

Inductive Definitions

Inductive definitions play a **central role** in the study of programming languages.

These aspects of a language are specified by inductive definitions:

- Its concrete syntax, using **context-free grammars**;
- Its abstract syntax, using **abstract syntax trees**;
- Its static semantics, using **typing rules**;
- Its dynamic semantics, using **evaluation rules**.

Inductive Definitions

An inductive definition consists of

- a set of **rules** for deriving . . .
- one or more **judgements**, or **assertions**.

For example, rules for deriving the judgement $n \text{ nat}$:

1. zero nat ;
2. if $n \text{ nat}$, then $\text{succ}(n) \text{ nat}$.

Inference Rule Notation

Inference rules are normally written in the form

$$\frac{J_1 \ \dots \ J_n}{J}$$

where J and J_1, \dots, J_n are judgements. (For axioms, $n = 0$.)

For example, the rules for deriving $a \text{ nat}$ are usually written:

$$\frac{}{\text{zero nat}} \quad \frac{a \text{ nat}}{\text{succ}(a) \text{ nat}}$$

Use of the meta-variable a in the second rule makes it a **rule scheme**, standing for the collection of all its **instances**.

Derivation of Judgements

A judgement J is **derivable** iff either

1. there is an axiom

$$\frac{}{J}$$

or

2. there is a rule

$$\frac{J_1 \ \dots \ J_n}{J}$$

such that each of the judgements J_1, \dots, J_n is derivable.

(The second condition subsumes the first, by taking $n = 0$.)

Derivation of Judgements

We may determine whether a judgement is derivable by working backward (or forward; top-down or bottom-up).

For example, the judgement $\text{succ}(\text{succ}(\text{zero})) \text{ nat}$ is derivable:

$$\frac{\frac{\text{zero nat}}{\text{succ}(\text{zero}) \text{ nat}}}{\text{succ}(\text{succ}(\text{zero})) \text{ nat}}$$

Such a composition of rules is called a **derivation**.

Binary Trees

Here is a set of rules (rule scheme) defining the judgement a tree stating that a is a binary tree:

$$\frac{}{\text{empty tree}} \quad \frac{a_1 \text{ tree} \quad a_2 \text{ tree}}{\text{node}(a_1; a_2) \text{ tree}}$$

It's easy to check that $\text{node}(\text{empty}; \text{node}(\text{empty}; \text{empty}))$ tree is derivable according to these rules.

6

Rule Induction

By definition, every derivable judgement

- is the consequence of some rule . . .
- whose premises are derivable.

That is, the rules are an **exhaustive** (i.e., necessary and sufficient) description of the derivable judgements.

7

Rule Induction

Thus to show that every derivable judgement has a property \mathcal{P} , it is enough to show that

if J_1, \dots, J_n have property \mathcal{P} , then J has property \mathcal{P} .

for every rule

$$\frac{J_1 \quad \dots \quad J_n}{J}$$

This is the principle of **rule induction**.

8

Example: Natural Numbers

Consider the rules defining the judgement a nat.

We can prove that the property \mathcal{P} holds of every a such that a nat by rule induction.

- Show that \mathcal{P} holds of zero nat;
- Assuming that \mathcal{P} holds of a nat, show that \mathcal{P} holds of $\text{succ}(a)$ nat.

This is just ordinary **mathematical induction**!

9

Example: Binary Trees

Similarly, we can prove that every binary tree a has a property \mathcal{P} by showing that

- empty tree has property \mathcal{P} ;
- if a_1 tree has property \mathcal{P} and a_2 tree has property \mathcal{P} , then $\text{node}(a_1; a_2)$ tree has property \mathcal{P} .

This is called **tree induction**.

10

Example: The Height of a Tree

Consider the following equations:

$$\begin{aligned} \text{hgt}(\text{empty}) &= 0 \\ \text{hgt}(\text{node}(a_1; a_2)) &= 1 + \max(\text{hgt}(a_1); \text{hgt}(a_2)) \end{aligned}$$

Claim: for every binary tree a there is a unique natural number b such that $\text{hgt}(a) = b$.

