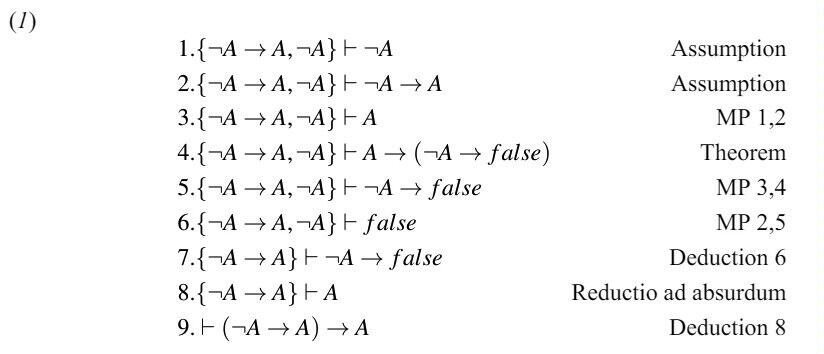
# Proof for 15 [>>](marginnote3app://note/77943505-2637-4696-BE50-DA1A7748246B)



# I2ML\_02\_preliminaries [>>](marginnote3app://note/FBB99DE8-684C-4EE2-BDAB-654DD1DA8B36)

## Set [>>](marginnote3app://note/ED487081-AF2B-4799-BE05-ABCFC1BDA077)

### A Set (集合) is a collection of objects. • A = Students in our class • B = {1, 5, 10, 20} • Z = Integers [>>](marginnote3app://note/A3C0AC8C-A54D-4F8C-AC83-FFDB1935E049)

#### Intension (内涵): The intension of a set is its description or defining properties, i.e., what is true about members of a set. (对概念的定义) [>>](marginnote3app://note/CCA4D101-C212-4A9A-B487-18AB3EA4157D)

#### Extension (外延): The extension of a set is its members or contents. (概念所代表的对象) [>>](marginnote3app://note/9D6591DF-645E-45E4-A24D-D3FF08D86FED)

#### Definition [>>](marginnote3app://note/4DA0461B-00B9-4A3A-976C-DE1852E82DF6) If φ(x) represents a property, then {x | φ(x)} denotes the set of all elements that have this property.

#### Axiom of Extension [>>](marginnote3app://note/ECEA2234-CA5A-4E5E-A721-90055AF57140)

##### The two sets A and B are equal (A = B) if and only if A and B have the same members. [>>](marginnote3app://note/96807E7D-DE64-48DD-B232-4D48E1A3567A)

###### Example: A and B are the same set: • A = {x ∈ R : x + y = y} for every real number y • B = {x ∈ R : x × z = x} for every real number z [>>](marginnote3app://note/08ED6350-40B3-43C8-AA74-BEE0D38CB5AE)

##### Order & Repetition Don’t Matter [>>](marginnote3app://note/D0842ABC-638B-45D7-A584-0D964314744B) • {a} = {a,a} • {a,b} = {b,a} = {a,b,b} = {a,b,b,a} • {a,b,c} = {c,b,a} = {b,c,b,a}

#### Subset [>>](marginnote3app://note/CAD7D283-8886-4320-9AE3-1F68C2EF0141) • A set A is a subset of a set B if all elements of A are also elements of B. • Formally, A ⊆ B iff for any x, if x ∈ A,then x ∈ B。 • For any set A,we have ∅ ⊆ A and A ⊆ A。 — {Aristotle, Russell} ⊆ {x | x is human} — {x | x is a prime number} ̸⊆ {x | x is an odd number} — {< 孟子,孟母 >,< 曹植,曹操 >} ⊆ {<x, y> | x is y’s son}

##### Proper Subset (真子集) [>>](marginnote3app://note/8EC1CD49-7DFB-4A68-87D3-9065477553B6) • If A ⊆ B and A ̸= B (i.e. there exists at least one element of B which is not an element of A),then A is a proper (or strict) subset of B, denoted by A ⊂ B. • ∅ (empty set, or {}) is a proper subset of any set except itself.

##### Power Set (幂集) [>>](marginnote3app://note/8A6F2F2C-6CC5-459D-90F1-826EE24922B3) If A is a set, then {X | X ⊆ A} is the power set of A (the set of all subsets of A), i.e., P(A). • P({a, b, c}) = {{a, b, c}, {a, b}, {a, c}, {b, c}, {a}, {b}, {c}, ∅} • P(∅) = {∅} • P({∅}) = {∅, {∅}}

#### Set Operations [>>](marginnote3app://note/A36AAE54-6969-410D-8AB7-879C28F88F5F) • A ∪ B denotes the union (并集) of set A and B: {x | x ∈ A or x ∈ B} • A ∩ B denotes the intersection (交集) of set A and B: {x | x ∈ A and x ∈ B} • A − B denotes the difference (差集) of set A and B: {x | x ∈ A and x ̸∈ B}

#### Set Properties [>>](marginnote3app://note/057D4317-F68B-458E-89EC-684DB5CBF234) • A ⊆ B iff. A ∩ B = A • A ⊆ B iff. A ∪ B = B

## Relation [>>](marginnote3app://note/9461912E-32DB-4585-BABF-79ABA65EF5AF)

### n-tuples (有序 n 元组) [>>](marginnote3app://note/C9816E63-9DA1-4D46-84CC-9FE307AC3A0E) In mathematics, a tuple is a finite sequence or ordered list of numbers. An n-tuple is a tuple of n elements, where n is a non-negative integer.

#### Properties of n-tuples [>>](marginnote3app://note/4AE0FB3E-7527-4C21-8D6E-DDDE70E54229) The general rule for the identity of two n − tuples: ⟨x1, x2, ..., xm⟩ = ⟨y1, y2, ..., yn⟩ iff m = n and x1= y1, x2= y2, ..., xn= yn A tuple has properties that distinguish it from a set: • A tuple may contain multiple instances of the same element, e.g., ⟨1, 2, 2, 3⟩ ̸= ⟨1, 2, 3⟩ • Tuple elements are ordered, e.g., ⟨1, 2, 3⟩ ̸= ⟨3, 2, 1⟩ • A tuple has a finite number of elements while a set may not.

### Binary Relation [>>](marginnote3app://note/C47428BB-C98D-42E8-94C6-50E83452CCF9) Intuitively, a binary relation from a set X to a set Y is a set of ordered pairs <x, y> where x is an element of X and y is an element of Y. A × B denotes the Cartesian product (笛卡尔积) of set A and B: {⟨x, y⟩ | x ∈ A and y ∈ B} (the set of all ordered pairs where x is in A and y is in B.) A binary relation R over sets X and Y is a subset of Cartesian product A × B, denoted as R ⊆ A × B.

