Symbolic Logic

First-Order Predicate Logic

Required Text: Yaqub, *Introduction to Metalogic*

1.1 The Syntax of Predicate Logic (*PL*)

- The **vocabulary** of *PL* consists of the following six categories.
- **1.1.1a Names**: The following lowercase italic letters: *a, b, c, …, r, s, t* (excluding *f, g*, and *h*) with numeric subscripts if needed.
- **1.1.1b Function symbols**: The following lowercase italic letters with numeric subscripts: f_1 g_1, h_1, f_2, g_2, h_2, f_3, g_3, h_3, ...
- **1.1.1c Predicates**: Uppercase italic letters with numeric subscripts: A_1, B_1, C_1, …, X_1, Y_1, Z_1; A_2, B_2, C_2, …, X_2, Y_2, Z_2; A_3, B_3, C_3, …, X_3, Y_3, Z_3; ….
- **1.1.1d Variables**: The following lowercase italic letters: *u, v, w, x, y, z* with numeric subscripts if needed.
- **1.1.1e Logical symbols**: ¬, &, v, →, ↔, ∀, ∃, =
- **1.1.1f Parentheses**: (,)

Object Language & Metalanguage

- The variables, logical symbols, and parentheses are referred to as the **logical vocabulary** of *PL*, and the names, function symbols, and predicates are referred to as the **extra-logical vocabulary** of *PL*.
- The language of *PL* generally is our **object language**, the language under study. Our **metalanguage**, the language we use to conduct the study, is English supplemented with various mathematical symbols.
- In order to talk about, i.e. **mention**, a symbol, like *f*, we should clarify that we are not **using** it (to talk about a function). In principle, we may place the symbol is quotation marks. In practice, the context makes it clear when we are mentioning rather than using a symbol.

Variables & Metavariables

- The variables in the language of *PL* range over objects in the *universe of discourse*, or *domain*, of our object language (to be defined). But we use variables in our metalanguage to speak of, e.g., all predicates.
- To speak of all *PL* predicates, function symbols, or variables, we use **metavariables** (i.e., variables in the metalanguage).
- The boldfaced letters ***P***, ***Q***, and ***R***, perhaps adjoined with numeric superscripts, range over *PL* predicates; the boldfaced letters ***f*** and ***g***, possibly adjoined with numeric superscripts range over *PL* function symbols; the boldfaced letters ***x***, ***y***, and ***z*** range over *PL* variables, and the boldfaced capital letters, ***X***, ***Y***, and ***Z*** range over *PL* formulas.

Intuitive Meaning of Non-Logical Symbols

- The non-logical symbols of *PL* are intuitively understood as follows. Names stand for individuals (like *George Washington*), function symbols stand for functions (like *the father of*), and predicates stand for properties and relations (like *is red* or *is to the left of*).
- Names are examples of **singular terms**. If we flank the name in *PL* for *George Washington* with the function symbol in *PL* for *the father of*, for example, the resulting term is the name of *George Washington's father*. This is an illustration of a **complex** singular term. A function symbol followed by a variable is an example of a **functional term**.
- *Note*: Singular, but not functional, terms are said to denote things.

Intuitive Meaning of Non-Logical Symbols

- Combining non-logical symbols with logical symbols results in complex expressions that correspond to *complex properties*, *relations*, or *states of affairs*, like the **statement** or **proposition** or **fact** *that Mary is running*.
- Predicates and function symbols have **arities**, as in A_1^3 and , f_2^3. *1-place* predicates signify **properties**, *2-place* predicates, **binary relations**, *3-place* predicates, **ternary relations**, and so on. *1-place* function symbols signify **monadic functions**, *2-place* symbols **binary functions**, 3-place symbols, **ternary functions** (*the product of x, y and* $\sqrt[3]{z}$), and so forth.
- *Note*: We often write A_1xyz or f_2^3xyz instead of A_1^3 or f_2^3 for readability.

Intuitive Meaning of Logical Symbols

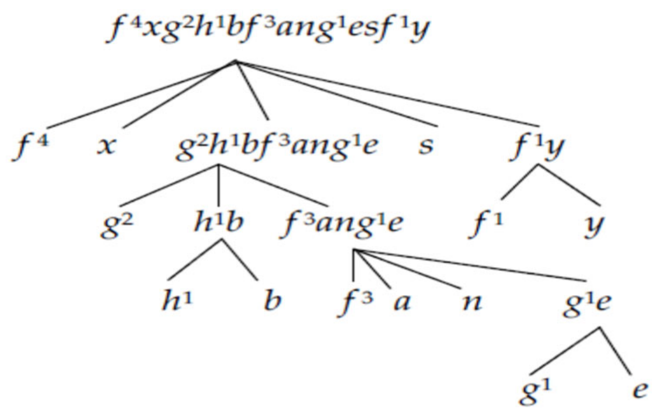
- The logical symbols signify the following **operators**:
- ¬ =: *negation* (‘it is not the case that’)
- ∧ =: *conjunction* (‘and’)
- ∨ =: *disjunction* (‘or’)
- → =: *material conditional* (‘if…then’)
- ↔ =: *material biconditional* (‘if and only if’)
- ∀ =: *universal quantification* (‘for all’)
- ∃ =: *existential quantification* (‘there is’)
- = =: *identity* (‘is numerically identical to’).

Terms

- A **term** of *PL* is either a *PL* name, variable, or complex expression that is obtained from the names and variables by applying the following rule some finite number of times:
- **Term-Formation Rule**: If f^n is an n-place function symbol and t_1, t_2, …, and t_n are *PL* terms, then $f^n\, t_1 t_2, \ldots\, t_n$ is also a *PL* term.
  - The variables here are metavariables. The *Term-Formation Rule* says that the result of flanking *any* n-place function symbol with *any* n terms is a term.
- *N.B*. Terms are not, in general, singular terms. So, under the intended interpretation, they typically do not denote. f^1x is a term with a ‘blank’.

