I'm not a robot

Mathematical logic deals with the logic in mathematics. Mathematical logic operators and laws define various statements in their mathematical form. In this article, we will explore mathematical logic along with the mathematical logic operators and types of mathematical logic. We will also solve some examples related to mathematical logic. What is Mathematical Logic?The study of mathematical logic in mathematics is called mathematical logic. The basic mathematical logic used are the conjunction (), disjunction (), and negation (). Some other mathematical logics are implication and double implication.Mathematical LogicMathematical Logic OperatorsThe basic mathematical logic operators are:ConjunctionDisjunctionNegationImplicationDouble ImplicationConjunctionIn mathematical logic conjunction of two statements results in true when both the statements are true otherwise false. Conjunction is also known as AND operator and is represented by . DisjunctionIn mathematical logic disjunction of two statements results in false if both the statements are false otherwise true. Disjunction is also known as OR operator and is represented by .NegationIn mathematical logic negation of two statements results in the not of the given statement i.e., if the statement is true it results in false and if the statement is false it results in true. Negation is also known as NOT operator and is represented by ~ or .ImplicationIn mathematical logic implication of two statements results in false if the first statement is true and second statement is false otherwise true. Implication is also known as conditional operator and is represented by or . Implication XY is read as If X and then Y.Double ImplicationIn mathematical logic double implication of two statements results in true when either both statements are true or both statements are false. Double implication is also known as biconditional operator and is represented by or . Double implication X Y is read as Y iff X or Y if and only if X. Mathematical Logic FormulasSome of the basic mathematical formulas are listed below:Formula NamesMathematical Logic FormulaIdentity LawDomination LawIdempotent lawDouble Negation LawCommutative Law(a b) (b a) (a b) (b a)Associative Law(a b) c a (b c)(a b) c a (b c)Distributive Lawa (b c) (a b) (a c)a (b c) (a b) (a c)De Morgan Law (a b) a b (a b) a bAbsorption Lawa (a b) aa (a b) aNegation LawTypes of Mathematical LogicThe different types of mathematical logic include:Set TheoryModel TheoryProof Theory Recursion TheorySet Theory: Set theory is a part of mathematical logic that deals with the sets which means collection of elements. The set theory is the theory consisting of sets, sets formulas and many more. Model Theory: Model theory is a part of mathematical logic that deals with the models of different theories of mathematics. The model theory provides different models describing the complex theories making it easy to understand.Proof Theory: Proof theory is a part of mathematical logic that deals with the proofs. The mathematical proofs provide easy analysis of different mathematical methods.Recursion Theory: Recursion theory is a part of mathematical logic used to construct computable functions, Turing machines and recursively enumerable sets.Mathematical Logic Truth Table The truth table in mathematical logic is a table which takes inputs and provides output when a logic is applied to it. The truth table for different mathematical logic operators are given below.NegationThe truth table for negation is given below.ConjunctionThe truth table for conjunction is given below.ABABTrueTrueTrueTrueFalseFalseFalseTrueFalseFalseFalseFalseDisjunctionThe truth table for disjunction is given below.ABABTrueTrueTrueTrueFalseFalseTrueTrueFalseFalseFalseImplicationThe truth table for implication is given below.ABABTrueTrueTrueTrueFalseFalseFalseTrueFalseFalseTrueDouble ImplicationThe truth table for double implication is given below.ABABTrueTrueTrueTrueFalseFalseFalseTrueFalseFalseFalseTrueMathematical Logic Solved ExamplesExample 1: Consider the statement x < 5 x - 2 < 5 is true or false?Solution:If x < 5 is true then, x - 2 < 5 is also true.T T is trueSo, the given statement x < 5 x - 2 < 5 is true.Example 2: For given two statements compute the truth table for conjunction.P: a is divisible by 4Q: a is divisible by 10Solution:Given the two statements P and Q.P: a is divisible by 4Q: a is divisible by 10Value of aPQP Q20TTT8TFF30FTF5FFFExample 3: Find the negation of the given statement P: It is a rainy day.Solution:Given statement,P: It is a rainy day.Negation of P = P: It is not a rainy day. While logic may simply refer to valid reasoning in everyday life, it is also one of the oldest and most foundational branches of mathematics, often blurring the boundaries between mathematics and philosophy. Logic is the study of Truth and how we can obtain universal Truths trough mathematical deduction. It is the most basic language of mathematics, and the underlying principle of proof. Aristotle384 322 BC Chrysippus of Solic. 