#### The statement ⟨x, y⟩ ∈ R reads ”x is R-related to y”, and is denoted by R(x, y) or xRy. When X = Y, we call a relation R from X to Y a (binary) relation over X. (R 是 X 中的一个二元关系). [>>](marginnote3app://note/B8F7F79E-BC9A-4190-AD00-CBCB96600B97)

### n-ary relation [>>](marginnote3app://note/AE8BD678-E968-4E86-9717-39F96DC8904A) Relation Cartesian product of sets A1, A2, ..., An(n ≥ 1): A1× A2× ... × An= {⟨x1, x2, ..., xn⟩ | x1∈ A1, x2∈ A2, ..., xn∈ An} Denoted as Anif A1= A2= ... = An= A. If R ⊆ An, R is denoted as an n-ary relation over A.

### Equivalence Relation [>>](marginnote3app://note/FA1E04F8-5B77-40F9-9E5B-DFE468DCD8A8) Relation Let R be a binary relation on a set A. • R is reflexive(自反) if for all x∈A, xRx. • R is symmetric(对称) if for all x, y∈A, if xRy, then yRx. • R is transitive(传递) if for all x, y, z∈A, if xRy and yRz, then xRz. R is an equivalence relation (等价关系) on A if A is nonempty and R is reflexive, symmetric and transitive.

#### Equivalence Class [>>](marginnote3app://note/98DF2655-B4A0-4D9E-87C3-FA29F552CC3A) Relation Given an equivalence relation R over a set A, for any x ∈ A, the equivalence class of x is the set [x]R= {y ∈ A | xRy} [x]Ris the set of all elements of A that are equivalent to x.

##### If R is an equivalence relation over A, then every a ∈ A belongs to exactly one equivalence class. [>>](marginnote3app://note/7CF34356-5C90-4F90-A4EC-83D7A531A3B9)

##### Given an equivalence relation on set A, the collection of equivalence classes forms a partition (划分) of set A. [>>](marginnote3app://note/AA059D41-336A-4F7F-B21E-965F018C5641)

### Partial Order Relation (偏序关系) [>>](marginnote3app://note/3583374D-E2F7-4054-8A58-4AABB6F584FE) A binary relation R on a set A is antisymmetric (反对称的) if for all x, y ∈ A,if xRy and yRx, then x=y. • A binary relation R on a set A is a partial order (偏序关系) if R is reflexive, antisymmetric, and transitive. • For x ∈ A, if there doesn’t exist another y ∈ A such that yRx, then x is the minimal element (极小元) of this partial order. • For x ∈ A, if there doesn’t exist another y ∈ A such that xRy, then x is the maximal element (极大元) of this partial order.

### Total Order Relation (全序关系) [>>](marginnote3app://note/C26D8AFE-90E2-4E64-9789-D217C3744D90) Formally, a partial order relation R on a set A is a total order (linear order), if for any x, y ∈ A, either xRy or yRx. Intuitively, a total order or linear order is a partial order in which any two elements are comparable.

## Function [>>](marginnote3app://note/16451634-CFEA-4698-9B62-E8AB7FB096A4)

### A function from a set X to a set Y is a binary relation R between X and Y that satisfies the two following conditions: • For any x ∈ X, there exists y ∈ Y such that xRy. • If y, z ∈ Y such that xRy and xRz, then y = z. A function from a set X to a set Y assigns to each element of X exactly one element of Y. [>>](marginnote3app://note/17586AE3-447B-46A0-97CA-D64ADF7B7516)

### We typically use f, g, h to represent functions. The notation f : A → B expresses that f is a function from set A to a subset of set B. • The domain (定义域) of f: the set of input values • The codomain(陪域、上域)of f: the set of possible output values • The range (值域) of f: the set of actual ouput values, i.e., a subset of the B • We typically use f(x) = y to denote ⟨x, y⟩ ∈ f. [>>](marginnote3app://note/835A987A-D86B-4C85-909B-42C3A799C269)

### n-ary function [>>](marginnote3app://note/AC8DAB52-3F84-44D5-9103-40B6272F554B) • If the domain of f is the Cartesian product A1× A2× ... × An(n ≥ 1), then f is called an n-ary function. • An n-ary function maps ordered n-tuples from its domain to elements in its codomain. • f : An→ A is called an n-ary, function in A. • For example, the addition function + from N2to N is a binary function. Its domain is N2, its codomain is N.

### Injective, Surjective, and Bijective Functions [>>](marginnote3app://note/ED78F1D5-197D-497F-A6AA-C645B6CB5DB8) Given a function f : X → Y: • Injective (one-to-one, 单射或一一映射): if each element of the codomain is mapped to by at most one element of the domain, i.e., for all x1, x2∈ X, if f(x1) = f(x2) then x1= x2. • Surjective (onto, 满射): if each element of the codomain is mapped to by at least one element of the domain, i.e., for any y ∈ Y, there exists x ∈ X such that f(x) = y. • Bijective (双射): if each element of the codomain is mapped to by exactly one element of the domain. That is, the function is both injective and surjective.

#### [>>](marginnote3app://note/6C822AD8-B6B8-4CF1-A42C-C35CE3F9D085)

### Composite Function [>>](marginnote3app://note/49001676-AB76-4925-8CD4-A7E99CA44630) Function Given f : X → Y, g : Y → Z. The composite function is denoted g ◦ f : X → Z, defined by (g ◦ f)(x) = g(f(x)) for all x in X.