Term Trees

- We can show that $\boldsymbol{f^4xg^2h^1bf^3ang^1esf^1y}$ is a *PL* term as follows:



Formulas

- Formulas, like terms, are also either basic or derived. We will call basic formulas **atomic** and the derived formulas **compound**.
- **Atomic formulas** of *PL* are expressions of the form $\boldsymbol{r = s}$ where $\boldsymbol{r}$ and $\boldsymbol{s}$ are (atomic or complex) *PL* <u>terms</u>, and expressions of the form $\boldsymbol{Q^n t_1 t_2 \ldots t_n}$ where $\boldsymbol{Q_n}$ is any *n*-place *PL* predicate (except for the identity predicate =) and $\boldsymbol{t_1 t_2 \ldots t_n}$ are <u>at most</u> *n* distinct *PL* terms.
- **Compound formulas** of *PL* are obtained from the atomic formulas by applying one or more **formation rules** a <u>finite</u> number of times.

Recursion Clauses

- Compound formulas are constructed from atomic ones by applying any of seven formation rules finitely-may times. If ***X*** and ***Y*** are *PL* formulas:
- (~): ~***X*** is a *PL* formula.
- (**&**): (***X & Y***) is a *PL* formula.
- (**v**): (***X v Y***) is a *PL* formula.
- (**→**): (***X → Y***) is a *PL* formula.
- (**←→**): (**X ←→ Y**) is a *PL* formula.
- (∀): If ***X*** contains occurrences of the *PL* variable ***z*** but no *z*-quantifiers, then (∀*z*)***X*** is a *PL* formula.
- (∃): If ***X*** contains occurrences of the *PL* variable ***z*** but no *z*-quantifiers, then (∃*z*)***X*** is a *PL* formula.

Use and Mention

- We have been fudging -- and will continue to fudge -- the distinction between *using* a symbol and *mentioning* it. Strictly speaking, the claim that if ***X*** is a formula, then so is ~***X*** is nonsensical. '~***X***' says that it is not the case that ***X***, where ***X*** is a variable, i.e., an example of a singular term.
- What we intend is that if ***X*** is a well-formed formula, then so is *the formula obtained by prefixing the formula denoted by* ***X*** *with the negation symbol, '~'*. That is, we mix a use of '***X***' with a mention of '~'. Since the symbol '***X***' is not in the *PL* language, we cannot express this as: if ***X*** is a *PL* formula, then so is '~ ***X***'. This would imply that since, e.g., (P_1 *v* ~~P_{211}) is a *PL* formula, *the negation sign followed by '**X**' is a PL formula as well*.
- If we wanted to be very careful (more careful than virtually any mathematical logic text), we would use **Quine quotes**. The following says exactly what we intend: If ***X*** is a well-formed formula, then so is ⌜ ~***X*** ⌝.

Types, Tokens, and Occurrences

- We will also be careless about the distinction between symbol types, tokens, and occurrences. A type is a kind of *universal* -- a multiply-instantiatable entity -- like the property, *the color red* or the relation, *being to the right of*. Logic is about symbol *types*. When we say that the formula (*) has *one bound variable*, we mean that the formula *type* does, not that a particular instance – i.e., *tokening* -- of the type does: (*) (∀x)((*F*x v *G*x) & *H*y)
- Now consider the claim that the universal quantifier in (*) binds *two occurrences* of (the one variable) *x*. This is about formula types. So, whatever occurrences are, they 'inhere' in types, not their tokens (like the string of symbols above). But there are *two* of them, and only one (universal) letter *x*. Thus, even the ontology of symbols is vexed!
- We will happily ignore these nuances, like (literally!) all logic textbooks.

Example of Formula Construction

- Consider the string: $\sim(((\sim A^1z \vee B^3xay) \leftrightarrow K^2zw) \,\&\, D^1y)$
- This is a *PL* formula, because we can construct it from atomic ones using the seven recursion clauses as follows:
  - A^1z, B^3xay, K^2zw, and D^1y are all atomic formulas.
  - $\sim A^1z$ is a formula by (~)
  - $(\sim A^1z \vee B^3xay)$ is a formula by (v)
  - $((\sim A^1z \vee B^3xay) \leftrightarrow K^2zw)$ is a formula by ($\leftrightarrow$)
  - $(((\sim A^1z \vee B^3xay) \leftrightarrow K^2zw) \,\&\, D^1y)$ is a formula by (&)
  - $\sim(((\sim A^1z \vee B^3xay) \leftrightarrow K^2zw) \,\&\, D^1y)$ is a formula by (~)

Formulas vs. Sentences

- *A **sentence** is a **formula** that contains **no free variables**. A formula that is not a sentence is called an **open formula** or **open sentence**.*
- A(n occurrence of) a variable in a *PL* formula is **free** if it is not **bound**. A(n occurrence of) a variable ***z*** in a *PL* formula is **bound** when it is inside a quantifier in the formula (as in (∀*z*)) or inside the **scope** of a ***z***-quantifier in the formula. The **scope** of a quantifier is the shortest (in terms of number of symbols from the *PL* vocabulary) formula that immediately follows the quantifier. Exactly one formula must immediately follow a(n occurrence of) a quantifier in a formula.

Illustration

- The *formula*, $\sim(((\sim A^1z \vee B^3xay) \leftrightarrow K^2zw) \,\&\, D^1y)$, is not a *sentence* because it contains free variables – namely, *z, x, y,* and *w* (*a* is a name).
- However, we could transform it into a sentence via four applications of quantifier rules. For instance, the following is a sentence:
  - $(\forall z) \sim(((\sim A^1z \vee B^3xay) \leftrightarrow K^2zw) \,\&\, D^1y)$ (∀)
  - $(\exists x)(\forall z) \sim(((\sim A^1z \vee B^3xay) \leftrightarrow K^2zw) \,\&\, D^1y)$ (∃)
  - $(\forall y)(\exists x)(\forall z) \sim(((\sim A^1z \vee B^3xay) \leftrightarrow K^2zw) \,\&\, D^1y)$ (∀)
  - $(\exists w)(\forall y)(\exists x)(\forall z) \sim(((\sim A^1z \vee B^3xay) \leftrightarrow K^2zw) \,\&\, D^1y)$ (∃)

Immediate Components

- We will say that ***X*** is the **immediate component** of (∀*z*)***X***, (∃**z**)***X***, and ~***X***, and that (∀*z*), (∃**z**), and ~ are their **main operators**, respectively.
- The **immediate components** of *PL* formulas of the forms (***X & Y***), (***X v Y***), (***X*** → ***Y***) and (**X** ↔ ***Y***) are just ***X*** and ***Y***, and their main operators are the binary connectives ***&***, ***v***, →, and ↔, respectively.
- Finally, if a formula ***X*** occurs in a formula ***Y*** (technically, the string, ***X***, is identical to a substring of the string, ***Y***), then ***X*** is a **subformula** of ***Y***. We say that ***X*** is a **proper subformula** of ***Y*** just in case ***X*** is a subformula of ***Y*** and it is not identical with ***Y***. An **atomic component** of a *PL* formula ***X*** is a subformula of ***X*** that is an atomic formula.

