Inductive Definitions with Inference Rules
Outline

Introduction

Specifying inductive definitions
   Inference rules in action
   Judgments, axioms, and rules

Reasoning about inductive definitions
   Direct proofs
   Admissibility
   Rule induction
What are inference rules?

**Inference rules** – a mathematical metalanguage

For specifying and formally reasoning about inductive definitions

**Inductive definition**

Recursively defines something in terms of itself

\[
\frac{\text{Human}(x) \to \text{Mortal}(x) \quad \text{Human}(x)}{\text{Mortal}(x)}
\]

premises

conclusion
Outline

Introduction

Specifying inductive definitions
  Inference rules in action
    Judgments, axioms, and rules

Reasoning about inductive definitions
  Direct proofs
  Admissibility
  Rule induction
Other metalanguages for inductive definitions

**Haskell data types**

```haskell
data Nat = Z | S Nat
data Exp = Add Exp Exp
       | Neg Exp
       | Lit Nat
```

**Recursive functions in Haskell**

```haskell
even :: Nat -> Bool
even Z = True
even (S Z) = False
even (S (S n)) = even n
```

**Grammars**

```
n ∈ Nat ::= Z | S n
e ∈ Exp ::= add e e
   | neg e
   | n
```

Can also define all of these with inference rules!
Example: defining syntax by inference rules

**Grammars**

\[ n \in \text{Nat} \quad ::= \quad Z \mid S \ n \]

\[ e \in \text{Exp} \quad ::= \quad \text{add} \ e \ e \]

\( \mid \)  \quad \text{neg} \ e

\( \mid \)  \quad n

**Rule schema**

\[ Z \in \text{Nat} \]

\[ n \in \text{Nat} \]

\[ S \ n \in \text{Nat} \]

**Axiom**

(no premises)

\[ n \in \text{Nat} \]

\[ e \in \text{Exp} \]

\[ \text{neg} \ e \in \text{Exp} \]

\[ e_1 \in \text{Exp} \quad e_2 \in \text{Exp} \]

\[ \text{add} \ e_1 \ e_2 \in \text{Exp} \]
Example: defining a predicate

Recursive function in Haskell

```haskell
even :: Nat -> Bool
even Z = True
even (S Z) = False
even (S (S n)) = even n
```

Option 1: Constructive judgment

\[
\begin{align*}
\text{Even}(Z) & \quad \text{Even}(n) \\
\downarrow & \quad \downarrow \\
\text{Even}(S (S n)) & \\
\end{align*}
\]

Option 2: Relate inputs to outputs

\[
\begin{align*}
\text{Even}(Z, \text{true}) & \quad \text{Even}(S Z, \text{false}) \\
\downarrow & \quad \downarrow \\
\text{Even}(n, b) & \quad \text{Even}(S (S n), b) \\
\end{align*}
\]
Outline

Introduction

Specifying inductive definitions
  Inference rules in action
  Judgments, axioms, and rules

Reasoning about inductive definitions
  Direct proofs
  Admissibility
  Rule induction
The structure of a definition

How to define a “concept” in three parts:

1. **syntax** – how to express the concept
2. **type** – what kind of information is it?
3. **content** – the definition itself

Example: dictionary definition

Syntax: `even`  
Type: adjective  
Content: (of a number) divisible by two without a remainder

Example: function definition

```
even :: Nat -> Bool
even Z = True
even (S Z) = False
even (S (S n)) = even n
```
How to define a concept using inference rules

1. Define a **judgment form** – syntax and type
   States that one or more values have some **property**
   or exist in some **relation** to each other

2. Write down the **rules** for the judgment – content
   - **axioms** – base cases, only conclusion
   - **proper rules** – recursive cases, premises + conclusion
# Judgments

1. Define a **judgment form** – syntax and type

States that one or more values have some property or exist in some relation to each other

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Type</th>
<th>Property or relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n \in \text{Nat}$</td>
<td>AST</td>
<td>$n$ is in the syntactic category $\text{Nat}$</td>
</tr>
<tr>
<td>Even($n$)</td>
<td>$\text{Nat}$</td>
<td>$n$ is an even number</td>
</tr>
<tr>
<td>$n_1 &lt; n_2$</td>
<td>$\text{Nat} \times \text{Nat}$</td>
<td>$n_1$ is less than $n_2$</td>
</tr>
<tr>
<td>$e : T$</td>
<td>$\text{Exp} \times \text{Type}$</td>
<td>$e$ has type $T$</td>
</tr>
<tr>
<td>$\Gamma \vdash e : T$</td>
<td>$\text{Env} \times \text{Exp} \times \text{Type}$</td>
<td>$e$ has type $T$ in environment $\Gamma$</td>
</tr>
</tbody>
</table>
A judgment is (conceptually) a **predicate** that indicates set membership.

