



INDUCTIVE DEFINITION

1

OUTLINE

- Judgements
- Inference Rules
- Inductive Definition
- Derivation
- Rule Induction

LANGUAGE AND META-LANGUAGE

- Language is the target programming language, e.g., Java, Python, ML.
 - Has its own identifiers, variables, etc.
- Meta-language is the language in which to describe the target language.

META-VARIABLES

- A symbol in a meta-language that is used to describe some element in an object (target) language
 - E.g., Let **a** and **b** be two sentences of a language \mathcal{L}
 - E.g., Let **n** be a number, **d** be a digit and **s** be a sign in the language of numerals
 - 435, 535.23, -3847 are all numbers in the language of numerals
 - meta-variable doesn't appear in the language itself.
- Meta- is a prefix used to indicate a concept, which is an abstraction from another concept, used to complete or add to the latter.
- Similar use in “meta-data”, “meta-theory”, etc.
 - The syntax, semantics, etc. about a PL (e.g., Java) is the meta-theory about that language

JUDGEMENTS

- A *judgement* is an *assertion* (in the meta-language) about one or more syntactic objects.

Judgement

n nat

$n = n_1 + n_2$

τ type

$e:\tau$

$e \Downarrow v$

Meaning

(n is a natural number)

(n is the sum of n_1 and n_2)

(τ is a type)

(expression e has type τ)

(expression e has value v)

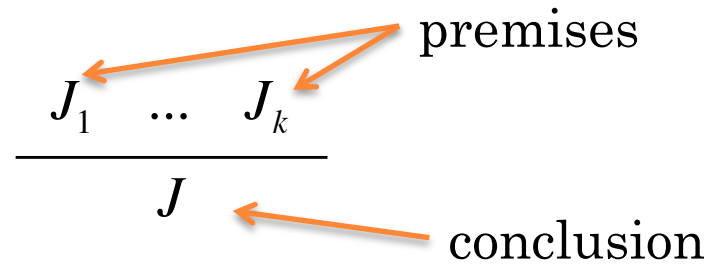
- “ n nat” can also be written as “ n isa nat”, “ n is a natural num”, etc. as long as it’s consistent.

JUDGEMENTS (II)

- A judgement states one or more syntactic objects have a property or have a relation among one another.
- The property or the relation itself is called *predicate*.
 - E.g., $n \text{ nat}$ (this judgement involves one object n)
- The abstract structure (schema) of a judgement is called *judgement form*.
 - E.g. $n \text{ nat}$.
- The judgement that a particular object or objects having that property is an *instance* of a judgement form.
 - E.g., 5 nat , $\text{succ}(n) \text{ nat}$ are all judgements
- W.L.O.G., we use “judgement” to mean the instance of judgement form usually.

INFERENCE RULES

- An inductive definition of a judgement form consists of a collection of rules of the form:



- To show J , it is sufficient to show J_1, \dots, J_k .
- A rule without premises is called an *axiom*;
- Otherwise it's called a *proper rule*.

INDUCTIVE DEFINITION

- Definition of judgement form $n \text{ nat}$:

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

- Definition of judgement form $t \text{ tree}$:

$$\frac{}{\text{empty tree}} \qquad \frac{t_1 \text{ tree} \quad t_2 \text{ tree}}{\text{node}(t_1; t_2) \text{ tree}}$$

Axioms!

Proper Rules!

DERIVATION

- To show an inductively defined judgement holds \rightarrow exhibit a derivation of the judgement.
- A derivation is an *evidence* for the validity of the defined judgement.
- Derivation of a judgement is the finite composition of rules starting from *axioms* and ending at *that judgement*.
- Usually a tree structure
 - In compiler, derivation of grammar in the form of a *parse tree*.

