Introduction to Logic

Logic in Logic

Michael Genesereth
Computer Science Department
Stanford University

Whole Numbers

Object Constant: 0

Unary Function Constant: *s*

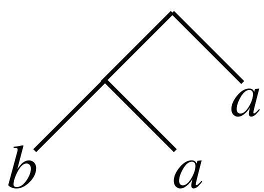
Binary Relation Constants:

same - the first and second arguments are identical
succ - the first argument immediately precedes second
less - the first argument less than or equal to second

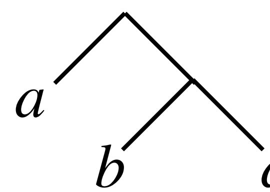
Trees

Object constants: a, b

Unary function constants: *cons*



$cons(cons(b, a), a)$



$cons(a, cons(b, a))$

Unary relation constants: *symmetric*, *uniform*, ...

Binary relation constant: *subtree*, *congruent*, *mirror*, ...

Lists

Object Constants: a, b, c, d, nil

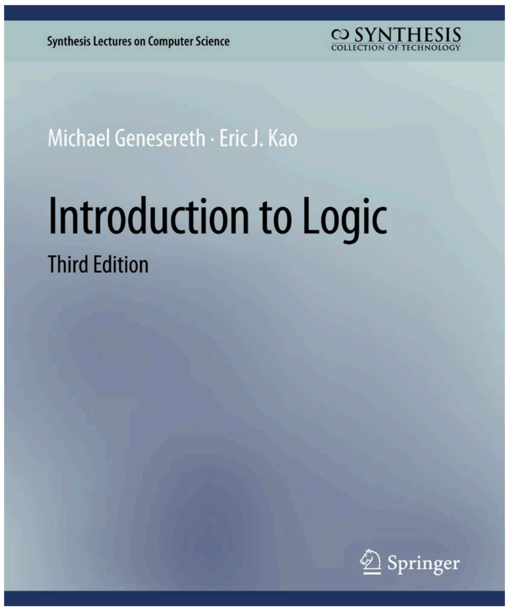
Binary Function Constant: *cons*

Binary Relation Constant: *member*
Ternary Relation Constant: *append*

$$member(b, [a, b, c])$$
$$append([a, b], [c, d], [a, b, c, d])$$

Metalevel Logic

Metalevel Logic

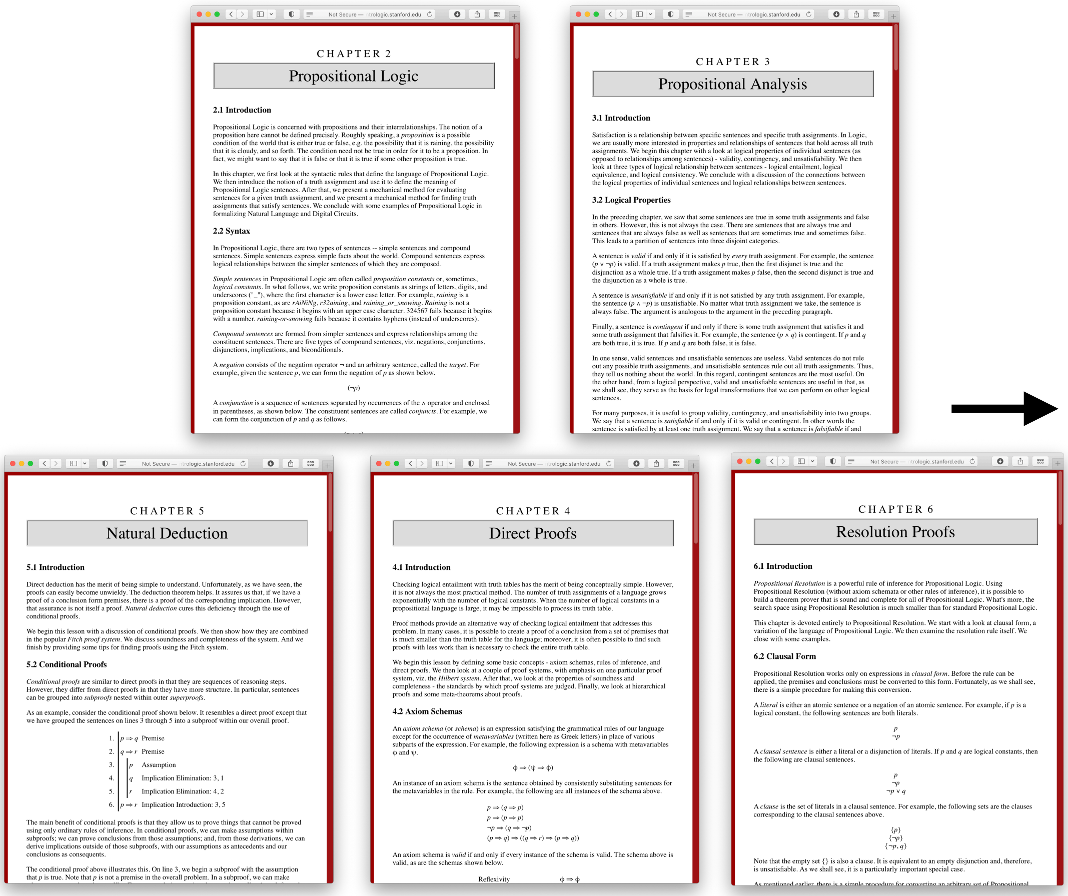


proposition(p)
proposition(q)
proposition(r)

negation(not(x)) ⇔ *sentence(x)*
conjunction(and(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*
disjunction(or(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*
implication(if(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*
biconditional(iff(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*

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

Propositional Logic in Term Logic



proposition(p)
proposition(q)
proposition(r)

negation(not(x)) ⇔ *sentence(x)*
conjunction(and(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*
disjunction(or(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*
implication(if(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*