Significance of Quantifier Scope

- The scope of a quantifier helps to determine the **truth-conditions** of the formula in which it appears.
- The existential quantifier in the sentence $(\exists z)(\forall x)S^2zx$ applies to the first 'slot' of the predicate S^2zx and the universal quantifier applies to the second. If we interpret S^2zx to mean that *z* hates *x*, then the sentence says that *there is someone who hates everyone*, while switching the variable order says that *there is someone who is hated by everyone*. Not the same!
  - *Note*: Occurrences of the same variable in different scopes of quantifiers are independent. Thus, $(\exists x)A^1x$ & $(\exists x)B^1x$ is equivalent to $(\exists x)A^1x$ & $(\exists y)B^1y$.

Conventions

- For sake of readability, we will make use of **conventions**, like that of dropping the outermost parentheses in formulas, writing (***s*** $\neq$ ***t***) instead of ~ (***s*** = ***t***), failing to indicate the arity of predicates and function symbols, writing *I* instead of I_Γ for an interpretation for a set of sentences, Γ (Gamma), italicizing or not officially unitalicized or italicized symbols, respectively, as this contributes to readability.
- These are conventions in the sense that what write will abbreviate the expressions whose grammar or meaning we described previously. Thus, for example, (***s*** $\neq$ ***t***) is really an abbreviation for ~ (***s*** = ***t***).

Semantics of *PL*

- A **semantics** for terms and formulas of a language is, intuitively, some specification of **meanings** for those terms and formulas.
- If Γ is a set of *PL* sentences, then the **full vocabulary** of Γ, *Voc*(Γ), is the **extra-logical vocabulary** of which members of Γ are composed plus the **logical vocabulary** of *PL* (the five sentential connectives and quantifiers, the identity predicate, the variables, and the parentheses).
- A *PL* **interpretation**, I_Γ, for Γ, then consists of ***Voc*(**I_Γ**)**, which includes a list of names, *LN*, a **universe of discourse**, *UD,* and **semantical assignments**, *SA*, satisfying the following three conditions.

Semantics of *PL*

- **(1)** *Voc*(I_Γ) includes *Voc*(Γ) and, perhaps, *PL* function symbols and predicates that are not in *Voc*(Γ)
- **(2)** *UD* is a nonempty (perhaps infinite) collection of individuals
- **(3)** *LN* consists of names for all individuals in *UD*, and, perhaps, names that are not listed in **1.1.1a** or complex *PL* singular terms.

- An **interpretation**, (I_Γ), according to which all of the members of Γ are true is called a **model** of Γ.

Semantics of *PL*

- The semantical assignments (*SA*) made by I_Γ are as follows:
- **1.2.1a** I_Γ assigns *exactly one* individual in *UD* to every *name* in *LN*; and every individual in *UD* is assigned by I_Γ to *at least* one such name.
  - The individual I_Γ assigns to the name ***s*** is then the **referent** of ***s*** on I_Γ, $I_\Gamma(\boldsymbol{s})$.
- **1.2.1b** To every *n*-place *function symbol* $\boldsymbol{f^n}$ that belongs to *Voc*(I_Γ), I_Γ assigns *exactly one* *n*-place function on *UD*.
  - The function I_Γ assigns to the *n*-place function symbol $\boldsymbol{f^n}$ is denoted, $I_\Gamma(\boldsymbol{f^n})$.
- **1.2.1c** To every *singular term* $\boldsymbol{f^n\ t_1 t_2, \ldots\ t_n}$, where $\boldsymbol{f^n}$ is an *n*-place function symbol and $\boldsymbol{t_1 t_2, \ldots\ t_n}$ are singular terms, I_Γ assigns the individual $F(\alpha^1, \alpha^2, \ldots, \alpha^n)$, which is unique, where F is the function that I_Γ assigns to the *n*-place function *symbol* $\boldsymbol{f^n}$, α^1 is the referent that I_Γ assigns to $\boldsymbol{t_1}$, α^2 is the referent that I_Γ assigns to $\boldsymbol{t_2}$, …, and α^n is the referent I_Γ assigns to $\boldsymbol{t_n}$.
  - That is: $I_\Gamma(\boldsymbol{f^n\ t_1 t_2, \ldots\ t_n}) = I_\Gamma(\boldsymbol{f^n})(I_\Gamma(\boldsymbol{t_1}), I_\Gamma(\boldsymbol{t_2}), \ldots I_\Gamma(\boldsymbol{t_n})) = F(\alpha^1, \alpha^2, \ldots, \alpha^n)$.

Semantics of *PL*

- **1.2.1d** I_Γ assigns to the *identity predicate*, '=', the binary relation of token identity, which holds between every individual in *UD* and itself and fails to hold between any distinct individuals in *UD.*
- **1.2.1e** To every *1-place predicate*, $\boldsymbol{P^1}$, that belongs to *Voc*(I_Γ), I_Γ assigns just one property (set) on *UD,* $I_\Gamma(\boldsymbol{P^1})$.
- **1.2.1f** To every *n-place predicate* ($n > 1$), $\boldsymbol{R^n}$, that belongs to *Voc*(I), I_Γ assigns just one *n*-place relation on (set of *n*-tuples from) *UD,* $I_\Gamma(\boldsymbol{R^n})$.
- *Note*: The individuals, functions, properties, and relations of a I_Γ are said to be the constituents of I_Γ, and the vocabulary of I_Γ is interpreted by I_Γ. If I_Γ is a *PL* interpretation for Γ, it is described as **relevant** to Γ.

