end{aligned}$$

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:

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

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:

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

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:

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

- (Base case):
- (Inductive case):

Example 2

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

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

$\square$

Example 3

Let $\boldsymbol{T}$ be the set of binary trees:

$$\frac{}{\textbf{leaf}} \qquad \frac{\boldsymbol{t}_1 \quad \boldsymbol{t}_2}{(\boldsymbol{n}, \boldsymbol{t}_1, \boldsymbol{t}_2)}\ \boldsymbol{n} \in \mathbb{Z}$$

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:

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

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

$$\begin{array}{rclcrcl} \boldsymbol{l}(\textbf{leaf}) & = & 1 & \qquad & \boldsymbol{i}(\textbf{leaf}) & = & 0 \\ \boldsymbol{l}(\boldsymbol{n}, \boldsymbol{t}_1, \boldsymbol{t}_2) & = & \boldsymbol{l}(\boldsymbol{t}_1) + \boldsymbol{l}(\boldsymbol{t}_2) & & \boldsymbol{i}(\boldsymbol{n}, \boldsymbol{t}_1, \boldsymbol{t}_2) & = & \boldsymbol{i}(\boldsymbol{t}_1) + \boldsymbol{i}(\boldsymbol{t}_2) + 1 \end{array}$$

- (Base Case):
- (Inductive Case):

Example 3

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:

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

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 $l((n, t_1, t_2)) = i((n, t_1, t_2)) + 1$:

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

□

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