biconditional(iff(x,y)) ⇔ *sentence(x)* ∧ *sentence(y)*

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

Basic Idea

(1) Represent Propositional Logic *sentences* as *terms* in Term Logic.

$$p \wedge \neg q \text{ represented as } and(p, not(q))$$

(2) Write Term Logic sentences to define the syntax and semantics of Propositional Logic.

$$conjunction(and(p, not(q)))$$

(3) Create Term Logic proofs of Propositional Logic metatheorems (e.g. soundness, completeness, deduction theorem, and so forth).

$$\forall x.\forall y.(entails(x,y) \Rightarrow proves(x,y))$$

Syntactic Metavocabulary

Object Constants (representing *propositions*):

p, q, r

Syntactic Metavocabulary

Object Constants (representing propositions):

p, *q*, *r*

Function constants (representing logical operators):

not(*x*)	*if*(*x*,*y*)
and(*x*,*y*)	*iff*(*x*,*y*)
or(*x*,*y*)	

*These are **terms**!*

***Not** sentences!*

Syntactic Metavocabulary

Object Constants (representing propositions):

p, *q*, *r*

Function constants (representing logical operators):

not(x)	*if(x,y)*	*These are **terms**!*
and(*x*,*y*)	*iff(x,y)*	***Not** sentences!*
or(*x*,*y*)		

Unary Relation Constants (properties of sentences):

proposition(*x*)	*implication*(*x*)	
negation(*x*)	*biconditional*(*x*)	
conjunction(*x*)	*sentence*(*x*)	
disjunction(*x*)		*These are sentences.*

Syntactic Metadefinitions

$proposition(p)$
$proposition(q)$
$proposition(r)$

$negation(not(x)) \Leftrightarrow sentence(x)$
$conjunction(and(x,y)) \Leftrightarrow sentence(x) \wedge sentence(y)$
$disjunction(or(x,y)) \Leftrightarrow sentence(x) \wedge sentence(y)$
$implication(if(x,y)) \Leftrightarrow sentence(x) \wedge sentence(y)$
$biconditional(iff(x,y)) \Leftrightarrow sentence(x) \wedge sentence(y)$

