Computational Logic

Lecture 16

Mathematical Induction

Michael Genesereth

Autumn 2007

Incomplete Induction

Definition

$f(1)=1$

$f(x+1)=f(x)+2*x+1$

Data

$$1 = 1 = 1^2$$
$$1+3 = 4 = 2^2$$
$$1+3+5 = 9 = 3^2$$
$$1+3+5+7 = 16 = 4^2$$
$$1+3+5+7+9 = 25 = 5^2$$

Conjecture

$$f(x) = x^2$$

In this case, the answer is correct. Lucky Guess.

Not So Lucky Guess

Data:

$$2^{2^1}+1 = 2^{2+1} = 5$$
$$2^{2^2}+1 = 2^{4+1} = 17$$
$$2^{2^3}+1 = 2^{8+1} = 257$$
$$2^{2^4}+1 = 2^{16+1} = 65537$$

"Theorem" by Fermat (1601-1665):

$$prime(2^{2^x}+1)$$

Fact discovered (mercifully) after his death:

$$2^{2^5}+1 = 4{,}294{,}967{,}297 = 641 * 6{,}700{,}417$$

Oops.

Basic Metavocabulary

Data

$$1 = 1 = 1^2$$
$$1+3 = 4 = 2^2$$
$$1+3+5 = 9 = 3^2$$
$$1+3+5+7 = 16 = 4^2$$

Base Case: Prove for n=1

$$f(1)=1=1^2$$

Inductive Case: Assume true for x; prove for x+1

$$f(x+1)=f(x)+2*x+1$$
$$f(x+1)=x^2+2*x+1$$
$$f(x+1)=(x+1)^2$$

Jules Henri Poincare (1854-1912) credited with invention.

Outline

Linear Induction
  Input: successor function on individuals
  Output: universally quantified conclusions

Structural Induction
  Input: constructor function for structures
  Output: universally quantified conclusions

Linear Induction

Linearly Structured World:

$$\bullet \rightarrow \bullet \rightarrow \bullet$$
$$e \quad f(e) \quad f(f(e))$$

In other words, there is a distinguished *base element* and there is a *successor function*, which, starting at the base element, *enumerates all elements* in the universe of discourse.

Base element: e
Successor function: f

Linear Induction Schema

If a property holds of the base element and if it holds of a successor whenever it holds of an element, we would like to assert that it holds of all elements in the universe of discourse.

$$\phi[e] \wedge \forall x.(\phi[x] \Rightarrow \phi[f(x)]) \Rightarrow \forall x.\phi[x]$$

Base case: $\phi[e]$
Inductive case: $\forall x.(\phi[x] \Rightarrow \phi[f(x)])$
  Inductive antecedent: $\phi[x]$
  Inductive consequent: $\phi[f(x)])$
Conclusion: $\forall x.\phi[x]$

Analogy to students in class passing on messages

Arithmetic Examples

Object constant: 0
Unary function constant: s (+1)
Unary relation constants: p, *even*, *odd*

Induction Schema:

$$\phi[0] \wedge \forall x.(\phi[x] \Rightarrow \phi[s(x)]) \Rightarrow \forall x.\phi[x]$$

Instances of Induction Schema:

$$p(0) \wedge \forall x.(p(x) \Rightarrow p(s(x))) \Rightarrow \forall x.p(x)$$

$$even(0) \wedge \forall x.(even(x) \Rightarrow even(s(x))) \Rightarrow \forall x.even(x)$$

$$(even(0) \vee odd(0))$$
$$\wedge \forall x.(even(x) \vee odd(x) \Rightarrow even(s(x)) \vee odd(s(x)))$$
$$\Rightarrow \forall x.(even(x) \vee odd(x))$$

Arithmetic Problem

Object constant: 0
Unary function constant: s (+1)
Unary relation constants: p

Axioms

$$p(0)$$
$$\forall x.(p(x) \Rightarrow p(s(x)))$$

Goal:

$$\forall x.p(x)$$

Induction Schema:

$$p(0) \wedge \forall x.(p(x) \Rightarrow p(s(x))) \Rightarrow \forall x.p(x)$$

Clausal Form

$p(0) \wedge \forall x.(p(x) \Rightarrow p(s(x))) \Rightarrow \forall x.p(x)$

I: $\neg(p(0) \wedge \forall x.(\neg p(x) \vee p(s(x)))) \vee \forall x.p(x)$

N: $\neg p(0) \vee \neg \forall x.(\neg p(x) \vee p(s(x))) \vee \forall x.p(x)$

$\neg p(0) \vee \exists x.\neg(\neg p(x) \vee p(s(x))) \vee \forall x.p(x)$

$\neg p(0) \vee \exists x.(\neg\neg p(x) \wedge \neg p(s(x))) \vee \forall x.p(x)$