#### [>>](marginnote3app://note/9AD7BDEF-9D38-41AD-991E-BAF438C72CAE)

## Mathematical Definitions & Proof [>>](marginnote3app://note/73760789-50C8-46C0-929A-01677E155C6D)

### Inductive Definition [>>](marginnote3app://note/819CE263-FC7E-4D51-8DB3-55D93B194DAA)

#### The set N of natural numbers is inductively defined by the following rules: (i) 0 ∈ N (ii) For any n, if n ∈ N, then n + 1 ∈ N (iii) Only n generated by (finite iterations of) (i) and (ii), n ∈ N [>>](marginnote3app://note/0566650E-B6B9-454B-BDDA-CE136671DF9A)

#### The above definition can be equivalently stated as follows: N is the smallest inductive subset of S that satisfies conditions (i) and (ii): (i) 0 ∈ S (ii) For any n, if n ∈ S, then n + 1 ∈ S An inductive definition always implies that we are looking for the smallest set such that the given rules hold. [>>](marginnote3app://note/C6B1A7ED-9F88-4B32-834F-1125609B088F)

### Proof by Induction (归纳证明) [>>](marginnote3app://note/FA73A973-D4CA-4FFF-97DD-45C89C9FDB63)

#### For a set defined inductively, to prove that all its elements have a certain property P, one can use the proof by induction method. Let P be a property, and P(x) denotes that x has property P. If (i) P(0); (ii) For any n ∈ N, if P(n), then P(n′) (n′is the successor of n) Then, for any n ∈ N, P(n). [>>](marginnote3app://note/0F70EC00-5AD3-4D35-9176-8D6A73C57D60)

#### Template for proof by induction [>>](marginnote3app://note/EF6EE1D7-519A-48A5-BD4B-C142DF997448) • Base case: We need to show that p(n) is true for the smallest possible value of n, e.g., p(n0) is true. • Induction Hypothesis: Assume that the statement p(k) is true for any positive integer k ≥ n0. • Inductive Step: Show that the statement p(k + 1) is true.

### Recursive Definition (递归定义) [>>](marginnote3app://note/04DB118B-967E-4271-8107-BCD1E430EC37)

#### A recursive definition of a function f, defines a value of function at some natural number n in terms of the function’s value at some previous point(s). For example, let g, h be known functions on N, which define a function f on N: f(0) = g(0), f(n′) = h(f(n)). For any n ∈ N, the value of f(n) can be computed from the above definition using f(0), f(1), ..., f(n − 1), and this type of definition is referred to as recursive definition. [>>](marginnote3app://note/99A75CD6-9987-44B2-959C-71EB962638A4)

### Proof by Contradiction [>>](marginnote3app://note/3A166BEE-6280-4E44-A040-6C276C7A3017)

# I2ML\_03\_PL\_syntax [>>](marginnote3app://note/0D7AC911-3CAE-474A-9F9B-C1BEC71871F9)

## Propositional Logic as a Formal Language [>>](marginnote3app://note/D87BCAA2-B5A6-4F5B-BC30-3929690602BC)

## Propositions & Connectives [>>](marginnote3app://note/6169FFFC-1ACB-4588-BDC1-90C94A0E1088)

### Proposition (命题) [>>](marginnote3app://note/85ADAF43-198A-4A30-B19C-9E69D6CF1356) A proposition is a declarative sentence that can be judged as either true or false.

#### Atomic Proposition (原子命题) [>>](marginnote3app://note/8AAA8C9C-C320-4791-860E-B7D9F6B56E63) A proposition that does not contain any smaller part that is still a proposition is called an atomic proposition.

#### Compound Proposition (复合命题) [>>](marginnote3app://note/F991D43E-08AF-4B4D-96EE-F2AF12D997F7) A proposition that involves the assembly of multiple propositions is called a compound proposition.

### Logical Connectives [>>](marginnote3app://note/EA69AF2C-7C5E-450F-B23A-5C75C6FF21A7) Propositions & Connectives Words that connect multiple propositions to form a compound proposition are called logical connectives. For example: • ... and ... (并且) • not ... (并非) • ... or ... (或者) • if ... then ... (如果... 那么...) • ... if and only if ... (当且仅当)

### Symbols [>>](marginnote3app://note/3BFC70F3-E176-4463-B2BA-CFA5E0572C1E)

#### • Atomic proposition: p, q, r, ...... • Compound proposition: A, B, C, ...... [>>](marginnote3app://note/594B49CF-2AC5-4E8D-A9CA-D4BD0428FA89)

#### [>>](marginnote3app://note/A401526F-42F6-4920-80C6-58F9F029F30D)

## Propositional Logic as a Formal Language [>>](marginnote3app://note/C841AED3-C342-4B47-A7E5-E19BA3DF2AE4)

### [>>](marginnote3app://note/C6BE5BD4-313E-42AF-BE69-86269F369C8A)

#### [>>](marginnote3app://note/03C9DE01-0142-4549-95D5-1C738F055BB7)

### [>>](marginnote3app://note/A1DA6398-3D05-4D9E-8848-99E2184F3BF5)

#### [>>](marginnote3app://note/01715A35-57E2-4D94-8442-7E8D2D0355F6)

### [>>](marginnote3app://note/10F73886-7124-4D23-81F9-B5D09DB99308)

#### [>>](marginnote3app://note/451C5867-D75F-4B2C-A3BA-2D6B8E9533B2)

#### [>>](marginnote3app://note/432CDE4E-33E5-4DD1-A06E-B4DE8592A95D)

### Parse Tree [>>](marginnote3app://note/DA78298D-51C1-488A-AA95-08C51A695927)

#### [>>](marginnote3app://note/AF36B7F8-E0BB-450A-B168-354CAB966D62)

##### [>>](marginnote3app://note/301669E8-5221-4305-B019-7CA25BC523BD)

###### [>>](marginnote3app://note/2EAE5362-D7F4-43BE-875F-CACD042B18BB)

#### [>>](marginnote3app://note/05674D3A-17FE-438A-89B0-FF7B9DA48356)

### Subformula [>>](marginnote3app://note/1BCFCBCC-9D31-42AB-8CFA-E4E73970B1C7)

#### [>>](marginnote3app://note/FCA57100-97F0-45E2-A3C0-BE65B7267B9D)

##### 简单来说 就是一个命题树的节点 proper sub formula 不包括根节点 [>>](marginnote3app://note/0A6C5299-CAB4-4039-9BBB-6B256DE020A7)

### The Structure of Formula [>>](marginnote3app://note/C1AEAC2D-FA5C-4921-BD82-1E0DBB2775A0)

#### [>>](marginnote3app://note/E03B0970-614E-4BA6-A9B0-B10B0B9A87D8)

#### • uv denotes the result of concatenating two expressions u, v in this order. Note that λu = uλ = u. • v is a segment of u if u = w1vw2where u, v, w1, w2are expressions. • v is a proper segment of u is v is non-empty and v ̸= u. • If u = vw, where u, v, w are expressions, then v is an initial segment (prefix) of u, w is a terminal segment (suffix) of u. • If u = vw, where u, v, w are expressions, and v, w are non-empty, then v is an proper prefix of u, w is a proper suffix of u. [>>](marginnote3app://note/E740A65E-75D1-4E0A-ACBF-6429F243DB12)