DERIVATION (II)

- Derivation of judgement $\text{succ}(\text{succ}(\text{succ}(\text{zero}))) \text{ nat}$:

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

- Derivation of $\text{node}(\text{node}(\text{empty}, \text{empty}), \text{empty}) \text{ tree}$:

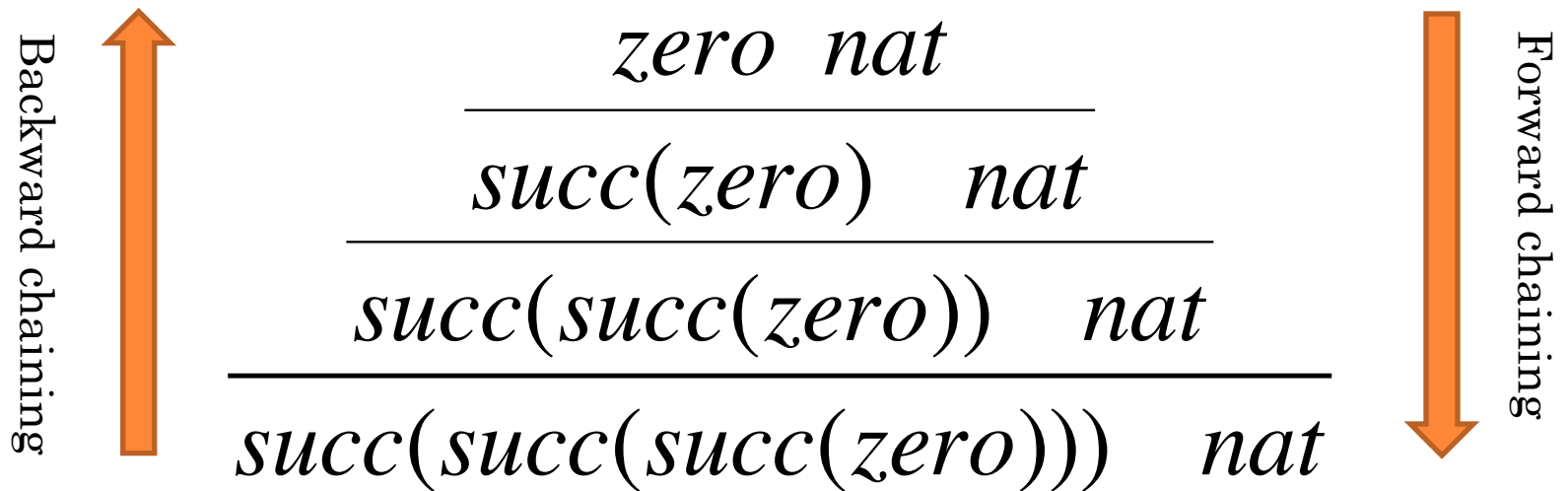
$$\frac{\frac{\frac{\overline{\text{empty tree}}}{\text{node}(\text{empty}; \text{empty}) \text{ tree}} \quad \frac{\overline{\text{empty tree}}}{\text{empty tree}}}{\text{node}(\text{node}(\text{empty}; \text{empty}); \text{empty}) \text{ tree}}$$

TYPES OF DERIVATION

- Forward chaining (bottom-up):
 - Starting from axioms, work up to the conclusion
- Backward chaining (top-down):
 - Start from the conclusion, work backwards toward axioms
- Note the terms bottom-up and top-down are exactly the opposite of the derivation tree we presented.

TYPE OF DERIVATION

- Derivation of judgement $\text{succ}(\text{succ}(\text{succ}(\text{zero}))) \text{ nat}$:



DEDUCTIVE SYSTEMS

- A deductive system has 2 parts:
 - Definition of a judgement form or a collection of judgement forms
 - A collection of inference rules about these judgement forms
- We have just introduced two deductive systems: nat and tree.
- A *programming language* can be represented by a deductive system, of course with many judgement forms!

RULE INDUCTION (I)

- Reason about rules under an inductive definition (or within a deductive system)
- Principle of rule induction:
 - To show property P holds of a judgement form J whenever J is derivable, it is enough to show that P is *closed under*, or *respects*, the rules defining J .
 - Write $P(J)$ to mean property P holds for J .
 - P respects the rule

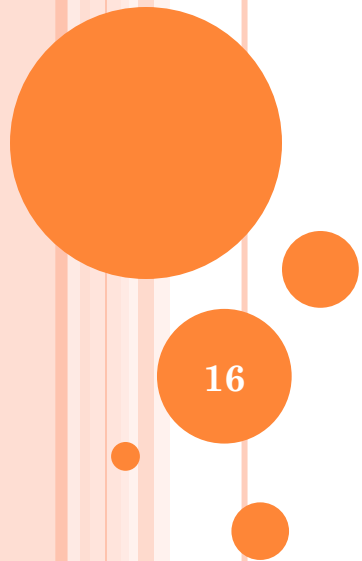
$$\frac{J_1 \quad \dots \quad J_k}{J}$$

if $P(J)$ holds whenever $P(J_1), \dots, P(J_k)$ hold.