Idealizations

- The Syntax and Semantics of *PL* **idealizes** from ordinary languages, somewhat as models in physics idealize from real physical systems.
  - Any string of symbols in the language of *PL* either is (determinately) a term, formula, or sentence, or it is not. There is no grammatical **vagueness**.
  - Any predicate of *PL* is either true or false of any object, i.e. is **bivalent**.
  - Every *PL* term uniquely **denotes**, i.e., refers to exactly one object, and every predicate and function symbol uniquely **applies**, on an interpretation.
  - Finally, we treat functions and predicates as **extensional** on an interpretation, I_Γ, i.e., as interchangeable with the sets to which they correspond on I_Γ.
- One motivation for non-classical logics is to avoid such idealizations.

Terminology

- The set of all the individuals that have a property P^1 is called the **extension** of P^1, and the extension of a *1-place PL predicate* on any relevant *PL* interpretation I_Γ is the extension of the property that it designates on I_Γ.
- The set of all *n-tuples*, $<a_1, a_2, a_3, \ldots, a_n>$, bearing the relation $I_\Gamma(\boldsymbol{R^n})$ to one another is the **extension** of $\boldsymbol{R^n}$. The extension of an *n-place predicate* on a relevant *PL* interpretation I_Γ is the extension of the relation that this predicate designates on I_Γ. We call the items, $\boldsymbol{a_1, a_2, a_3, \ldots, a_n}$, **coordinates**.
- *Note*: Because we treat functions, properties, and relations as extensional, we speak indifferently of them and their extensions.

Terminology

- An *n-place* **function** on *UD* is a kind of relation - a 'rule' that assigns to every *n-tuple* of individuals in *UD* exactly one individual also in *UD*. The extension of an *n-place function* is thus a set of *n+1-tuples*.
- If *A* is a function on *UD*, then the *n-tuples* to which it assigns individuals are said to be the **arguments** of *A,* and the individuals assigned are said to be the **values** of the corresponding arguments.
- Hence, a (total) *n-place function* $\boldsymbol{F}$ on a set *UD* is an $(n+1)$-place relation such that for every *n-tuple* $<\boldsymbol{a_1, a_2, a_3, \ldots, a_n}>$ of *coordinates* in *UD*, there is *exactly one* individual $\boldsymbol{a_{n+1}}$ in *UD* where the $(n+1)$-tuple $<\boldsymbol{a_1, a_2, a_3, \ldots, a_n, a_{n+1}}>$ is in the extension of the function, $\boldsymbol{F}$.

Example of an Interpretation

- Suppose that we wish to construct an interpretation, I_Γ for the following *PL* sentences with an infinite universe of discourse, *UD,* making each true.
  - **S1** $(\forall x)\ o \neq gx$
  - **S2** $(\forall x)(\forall y)(gx = gy \rightarrow x = y)$
  - **S3** $(\forall x)(x \neq o \rightarrow (\exists z)\ x = gz)$
- Then here is a natural construction:
  - *UD*: The set of all the natural numbers: {*0, 1, 2, 3, 4, …*}
  - *LN*: $0, a_1, a_2, a_3, \ldots, a_n, \ldots$
- Semantical Assignments:
  - $I(o)$: 0; $I(a_1)$: 1; $I(a_2)$: 2; $I(a_3)$: 3; ... (in general, $I(a_n)$: n)
  - $I(gx)$: $x+1$ (that is, $I(gx)$ is the **successor function**)

Informal Reading

- Informally, we have interpreted **(S1) - (S3)** to mean the following:

- **(S1)** *0* is not the successor of any natural number.
- **(S2)** Any two numbers that have the same successor are identical.
- **(S3)** Every natural number that is not *0* is the successor of some natural number.

- Because each of **(S1)** – **(S3)** is true on I_Γ, and the universe, *UD,* is infinite, we have constructed an interpretation of the desired sort.

Substitutional vs. Objectual Interpretations

- The informal reading we gave to **(S1) - (S3)** suggests that the quantifiers range over the universe of discourse of the interpretation, not all things. '$(\forall x)$' means *for all natural numbers*, not for all *period*.
- **Objectual quantifiers** are interpreted in this way. But **substitutional quantifiers** on which we rely hence range over 'basic' *names in LN*.
- The substitutional interpretation of $(\forall z)\boldsymbol{X}$ is *that every 'basic' substitutional instance of* $(\forall z)\boldsymbol{X}$ *is true*; and the substitutional interpretation of $(\exists z)\boldsymbol{X}$ is *that there is at least one 'basic' substitutional instance of* $(\exists z)\boldsymbol{X}$ *that is true*. Let us explain the operative ideas.

Substitution Instance

- The sentence $\boldsymbol{X}[\boldsymbol{t}]$ is a **substitution instance** of the quantified sentence $(\boldsymbol{\Omega z})\boldsymbol{X}$ (where $\boldsymbol{\Omega}$ is the <u>universal quantifier</u> symbol $\forall$ or the <u>existential quantifier</u> symbol $\exists$) just in case $\boldsymbol{X}[\boldsymbol{t}]$ is obtained from $\boldsymbol{X}$ by replacing <u>all occurrences</u> of the variable $\boldsymbol{z}$ in $\boldsymbol{X}$ with the <u>singular term</u> $\boldsymbol{t}$.
- If $\boldsymbol{t}$ is a **name** listed in the *LN* of a *PL* interpretation, I_Γ, that is relevant to $\boldsymbol{X}[\boldsymbol{t}]$, then $\boldsymbol{X}[\boldsymbol{t}]$ is a **basic** substitutional instance of $(\boldsymbol{\Omega z})\boldsymbol{X}$ on I_Γ.
- *Example*: $(ga_1 = ga_2 \rightarrow a_1 = a_2)$ is a basic substitution instance of **(S2)** $(\forall x)(\forall y)(gx = gy \rightarrow x = y)$. According to our definition, $(\forall x)(\forall y)(gx = gy \rightarrow x = y)$ is true on I_Γ just in case every such instance is true on I_Γ.