That is, the above equations define a **function**.

Or, the corresponding judgement form $\text{hgt}(a; b)$ has **mode** $(\forall, \exists!)$.

11

Example: The Height of a Tree

Here is a set of rules defining the judgement form $\text{hgt}(a; b)$:

$$\frac{}{\text{hgt}(\text{empty}; \text{zero})} \quad \frac{\text{hgt}(a_1; b_1) \quad \text{hgt}(a_2; b_2) \quad \max(b_1, b_2; b)}{\text{hgt}(\text{node}(a_1; a_2); \text{succ}(b))}$$

12

Example: The Height of a Tree

We will prove the claim by rule induction:

- If a is derivable by the axiom

$$\frac{}{\text{empty tree}}$$

then $b = 0$ is uniquely determined by the first rule ($\text{hgt}(\text{empty}; \text{zero})$)

- ...

13

Example: The Height of a Tree

Continuing, ...

- If a is derivable by the rule

$$\frac{a_1 \text{ tree} \quad a_2 \text{ tree}}{\text{node}(a_1; a_2) \text{ tree}},$$

then we may assume that

1. there exists a unique b_1 such that $\text{hgt}(a_1; b_1)$;
2. there exists a unique b_2 such that $\text{hgt}(a_2; b_2)$;

Hence it follows that there exists a unique b , namely $1 + \max(n_1, n_2)$, such that $\text{hgt}(a) = b$ (i.e., $\text{hgt}(\text{node}(a_1; a_2); \text{succ}(b))$).

14

Example: The Height of a Tree

This is awfully pedantic, but it is useful to see the details at least once.

- It is not obvious **a priori** that a tree has a well-defined height!
- Rule induction justifies the existence of the function hgt , i.e., that judgement form $\text{hgt}(a; b)$ has **mode** $(\forall, \exists!)$

It is obvious from the equations that there is **at most one** b such that $\text{hgt}(a) = b$. The proof shows that there exists **at least one**.

15

Simultaneous Inductive Definitions

We often define several judgements simultaneously by a set of (mutually recursive) rules.

For example, these rules define the **variadic trees**:

$$\frac{f \text{ forest}}{\text{node}(f) \text{ vtree}} \quad \frac{}{\text{nil forest}} \quad \frac{t \text{ vtree} \quad f \text{ forest}}{\text{cons}(t; f) \text{ forest}}$$

(Variadic = variable number of children at any node.)

16

Simultaneous Rule Induction

We may show that every vtree t has property \mathcal{P} and every forest f has property \mathcal{P} by showing

- $\text{node}(f)$ has property \mathcal{P} , assuming that f has property \mathcal{P} .
- nil has property \mathcal{P} .
- $\text{cons}(t; f)$ has property \mathcal{P} , assuming that t has property \mathcal{P} and f has property \mathcal{P} .

17

Hypothetical Judgements

One form of **hypothetical judgement** expresses the **derivability** of a conclusion from a set of assumptions (hypotheses). For instance:

$$\Gamma \vdash A$$

says that

1. given the set of **assumptions**, or **hypotheses**, Γ
2. the **conclusion** judgment A holds

For example:

$$n \text{ nat} \vdash \text{succ}(\text{succ}(n)) \text{ nat}$$

18

Hypothetical Judgements

A valid derivability judgement $\Gamma \vdash A$ corresponds to a derivation, using some set of rules, of judgement A when Γ is taken as an axiom.