$\neg p(0) \vee \exists x.(p(x) \wedge \neg p(s(x))) \vee \forall x.p(x)$

S:

E: $\neg p(0) \vee (p(a) \wedge \neg p(s(a))) \vee \forall x.p(x)$

A: $\neg p(0) \vee (p(a) \wedge \neg p(s(a))) \vee p(x)$

D: $\neg p(0) \vee p(a) \vee p(x)$

$\neg p(0) \vee \neg p(s(a)) \vee p(x)$

O: $\{\neg p(0), p(a), p(x)\}$

$\{\neg p(0), \neg p(s(a)), p(x)\}$

Resolution Proof

1.	$\{p(0)\}$	Premise
2.	$\{\neg p(x), p(s(x))\}$	Premise
3.	$\{\neg p(0), p(a), p(x)\}$	Induction
4.	$\{\neg p(0), \neg p(s(a)), p(x)\}$	Induction
5.	$\{\neg p(c)\}$	Goal
6.	$\{p(a), p(x)\}$	1,3
7.	$\{\neg p(s(a)), p(x)\}$	1,4
8.	$\{p(s(a)), p(x)\}$	2,6
9.	$\{p(x)\}$	7,8
10.	$\{\}$	5,9

Arithmetic Problem

Object constant: 0
Unary function constant: *s*
Binary relation constants: *even*, *odd*

Axioms

$$even(0)$$
$$\forall x.(even(x) \Rightarrow odd(s(x)))$$
$$\forall x.(odd(x) \Rightarrow even(s(x)))$$

Goal:

$$\forall x.(even(x) \vee odd(x))$$

Induction Schema:

$$(even(0) \vee odd(0))$$
$$\wedge \forall x.(even(x) \vee odd(x) \Rightarrow even(s(x)) \vee odd(s(x)))$$
$$\Rightarrow \forall x.(even(x) \vee odd(x))$$

Clausal Form

$(even(0) \lor odd(0))$
$\quad \land \forall x.(even(x) \lor odd(x) \Rightarrow even(s(x)) \lor odd(s(x)))$
$\quad \Rightarrow \forall x.(even(x) \lor odd(x))$

$\{\neg even(0), even(a), odd(a), even(x), odd(x)\}$
$\{\neg odd(0), even(a), odd(a), even(x), odd(x)\}$
$\{\neg even(0), \neg even(s(a)), even(x), odd(x)\}$
$\{\neg even(0), \neg odd(s(a)), even(x), odd(x)\}$
$\{\neg odd(0), \neg even(s(a)), even(x), odd(x)\}$
$\{\neg odd(0), \neg odd(s(a)), even(x), odd(x)\}$

Induction and Resolution

Understandability
Okay for computers
Terrible for humans

Generality:
Induction is an axiom schema.
Resolution works on sentences.

Linear Induction Method

Using the Induction schema to prove a universally quantified formula.

(1) Base Case. Prove the base case.

(2) Inductive Case. Assume ground version of induction antecedent (induction hypothesis) and prove corresponding version of induction consequent.

If successful, the universally quantified conclusion holds. Why? Deduction Theorem.

Even and Odd

Axioms

$$even(0)$$
$$even(x) \Rightarrow odd(s(x))$$
$$odd(x) \Rightarrow even(s(x))$$

Desired Conclusion:

$$\forall x.(even(x) \vee odd(x))$$

Induction Axiom:

$$(even(0) \vee odd(0))$$
$$\wedge \forall x.(even(x) \vee odd(x) \Rightarrow even(s(x)) \vee odd(s(x)))$$
$$\Rightarrow \forall x.(even(x) \vee odd(x))$$

Even and Odd Problem

1.	$\{even(0)\}$	Premise
2.	$\{\neg even(x), odd(s(x))\}$	Premise
3.	$\{\neg odd(x), even(s(x))\}$	Premise
4.	$\{even(a), odd(a)\}$	Hypothesis
5.	$\{\neg even(s(a))\}$	Goal
6.	$\{\neg odd(s(a))\}$	Goal
7.	$\{\neg odd(a)\}$	2,6
8.	$\{\neg even(a)\}$	3,5
9.	$\{even(a)\}$	4,7
10.	$\{\}$	8,9