Substitutional Interpretation

- A ***PL Interpretation***, I_{Γ}, is thus, a triple, <*UD*, *V*(I_{Γ}), *SA*>, where *UD* is the universe of discourse, *V*(I_{Γ}), is the vocabulary of the interpretation -- including the list of names, *LN* -- and *SA* are the assignments.
  - *Note*: Every *PL* interpretation has its own vocabulary, *V*(I_{Γ}), which is not, in general, just the vocabulary of the interpreted set of sentence, *Voc*(Γ)!
- So defined, a *PL* interpretation is **substitutional**. We demand that every member of *UD* has a name from *LN*. Individuals come labeled.
- Substitutional interpretations contrast with **objectual** interpretations, which do not require that the objects of *UD* be labeled. Before we discuss the difference, we specify truth conditions for sentences.

Truth Conditions

- Given the definitions of an interpretation, I_{Γ}, and a basic substitution instance, we can specify **truth conditions** for every sentence of *PL*.
- If ***X*** and ***Y*** are any *PL* sentences and I_{Γ} is a *PL* interpretation relevant to them (i.e., it is an interpretation for a set containing ***X*** and ***Y***), then:
- **1.2.5a** (Atomic Clause a) If ***X*** is of the form ***r*** = ***s*** where ***r*** and ***s*** are *PL* singular terms, then **X** is true on I_{Γ} if and only if $I_{\Gamma}(\boldsymbol{s}) = I_{\Gamma}(\boldsymbol{r})$, i.e., the referents of ***r*** and ***s*** on I_{Γ} are numerically identical (the same individual).
- **1.2.5b** (Atomic Clause b) If ***X*** is of the form $\boldsymbol{P^1 s}$ where $\boldsymbol{P^1}$ is a *1*-place PL predicate and ***s*** is a *PL* singular term, then ***X*** is true on I_{Γ} if and only if $I_{\Gamma}(\boldsymbol{s}) \in I_{\Gamma}(\boldsymbol{P^1})$, i.e., the referent of ***s*** on I_{Γ} has the property that I_{Γ} assigns to $\boldsymbol{P^1}$.

Truth Conditions

- **1.2.5c** (Atomic Clause c) If ***X*** is of the form $\boldsymbol{R^n a_1, a_2, a_3, \ldots, a_n}$, where $\boldsymbol{R^n}$ is an <u>*n*-place *PL* predicate</u> (*n* >*1*) and $\boldsymbol{t_1, t_2, \ldots t_n}$ are (perhaps not distinct) *PL* <u>singular terms</u>, then ***X*** is true on I_Γ if and only if $<(I_\Gamma(\boldsymbol{t_1}), I_\Gamma(\boldsymbol{t_2}), \ldots I_\Gamma(\boldsymbol{t_n}))> \in I_\Gamma(\boldsymbol{R^n})$, i.e., the referents of $\boldsymbol{t_1, t_2, \ldots t_n}$, in the order specified, are related to each other according to the relation that I_Γ assigns to $\boldsymbol{R^n}$.
- **1.2.5d** (Truth Functions) If ***X*** and ***Y*** are <u>any</u> *PL* sentences, then:
  - ~***X*** is true on I_Γ if and only if ***X*** is false on I_Γ.
  - (***X*** & ***Y***) is true on I_Γ if and only if ***X*** is true on I_Γ and ***Y*** is true on I_Γ.
  - (***X*** v ***Y***) is true on I_Γ if and only if ***X*** is true on I_Γ or ***Y*** is true on I_Γ (or both).
  - (***X***→Y) is true on I_Γ if and only if ***X*** is false on I_Γ or ***Y*** is true on I_Γ (or both).
  - (***X***←→***Y***) is true on I_Γ if and only if ***X*** and ***Y*** are both true or both false on I_Γ.

Truth Tables

X	¬X
T	F
F	T

Truth table for ¬

X	Y	X∧Y
T	T	T
T	F	F
F	T	F
F	F	F

Truth table for ∧

X	Y	X∨Y
T	T	T
T	F	T
F	T	T
F	F	F

Truth table for ∨

X	Y	X→Y
T	T	T
T	F	F
F	T	T
F	F	T

Truth table for →

X	Y	X↔Y
T	T	T
T	F	F
F	T	F
F	F	T

Truth table for ↔

The Conditional

- Most of the entries in the truth tables are self explanatory. But one row of the truth table for the conditional, →, might seem puzzling.
- Why do we consider the conditional true when its **antecedent**, ***X***, is false and its **consequent**, ***Y***, is true? There are three reasons.
  - First, (***X*** → ***Y***) must be either true or false given our assumption of **bivalence**.
  - Second, whatever truth-value (***X*** → ***Y***) has in a row must be a function of the truth-values of its constituents, ***X*** and ***Y***, since → is a **truth function**.
  - Finally, → is weaker than ←→. But if we assigned F to (***X*** → ***Y***) when ***X*** was false and ***Y*** was true, then the truth-tables for → and ←→ would be the same.
- *Note*: We have defined → such that (***X*** → ***Y***) is equivalent to (~***X*** v ***Y***).

Truth Conditions Continued

- **1.2.5e** (Universal Quantifier) If ***X*** is of the form (∀y)***Z***, then ***X*** is true on I_Γ if and only if, for every name **s** in *LN*, the sentence ***Z***[***s***] is true on I_Γ, where ***Z***[***s***] is formed by replacing all the occurrences of ***y*** in ***Z*** by ***s*** (i.e., just in case every basic substitution instance of ***Z*** is true on I_Γ).
- **1.2.5f** (Existential Quantifier) If ***X*** is of the form (∃y)***Z***, then ***X*** is true on I_Γ if and only if, for some name **s** in *LN*, the sentence ***Z***[***s***] is true on I_Γ, where ***Z***[***s***] is formed by replacing all the occurrences of ***y*** in ***Z*** by ***s*** (i.e., just in case some basic substitution instance of ***Z*** is true on I_Γ).
- I_Γ **satisfies** Γ and is a **model** of Γ if and only if every *PL* sentence in the set Γ is true on I_Γ. (This is not the same as objectual satisfaction!)