$sentence(x) \Leftrightarrow$
$proposition(x) \vee negation(x) \vee conjunction(x) \vee$
$disjunction(x) \vee implication(x) \vee biconditional(x)$

Semantic Metavocabulary

Unary Relation Constants (properties of sentences):
valid(*x*) - validity
contingent(*x*) - contingency
unsatisfiable(*x*) - unsatisfiability

Binary Relation Constants (relations among sentences):
equivalent(*x*,*y*) - logical equivalence
entails(*x*,*y*) - logical entailment
consistent(*x*,*y*) - consistency

We also need to talk about truth assignments in order to define these notions. Doable but messy; skipping here.

Semantic Metatheorems

Validity of Axiom Schemata:

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

Equivalence and Entailment:

$$equivalent(x,y) \Leftrightarrow entails(x,y) \wedge entails(y,x)$$

Deduction Theorem:

$$entails(and(x,y),z) \Leftrightarrow entails(x,if(y,z))$$

Rules of Inference

And Introduction:

$$\forall x.\forall y.(sentence(x) \wedge sentence(y) \Leftrightarrow ai(x,y,and(x,y)))$$

And Elimination:

$$\forall x.\forall y.(sentence(x) \wedge sentence(y) \Leftrightarrow ae(and(x,y),x))$$
$$\forall x.\forall y.(sentence(x) \wedge sentence(y) \Leftrightarrow ae(and(x,y),y))$$

More Metatheorems

Soundness:

$$\forall x.\forall y.(proves(x,y) \Rightarrow entails(x,y))$$

Completeness:

$$\forall x.\forall y.(entails(x,y) \Rightarrow proves(x,y))$$

Term Logic in Term Logic

Can we define Term Logic in Term Logic?

Basic idea: represent Term Logic expressions as terms in Term Logic, write sentences to define syntax and semantics, prove metatheorems.

Syntactic Metavocabulary

NB: We need terms to represent *functional terms* and *relational sentences*.

$$p(a,f(a)) \qquad relsent(p,a,funterm(f,a))$$

NB: We need *constants* in our language to refer to *variables* in the language we are describing.

$$\forall y.p(y,f(y)) \qquad forall(ny,relsent(p,ny,funterm(f,ny)))$$

Syntactic Metadefinitions

$obconst(a)$
$funconst(f)$
$relconst(r)$
$variable(nx)$

$functionalterm(funterm(w,x)) \Leftrightarrow funconst(w) \wedge term(x)$
$relationalsentence(relsent(w,x)) \Leftrightarrow relconst(w) \wedge term(x)$

$negation(not(x)) \Leftrightarrow sentence(x)$
$conjunction(and(x,y)) \Leftrightarrow sentence(x) \wedge sentence(y)$
$disjunction(or(x,y)) \Leftrightarrow sentence(x) \wedge sentence(y)$
$implication(if(x,y)) \Leftrightarrow sentence(x) \wedge sentence(y)$
$biconditional(iff(x,y)) \Leftrightarrow sentence(x) \wedge sentence(y)$
$universal(forall(v,x) \Leftrightarrow variable(v) \wedge sentence(x)$
$universal(exists(v,x) \Leftrightarrow variable(v) \wedge sentence(x)$

Cardinality Problem

In formalizing Propositional Logic, we *can* talk about truth assignments. The Herbrand base is always finite, and so there are only finitely many truth assignments.

In formalizing Term Logic, things are more difficult. The Herbrand base can be infinite (though it is always *countable*). However, the number of truth assignments can be *uncountable*. Unfortunately, we have only countably many terms!

Term Logic in Another Logic

Can we define the semantics of Term Logic in some ***other*** *logic?*

Good News / Bad News: First-Order Logic (FOL) allows for uncountable universes and so in principle can be used. Unfortunately, FOL theories with infinite universes have *nonstandard models* (unintended models that *cannot be excluded*).

NB: FOL is *weaker* than Term Logic. Some notions that can be defined exactly in Term Logic cannot be defined in FOL without allowing nonstandard models, e.g. Peano Arithmetic, transitive closure.