Binary Addition

Object constant: 0
Unary function constant: s
Binary function constant: +

Chain Axioms:

$$s(x)=s(y) \Rightarrow x=y$$

Addition Axioms:

$$x+0=x$$
$$x+s(y)=s(x+y)$$
$$s(x)+y=s(x+y)$$

Binary Addition Problem

Question:

$$\forall x.0+x=x$$

Axioms:

$$s(x)=s(y) \Rightarrow x=y$$
$$x+0=x$$
$$x+s(y)=s(x+y)$$
$$s(x)+y=s(x+y)$$

Goal:

$$\forall x.0+x=x$$

Induction Axiom:

$$0+0=0 \wedge \forall x.(0+x=x \Rightarrow 0+s(x)=s(x)) \Rightarrow \forall x.0+x=x$$

Binary Addition Problem

1.	$\{\neg s(x) = s(y), x = y\}$	Axiom
2.	$\{x + 0 = x\}$	Axiom
3.	$\{x + s(y) = s(x + y)\}$	Axiom
4.	$\{s(x) + y = s(x + y)\}$	Axiom
5.	$\{0 + a = a\}$	Induction
6.	$\{x = x\}$	Equality
7.	$\{\neg 0 + s(a) = s(a)\}$	Goal
8.	$\{\neg s(0 + a) = s(a)\}$	3,7
9.	$\{\neg s(a) = s(a)\}$	5,8
10.	$\{\}$	6,9

Binary Addition Problem

Axioms:

$$s(x)=s(y) \Rightarrow x=y$$
$$x+0=x$$
$$x+s(y)=s(x+y)$$
$$s(x)+y=s(x+y)$$

Question:

$$\forall x.x+y=y+x$$

Induction Axiom:

$$0+y=y+0 \wedge \forall x.(x+y=y+x \Rightarrow s(x)+y=y+s(x)) \Rightarrow \forall x.x+y=y+x$$

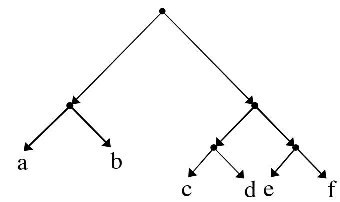
11/15/07 21

Binary Addition Problem

1.	$\{\neg s(x) = s(y), x = y\}$	Axiom
2.	$\{x + 0 = x\}$	Axiom
3.	$\{x + s(y) = s(x + y)\}$	Axiom
4.	$\{s(x) + y = s(x + y)\}$	Axiom
5.	$\{a + y = y + a\}$	Induction
6.	$\{x = x\}$	Equality
7.	$\{\neg s(a) + y = y + s(a)\}$	Goal
8.	$\{\neg s(a + y) = y + s(a)\}$	4,7
9.	$\{\neg s(y + a) = y + s(a)\}$	5,8
10.	$\{\neg y + s(a) = y + s(a)\}$	3,9
11.	$\{\}$	6,10

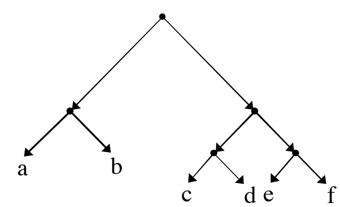
11/15/07 22

Intuition for Structural Induction



Binary Tree Representation

Tree:



Representation as a term:

pair(*pair*(*a*,*b*),*pair*(*pair*(*c*,*d*),*pair*(*e*,*f*)))

Infix version:

(*a***b*)*((*c***d*)*(*e***f*))

Induction and Resolution

Base elements: *a*, *b*

Successor function: *

Induction Schema:

$$\phi[a] \wedge \phi[b] \wedge \forall x.\forall y.(\phi[x] \wedge \phi[y] \Rightarrow \phi[x*y]) \Rightarrow \forall x.\phi[x]$$

Reverse

Object Constants: *a*, *b*
Unary Function Constant: *rev*
Binary Function Constant: *

Axioms:

$rev(a)=a$
$rev(b)=b$
$rev(x*y)=rev(y)*rev(x)$

Reverse Problem

Desired Conclusion:

$$rev(rev(x))=x$$

Axioms:

$$rev(a)=a$$
$$rev(b)=b$$
$$rev(x*y)=rev(y)*rev(x)$$

Induction Axiom:

$$rev(rev(a))=a \wedge rev(rev(b))=b$$
$$\wedge \forall x. \forall y.(rev(rev(x))=x \wedge rev(rev(y))=y \Rightarrow rev(rev(x*y))=x*y)$$
$$\Rightarrow \forall x.\ rev(rev(x))=x$$