#### [>>](marginnote3app://note/9DC62E11-D7DA-42EF-95B7-EA95DC0D5D3C)

#### [>>](marginnote3app://note/3728BF61-8952-40FB-819C-78ADCC6D865B)

### Conventions [>>](marginnote3app://note/6A86F105-3D02-48CB-9D32-9F9A907B162A)

#### For readability, we employ the following convention, which states that we may drop the outermost parentheses. If F or (F) is a formula, then we view F and (F) as the same formula. [>>](marginnote3app://note/FFD6B5C3-D4A0-4A7F-995E-EC933F119EEB)

### Precedence [>>](marginnote3app://note/E9BA65F3-03EB-4C83-8401-6D617BBF487E) Parentheses are used to resolve ambiguity. But they are hard to read. If no parentheses are present, we could use precedence and associativity to disambiguate formulas. • Each connective on the left has priority over those on the right. ¬, ∧, ∨, →, ↔. • Parentheses take the highest precedence. • Connectives are assumed to associate to the right (right associative), i.e., first group the rightmost occurrence. For example, p → q → r means p → (q → r)

# I2ML\_04\_PL\_semantics [>>](marginnote3app://note/3B6385BE-9B68-46B1-AD59-BFD78F149415)

## Truth Table [>>](marginnote3app://note/BF75855A-9776-433C-8DB0-98A6009C4C74)

### [>>](marginnote3app://note/7FDC38E1-1DCF-451C-A988-B1EADAE7BE71)

### [>>](marginnote3app://note/D84633C1-3CD0-433B-ADB9-88D49415AF5B)

### [>>](marginnote3app://note/90B2FC70-D05A-4123-B8B7-89C8139B1A6E)

自然语言中的 or 是 异或
而这里是 同或+异或

### [>>](marginnote3app://note/C8C3C86D-2B0E-4B4B-96D2-5F80F31226AA)

### [>>](marginnote3app://note/569D6D78-A0F2-4EE7-B1DA-5CDF192BC4E3)

## Truth Valuation(真值指派) [>>](marginnote3app://note/E7A88627-41B0-42E4-8095-1C0B84EAFA7F)

### [>>](marginnote3app://note/B75BE8F4-17A9-4A7D-98A3-6E4F259D6CA7)

### [>>](marginnote3app://note/685587CC-0C48-427B-96E1-B217295890C7)

### [>>](marginnote3app://note/FD0355E4-10E9-44DC-BAB7-FFA0381EBCE3)

## Properties of a formula [>>](marginnote3app://note/02DEE048-1889-4CE6-9970-5BE3E218004B) Let A ∈ Form(Lp). • If for every truth valuation v, Av= 1, then A is tautology (永真式或重言式) • If for every truth valuation v, Av= 0, then A is contradiction (永假式或矛盾式) • If there exists a truth value v such that Av= 1, then A satisfiable (可满足的)

### Truth table [>>](marginnote3app://note/986749C1-6AFA-453D-80C7-1928D637A732)

We could use truth table to determine the properties of a formula, e.g., the truth table of p ∨ q → q ∧ r

#### [>>](marginnote3app://note/90E25A88-1C6E-45D1-B877-ED21500D059A)

### Valuation Tree [>>](marginnote3app://note/0A3D1F1A-8D03-41CB-A9B5-D58CB73FAA87) Rather than fill out an entire truth table, we can analyze what happens if we plug in a truth value for one variable. We can evaluate a formula by using these rules to construct a valuation tree.

#### [>>](marginnote3app://note/9353C89D-E47F-41A2-9CFF-199CA6B3A5FE)

##### [>>](marginnote3app://note/B4327B71-E1D1-4D25-A48D-1BF0BF82F105)

## Satisfiability of sets of formulas [>>](marginnote3app://note/F54DB72F-8A01-4178-A2AE-5A4E81BDB1F9) Let Σ ⊆ Form(Lp) (a set of well-formed formulas). v is a truth valuation. Define: Σv= { 1 if for each B ∈ Σ, Bv= 1 0 else Σ is satisfiable iff there is some valuation v such that Σv= 1; we say v satisfies Σ.

## I2ML\_04\_PL\_semantics 2 [>>](marginnote3app://note/F6817643-041A-4685-8A45-B3CADC23D23F)

### Logical Equivalence [>>](marginnote3app://note/0ACDC99F-504A-44EF-9D61-56F6C84D6310)

#### Definition of Logical Equivalence [>>](marginnote3app://note/AECD0C7D-01E5-4EBC-95F2-4BB2FB5E77C5) Two formulas A and B are logically equivalent if and only if they have the same value under any valuation. • Av= Bv for every truth valuation v. • A and B must have the same final column in their truth tables. • A ↔ B is a tautology.

##### [>>](marginnote3app://note/085D5F82-C5BD-4D8E-A218-34A727F145A8)

##### [>>](marginnote3app://note/F7A96C15-9C7A-46EF-91CF-F43981E37F0C)

##### [>>](marginnote3app://note/7FDC30B5-93D9-4EFD-AEB1-B0582BEDBCE9)

### Substitution [>>](marginnote3app://note/8AACAE88-1002-4B15-8B63-FAF343B4ED7E) A substitution is a syntactic transformation on formal expressions.

#### [>>](marginnote3app://note/09637CE7-FB4E-44E8-B52A-069EFBDAEDE3)

#### [>>](marginnote3app://note/738863D8-1DCA-4EEE-A0AE-5774CA2CA59C)

#### [>>](marginnote3app://note/C3394532-295A-44F4-954D-FB242D163991)

把一个等价的命题中的一小部分等价替换 得到的新命题和原来等价。

#### [>>](marginnote3app://note/BE841DCC-DDD8-4222-A5E5-1697719B3C2D)

# I2ML\_05\_PL\_entailment [>>](marginnote3app://note/53B8131F-7F6C-46A5-A8DE-4B6A31A26588)

## Semantic Entailment [>>](marginnote3app://note/0FFFD93B-6B2D-42EF-A1AE-A7861F772A6E)

### Definition [>>](marginnote3app://note/E9A3C52D-711F-4300-8E68-22CE7E8163FC) Let Σ be a set of formulas (Σ ⊆ Form(Lp)), A be a formula (A ∈ Form(Lp)). We say: • A is a logical consequence (逻辑推论) of Σ, or • Σ (semantically) entails (逻辑蕴含) A, or • Σ ⊨ A if and only if For all truth valuation v, if Σv= 1 then Av= 1.