For example, $n \text{ nat} \vdash \text{succ}(\text{succ}(n)) \text{ nat}$ is valid under the rules for the nat judgement form, since taking $n \text{ nat}$ as an axiom allows us to derive:

$$\frac{\frac{n \text{ nat}}{\text{succ}(n) \text{ nat}}}{\text{succ}(\text{succ}(n)) \text{ nat}}$$

In other words, the following rule is derivable:

$$\frac{n \text{ nat}}{\text{succ}(\text{succ}(n)) \text{ nat}}$$

19

Structural Properties of Derivability

- Reflexivity: $\overline{\Gamma, A \vdash A}$
- Weakening: $\frac{\Gamma \vdash A}{\Gamma, A' \vdash A}$
- Transitivity: $\frac{\Gamma \vdash A' \quad \Gamma, A' \vdash A}{\Gamma \vdash A}$

Also Exchange and Contraction, allowing us to treat Γ as a finite set.

20

Conditional Rules

Note the use of derivability judgements in the premises of the rules on the previous slide, making them **conditional rules**.

A **conditional inductive definition** consists of a set of conditional rules.

The hypotheses in the premises are **local hypotheses**, local to a specific judgement in the premises.

21

Hypothetical Judgements

Another form of **hypothetical judgement** expresses the **admissibility** of a conclusion from a set of assumptions (hypotheses).

Admissibility is strictly weaker than derivability.

The same structural properties hold for admissibility as for derivability.

Derivability **is** stable under addition of rules to the rule set.

Admissibility **is not** stable under addition of rules to the rule set.

22

Parametric Judgements

A **parametric** judgement expands the universe of objects through the addition of a finite set of "names" for unspecified objects. For instance:

$$\mathcal{X} \mid A$$

says that

1. given the finite set of **parameters** \mathcal{X}
2. the judgment A holds **uniformly** or **parametrically**

For example:

$$n \mid n \text{ nat} \vdash \text{succ}(\text{succ}(n)) \text{ nat}$$

23

Structural Properties of Parametric Judgements

- Proliferation: $\frac{\mathcal{X}|A}{\overline{\mathcal{X}}, x|A}$ assuming $x \notin \mathcal{X}$
- Swapping: $\frac{\mathcal{X}_1, x_2, x_1, \mathcal{X}_2|A}{\overline{\mathcal{X}_1}, x_1, x_2, \overline{\mathcal{X}_2}|A}$
- Duplication: $\frac{\mathcal{X}, x, x|A}{\overline{\mathcal{X}}, x|A}$
- Renaming: $\frac{\mathcal{X}, y|A_y}{\overline{\mathcal{X}}, x|A_x}$ provided $y \notin \mathcal{X}$

24

Transition Systems

A **transition system** is specified by the following:

- Set of states S , corresponding to the judgement s state.
- Initial states $I \subseteq S$, corresponding to the judgement s initial.
- Final states $F \subseteq S$, corresponding to the judgement s final.
- Relation $\mapsto \subseteq S \times S$ corresponding to the judgement $s \mapsto s'$ where s state and s' state.

25

Stuck States

A state s for which there exists no s' such that $s \mapsto s'$ is said to be **stuck**.

We write:

$$s \not\mapsto$$

All final states are stuck, but not all stuck states are final.

26

Iterated Transition

The relation $s \mapsto^* s'$ is inductively defined by the following rules:

$$\frac{}{s \mapsto^* s} \quad \frac{s \mapsto s' \quad s' \mapsto^* s''}{s \mapsto^* s''}$$

That is, $s \mapsto^* s'$ iff $s = s_0 \mapsto s_1 \mapsto \dots \mapsto s_n = s'$ for some $n \geq 0$.

27

Induction on Iterated Transition

Since iterated transition is inductively defined, there is an associated principle of induction, called **induction on iterated transition**.

To show that $s \mapsto^* s'$ implies $P(s, s')$, it suffices to show

1. $P(s, s)$, i.e. that P is **reflexive**.
2. If $s \mapsto s' \mapsto^* s''$ and $P(s', s'')$, then $P(s, s'')$. This is called **closure under head expansion** or **closure under converse evaluation**.

28

Summary

An inductive definition consists of a set of rules defining the derivability of one or more judgements.

A judgement is derivable iff it may be obtained by composing rules.

Rule induction allows us to show that every derivable judgement has a specified property by considering each rule in turn.

29