**Example:** \( \text{Even}(n) \subseteq \text{Nat} \)

\[
\text{Even} : \text{Nat} \to \mathbb{B} \\
= \{ (\mathbb{Z}, \text{true}), (\mathbb{S} \mathbb{Z}, \text{false}), (\mathbb{S} (\mathbb{S} \mathbb{Z}), \text{true}), \ldots \} \\
\equiv \{ \mathbb{Z}, \mathbb{S} (\mathbb{S} \mathbb{Z}), \mathbb{S} (\mathbb{S} (\mathbb{S} (\mathbb{S} \mathbb{Z}))), \ldots \} \subseteq \text{Nat}
\]

**Example:** \( n_1 < n_2 \subseteq \text{Nat} \times \text{Nat} \)

\[
< : \text{Nat} \times \text{Nat} \to \mathbb{B} \\
= \{ ((\mathbb{0}, \mathbb{0}), \text{false}), ((\mathbb{0}, \mathbb{1}), \text{true}), \ldots ((\mathbb{5}, \mathbb{3}), \text{false}), \ldots ((\mathbb{5}, \mathbb{7}), \text{true}), \ldots \} \\
\equiv \{ (\mathbb{0}, \mathbb{1}), \ldots (\mathbb{5}, \mathbb{7}), \ldots \} \subseteq \text{Nat} \times \text{Nat}
\]
2. Write down the **rules** of the judgment – content

- **axioms** – base cases, only conclusion
- **proper rules** – recursive cases, premises + conclusion

Inductively defines the **instances** of a judgment (i.e. members of its set)

---

### Rules for: $\text{Even}(n) \subseteq \text{Nat}$

<table>
<thead>
<tr>
<th>Rule</th>
<th>Precondition</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Even}(Z)$</td>
<td>$\text{Even}(n)$</td>
<td>$\text{Even}(S \ (S \ n))$</td>
</tr>
</tbody>
</table>

### Rules for: $n_1 < n_2 \subseteq \text{Nat} \times \text{Nat}$

<table>
<thead>
<tr>
<th>Rule</th>
<th>Precondition</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z &lt; S \ Z$</td>
<td>$n_1 &lt; n_2$</td>
<td>$n_1 &lt; S \ n_2$</td>
</tr>
<tr>
<td>$n_1 &lt; S \ n_2$</td>
<td>$S \ n_1 &lt; S \ n_2$</td>
<td></td>
</tr>
</tbody>
</table>
Exercises

1. Define the judgment: \( \text{Odd}(n) \subseteq \text{Nat} \)

2. Define the judgment: \( n_1 + n_2 = n_3 \subseteq \text{Nat} \times \text{Nat} \times \text{Nat} \)

For reference:

Rules for: \( \text{Even}(n) \subseteq \text{Nat} \)

\[
\begin{array}{c}
\text{Even}(\mathbb{Z})
\end{array}
\]

\[
\begin{array}{c}
\text{Even}(n) \\
\hline
\text{Even}(S(Sn))
\end{array}
\]

Rules for: \( n_1 < n_2 \subseteq \text{Nat} \times \text{Nat} \)

\[
\begin{array}{c}
\text{Z} < S\text{Z}
\end{array}
\]

\[
\begin{array}{c}
\frac{n_1 < n_2}{n_1 < S\text{ }n_2}
\end{array}
\]

\[
\begin{array}{c}
\frac{n_1 < n_2}{S\text{ }n_1 < S\text{ }n_2}
\end{array}
\]
Outline

Introduction

Specifying inductive definitions
- Inference rules in action
- Judgments, axioms, and rules

Reasoning about inductive definitions
- Direct proofs
- Admissibility
- Rule induction
Expressing claims

We can also use inference rules to express **claims** about judgments

**Examples**

<table>
<thead>
<tr>
<th>S (S Z) ∈ Nat</th>
<th>Even(S n)</th>
<th>n₁ &lt; n₂</th>
<th>n₂ &lt; n₃</th>
<th>n₁ + n₂ = n₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>Odd(n)</td>
<td></td>
<td>n₁ &lt; n₃</td>
<td></td>
<td>n₂ + n₁ = n₃</td>
</tr>
</tbody>
</table>

How can we **prove** these claims?