### We use Σ ̸⊨ A to denote “not Σ ⊨ A”, which is: There exists a truth valuation v such that Σv= 1 and Av= 0. [>>](marginnote3app://note/3C63AB7F-22AE-414F-9E49-CED29E952D1C)

### How do we prove Σ ⊨ A? [>>](marginnote3app://note/FA32D819-BA72-4903-8A22-A14C44D56BF5) • Direct proof: For every truth valuation under which all of the premises are true, show that the conclusion is also true under this valuation. • Using a truth table: Consider all rows of the truth table in which all of the formulas in Σ are true. Verify that A is true in all of these rows.

### Proof by contradiction: Assume that the entailment does not hold, which means that there is a truth valuation under which all of the premises are true and the conclusion is false. Derive a contradiction. [>>](marginnote3app://note/D0C2BF58-1E95-4B76-852E-B2CAFD09B9A7)

先假设不行 则必有 Σ v = 1 and A v = 0 找到对应的取值 接着让右侧满足得到一组取值，接着向左回代得到所有取值推出矛盾

### How do we prove Σ ̸⊨ A? [>>](marginnote3app://note/5AD4D1D6-0650-408F-8AB8-6DA7443C674A) • Find one truth valuation v under which all of the premises in Σ are true and the conclusion A is false.

### [>>](marginnote3app://note/48CF967E-A0CC-4652-A993-EF9DDC85E1EC)

## Logical Connectives [>>](marginnote3app://note/080EB2F5-F182-402B-BA82-C728BEC19F28)

### [>>](marginnote3app://note/0986224F-F510-4F4B-B50B-0F5393AB81F4)

#### [>>](marginnote3app://note/60D58F1E-D3BE-4E0E-8C04-7ED3D1A7391B)

#### [>>](marginnote3app://note/C4DBE300-796F-4B09-811E-240D3DDBA67F)

##### • g9: Joint denial (或非词, neither...nor...), denoted by ↓, p ↓ q ≡ ¬(p ∨ q) • g15:Alternative denial (与非词, not both), denoted by ↑, p ↑ q ≡ ¬(p ∧ q) • g7:Exclusive disjunction (异或词, either...or...), denoted by ⊗, p ⊗ q ≡ ¬(p ↔ q) [>>](marginnote3app://note/DFA45CD4-94CF-4DBA-B848-D21AF3575B04)

### Adequate Set [>>](marginnote3app://note/811E7DDE-9AEA-4C03-89DB-D8144DFBBF8B) A set of connectives is said to be adequate (完备的) iff every well-formed formula is logically equivalent to a well-formed formula using only connectives from the set. Or, every n-ary connectives is definable in terms of only the connectives from the adequate set.

#### [>>](marginnote3app://note/879AFF01-66A1-4091-9677-9D06119291A9)

#### [>>](marginnote3app://note/1F2C16C2-374F-40B2-87F1-5498FEF5EE70)

# I2ML\_06\_PL\_Hibert [>>](marginnote3app://note/4AE5CDE5-D4AC-431F-A5F6-D886051BB9BC)

## Overview of Proof Systems [>>](marginnote3app://note/45798B53-8BD9-4969-9584-C604B5F5582A)

### A formal proof system (deductive system, 形式推演系统) consists of the language part and the inference part. • Language part — Symbols, alphabet — Set of formulas • Inference part — Set of axioms — Inference rules [>>](marginnote3app://note/985CAA02-61BF-438A-A01E-24084C08B15E)

### • Hilbert-style system (Σ ⊢HA): many axioms and only one rule. The deduction is linear. • Natural Deduction System (Σ ⊢NDA): Few axioms (even none) and many rules. The deductions are tree-like. • Resolution (Σ ⊢ResA): used to prove contradictions. [>>](marginnote3app://note/7FA88806-49C6-41EC-8872-A90E5DC90D52)

#### We notate ”there is a proof with premises Σ and conclusion A” by Σ ⊢ A [>>](marginnote3app://note/8A67955B-FA4E-47EC-96FE-BBC96F2E4969)

Proves a or a proves

## The Hilbert-style Proof System [>>](marginnote3app://note/994A0955-6F61-40EC-913B-693D10A3C656)

### Language [>>](marginnote3app://note/2CE51A7C-1A8B-4902-93F1-ABD74A8097AB)

#### [>>](marginnote3app://note/2D6E47D4-74AD-4AD7-B509-9F206500B3EE)

#### [>>](marginnote3app://note/4F910DED-A5E7-45F0-A381-08332499DFB3)

### Axioms & Inference Rules [>>](marginnote3app://note/2B7EC3E4-AEB4-4E32-B73F-D55E9F238F84)

公理

#### [>>](marginnote3app://note/E3E2CB51-8C81-4ADE-BC36-AEA46EC0A4A3)

### Theorem [>>](marginnote3app://note/7FF69D34-C3B0-4301-8A44-4A684A38A50C)

#### [>>](marginnote3app://note/D0F5A2E6-07E9-4644-BD72-81DFFBC70922)

#### [>>](marginnote3app://note/C52C3A98-B988-4210-A121-0CFF135D14F5)

#### [>>](marginnote3app://note/2185E327-B48A-47DA-B3C7-D1D620911B91)

#### [>>](marginnote3app://note/E388F6B6-CA44-4832-8876-B6B65C27DFEC)

这句话的意思是说，如果你能证明一个命题和他的否命题同时成立，那么任意命题B都成立

#### [>>](marginnote3app://note/501460DB-9F50-4B7F-9367-48AB9CBAA455)

#### [>>](marginnote3app://note/5EF0D5C1-FE2E-45D9-B4D6-39C78B134A22)

#### [>>](marginnote3app://note/34F85643-D8D6-4E45-AAAA-46C4EF1316A1)

#### [>>](marginnote3app://note/E612633E-88CB-48A3-854F-8DE3D75B2674)

### Derived rules in H [>>](marginnote3app://note/FAA6D930-F6ED-4047-96A3-E5337AC39A43)

#### [>>](marginnote3app://note/16BDA93C-8863-4489-AA61-6B0CE063002A)

#### [>>](marginnote3app://note/C29FC342-D83A-4989-B76E-0E4E2A1F199C)

它主要的用途是将左面的公式移到右边

#### [>>](marginnote3app://note/AACA6860-B4C7-449A-AFCC-83F115BAD7BD)

#### [>>](marginnote3app://note/B611C1FA-D0E3-422D-940E-84AE7AEB013E)

#### [>>](marginnote3app://note/5819CD43-D54A-45C3-84A9-70617E1346B1)

#### [>>](marginnote3app://note/28D23950-43E8-4A7F-B2B3-65F10BFA298D)

## [>>](marginnote3app://note/9E8411B2-9106-4CC3-B222-0E0B4236E851)

## Formal Proof [>>](marginnote3app://note/10587B5D-0C0C-4D46-A5B9-5C302373A0AA) A formal proof is a “logical chain” from assumptions to conclusions. • First, the “chain” must be finite. • Second, each “link” in the ”chain” may be: — Axioms (common sense) — Assumptions (premises) — Intermediate conclusions derived by using inference rules.

