Next-Next-Gen Notes
Object-Oriented Maths

Dark JP

March 28, 2018

Model Theory: semantics; Proof Theory: syntax

1 Kleene

1.1 Linguistic considerations: formulas

undefined terms: *lolm2k*

(1)

paradox: logic in terms on logic; solution: compartmentalize logic within "languages" (2)

object language/logic: the particular logic to be studied (3)

observer's language/logic: the logic used in studying the object language/logic (4)

sentences - declarative: a proposition; interrogative: a question; imperative: a command (5)

assume that object languages have a class of declarative sentences which serves as the building blocks, (6)

– and other sentences can be built from them by certain operations which are called "formulas" (A, ..., O) (7)

a language has "prime formulas"/"atoms" (P, ..., Z) which are distinct sentences that don't change meanings (8)

a language has 5 operations for building "composite formulas"/"molecules", (9)

– and these are - $\sim$: equivalence; $\supset$: implication; $\&$: conjunction; $\vee$: disjunction; $\neg$: negation (10)

(P, ..., Z) represent distinct prime formulas; (A, ..., O) represent formulas (11)

operator precedence: $\sim, \supset, \&, \vee, \neg, ..., (\_)$; – where the higher ranks are evaluated first, same ranks right first (12)

the "scope" of an operator is the parts of the formula where it acts upon (13)

1.2 Model theory: truth tables, validity

undefined terms: *lolm2k*

$$ \tag{14} $$

this chapter discusses the system of logic called classical logic (15)

different systems of logic are conceptually equally possible, but classical logic is the simplest (16)

classical logic: assumes that atom/declarative sentence/proposition can either be true or false, but not both (17)

do truth table for: $\sim, \supset, \&, \vee, \neg$ (18)

"valid"/"identically true"/"tautology"/"⊨" formulas evaluate to true independent of its prime formula values (19)

1.3 Model theory: the substitution rule, a collection of valid formulas

undefined terms: *lolm2k*

$$ \tag{20} $$

Theorem 1: let E be a formula consisting of the atoms $P_1, \ldots, P_n$, (21)

– and let E^* be a the formula E where atoms $P_1, \ldots, P_n$ are substituted by the formulas $A_1, \ldots, A_n$ (22)

– if $\vDash E$, then $\vDash E^*$, since formulas reduce to truth values which valid formulas are indifferent towards (23)

Note: Operators (op)s preserve type; Relations (rel)s return truths; include setOps; fix

2 Logic and Set Theory

2.1 D: Logical Truths and Operators

undefined terms: $:=, =, (\_), ,, `, .,$

$$ \tag{24} $$

$$ truth[t][] :=_{or} \begin{Bmatrix} t = T \\ t = F \end{Bmatrix} \tag{25} $$

$$ operatorLogic[\odot][x,y] :=_{and} \left\{ \begin{matrix} (truth[x][]) \\ (truth[y][]) \\ (truth[x \odot y][]) \end{matrix} \right) \tag{26} $$

$$ operatorOR[\vee][x,y] :=_{,} \left(truth[x][]\right),_{,} \left(truth[y][]\right),_{,} \left(truth[x \vee y][] = \begin{cases} F & x = F, y = F \\ T & x = F, y = T \\ T & x = T, y = F \\ T & x = T, y = T \end{cases} \right) ._{,} \tag{27} $$

$$operatorAND[\wedge][x,y]:=_1 \left(truth[x][]\right),_1 \left(truth[y][]\right),_1 \left(truth[x\wedge y][]=\begin{cases}F & x=F,y=F\\ F & x=F,y=T\\ F & x=T,y=F\\ T & x=T,y=T\end{cases}\right)._1 \tag{28}$$

$$operatorNOT[\neg][x]:=_1 \left(truth[x][]\right),_1 \left(truth[\neg x][]=\begin{cases}T & x=F\\ F & x=T\end{cases}\right)._1 \tag{29}$$

$$operatorXOR[\veebar][x,y]:=_1 \left(truth[x][]\right),_1 \left(truth[y][]\right),_1 \left(truth[x\veebar y][]=\begin{cases}F & x=F,y=F\\ T & x=F,y=T\\ T & x=T,y=F\\ F & x=T,y=T\end{cases}\right)._1 \tag{30}$$

$$operatorIF[\Longrightarrow][x,y]:=_1 \left(truth[x][]\right),_1 \left(truth[y][]\right),_1 \left(truth[x\Longrightarrow y][]=(\neg x)\vee y=\begin{cases}T & x=F,y=F\\ T & x=F,y=T\\ F & x=T,y=F\\ T & x=T,y=T\end{cases}\right)._1$$

$$\#\text{ a counterexample cannot follow from a false precedence, thus the conditional cannot be false} \tag{31}$$

$$operatorOIF[\Longleftarrow][x,y]:=_1 \left(truth[x][]\right),_1 \left(truth[y][]\right),_1 \left(truth[x\Longleftarrow y][]=(\neg y)\vee x=\begin{cases}T & x=F,y=F\\ F & x=F,y=T\\ T & x=T,y=F\\ T & x=T,y=T\end{cases}\right)._1 \tag{32}$$

$$\begin{aligned}&operatorIIF[\iff][x,y]:=_1 \left(truth[x][]\right),_1 \left(truth[y][]\right),_1\\ &\left(truth[x\iff y][]=(x\Longrightarrow y)\wedge(y\Longrightarrow x)=\begin{cases}T & x=F,y=F\\ F & x=F,y=T\\ F & x=T,y=F\\ T & x=T,y=T\end{cases}\right)._1\end{aligned} \tag{33}$$

P

2.2 P: Boolean Algebra

2.3 Predicates, Sets, Tuples

$arg\_(\_), set, \in, \{\_\},$

$$predicate[P][]:=truth[P(v_{free})][] \tag{45}$$

$$\begin{aligned}&universalQuantifier[\forall][P]:=_1 (predicate[P][]),_1\\ &(\forall_{x_{free}}(P(x_{free}))=P(y_{free}))._1\end{aligned} \tag{46}$$

$$existentialQuantifier[\exists][Q,P] := (\exists_{arg_x(Q(x))}(P(x)) = \neg\forall_{arg_x(Q(x))}(\neg P(x))) \tag{47}$$

$$uniquenessQuantifier[\exists!][Q,P] := (\exists!_{arg_x(Q(x))}(P(x)) = \exists_{arg_x(Q(x))}(P(x) \wedge \neg\exists_{arg_y(Q(y))}(P(y) \wedge \neg(y = x)))) \tag{48}$$

$$relationSetEq[=][X,Y] := (\forall_{arg_z(z \in X \vee z \in Y)}(z \in X \wedge z \in Y)) \tag{49}$$

$$operatorIntersection[\bigcap][X] := (z \in \bigcap(X) \iff \forall_{x \in X}(z \in x)) \tag{50}$$

$$operatorUnion[\bigcup][X] := (z \in \bigcup(X) \iff \exists_{x \in X}(z \in x)) \tag{51}$$

$$orderedPair[<x,y>][] == <x,y> = <a,b> iff x = a and y = b == \{\{x\},\{x,y\}\} \tag{52}$$