Term Logic in Another Logic

Can we define the semantics of Term Logic in some other logic?

Good News / Bad News: First-Order Logic (FOL) allows for uncountable universes and so in principle can be used. Unfortunately, FOL theories with infinite universes have *nonstandard models* (unintended models that *cannot be excluded*).

Good News / Bad News: Second-Order Logic (SOL) allows us to eliminate these nonstandard models, but it is more complicated and there is no complete proof procedure.

Self-Referential Logic

Can we use this "metalevel" approach to relate the truth of sentences described in a metalanguage to sentences describing those sentences?

Truth Predicate

Can we use this "metalevel" approach to relate the truth of sentences described in a metalanguage to sentences describing those sentences?

Example: If so, can we define a *truth predicate* that allows us to say whether or not a sentence is true?

$$\forall x.\forall y.(true(relsent(p,x,y)) \Leftrightarrow p(x,y))$$

Beliefs

Can we use this "metalevel" approach to relate the truth of sentences described in a metalanguage to sentences describing those sentences?

Example: Can we use our truth predicate to formalize the truth of people's beliefs, beliefs about those beliefs, etc.?

$$\forall x.(believes(john, x) \Leftrightarrow true(x))$$

Disinformation

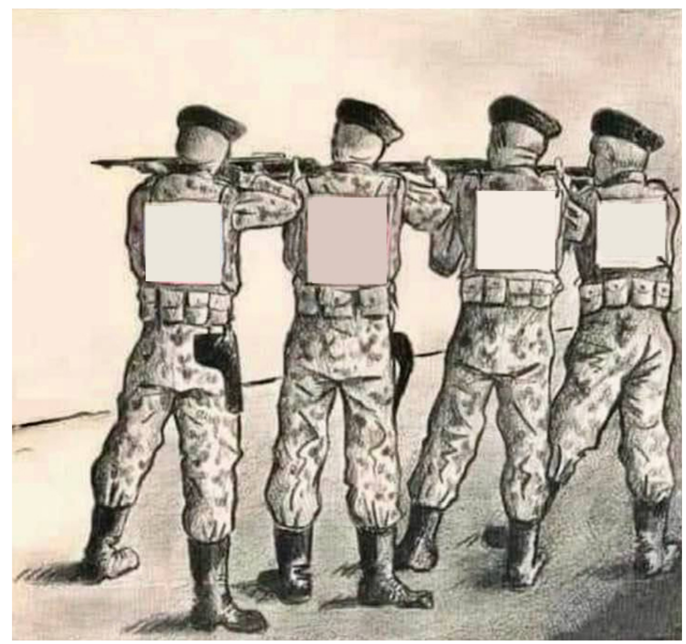
Can we use this "metalevel" approach to relate the truth of sentences described in a metalanguage to sentences describing those sentences?

Example: Can we use our truth predicate to formalize the truth or falsehood of people's statements?

$$\forall x.(says(john,x) \Rightarrow true(x))$$

Puzzle

You are taken prisoner by a drug cartel and told: *If you tell a lie, we will hang you. If you tell the truth, we will shoot you.* What do you say?



You say: *You will hang me.*
Result: They hang you *and* shoot you!
Suggestion: You should have asked if they meant *if and only if*.

Paradoxes

Unfortunately, trying to use a logic to define a truth predicate is problematic.

We run the risk of *paradoxes* (sentences that are both true and false / neither true nor false).

This sentence is false.

Also *nonsense terms* (terms that do not refer to anything).

The set of all sets that do not contain themselves

Results

We *can* completely formalize Propositional Logic in Term Logic.

(1) We can formalize *some details* of Term Logic in Term Logic but not everything. (2) We can formalize *more* of Term Logic in FOL, but we end up with *nonstandard models*. (3) We can eliminate nonstandard models using SOL, but it is complicated and there is no complete proof procedure.

We can axiomatize a metalevel truth predicate; but, unless we are very, very careful, this can lead to unpleasant complications, e.g. paradoxes.