Basic Substitution Instances

- Why does it suffice to specify truth-conditions in terms of basic substitution instances? Because if all the basic substitutional instances of (***Ωz***)***X*** are true on I_Γ, then the formula ***X*** is true of all the individuals in *UD*. But all the individuals in *UD* have names in *LN*, and all singular terms, basic or not, refer to unique individuals in *UD*. Thus if we substitute a singular term for ***z*** in ***X***, we get a true sentence on I_Γ.
- Conversely, if there is some true substitution instance of (***Ωz***)***X*** on I_Γ, then there is some individual in *UD* of which the formula ***X*** is true. (Every singular term has a unique referent on I_Γ). But this individual must have a name in *LN*. Hence, if we substitute this name for ***z*** in ***X***, then we obtain a true sentence on I_Γ. The case of falsehood is similar.

Questions about Substitutional Quantifiers

- **Question 1**: Do substitutional and objectual interpretations agree on the truth-values of all sentences in a *PL* language?
- **Answer**: Yes, but only if every individual in *UD* has a name in *LN*.
- **Question 2**: Some infinite sets, like $\mathbb{R}$, are not **countable**. They cannot be placed in one-to-one-correspondence with (a subset of) the natural numbers. Formal languages are ordinarily assumed to be countable. So, what happens if our Universe of Discourse, *UD*, is **uncountable**?
- **Answer**: We let *LN* be uncountable as well! We require that a formal language, like *PL,* be countable. It is only our **metalanguage** that may not be. However, in what follows, we use a countable metalanguage.

Metalogical Concepts

- Much as there is a distinction between an **object language** (in this case a *PL* language) and a **metalanguage** (in this case English + technical symbols), there is a distinction between **logic** and **metalogic**.
- Confusingly, Symbolic Logic is about the latter. It is about what is a valid argument, what is consistent, what is a logical truth, and so forth -- not (primarily) about *what non-metalogical claims are true*.
- *Example*: The claim *that* $(\forall x)\ x = x$ *is a logical truth* is itself a metalogical claim which is **not** a logical truth. It is about a string of symbols. By contrast, the claim that $(\forall x)\ x = x$ is a logical truth. The claim that $(\forall x)\ x = x$ is about everything (whatever it is) and happens to be such that it cannot -- as a matter of classical logic -- be false.

Metalogical Concepts

- **1.3.1** A *PL* **argument** is a nonempty collection of *PL* sentences, one of which is the **conclusion** and the others of which are its **premises**.
  - If Γ is the set of the premises of an argument whose conclusion is $\boldsymbol{X}$, we write: Γ / $\boldsymbol{X}$. An argument in *PL* has exactly one conclusion. But its set of premises Γ may be empty or contain finitely or even infinitely-many *PL* sentences.
- **1.3.2** An argument Γ / $\boldsymbol{X}$ is **valid** if and only if its conclusion, $\boldsymbol{X}$, is a **consequence** of its set of premises, Γ. We rely on some symbolism.
- **1.3.2a** The string, $\Gamma \models \boldsymbol{X}$, is read: ***X*** *is a logical consequence of* Γ, or ***X*** *logically follows from* Γ, or s Γ *logically implies* ***X***.
- **1.3.2b** The string, $\Gamma \not\models \boldsymbol{X}$ is read: ***X*** *is not a logical consequence of* Γ, or ***X*** *does not logically follow from* Γ, or Γ *does not logically imply* ***X***.

Metalogical Concepts

- **1.3.3a** A *PL* argument Γ / ***X*** is **valid**, so that Γ |= ***X***, if and only if ***X*** is true in every model of Γ that is relevant to ***X*** (that is, on every *PL* interpretation for Γ / ***X*** on which all the members of Γ are true ***X*** is true as well).

Equivalently:

- **1.3.3b** A *PL* argument Γ / ***X*** is **valid**, so that Γ |= ***X***, if and only if there is no model of Γ that is relevant to ***X*** on which ***X*** is false (that is, there is no *PL* interpretation on which all the members of Γ are true and ***X*** is false).

Metalogical Concepts

- **1.3.4** A *PL* argument Γ / ***X*** is **invalid**, so that Γ |=/ ***X***, if and only if there is a model of Γ on which ***X*** is false (that is, there is a *PL* interpretation on which the members of Γ are all true and ***X*** is false).
- We may also speak of the validity (or **logical truth**) and invalidity (of **logical falsehood**) of *sentences*, as follows.
- **1.3.5a** A *PL* sentence is **valid** if and only if it is true on every *PL* interpretation relevant to that sentence.
- **1.3.5b** A *PL* sentence is **valid** if and only if there is no *PL* interpretation on which it is false.
- **1.3.6a** A *PL* sentence is **contradictory** (or **logically false**) if and only if it is false on every *PL* interpretation relevant to that sentence.
- **1.3.6b** A *PL* sentence is **contradictory** (or **logically false**) if and only if there is no *PL* interpretation on which it is true.

Metalogical Concepts

- **1.3.7** A *PL* sentence is **contingent** if and only if it is true on at least one *PL* interpretation and false on at least one interpretation.
- **1.3.8a** Two *PL* sentences are **logically equivalent** if and only if they have identical truth values on every *PL* interpretation relevant to them.
- **1.3.8b** Two *PL* sentences are **logically equivalent** if and only if there is no relevant *PL* interpretation on which they disagree in truth value.
- **1.3.9** A set of *PL* sentences is **satisfiable** if and only if it has a model (that is, there is a *PL* interpretation that 'satisfies' it, on which every member of the set is true).
- **1.3.10a** A set of *PL* sentences is **unsatisfiable** if and only if on every relevant *PL* interpretation, a member of the set is false.
- **1.3.10b** A set of *PL* sentences is **unsatisfiable** if and only if it lacks a model (that is, there is no *PL* interpretation on which all the members of the set are true).