### [>>](marginnote3app://note/D61FF5F1-E83C-49AF-85A9-5BB45C4D2B8D)

证明序列的每一步可以是一个公理(Axiom) 也可以是一个定理

### [>>](marginnote3app://note/A9AFB890-F24B-4C8C-AD6D-564D48CD8B7D)

一个横：形式推演
两个横 逻辑推论

# I2ML\_07\_PL\_ND [>>](marginnote3app://note/8790CBC2-62C2-4AC5-9076-1B8ED6FFAED6)

## Language [>>](marginnote3app://note/4EC1FA61-6F35-46BC-AAF0-D636D8261357)

### [>>](marginnote3app://note/34025BBC-20F6-4DD9-B172-CD41EC900228)

## Inference Rules [>>](marginnote3app://note/31DCFE28-9C54-4598-85A1-D77F25CC56CD)

### [>>](marginnote3app://note/60806487-8D8B-4513-A4AF-1C7A9E27FFEA)

### For each logical symbol, the rules come in pairs. • An “introduction rule” adds the symbol to the formula. • An “elimination rule” removes the symbol from the formula. [>>](marginnote3app://note/D5BD5295-FEB0-423D-A783-2B6448D14A90)

#### Rules for Conjunction [>>](marginnote3app://note/3FFB8358-F9EE-4A30-BCB9-3E6203468B89)

##### [>>](marginnote3app://note/9BA906F1-BFB1-4394-8836-519143B992E9)

#### Rules for Implication [>>](marginnote3app://note/C470E996-20C0-4324-BAC5-8B85CE6A760F)

##### [>>](marginnote3app://note/F5F81F05-6903-41C7-8E1B-4A3FE04581FE)

##### [>>](marginnote3app://note/79D2D7B0-1A15-458C-901E-BB70D16485CB)

#### Inference Rules for Disjunction [>>](marginnote3app://note/07E30F9D-9705-4633-83E5-3A331F44D9A3)

##### [>>](marginnote3app://note/14818EFB-A4E1-4C23-965A-E45A8535BD6B)

#### Inference Rules for Negation [>>](marginnote3app://note/FFE61B38-16E1-46BB-AC1A-8992AA86A3FA)

##### [>>](marginnote3app://note/3D43D8C4-9692-404C-A12A-F82EFF6B4EDE)

##### [>>](marginnote3app://note/4060DF50-E25F-48A2-99DE-02ECB6CE13A7)

##### [>>](marginnote3app://note/63FC1EEE-0396-46F5-9046-93C6B072A4B6)

##### [>>](marginnote3app://note/64EC4152-9234-46C9-AAD2-BF5678CEF9B5)

### Derived Rules [>>](marginnote3app://note/8DE04BE2-CC69-4CED-A1C6-2E3EF61E4593)

#### [>>](marginnote3app://note/3A09CA08-5CAA-4011-9F2F-3B176C068025)

##### [>>](marginnote3app://note/59DE8BDE-4B11-4E7D-B6F8-9691EF2BF2C0)

#### [>>](marginnote3app://note/8F9389B9-9170-4E74-B71C-FC68A5CE7258)

#### [>>](marginnote3app://note/583CABC7-021C-4AC9-A50D-97CB872D960E)

#### [>>](marginnote3app://note/2695BC26-FAA1-4215-9B09-00B5D5D0C967)

# I2ML\_08\_PL\_Resolution [>>](marginnote3app://note/20B9DC98-7CE7-42DB-AEF8-D8BE980112B5)

## Normal Form [>>](marginnote3app://note/72B16E79-B91D-4933-B06A-8B08DEC1A7DF) a standardized representation of logical formulas

### [>>](marginnote3app://note/A44C3DED-FEDA-4CE3-8646-F021CC867040)

### [>>](marginnote3app://note/DFD2A0B4-D259-411C-A8C2-62DE8DFC7FA1)

### [>>](marginnote3app://note/40723BF1-611D-444A-8FA6-473731A5087E)

### Recipe [>>](marginnote3app://note/ACBFBEA9-A4ED-44BC-A72C-969DE7364C09)

#### [>>](marginnote3app://note/4C0C97E8-B346-4239-9C93-811A17C10054)

### [>>](marginnote3app://note/60417BA2-EABD-4E5D-99F9-0518DC1A30F8)

## Resolution (归结/消解原理) [>>](marginnote3app://note/FABD7167-304F-45C7-A4A9-8E547F912217) Resolution (归结/消解原理) is one of the most widely used systems for computer-aided proofs. It has two distinctive features. • It applies only to formulas in CNF. Thus we do some preliminary work before starting an actual proof. • It is used to prove contradictions. That is, a proof aims to conclude a special “contradiction formula” ⊥. For this reason, Resolution is sometimes called a refutation (反驳) system.

### Inference Rules [>>](marginnote3app://note/FAABE2CA-26AF-46CB-9BBD-2DF66583B4B2)

#### [>>](marginnote3app://note/02861179-C460-451C-AD7C-917556C91D96)

#### [>>](marginnote3app://note/3A521777-CDCB-4FC3-903E-0DD1A4C97277)

### The Resolution Proof Procedure [>>](marginnote3app://note/C2D4E250-AB28-423B-9B61-89FAA6709DBD)

#### [>>](marginnote3app://note/2A0C1200-03EE-4FBA-A931-5E0BCB066F3F)

#### Set notation [>>](marginnote3app://note/1CD531C6-C38D-431E-947A-1E33D2823F48)

# I2ML\_09\_PL\_Soundness\_Completeness [>>](marginnote3app://note/8C5FF155-B4CA-47EC-8A5D-EC377E00EB9D)

## [>>](marginnote3app://note/DCBA1D69-9755-40EF-B32B-EEFD448FB510)

## [>>](marginnote3app://note/BA2B04C1-EE40-452D-AA8E-B53ACBAE6A38)

# I2ML\_10\_FOL\_syntax [>>](marginnote3app://note/87D967E8-7562-42A8-9850-E8A55B557D32)

## Basic Concepts of FOL [>>](marginnote3app://note/379DC0C9-3DAE-4237-8EB0-B54A4459F9B5)

### Domain [>>](marginnote3app://note/05595829-112D-46F2-ACEB-00EA60AE8DA2) A domain (论域) is a non-empty set of objects. It is a world that our statement is situated within.