Reverse Problem

1.	$\{rev(a) = a\}$	Axiom
2.	$\{rev(b) = b\}$	Axiom
3.	$\{rev(x * y) = rev(y) * rev(x)\}$	Axiom
4.	$\{rev(rev(c)) = c\}$	Induction
5.	$\{rev(rev(d)) = d\}$	Induction
6.	$\{x = x\}$	Equality
7.	$\{\neg rev(rev(x * y)) = x * y\}$	Goal
8.	$\{\neg rev(rev(y) * rev(x)) = x * y\}$	3,7
9.	$\{\neg rev(rev(x)) * rev(rev(y)) = x * y\}$	3,8
10.	$\{\neg c * rev(rev(y)) = c * y\}$	4,9
11.	$\{\neg c * d = c * d\}$	5.10
12.	$\{\}$	6,11

Metalevel Logic

Basic idea: represent expressions in Propositional Logic as terms in Relational Logic, write Relational sentences to define basic concepts of Propositional Logic, prove metatheorems.

NB: We can extend to Relational Logic as well. The formalization is messier, and some nasty problems need to be handled (notably paradoxes).

Syntactic Metavocabulary

Object Constants (propositions)
p, *q*, *r*

Function constants

not(p)	*implies(p,q)*
and(*p*,*q*)	*impliedby(p,q)*
or(*p*,*q*)	*iff(p,q)*

Relation Constants
proposition(*p*)
sentence(*and(p*,*q*))
proves(*and(p*,*q), or*(*p*,*q*))

Syntactic Metadefinitions

negation(not(x)) ⇔ *sentence(x)*
conjunction(and(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*
disjunction(or(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*
implication(implies(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*
reduction(impliedby(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*
equivalence(iff(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*

sentence(x) ⇔
proposition(x) ∨
negation(x) ∨ *conjunction(x)* ∨ *disjunction(x)* ∨
implication(x) ∨*reduction(x)* ∨ *equivalence(x)*

proves(x,y) ⇔ …

Semantic Metavocabulary

Object Constants (interpretations)
i, j, k

Function constants

not(p)
and(p,q)
or(p,q)
implies(p,q)
impliedby(p,q)
iff(p,q)

Relation Constants
interpretation(i)
satisfies(i,p)
valid(or(p,not(p))
entails(and(p,q), or(p,q))

Semantic Metadefinitions

$satisfies(i,not(x)) \Leftrightarrow \neg satisfies(i,x)$
$satisfies(i,and(x,y)) \Leftrightarrow satisfies(i,x) \wedge satisfies(i,y)$
$satisfies(i,or(x,y)) \Leftrightarrow satisfies(i,x) \vee satisfies(i,y)$
$satisfies(i,implies(x,y)) \Leftrightarrow \neg satisfies(i,x) \vee satisfies(i,y)$
$satisfies(i,impliedby(x,y)) \Leftrightarrow satisfies(i,x) \vee \neg satisfies(i,y)$
$satisfies(i,iff(x,y)) \Leftrightarrow satisfies(i,x) \wedge satisfies(i,y) \vee$
$\neg satisfies(i,x) \wedge \neg satisfies(i,y)$

$valid(z) \Leftrightarrow \forall x.(interpretation(x) \Rightarrow satisfies(x,z))$

$entails(y,z) \Leftrightarrow$
$\forall x.(interpretation(x) \wedge satisfies(x,y) \Rightarrow satisfies(x,z))$

Metatheorems

Validity of Axiom Scemata:

$$valid(or(x,not(x)) \Leftrightarrow sentence(x)$$

Soundness:

$$proves(x,y) \Leftrightarrow entails(x,y)$$

Deduction Theorem:

$$proves(and(x,y),z) \Leftrightarrow proves(x,implies(y,z))$$

Goldbach's Conjecture

Data

$$
\begin{aligned}
6 &= 3+3 \\
8 &= 3+5 \\
10 &= 5+5 \\
12 &= 5+7 \\
14 &= 7+7 \\
16 &= 5+11
\end{aligned}
$$

Conjecture (1742):

$$\forall z.(even(z) \wedge z > 4 \Rightarrow \exists x.\exists y.(prime(x) \wedge prime(y) \wedge x + y = z))$$

As of late 60's, still not proved.
Single counterexample would disprove.

Summary

The key to induction is having some local way of working through space of all objects and using that to establish arbitrary formulas.

Reasoning by Induction:
- Instance of schema can be used by standard proof methods.
- Induction schema cannot be written as finite set of axioms.
- Induction method works when schema true, and it's simple.

Structural Induction useful in writing formal proofs of metatheorems.