Three main techniques:

1. **direct proof** – derive conclusion from premises using the definition
2. **admissibility** – derive conclusion from derivations of premises
3. **rule induction** – reason inductively using the definition
Introduction

Specifying inductive definitions
  Inference rules in action
  Judgments, axioms, and rules

Reasoning about inductive definitions
  Direct proofs
  Admissibility
  Rule induction
Direct proof by derivation

**Definition:** \( n \in \text{Nat} \)

- \( Z \in \text{Nat} \)
- \( \text{Succ} \quad n \in \text{Nat} \rightarrow Sn \in \text{Nat} \)
- \( Z \in \text{Nat} \)
- \( \text{Succ} \quad S Z \in \text{Nat} \)
- \( \text{Succ} \quad S (S Z) \in \text{Nat} \)

**Definition:** \( n_1 < n_2 \subseteq \text{Nat} \times \text{Nat} \)

- \( Z < S Z \)
  - \( S \quad n_1 < n_2 \rightarrow n_1 < S n_2 \)
  - \( +1 \quad n_1 < n_2 \rightarrow Sn_1 < S n_2 \)
- \( S Z < S (S Z) \)
  - \( S \quad Z < S Z \rightarrow S Z < S (S Z) \)
  - \( +1 \quad Z < S (S Z) \rightarrow SZ < S (S (S Z)) \)
Proof trees

Definition: \( e \in \text{Exp} \)

Axioms: \( 0 \in \text{Nat}, \; 1 \in \text{Nat}, \; 2 \in \text{Nat}, \; \ldots \)

\[
\begin{align*}
\text{lit} & \quad n \in \text{Nat} \quad \quad \text{lit} & \quad \text{lit} 2 \in \text{Nat} \\
\text{neg} & \quad e \in \text{Exp} \quad \quad \text{neg} & \quad \text{lit} 4 \in \text{Nat} \\
\text{add} & \quad e_1 \in \text{Exp} \quad \quad \text{add} & \quad \text{lit} 3 \in \text{Exp} \\
\text{neg} & \quad \text{neg} e \in \text{Exp} \quad \quad \text{add} & \quad \text{add} e_1 e_2 \in \text{Exp} \\
\text{add} & \quad \text{add} 2 3 \in \text{Exp} \quad \quad \text{neg} & \quad \text{neg} 4 \in \text{Exp} \\
\text{add} & \quad \text{add} (\text{add} 2 3) (\text{neg} 4) \in \text{Exp}
\end{align*}
\]
Exercises

Prove that the following expressions are valid terms in $Exp$

1. $\text{neg} \ (\text{add} \ 5 \ (\text{neg} \ 2))$
2. $\text{add} \ (\text{neg} \ (\text{neg} \ 3)) \ 4$

Definition: $e \in Exp$

Axioms: $0 \in \text{Nat}$, $1 \in \text{Nat}$, $2 \in \text{Nat}$, ...
Outline

Introduction

Specifying inductive definitions
- Inference rules in action
- Judgments, axioms, and rules

Reasoning about inductive definitions
- Direct proofs
- Admissibility
- Rule induction
Admissibility

Construct proofs from assumed *derivations of the premises*

**Insights:**
- If the premise of a claim is satisfied, it must have a derivation
- Can use information in the derivations to prove the conclusion

**Proof technique**
Show that all possible derivations of premises yield a proof of the conclusion

Apply definition rules backwards on the premises, prove for each case!
Super simple example

Definition:  \( n \in \text{Nat} \subseteq \text{AST} \)

- \( Z \in \text{Nat} \)
- \( \text{Succ} \quad n \in \text{Nat} \quad S \ n \in \text{Nat} \)

Bold claim

\[
\begin{align*}
S \ (S \ n) & \in \text{Nat} \\
n & \in \text{Nat}
\end{align*}
\]

Proof sketch:
- Enumerate derivations of premise
- Show that each derivation proves the conclusion

Only possible derivation

\[
\begin{align*}
\text{Succ} \quad n & \in \text{Nat} \\
\text{Succ} \quad S \ n & \in \text{Nat} \\
S \ (S \ n) & \in \text{Nat}
\end{align*}
\]
Outline

Introduction

Specifying inductive definitions
  Inference rules in action
  Judgments, axioms, and rules

Reasoning about inductive definitions
  Direct proofs
  Admissibility
  Rule induction
Rule induction

Just like structural induction on inductive data types!

Definition: $e \in \text{Exp} \subseteq \text{AST}$

- $n \in \text{Nat} \Rightarrow n \in \text{Exp}$
- $e \in \text{Exp} \Rightarrow \text{neg} e \in \text{Exp}$
- $e_1 \in \text{Exp}$, $e_2 \in \text{Exp} \Rightarrow \text{add } e_1 e_2 \in \text{Exp}$

Suppose I want to prove property $P$ on all $\text{Exp}$. Just prove:

- $\forall n \in \text{Nat}, P(n)$
- $P(e) \rightarrow P(\text{neg } e)$
- $P(e_1) \rightarrow P(e_2) \rightarrow P(\text{add } e_1 e_2)$