#### The same statement can have different truth values in different domains. [>>](marginnote3app://note/040876AB-9AC0-429F-A409-29AAD541BB32)

### Constants [>>](marginnote3app://note/9E51D936-BBBF-4A9E-BE2B-7EC13D7AE38D) Constants: concrete objects in the language (i.e., domain elements) • Example ?: Constants in “Alice is married to Jay and Alice is not married

### Variables [>>](marginnote3app://note/79020B84-3F9B-4176-AF0A-E4FDBA6F7F48) Variables: “place holders” for concrete values. • Variables are written u, v, w, x, y, z, ... or x1, y3, u5, ... • A variable lets us refer to an object without specifying which particular object it is (e.g., a student).

### Predicates [>>](marginnote3app://note/898328AB-BA02-4280-943C-D83834BEE4D4)

#### • A predicate (谓词) represents: — A property of an individual object in the domain, or — a relationship among multiple individuals [>>](marginnote3app://note/78D50231-AE85-4913-AD4E-81C9C5276CB5)

#### • A predicate can have a different number of arguments. S and I have just one (unary predicates), Y has two (binary predicate). [>>](marginnote3app://note/BF18FE5F-2078-450F-AF2C-6AFDAB016CDA)

### Quantifiers [>>](marginnote3app://note/20F86B40-14EE-481F-A50E-90B6D82DF255)

#### Quantifiers (量词): the quantity of objects • The universal quantifier ∀ (全称量词): the statement is true for every object in the domain. • The existential quantifier ∃ (存在量词): the statement is true for one or more objects in the domain. [>>](marginnote3app://note/06E2841C-5D6E-4611-A533-5D24088073CD)

### Functions [>>](marginnote3app://note/FDE3C933-A0A7-4F5C-8204-61EB236847E9)

#### The symbol m is a function symbol: a function has arity n and sometimes denoted as f(n). • In the example, m is a unary function: it takes one argument and returns the mother of that argument. [>>](marginnote3app://note/1E8EE95E-5C00-4183-AC36-52BD3C827BF7)

## FOL as a Formal Language [>>](marginnote3app://note/5604DC56-9AC6-48A4-95E8-5EA4E8F48FBF)

### Alphabet [>>](marginnote3app://note/AD4553DC-6E84-4EE2-B335-AA7EA3A084CF)

#### [>>](marginnote3app://note/6F3643EF-C6D0-4C1E-84F0-5B46B3E91E03)

#### [>>](marginnote3app://note/E306FE19-3F4C-407A-A8EE-1C22C4D4A28A)

### Terms [>>](marginnote3app://note/91D90B3C-9499-435A-BCCA-E1E63FE907AA)

#### [>>](marginnote3app://note/77A50745-C664-4396-AC7D-22B2183B6A16)

### Atoms [>>](marginnote3app://note/6A47B83B-69D5-457B-957C-D2B9D2373C66)

#### [>>](marginnote3app://note/73E1CB00-E5AC-401D-A9C2-8EB13DFDD0E9)

### Formulas [>>](marginnote3app://note/B9FDBF78-50B2-47EB-8164-CBD0B8BDEA2C)

#### [>>](marginnote3app://note/DBAAE524-959D-49F2-8F6F-AEC10D50D1CF)

### Precedence and Conventions [>>](marginnote3app://note/5487C118-3240-4370-9CC7-03383E512BFD)

#### [>>](marginnote3app://note/75B08ED0-3D02-449E-8490-EDC826003E1E)

### Parse Trees [>>](marginnote3app://note/2EC4FF83-D526-4020-AB33-F8DA71751E49)

#### [>>](marginnote3app://note/154E3EA9-A774-4290-A0E8-DA02D06C04ED)

### [>>](marginnote3app://note/9FE53405-FB85-4129-871E-241E4BDFB100)

#### [>>](marginnote3app://note/4D56BA5E-957E-4300-94A4-522F6FEC44DE)

##### [>>](marginnote3app://note/92D0B879-C463-417D-A910-EFA37EEDCE36)

# I2ML\_11\_FOL\_semantics [>>](marginnote3app://note/1665334B-EC3E-4F2B-B87A-9727BB2396F8)

## Scope [>>](marginnote3app://note/D1C1E4C4-F310-4FB6-8E2F-0108FD7C922A)

### [>>](marginnote3app://note/27808A8D-2B9F-462A-8BAD-3A7E78A9E8EF)

## Free and Bound Variables [>>](marginnote3app://note/6C358A38-2F09-4CEA-8759-13F544F3FD67)

### [>>](marginnote3app://note/DD5CAD10-E7C8-4437-8F54-644D99AC42D8)

## Sentence [>>](marginnote3app://note/32C4A1FC-77F4-4D85-9DAF-847F6078AE28)

### [>>](marginnote3app://note/D3AEE161-2089-4136-B255-EB1C2CDFD6D0)

## Closure [>>](marginnote3app://note/ADA7DBB9-8862-4926-ACCE-390CFE697094)

### [>>](marginnote3app://note/203123BA-FEAF-4EE7-8EA6-D6D828F50A56)

## interpretation [>>](marginnote3app://note/F0EB0897-162D-4A45-8354-E1A9199C3435)

### An interpretation I (or structure) consists of: • A non-empty set D, called the domain (or universe) of I. • For each constant symbol c, a member cIof D. • For each function symbol f(i), an i-ary function fI. • For each predicate symbol P(i), an i-ary predicate (relation) PI. [>>](marginnote3app://note/20E56A54-F3E2-400B-8EB3-B9A6DEA3A490)