Decidability and Effectiveness

- **1.3.11** A concept (property or predicate) is **decidable** if and only if there is an **effective decision procedure** for determining whether or not something is subsumed under the concept.
  - A procedure is **effective** if and only if it is **mechanical** (involving no creative steps) and generates the desired result after **finitely-many deterministic steps**.
- *Note*: *Not all effective procedures are decision procedures*. There are effective procedures that produce the answer 'Yes' when and only when the correct answer is 'Yes,' but that do not produce any answer when the correct answer is 'No'. There are also effective procedures that produce the answer 'No' when and only when the correct answer is 'No,' but do not produce any answer when the correct answer is 'Yes.' We will call the former kind of procedure a **Yes-Procedure** and will call the latter kind a **No-Procedure**.

Semidecidability

- A concept (property, predicate) with only a Yes-procedure is **semidecidable** (i.e., **recursively** or **computably enumerable**).
- A landmark limitation of Predicate Logic to which we return is that *there is no effective decision procedure for answering all the questions of the form*: '*Is this PL sentence, or is this set of PL sentences, valid?*' (likewise for 'satisfiable', 'contradictory', 'contingent' and so forth).
- We will find that some metalogical concepts of *PL*, like **validity** and **unsatisfiability**, are semidecidable. But this means that their compliments, **invalidity** and **satisfiability**, are not even semidecidable.

Proof Theory

- We have been discussing **semantic** concepts, like meaning (or reference), truth, validity, satisfiability and logical consequence. These have to do with the **interpretation** of strings of symbols.
- Proof Theory is a **syntactic** idea. Much as we gave formation rules for terms and formulas in *PL*, we must give **derivation rules** for proofs.
- We hope that (*) *a sentence* ***X*** *is derivable from a set of them, Γ, according to our rules just in case* Γ / ***X*** *is a valid argument.*
- Later in the course, we will find that this is indeed the case.

Proof Theory

- **1.4.1** A proof theory makes rigorous (some relevant notion of) **demonstrative proof**. A formal proof is called a **derivation**.
  - *Note*: Even bracketing non-classical (e.g., *intuitionistic*) notions of logical consequence, the word 'proof' is ambiguous. The kind of proof that we are interested in, which makes rigorous the kind in pure mathematics, does **not** categorically establish its conclusion. At best, it establishes that *if* the premises (axioms) are true, *then* so is the conclusion. When we ask for 'proof' that the defendant is guilty, we seek something categorical. We want to know whether they are guilty – period!
- A demonstrative proof consists of: (1) the **inferential antecedents**, (2) the **inferential conclusion**, and (3) the **inferential license** (where (3) refers to the inferential antecedents and the rules of inference that were used).

Proof Theory

- **Formal derivations** consist *solely* of symbolic sentences.
- The number associated with a stage or its inferential license is strictly extraneous to the derivation. But since we are humans (!) who require a degree of narrative, we will typically write formal derivations as a series of inferential steps. This is another one of our conventions.
- Sentences in a derivation are either **premises**, **assumptions** or **conclusions** that are licensed by some formal rules of inference.
- If L is a **logical syntax** and a DS a **Deduction System** (a collection of formal rules of inference), then if $\Gamma + \boldsymbol{X}$ is a set of L sentences, we say:

Proof Theory

- An *L* derivation of ***X*** from Γ is a finite sequence of *L* sentences such that the last sentence of the sequence is ***X*** and every sentence in the sequence is either a member of Γ or is licensed by a rule of *DS*.
- If there is an *L* derivation of ***X*** from Γ, we will say that ***X*** is **derivable** from Γ in *L* or, instead, that ***X*** is a **theorem** of Γ in *L*, written $\mathbf{\Gamma} \vdash_{\mathbf{L}} \boldsymbol{X}$.
- As *PL* is our default system, we write, $\vdash$ instead of $\vdash_{\mathbf{PL}}$ when *L* is *PL*.
- The lengthy collection of inference rules that we use is called the **Natural Deduction System** (*NDS*). We will outline its details shortly.

Proof Theory

- An *L* derivation of ***X*** from Γ is a finite sequence of *L* sentences such that the last sentence of the sequence is ***X*** and every sentence in the sequence is either a member of Γ or is licensed by a rule of *DS*.
- The premises are listed at the start of the derivation, called the **zero stage**. Sentences of **non-zero stages** are licensed by formal rules. Their applicability is determined by the (generally coarsest) **syntactical forms** of the inferential antecedents or conclusion.
- We can now restate the hope that we labeled (*) in an earlier slide.

Soundness and Completeness

- **1.4.2** Desideratum (*) consists of the following two conditionals:
- **Soundness Theorem for *PL***: For every set Γ of *PL* sentences and every sentence ***X*** of *PL*, ***if* Γ |– *X*, *then* Γ |= *X*** (that is, if ***X*** is a theorem of Γ, then ***X*** is also a logical consequence of Γ).
- **The Completeness Theorem for *PL***: For every set Γ of *PL* sentences and every sentence ***X*** of *PL*, ***if* Γ |= *X*, *then* Γ |– *X*** (that is, if ***X*** is a logical consequence of Γ, then ***X*** is also a **theorem** of Γ).

Formalizability

- The **Soundness** and **Completeness** theorems together mean that the semantic relation of *PL,* |=, is **formalized** by the syntactic relation, |–. That is, |=, is equivalent to a formal notion, |–. Every proof registers a real implication; and every implication is witnessed by some proof.
- *PL* is an example of a **formal logic** because the *PL* consequence relation is formalizable. By contrast, the notion of logical consequence for **Second-Order Logic** (PL^2), to which we return later in the course, is not formalizable. Hence, PL^2 is not a formal logic.
- This fact about PL^2 can be described by saying that PL^2 is an **incomplete** logic or, alternatively, that it is **essentially semantical**.