- $P(J_1), \dots, P(J_k)$ are *inductive hypothesis*.
- $P(J)$ is *inductive conclusion*.

RULE INDUCTION (II)

- For the judgement $n \text{ nat}$, to show $P(n \text{ nat})$, it is sufficient to show:
 1. $P(\text{zero nat})$.
 2. For every n , if $P(n \text{ nat})$, then $P(\text{succ}(n) \text{ nat})$.
- Looks familiar?
- This is just a generalized version of *mathematical induction*.
- Step 1 is called the basis; step 2 is called the induction step.
- Similar induction can be applied on $\text{node}(t_1, t_2)$ tree \rightarrow “*tree induction*”.



16

PROOF BY INDUCTION

OUTLINE

- Proof Principles
- Natural Numbers
- List
- Proof Structure

PROOF PRINCIPLE (RULE INDUCTION)

- Recall that...
- To show every derivable judgement has some property P, show for every rule in the deductive system:

$$\frac{J_1 \dots J_n}{J} [name]$$

- If J_1, \dots, J_n have property P then J has property P.

EXAMPLE (NATURAL NUMBERS)

- Given a property P , we know that P is true for all natural numbers, if we can prove:
 - P holds unconditionally for Z . Corresponds to rule Z :

$$\frac{}{Z \text{ nat}} Z$$

- Assuming P holds for n , then P holds for $(S \ n)$. Corresponds to rule S :

$$\frac{n \text{ nat}}{S \ n \text{ nat}} S$$

- Also called “induction on the structure of natural numbers”.

NATURAL NUMBERS

- Natural numbers:

$$\frac{}{Z \text{ nat}} Z$$

$$\frac{n \text{ nat}}{S n \text{ nat}} S$$

- Addition:

- Judgement: $\text{add } n1 \ n2 \ n3$

$$\frac{}{\text{add } Z \ n \ n} \text{addZ}$$

$$\frac{\text{add } n1 \ n2 \ n3}{\text{add } (S \ n1) \ n2 \ (S \ n3)} \text{addS}$$

Theorem 1: For all n_1, n_2 , there exists n_3 such that $\text{add } n_1 \ n_2 \ n_3$.
(if $n_1 \text{ nat}$, $n_2 \text{ nat}$, then there exists $n_3 \text{ nat}$ such that $\text{add } n_1 \ n_2 \ n_3$)

Proof: **By induction on the derivation of $n \text{ nat}$.**

Case: $\frac{}{Z \text{ nat}} Z$

Need to prove $\text{add } n_1 \ n_2 \ n_3$ where $n_1 = Z$

- (1) $\text{add } Z \ n_2 \ n_2$ (by addZ , and let $n=n_2$)
- (2) $\text{add } n_1 \ n_2 \ n_3$ (by letting $n_1=Z$, $n_3=n_2$)

Renaming!

(Case proved)

Case: $\frac{n \text{ nat}}{S \ n \text{ nat}} S$

Need to prove $\text{add } n_1 \ n_2 \ n_3$ where $n_1 = (S \ n)$

- (1) $\text{add } n \ n_2 \ n_3'$ (by I.H. and let $n = n_1$, $n_3'=n_3$)
- (2) $\text{add } (S \ n) \ n_2 \ (S \ n_3')$ (by (1), addS , and
let $(S \ n) = n_1$, $(S \ n_3')=n_3$)

(Case proved) QED.

$$\frac{\frac{}{\text{add } Z \ n \ n} \text{addZ}}{\text{add } n_1 \ n_2 \ n_3} \text{addS}$$

EVEN/ODD NUMBERS

○ Judgements:

- $\text{even } n$ “ n is an even number”
- $\text{odd } n$ “ n is an odd number”

$$\frac{}{\text{even } Z} \text{even}Z$$

$$\frac{\text{odd } n}{\text{even } (S n)} \text{even}S$$

$$\frac{\text{even } n}{\text{odd } (S n)} \text{odd}S$$

Theorem 2: If n nat, then either even n or odd n .