279 206 BC Gottfried Wilhelm Leibniz1646 1716 Bertrand Russell1872 1970 Mathematical logic and reasoning dates back many thousand years, to ancient Egyptian architects and Babylonian astronomers. Logical thinking also developed independently in India and China. Centuries later, a variety of groups of Greek mathematicians and philosophers were discussing the nature of truth, trying to develop a formal system of mathematical logic and deduction. The ideas of Plato, Aristotle and many others carried through the Middle Ages, and were revived by scholars like Saint Thomas Aquinas and various Arabic mathematicians.Gottfried Leibniz was one of the first mathematicians to symbolic language for logic, similar to what we use today. Since then, logic has become closely entwined with concepts like axioms and proof, infinity, or number sets. Propositional Calculus There are many examples of mathematical statements or propositions. For example, 1 + 2 = 3 and 4 is even are clearly true, while all prime numbers are even is false. In logic we are often not interested in these statements themself, but how true and false statements are related to each other. Therefore we represent the propositions simply by placeholders like P and Q. All these propositions have to be either true (T) or false (F). If we have two propositions, P and Q, we can form a third one P Q. The small wedge means and, and P Q is only true if both P and Q are true. This relationship can be represented in a truth table, where we can look up the value of P Q given the values of P and Q. There are four possibilities: Similarly, the proposition P Q is true if either one or both of the conditions P and Q are true. The upside-down wedge means or, and here the corresponding truth table: Other operations which can be applied to logical propositions are equality, P Q, which is true if both operands have the same value (both true or both false), as well as exclusive or, P Q, which is true if one but not both of the operands are true: Finally there is the implication P Q. A false proposition P can imply anything, but a true proposition P can only imply another true proposition Q. Note that, unlike all the operations above, the implication is not symmetric: we cant just swap P and Q. PQP Q TTT TFF FTT FFT It is also possible to swap the value of a proposition, using the not operator . For example, if P is true then P is false. Applying the not operator twice gives us P = P. Having defined the varios logical operators, we can now combine them to form more complex expressions. For example, A (B C) is true if A is true and at least one of B and C is true. Notice that we need parenthesis, because A (B C) can have different values than (A B) C! We can combine logical operators in the same way we are used from the arithmetic operators +, , and . Together, the logical operators form a new kind of algebra, called Boolean algebra, named after mathematician George Boole (1815 1864). Many of the rules we know from elementary algebra translate into Boolean algebra: X (Y Z) = (X Y) Z x + (y + x) = (x + y) + z X (Y Z) = (X Y) Z x (y x) = (x y) z X (Y Z) = (X Y) (X Z) x (y + z) = x y + x z Of Knights and Knaves Lewis Carroll (aka CharlesLutwidge Dodgson, 1832 1898) Logic and truth tables can be used to solve a variety of problems of puzzles, many famous of which were invented by the writer of Alice in Wonderland, Lewis Carroll (1832 1898). A shipwrecked sailor arrives on a mysterious island inhabited by knights, who always tell the truth, and knaves, who always lie. Trying to get home, he knocks at the door of one of the huts. The islander who opens the door says that he and his wife are both knaves. Who should the sailor ask to find the right way home? This problem is easy to solve using logic and truth tables. Let H and W be two propositions which are true if the husband or wife are knights, respectively, and false if they are knaves. The husban said that both he and his wife are knaves, i.e. that H W. If he is a knight, this proposition is true, and if he is a knave, it is false. In particular, we know that H (H W) must be true. We can draw a truth table to determine the value of this proposition depending on the possible values of H and W: HWH WH (H W) TTFF TFFF FTFT FFTF The only row in which H (H W) is true is the second row, where the husband is a knave and the wife is a knight. Therefore the sailor should ask the wife for directions home. Why is a raven like a writing desk? You could have easily solved the Knights and Knaves puzzle above without creating an entire truth table, but maybe not this much more complex problem: The sailor meets three different islanders, which we shall simply call A, B and C. Islander A claims that B and C are knights, and Bsays that A is a knave and C is a knight. Who is telling the truth? Since we have an additional person there will be twice as many rows in the truth table, but the principles remain the same. In this example we have to find the row where (A (B C)) (B (A C)) is true. Can you work out how this proposition corresponds to the original problem? A B C B C A (B C) A C B (A C) (A (B C)) (B (A C)) T T T T T F F F T T F F F F F F T F T F F F T F T F F F F F T F F T T T F T T F F T F T F T F F F F F T F T T F F F F F F T F T T The only possible solution is when A, B and C are all knaves. Unfortunately, there is no one the sailor can trust Your browser does not support HTML video. 