## Environment [>>](marginnote3app://note/1BBA983A-A1EA-4241-A2B6-AC476CBA903F)

### [>>](marginnote3app://note/0EBAD924-709E-4488-99C7-A66716800DEA)

## The value of terms [>>](marginnote3app://note/5ABB2681-8D2C-4CF6-948D-82A1737E58D5)

### [>>](marginnote3app://note/2D2CF56F-3548-4840-8F79-1ED24D8B6628)

## The value of atomic formulas [>>](marginnote3app://note/962F02AF-E975-4E56-9E91-B80C1FEB5EF3)

### [>>](marginnote3app://note/CE06A9E8-7439-48AC-8F7F-3229E7205BB1)

## The value of well-formed formulas [>>](marginnote3app://note/D9494715-D98E-4772-A0B5-5662D69C7425)

### [>>](marginnote3app://note/CE5038FF-062E-4F1B-862F-0D6D5877A78F)

### [>>](marginnote3app://note/23AE642D-A4EC-4C82-969F-CAE84B209979)

### [>>](marginnote3app://note/1295561A-164A-42B5-A167-13C03FB7208B)

### [>>](marginnote3app://note/5230559F-D4EE-40CE-9040-C9E6AAA8B930)

### [>>](marginnote3app://note/9A0CF798-2796-414A-BE6F-04C02AF07921)

# I2ML\_12\_FOL\_entailment [>>](marginnote3app://note/03409E80-DE43-478E-B8D8-3C1B59410FAB)

## Satisfiability and Validity [>>](marginnote3app://note/EB5F442D-D97B-4E1E-ABDC-68021DA3796A)

### E satisfy a formula α, denoted I ⊨Eα, iff α(I,E)= 1. They do not satisfy α, denoted I ?⊨Eα, if α(I,E)= 0. [>>](marginnote3app://note/627CEED2-B117-47DC-B89B-F365C55B6FA8)

### Validity and Satisfiability [>>](marginnote3app://note/5C8D1D09-2C99-4F51-AFD1-6F96517F3D09)

#### [>>](marginnote3app://note/E9DB2407-6512-4235-8EB7-2AE9E5A64C2B)

## Semantic Entailment [>>](marginnote3app://note/B24069F6-66D2-469E-BA17-2E40502E99B3)

### [>>](marginnote3app://note/671960B3-8608-4081-B01E-6C55640FEF8E)

# I2ML\_14\_ProgramVerification [>>](marginnote3app://note/4FCC8D1B-5731-48E8-B142-1C106B429E57)

## Program Verification [>>](marginnote3app://note/28EE118C-4E1D-4537-8B28-928259EBBCD8)

### [>>](marginnote3app://note/503A8D29-57E3-4EAA-9C65-82046BE94E69)

### [>>](marginnote3app://note/DEE2FB7C-F2A2-4395-83BC-C69772021AF2)

### Formal specification [>>](marginnote3app://note/7CB4F9FA-5E79-4419-B21A-4EB99B6D0DAA)

#### [>>](marginnote3app://note/EF49E7E8-F0AD-4D2B-B9C1-A68FBDC4C9D1)

#### [>>](marginnote3app://note/61B3DE1F-295B-4CD9-B461-A0F72D22C163)

### Hoare Triples [>>](marginnote3app://note/C2CBE785-F908-4AD6-B86B-56752FD977E3)

#### [>>](marginnote3app://note/D8A95E65-9DAC-4200-9B65-1AF46285B7AA)

## Partial and Total Correctness [>>](marginnote3app://note/E1B7D5A7-4FBB-4720-AB2E-A37039C7E7CF)

### [>>](marginnote3app://note/8C0C5CF9-6CE0-4C18-9D5A-82C26C4C19DC)

### [>>](marginnote3app://note/FBAFA1BC-B4F9-4A6B-85B5-9D21EA15EB12)

# I2ML\_13\_FOL\_ND [>>](marginnote3app://note/B248A673-E2F4-4E86-AE3D-DC39D20344D4)

## Substitution [>>](marginnote3app://note/7FDFC0AD-FF19-4CBF-A99B-33D7B40A90AC)

### [>>](marginnote3app://note/E5F04007-906C-4875-8537-F590570C85F0)

### 如果出现重名 就对原命题换名 [>>](marginnote3app://note/CEDB0B26-AF47-42A9-B299-6D7ACDE5EACE)

## Natural Deduction for FOL [>>](marginnote3app://note/1706493D-11DF-4CD3-8F24-98277044717F)

### [>>](marginnote3app://note/42E9D562-5171-4BE7-9053-5F1D7F9B8F56)

### [>>](marginnote3app://note/3F216201-2537-492C-B3E2-C5493EE22BED)

### [>>](marginnote3app://note/66E56D1A-9780-4BFC-B621-B77104184402)

#### [>>](marginnote3app://note/1422C255-C61B-4E38-AC2F-E3C874E09EC7)

### [>>](marginnote3app://note/2CE907E9-1819-4453-979C-BB6A3462B7D6)

### [>>](marginnote3app://note/DBE2CE70-2447-4113-80C7-9B124C2BBC64)

# I2ML\_15\_HoareLogic [>>](marginnote3app://note/35E772B9-1328-4E9F-BB11-75FF5AAB1E60)

## Axioms and Inference Rules [>>](marginnote3app://note/C2596F14-5479-489F-BDD0-DA822283DC34)

### rule for assignment [>>](marginnote3app://note/3CEB3703-6D9F-4176-8FDF-FAF7BE149D7B)

#### [>>](marginnote3app://note/CD86E41F-92A0-48F1-A21B-9D63A5300D96)

### implied rule [>>](marginnote3app://note/CE807464-2481-48FA-98FD-95D58A52D538)

#### [>>](marginnote3app://note/F4338890-CDE7-4351-B416-41617131B1D3)

### Composition [>>](marginnote3app://note/9D9E9BC5-B765-49D9-A648-184C9031C05F)

#### [>>](marginnote3app://note/696D99DF-DDA8-44AA-9939-69DF9F92175D)

### If statements [>>](marginnote3app://note/C88C958A-D165-41A8-AC46-F62E673537BC)

#### [>>](marginnote3app://note/6092A384-574D-4F53-AD9A-8BE5F00F2829)

### While statements [>>](marginnote3app://note/EED24915-F483-4ECC-8DFD-E3686F0E5B8C)

#### [>>](marginnote3app://note/8E2814B1-CF4C-4DE0-99EF-AD16990AD3A5)

### A note on loop invariant [>>](marginnote3app://note/8BE11495-CB72-4AA9-9905-7F768AD66756)

#### [>>](marginnote3app://note/40B90774-E4B8-4B70-9AB2-D7C1CA555B79)

# Made By @izumidonabe [>>](marginnote3app://note/D742B9F5-44F0-4D2F-8218-C6003011C917)

## Open Source On Github [>>](marginnote3app://note/72513553-939D-4661-9584-5BCF946D9DD3)

### https://github.com/izumidonabe/SUSTech-CS104-I2ML-Notes/ [>>](marginnote3app://note/A3664567-B08C-4AD8-8DC3-7086E8DEF413)