Proof: **By induction on the derivation of n nat.**

Case: $\frac{}{Z \text{ nat}} Z$

even Z (By rule evenZ)

Case: $\frac{n \text{ nat}}{S n \text{ nat}} S$

(1) even n or (2) odd n (By I.H.)

Need to prove: even $(S n)$ or odd $(S n)$

Assuming (1):

odd $(S n)$ (By (1) and rule oddS)

Assuming (2):

even $(S n)$ (By (2) and rule evenS)

QED.

$$\frac{}{\text{even } Z} \text{evenZ}$$

$$\frac{\text{odd } n}{\text{even } (S n)} \text{evenS}$$

$$\frac{\text{even } n}{\text{odd } (S n)} \text{oddS}$$

EVEN/ODD NUMBER (ALT. DEFINITION)

$$\frac{}{\text{even2 } Z} \text{even2Z}$$

$$\frac{\text{even2 } n}{\text{even2 } (S (S n))} \text{even2S}$$

$$\frac{}{\text{odd2 } (S Z)} \text{odd2Z}$$

$$\frac{\text{odd2 } n}{\text{odd2 } (S (S n))} \text{odd2S}$$

Theorem 3: If $\text{even2 } n$, then $\text{even } n$.

Proof: By induction on the derivation of $\text{even2 } n$.

Case:
$$\frac{}{\text{even2 } Z} \text{even2}Z$$

$\text{even } Z$ (by rule $\text{even}Z$)

Case:
$$\frac{\text{even2 } n}{\text{even2 } (S (S n))} \text{even2}S$$

(1) $\text{even } n$ (by I.H.)

Need to prove: $\text{even } (S (S n))$

(2) $\text{odd } (S n)$ (by (1), $\text{odd}S$)

(3) $\text{even } (S (S n))$ (by (2), $\text{even}S$)

QED.

$$\frac{}{\text{even } Z} \text{even}Z$$

$$\frac{\text{odd } n}{\text{even } (S n)} \text{even}S$$

$$\frac{\text{even } n}{\text{odd } (S n)} \text{odd}S$$

LIST OF NATURAL NUMBERS

- Judgement Form:

- $l \text{ list}$ “ l is a list”

$$\frac{}{nil \text{ list}} nil$$

$$\frac{n \text{ nat} \quad l \text{ list}}{cons(n, l) \text{ list}} cons$$

- Cons stands for “CONcateenateS”
- Means concatenation of a *head* and a *tail* of a list.
- In $cons(n, l)$, n is the head and l is the tail.
- $cons(1, cons(2, cons(3, nil))) = 1::2::3::nil = [1,2,3]$

Lemma 1: $\text{cons}((S\ Z), \text{cons}(Z, \text{nil}))$ is a list.

Proof: By giving a derivation of $\text{cons}((S\ Z), \text{cons}(Z, \text{nil}))$ list.

$$\begin{array}{c} \begin{array}{ccc} Z\ \text{nat} & (\text{by } Z) & \\ \hline (S\ Z)\ \text{nat} & & \end{array} & \begin{array}{ccc} Z\ \text{nat} & (\text{by } Z) & \text{nil list} & (\text{by nil}) \\ \hline \text{cons}(Z, \text{nil})\ \text{list} & & \end{array} \\ \hline \text{cons}((S\ Z), \text{cons}(Z, \text{nil}))\ \text{list} \end{array}$$

LIST - LEN

- Judgment Form: $\text{len } l \text{ } n$.
 - “the length of l is n ”.

$$\frac{}{\text{len nil } Z} \text{len} - \text{nil}$$

$$\frac{\text{len } l \text{ } n}{\text{len cons}(n_1, l) (S \text{ } n)} \text{len} - \text{cons}$$

LIST - APPEND

- Judgment Form: $\text{append } l_1 \text{ } n \text{ } l_2$.
 - “ l_2 is the result of appending n to l_1 ”.

$$\frac{}{\text{append } nil \text{ } n \text{ } cons(n, nil)} \text{append} - nil$$

$$\frac{\text{append } l \text{ } n_2 \text{ } l_1}{\text{append } cons(n_1, l) \text{ } n_2 \text{ } cons(n_1, l_1)} \text{append} - cons$$