1986 Jim Henson Productions, Lucasfilm, TriStar Pictures After years of sailing across the ocean, Odysseus arrives in the remote city of Mathigon. He wants to ask the Oracle for the directions home, but the gate to the temple of wisdom is guarded by three gods: Apollo, Bacchus, and Chronos. One of the gods always tells the truth, one of them always lies, and one god tells the truth and lies at random. In order to pass the gate, Odysseus has to determine which god tells the truth, which one lies, and which one responds randomly by asking three yes/no questions, each directed to a single god. The gods understand English, but they always answer in Olympian. The Olympian words for yes and no are ho and to, but it is unknown which way round. How should Odysseus proceed? Note that you can ask multiple questions to the same god, but you cant direct a question to multiple gods at once. The second and third question may depend on the previous answers 1. Module #1 - Logic1Module #1:The Fundamentals of LogicRosen 5th ed., 1.1-1.4~81 slides, ~4 lectures 2. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 2Module #1: Foundations of Logic(1.1-1.3, ~3 lectures)Mathematical Logic is a tool for working withelaborate compound statements. It includes: A formal language for expressing them. A concise notation for writing them. A methodology for objectively reasoning abouttheir truth or falsity. It is the foundation for expressing formal proofs inall branches of mathematics. 3. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 3Foundations of Logic: Overview Propositional logic (1.1-1.2): Basic definitions. (1.1) Equivalence rules & derivations. (1.2) Predicate logic (1.3-1.4) Predicates. Quantified predicate expressions. Equivalences & derivations. 4. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 4Propositional Logic (1.1)Propositional Logic is the logic of compoundstatements built from simpler statementsusing so-called Boolean connectives.Some applications in computer science: Design of digital electronic circuits. Expressing conditions in programs. Queries to databases & search engines.Topic #1 Propositional LogicGeorge Boole(1815-1864)Chrysippus of Soli(ca. 281 B.C. 205 B.C.) 5. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 5Definition of a PropositionDefinition: A proposition (denoted p, q, r,) is simply: a statement (i.e., a declarative sentence) with some definite meaning, (not vague or ambiguous) having a truth value thats either true (T) or false (F) it is never both, neither, or somewhere in between! However, you might not know the actual truth value, and, the truth value might depend on the situation or context. Later, we will study probability theory, in which we assign degreesof certainty (between T and F) to propositions. But for now: think True/False only!Topic #1 Propositional Logic 6. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 6Examples of Propositions It is raining. (In a given situation.) Beijing is the capital of China. 1 + 2 = 3But, the following are NOT propositions: Whos there? (interrogative, question) La la la la la. (meaningless interjection) Just do it! (imperative, command) Yeah, I sorta dunno, whatever... (vague) 1 + 2 (expression with a non-true/false value)Topic #1 Propositional Logic 7. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 7An operator or connective combines one ormore operand expressions into a largerexpression. (E.g., + in numeric exprs.) Unary operators take 1 operand (e.g., 3);binary operators take 2 operands (eg 3 4). Propositional or Boolean operators operateon propositions (or their truth values)instead of on numbers.Operators / ConnectivesTopic #1.0 Propositional Logic: Operators 8. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 8Some Popular Boolean OperatorsFormal Name Nickname Arity SymbolNegation operator NOT Unary Conjunction operator AND Binary Disjunction operator OR Binary Exclusive-OR operator XOR Binary Implication operator IMPLIES Binary Biconditional operator IFF Binary Topic #1.0 Propositional Logic: Operators 9. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 9The Negation OperatorThe unary negation operator (NOT)transforms a prop. into its logical negation.E.g. If p = I have brown hair.then p = I do not have brown hair.The truth table for NOT: p pT FF TT : True; F : False: means is defined asOperandcolumnResultcolumnTopic #1.0 Propositional Logic: Operators 10. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 10The Conjunction OperatorThe binary conjunction operator (AND)combines two propositions to form theirlogical conjunction.E.g. If p=I will have salad for lunch. andq=I will have steak for dinner., thenpq=I will have salad for lunch andI will have steak for dinner.Remember: points up like an A, and it means NDNDTopic #1.0 Propositional Logic: Operators 11. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 11 Note that aconjunctionp1 p2 pnof n propositionswill have 2n rowsin its truth table. Also: and operations together are suffi-cient to express any Boolean truth table!Conjunction Truth Tablep q pqF F FF T FT F FT T TOperand columnsTopic #1.0 Propositional Logic: Operators 12. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 12The Disjunction OperatorThe binary disjunction operator (OR)combines two propositions to form theirlogical disjunction.p=My car has a bad engine.q=My car has a bad carburetor.pq=Either my car has a bad engine, ormy car has a bad carburetor. After the downward-pointing axe of splits the wood, youcan take 1 piece OR theother, or both.Topic #1.0 Propositional Logic: OperatorsMeaning is like and/or in English. 13. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 13 Note that pq meansthat p is true, or q istrue, or both are true! So, this operation isalso called inclusive or,because it includes thepossibility that both p and q are true. and together are also universal.Disjunction Truth Tablep q pqF F FF T TT F TT T TNotedifferencefrom ANDTopic #1.0 Propositional Logic: Operators 14. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 14Nested Propositional Expressions Use parentheses to group sub-expressions:I just saw my old friend, and either hesgrown or Ive shrunk. = f (g s) (f g) s would mean something different f g s would be ambiguous By convention, takes precedence overboth and . s f means (s) f , not (s f)Topic #1.0 Propositional Logic: Operators 15. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 15A Simple ExerciseLet p=It rained last night,q=The sprinklers came on last night,r=The lawn was wet this morning.Translate each of the following into English:p =r p = r p q =It didnt rain last night.The lawn was wet this morning, andit didnt rain last night.Either the lawn wasnt wet thismorning, or it rained last night, orthe sprinklers came on last night.Topic #1.0 Propositional Logic: Operators 16. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 16The Exclusive Or OperatorThe binary exclusive-or operator (XOR)combines two propositions to form theirlogical exclusive or (exjunction?).p = I will earn an A in this course,q = I will drop this course,p q = I will either earn an A in this course,or I will drop it (but not both!)Topic #1.0 Propositional Logic: Operators 17. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 17 Note that pq meansthat p is true, or q istrue, but not both! This operation iscalled exclusive or,because it excludes thepossibility that both p and q are true. and together are not universal.Exclusive-Or Truth Tablep q pqF F FF T TT F TT T F Notedifferencefrom OR.Topic #1.0 Propositional Logic: Operators 18. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 18Note that English or can be ambiguousregarding the both case!Pat is a singer orPat is a writer. -Pat is a man orPat is a woman. -Need context to disambiguate the meaning!For this class, assume or means inclusive.Natural Language is Ambiguousp q p "or" qF F FF T TT F TT T ?Topic #1.0 Propositional Logic: Operators 19. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 19The Implication OperatorThe implication p q states that p implies q.I.e., If p is true, then q is true; but if p is nottrue, then q could be either true or false.E.g., let p = You study hard.q = You will get a good grade.p q = If you study hard, then you will geta good grade. (else, it could go either way)Topic #1.0 Propositional Logic: Operatorsantecedent consequent 20. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 20Implication Truth Table p q is false only whenp is true but q is not true. p q does not saythat p causes q! p q does not requirethat p or q are ever true! E.g. (1=0) pigs can fly is TRUE!p q pqF F TF T TT F FT T TTheonlyFalsecase!Topic #1.0 Propositional Logic: Operators 21. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 21Examples of Implications If this lecture ever ends, then the sun willrise tomorrow. True or False? If Tuesday is a day of the week, then I ama penguin. True or False? If 1+1=6, then Bush is president.True or False? If the moon is made of green cheese, then Iam richer than Bill Gates. True or False?Topic #1.0 Propositional Logic: Operators 22. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 22Why does this seem wrong? Consider a sentence like, If I wear a red shirt tomorrow, then Osama bin Ladenwill be captured! In logic, we consider the sentence True so long aseither I dont wear a red shirt, or Osama is caught. But, in normal English conversation, if I were tomake this claim, you would think that I was lying. Why this discrepancy between logic & language? 23. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 23Resolving the Discrepancy In English, a sentence if p then q usually reallyimplicitly means something like, In all possible situations, if p then q. That is, For p to be true and q false is impossible. Or, I guarantee that no matter what, if p, then q. This can be expressed in predicate logic as: For all situations s, if p is true in situation s, then q is alsotrue in situation s Formally, we could write: s, P(s) Q(s) That sentence is logically False in our example,because for me to wear a red shirt and for Osama tostay free is a possible (even if not actual) situation. Natural language and logic then agree with each other. 24. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 24English Phrases Meaning p q p implies q if p, then q if p, q when p, q whenever p, q q if p q when p q whenever p p only if q p is sufficient for q q is necessary for p q follows from p q is implied by pWe will see some equivalentlogic expressions later.Topic #1.0 Propositional Logic: Operators 25. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 25Converse, Inverse, ContrapositiveSome terminology, for an implication p q: Its converse is: q p. Its inverse is: p q. Its contrapositive: q p. One of these three has the same meaning(same truth table) as p q. Can you figureout which?Topic #1.0 Propositional Logic: Operators 26. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 26How do we know for sure?Proving the equivalence of p q and itscontrapositive using truth tables:p q q p pq q pF F T T T TF T F T T TT F T F F FT T F F T TTopic #1.0 Propositional Logic: Operators 27. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 27The biconditional operatorThe biconditional p q states that p is true if andonly if (IFF) q is true.p = Bush wins the 2004 election.q = Bush will be president for all of 2005.p q = If, and only if, Bush wins the 2004election, Bush will be president for all of 2005.Topic #1.0 Propositional Logic: Operators2004 Im stillhere!2005 28. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 28Biconditional Truth Table p q means that p and qhave the same truth value. Note this truth table is theexact opposite of s!Thus, p q means (p q) p q does not implythat p and q are true, or that either of them causesthe other, or that they have a common cause.p q p qF F TF T FT F FT T TTopic #1.0 Propositional Logic: Operators 29. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 29Boolean Operations Summary We have seen 1 unary operator (out of the 4possible) and 5 binary operators (out of the16 possible). Their truth tables are below.p q p pq pq pq pq pqF F T F F F T TF T T F T T T T FT F F F T T F FT T F T T TTopic #1.0 Propositional Logic: Operators 30. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 30Some Alternative NotationsName: not and or xor implies iffPropositional logic: Boolean algebra: p pq + C/C++/Java (wordwise): ! && || != ==C/C++/Java (bitwise): ~ & | ^Logic gates:Topic #1.0 Propositional Logic: Operators 31. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 31Bits and Bit Operations A bit is a binary (base 2) digit: 0 or 1. Bits may be used to represent truth values. By convention:0 represents false; 1 represents true. Boolean algebra is like ordinary algebraexcept that variables stand for bits, + meansor, and multiplication means and. See module 23 (chapter 10) for more details.Topic #2 BitsJohn Tukey(1915-2000) 32. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 32Bit Strings A Bit string of length n is an ordered sequence(series, tuple) of n0 bits. More on sequences in 3.2. By convention, bit strings are (sometimes) writtenleft to right: e.g. the first bit of the bit string 1001101010 is 1. Watch out! Another common convention is that therightmost bit is bit #0, the 2nd-rightmost is bit #1, etc. When a bit string represents a base-2 number, byconvention, the first (leftmost) bit is the mostsignificant bit. Ex. 11012=8+4+1=13.Topic #2 Bits 33. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 33Counting in Binary Did you know that you can countto 1,023 just using two hands? How? Count in binary! Each finger (up/down) represents 1 bit. To increment: Flip the rightmost (low-order) bit. If it changes 10, then also flip the next bit to the left, If that bit changes 10, then flip the next one, etc. 0000000000, 0000000001, 0000000010, , 1111111101, 1111111110, 1111111111Topic #2 Bits 34. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 34Bitwise Operations Boolean operations can be extended tooperate on bit strings as well as single bits. E.g.:01 1011 011011 0001 110111 1011 1111 Bit-wise OR01 0001 0100 Bit-wise AND10 1010 1011 Bit-wise XORTopic #2 Bits 35. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 35End of 1.1You have learned about: Propositions: Whatthey are. Propositional logicoperators Symbolic notations. English equivalents. Logical meaning. Truth tables. Atomic vs. compoundpropositions. Alternative notations. Bits and bit-strings. Next section: 1.2 Propositionalequivalences. How to prove them. 36. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 36Propositional Equivalence (1.2)Two syntactically (i.e., textually) differentcompound propositions may be thesemantically identical (i.e., have the samemeaning). We call them equivalent. Learn: Various equivalence rules or laws. How to prove equivalences using symbolicderivations.Topic #1.1 Propositional Logic: Equivalences 37. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 37Tautologies and ContradictionsA tautology is a compound proposition that istrue no matter what the truth values of itsatomic propositions are!Ex. p p [What is its truth table?]A contradiction is a compound propositionthat is false no matter what! Ex. p p[Truth table?]Other compound props. are contingencies.Topic #1.1 Propositional Logic: Equivalences 38. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 38Logical EquivalenceCompound proposition p is logicallyequivalent to compound proposition q,written pq, IFF the compoundproposition pq is a tautology.Compound propositions p and q are logicallyequivalent to each other IFF p and qcontain the same truth values as each otherin all rows of their truth tables.Topic #1.1 Propositional Logic: Equivalences 39. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 39Ex. Prove that pq (p q).p q ppqq pp qq pp qq ((pp qq))F FF TT FT TProving Equivalencevia Truth TablesFTTTTTTTTTFFFFFFFFTTTopic #1.1 Propositional Logic: Equivalences 40. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 40Equivalence Laws These are similar to the arithmetic identitiesyou may have learned in algebra, but forpropositional equivalences instead. They provide a pattern or template that canbe used to match all or part of a much morecomplicated proposition and to find anequivalence for it.Topic #1.1 Propositional Logic: Equivalences 41. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 41Equivalence Laws - Examples Identity: pT p pF p Domination: pT T pF F Idempotent: pp p pp p Double negation: p p Commutative: pq qp pq qp Associative: (pq)r p(qr)(pq)r p(qr)Topic #1.1 Propositional Logic: Equivalences 42. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 42More Equivalence Laws Distributive: p(qr) (pq)(pr)p(qr) (pq)(pr) De Morgans:(pq) p q(pq) p q Trivial tautology/contradiction:p p T p p FTopic #1.1 Propositional Logic: EquivalencesAugustusDe Morgan(1806-1871) 43. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 43Defining Operators via EquivalencesUsing equivalences, we can define operatorsin terms of other operators. Exclusive or: pq (pq)(pq)pq (pq)(qp) Implies: pq p q Biconditional: pq (pq) (qp)pq (pq)Topic #1.1 Propositional Logic: Equivalences 44. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 44An Example Problem Check using a symbolic derivation whether(p q) (p r) p q r.(p q) (p r) [Expand definition of] (p q) (p r) [Expand defn. of] (p q) ((p r) (p r)) [DeMorgans Law] (p q) ((p r) (p r))cont.Topic #1.1 Propositional Logic: Equivalences 45. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 45Example Continued...(p q) ((p r) (p r)) [commutes] (q p) ((p r) (p r)) [associative] q (p ((p r) (p r))) [distrib. over] q (((p (p r)) (p (p r)))[assoc.] q (((p p) r) (p (p r)))[trivail taut.] q ((T r) (p (p r))) [domination] q (T (p (p r)))[identity] q (p (p r)) cont.Topic #1.1 Propositional Logic: Equivalences 46. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 46End of Long Exampleq (p (p r))[DeMorgans] q (p (p r))[Assoc.] q ((p p) r)[Idempotent] q (p r)[Assoc.] (q p) r[Commut.] p q rQ.E.D. (quod erat demonstrandum)Topic #1.1 Propositional Logic: Equivalences(Which was to be shown.) 47. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 47Review: Propositional Logic(1.1-1.2) Atomic propositions: p, q, r, Boolean operators: Compound propositions: s : (p q) r Equivalences: pq (p q) Proving equivalences using: Truth tables. Symbolic derivations. p q r Topic #1 Propositional Logic 48. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 48Predicate Logic (1.3) Predicate logic is an extension ofpropositional logic that permits conciselyreasoning about whole classes of entities. Propositional logic (recall) treats simplepropositions (sentences) as atomic entities. In contrast, predicate logic distinguishes thesubject of a sentence from its predicate. Remember these English grammar terms?Topic #3 Predicate Logic 49. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 49Applications of Predicate LogicIt is the formal notation for writing perfectlyclear, concise, and unambiguousmathematical definitions, axioms, andtheorems (more on these in module 2) forany branch of mathematics.Predicate logic with function symbols, the = operator, and afew proof-building rules is sufficient for defining anyconceivable mathematical system, and for provinganything that can be proved within that system!Topic #3 Predicate Logic 50. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 50Other Applications Predicate logic is the foundation of thefield of mathematical logic, whichculminated in Gdels incompletenesstheorem, which revealed the ultimatelimits of mathematical thought: Given any finitely describable, consistentproof procedure, there will always remain sometrue statements that will never be provenby that procedure. I.e., we cant discover all mathematical truths,unless we sometimes resort to making guesses.Topic #3 Predicate LogicKurt Gdel1906-1978 51. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 51Practical Applicationsof Predicate Logic It is the basis for clearly expressed formalspecifications for any complex system. It is basis for automatic theorem provers and manyother Artificial Intelligence systems. E.g. automatic program verification systems. Predicate-logic like statements are supported bysome of the more sophisticated database queryengines and container class libraries these are types of programming tools.Topic #3 Predicate Logic 52. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 52Subjects and Predicates In the sentence The dog is sleeping: The phrase the dog denotes the subject -the object or entity that the sentence is about. The phrase is sleeping denotes the predicate-a property that is true of the subject. In predicate logic, a predicate is modeled asa function P() from objects to propositions. P(x) = x is sleeping (where x is any object).Topic #3 Predicate Logic 53. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 53More About Predicates Convention: Lowercase variables x, y, z... denoteobjects/entities; uppercase variables P, Q, Rdenote propositional functions (predicates). Keep in mind that the result of applying apredicate P to an object x is the proposition P(x).But the predicate P itself (e.g. P=is sleeping) isnot a proposition (not a complete sentence). E.g. if P(x) = x is a prime number,P(3) is the proposition 3 is a prime number.Topic #3 Predicate Logic 54. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 54Propositional Functions Predicate logic generalizes the grammaticalnotion of a predicate to also includepropositional functions of any number ofarguments, each of which may take anygrammatical role that a noun can take. E.g. let P(x,y,z) = x gave y the grade z, then ifx=Mike, y=Mary, z=A, then P(x,y,z) =Mike gave Mary the grade A.Topic #3 Predicate Logic 55. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 55Universes of Discourse (U.D.s) The power of distinguishing objects frompredicates is that it lets you state thingsabout many objects at once. E.g., let P(x)=x+1>x. We can then say,For any number x, P(x) is true instead of(0+1>0) (1+1>1) (2+1>2) ... The collection of values that a variable xcan take is called xs universe of discourse.Topic #3 Predicate Logic 56. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 56Quantifier Expressions Quantifiers provide a notation that allowsus to quantify (count) how many objects inthe univ. of disc. satisfy a given predicate. is the FORLL or universal quantifier.x P(x) means for all x in the u.d., P holds. is the XISTS or existential quantifier.x P(x) means there exists an x in the u.d.(that is, 1 or more) such that P(x) is true.Topic #3 Predicate Logic 57. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 57The Universal Quantifier Example:Let the u.d. of x be parking spaces at UF.Let P(x) be the predicate x is full.Then the universal quantification of P(x),x P(x), is the proposition: All parking spaces at UF are full. i.e., Every parking space at UF is full. i.e., For each parking space at UF, that space is full.Topic #3 Predicate Logic 58. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 58The Existential Quantifier Example:Let the u.d. of x be parking spaces at UF.Let P(x) be the predicate x is full.Then the existential quantification of P(x),x P(x), is the proposition: Some parking space at UF is full. There is a parking space at UF that is full. At least one parking space at UF is full.Topic #3 Predicate Logic 59. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 59Free and Bound Variables An expression like P(x) is said to have afree variable x (meaning, x is undefined). A quantifier (either or) operates on anexpression having one or more freevariables, and binds one or more of thosevariables, to produce an expression havingone or more bound variables.Topic #3 Predicate Logic 60. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 60Example of Binding P(x,y) has 2 free variables, x and y. x P(x,y) has 1 free variable, and one boundvariable. [Which is which?] P(x), where x=3 is another way to bind x. An expression with zero free variables is a bona-fide (actual) proposition. An expression with one or more free variables isstill only a predicate: e.g. let Q(y) = x P(x,y)Topic #3 Predicate Logic 61. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 61Nesting of QuantifiersExample: Let the u.d. of x & y be people.Let L(x,y)=x likes y (a predicate w. 2 f.v.s)Then y L(x,y) = There is someone whom xlikes. (A predicate w. 1 free variable, x)Then x (y L(x,y)) =Everyone has someone whom they like.(A _____ with ___ free variables.)Topic #3 Predicate Logic 62. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 62Review: Predicate Logic (1.3) Objects x, y, z, Predicates P, Q, R, are functionsmapping objects x to propositions P(x). Multi-argument predicates P(x, y). Quantifiers: [x P(x)] : For all xs, P(x).[x P(x)] : There is an x such that P(x). Universes of discourse, bound & free vars. 63. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 63Quantifier ExerciseIf R(x,y)=x relies upon y, express thefollowing in unambiguous English:x(y R(x,y))=y(x R(x,y))=x(y R(x,y))=y(x R(x,y))=x(y R(x,y))=Everyone has someone to rely on.Theres a poor overburdened soul whomeveryone relies upon (including himself)!Theres some needy person who reliesupon everybody (including himself).Everyone has someone who relies upon them.Everyone relies upon everybody,(including themselves)!Topic #3 Predicate Logic 64. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 64Natural language is ambiguous! Everybody likes somebody. For everybody, there is somebody they like, x y Likes(x,y) or, there is somebody (a popular person) whomeveryone likes? y x Likes(x,y) Somebody likes everybody. Same problem: Depends on context, emphasis.[Probably more likely.]Topic #3 Predicate Logic 65. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 65Game Theoretic Semantics Thinking in terms of a competitive game can help you tellwhether a proposition with nested quantifiers is true. The game has two players, both with the same knowledge: Verifier: Wants to demonstrate that the proposition is true. Falsifier: Wants to demonstrate that the proposition is false. The Rules of the Game Verify or Falsify: Read the quantifiers from left to right, picking values of variables. When you see , the falsifier gets to select the value. When you see , the verifier gets to select the value. If the verifier can always win, then the proposition is true. If the falsifier can always win, then it is false.Topic #3 Predicate Logic 66. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 66Lets Play, Verify or Falsify!Let B(x,y) : xs birthday is followed within 7 daysby ys birthday.Suppose I claim that among you:x y B(x,y)Your turn, as falsifier:You pick any x (so-and-so)y B(so-and-so,y)My turn, as verifier:I pick any y (such-and-such)B(so-and-so,such-and-such) Lets play it in class. Who wins this game? What if I switched thequantifiers, and Iclaimed thaty x B(x,y)?Who wins in thatcase?Topic #3 Predicate Logic 67. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 67Still More Conventions Sometimes the universe of discourse isrestricted within the quantification, e.g., x>0 P(x) is shorthand forFor all x that are greater than zero, P(x).=x (x>0 P(x)) x>0 P(x) is shorthand forThere is an x greater than zero such that P(x).=x (x>0 P(x))Topic #3 Predicate Logic 68. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 68More to Know About Binding x x P(x) - x is not a free variable inx P(x), therefore the x binding isnt used. (x P(x)) Q(x) - The variable x is outsideof the scope of the x quantifier, and istherefore free. Not a complete proposition! (x P(x)) (x Q(x)) This is legal,because there are 2 different xs!Topic #3 Predicate Logic 69. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 69Quantifier Equivalence Laws Definitions of quantifiers: If u.d.=a,b,c,x P(x) P(a) P(b) P(c) x P(x) P(a) P(b) P(c) From those, we can prove the laws:x P(x) x P(x)x P(x) x P(x) Which propositional equivalence laws canbe used to prove this?Topic #3 Predicate Logic 70. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 70More Equivalence Laws x y P(x,y) y x P(x,y)x y P(x,y) y x P(x,y) x (P(x) Q(x)) (x P(x)) (x Q(x))x (P(x) Q(x)) (x P(x)) (x Q(x)) Exercise:See if you can prove these yourself. What propositional equivalences did you use?Topic #3 Predicate Logic 71. Module #1 - Logic2/19/2024 (c)2001-2004, Michael P. Frank 71Review: Predicate Logic (1.3) Objects x, y, z, Predicates P, Q, R, are functionsmapping objects x to propositions P(x). Multi-