Corollaries

- **1.4.3** (Vocabulary) A *PL* sentence that is derivable from the empty set, ∅, is a **logical theorem**. A set of *PL* sentences from which a sentence and its negation are both derivable is **inconsistent**. A **consistent** set of *PL* sentences is a set that is not inconsistent. And sentences ***X*** and ***Y*** are **interderivable** when ***X*** is derivable from ***Y*** and ***Y*** is derivable from ***X***.
- *The Soundness and Completeness Theorems have the following corollaries*:
- **1.4.3a** A set of *PL* sentences is **unsatisfiable** if and only if it is **inconsistent**.
- **1.4.3b** Two *PL* sentences are **logically equivalent** if and only if they are **interderivable**.
- **1.4.3c** A *PL* sentence is **valid** if and only if it is a **logical theorem**.
- **1.4.3d** A *PL* sentence is **contradictory** if and only if a sentence and its negation are both **derivable** from it.

Proof of **1.4.3a**

1.4.3a A set of *PL* sentences is **unsatisfiable** if and only if it is **inconsistent**.

- *Part 1* (**Inconsistency → Unsatisfiability**):
- 1) Let Γ be an **inconsistent** *PL* set, i.e., **Γ |–** ***X*** and **Γ |–** ~***X***, for some *PL* sentence ***X***.
- 2) **Γ |=** ***X*** and **Γ |=** ~***X***. [From 1) by the **Soundness Theorem**]
- 3) Assume for *reductio* that Γ is **satisfiable**.
- 4) Then there is a *PL* **interpretation** I_Γ that is a model of Γ. [From 3), by the definition of **satisfiability**]
- 5) If I_Γ is not relevant to ***X***, expand I_Γ into I_Γ* such that I_Γ* interprets all the vocabulary in ***X*** without changing any of the semantical assignments made by I_Γ. Since I_Γ is a model of Γ, I_Γ* is also a model of Γ.
- 6) ***X*** and ~***X*** are both true on I_Γ*. [From 2) and 5) by the definition of **logical consequence**)
- 7) However, by definition **1.2.5d**, this is a contradiction. Consequently, by *reductio ad absurdum*, premise 3) must be false; that is, Γ is **unsatisfiable**. (From 3) through 6)

Proof of **1.4.3a**

1.4.3a A set of *PL* sentences is **unsatisfiable** if and only if it is **inconsistent**.

- *Part 2* (**Unsatisfiability → Inconsistency**):
- 1) Let Γ be an **unsatisfiable** set of *PL* sentences, and let ***X*** be any *PL* sentence.
- 2) Then <u>there is no</u> *PL* **interpretation** that 'satisfies' Γ, i.e., on which every member of Γ is true. [From 1), by the definition of **unsatisfiability**]
- 3) If ***X*** is <u>not</u> a **logical consequence** of Γ, ***X*** must be false on some *PL* interpretation that satisfies (i.e., makes true all the members of) Γ.
- 4) Γ |= ***X*** and Γ |= ~***X***. [From 2) and 3), since Γ has no model]
- 5) Γ |– ***X*** and Γ |– ~***X***. [From 4) by the **Completeness Theorem**]
- 6) So, Γ is **inconsistent**. [From 5) by the definition of **inconsistency**]

Proof of **1.4.3b**

1.4.3b Two *PL* sentences are **logically equivalent** if and only if they are **interderivable**.

- *Part 1* (**Interderivability → Logical Equivalence**):
- 1) Suppose that ***X*** and ***Y*** are any **interderivable** *PL* sentences, i.e., {***X***}|– ***Y*** and {***Y***}|– ***X***.
- 2) **{*X*}|= *Y*** and **{*Y*}|= *X***. [From 1) by the **Soundness Theorem**]
- 3) Let I_Γ be any *PL* interpretation for ***X*** and ***Y*** on which ***X*** is <u>true</u>.
- 4) Then ***Y*** is true on I_Γ too . [From 2), i.e., {***X***}|= ***Y***, and 3) by the definition of **logical consequence**
- 5) Now let I_Γ be a *PL* interpretation for ***X*** and ***Y*** on which ***X*** is <u>false</u>.
- 6) Then ***Y*** is false on I_Γ too. [From 2), i.e., {***Y***}|= ***X***, and 5) by the definition of **logical consequence**]
- 7) So, ***X*** and ***Y*** have identical truth values on <u>every</u> *PL* interpretation [From 3) – 6)]
- for them.
- 8) ***X*** and ***Y*** are **logically equivalent**. [From 7) by the definition of **logical equivalence**]

Proof of **1.4.3b**

1.4.3b Two *PL* sentences are **logically equivalent** if and only if they are **interderivable**.

- *Part 2* (**Logical Equivalence → Interderivability**):
- 1) Suppose that ***X*** and ***Y*** are **logically equivalent** *PL* sentences.
- 2) Then there is <u>no</u> *PL* interpretation on which ***X*** is true and ***Y*** is false, or ***Y*** is true and ***X*** is false. [From 1) by the definition of **logical equivalence**]
- 3) So,**{*X*}**|= ***Y*** and **{*Y*}**|= ***X***. [From 2) by the definition of **logical consequence**]
- 4) Thus,**{*X*}**|– ***Y*** and **{*Y*}**|– ***X***. [From 3) by the **Completeness Theorem**]
- 5) ***X*** and ***Y*** are **interderivable**. [From 4) by the definition of **interderivability**]

Proof of **1.4.3c**

1.4.3c A *PL* sentence is **valid** if and only if it is a **logical theorem**.

- *Part 1* (**Theoremhood → Validity**):
- 1) Let ***X*** be any ***PL*** theorem, that is**,** ∅ |– ***X***.
- 2) ∅ |= ***X***. [From 1) by the **Soundness Theorem**]
- 3 Let I_Γ be <u>any</u> *PL* interpretation for ***X***. Since there is no sentence in ∅ that is false on I_Γ (because there are no sentences in ∅!), I_Γ <u>satisfies</u> ∅.
- 4) So, ***X*** is <u>true</u> on I_Γ. [From 2) and 3) by the definition of **logical consequence**]
- 5) Since I_Γ is arbitrary, ***X*** is true on <u>every</u> *PL* interpretation of it. [From 3) and 4)]
- 6) ***X*** is **valid**. [From 5) by the definition of valid sentence]

Proof of **1.4.3c**

1.4.3c A *PL* sentence is **valid** if and only if it is a **logical theorem**.

- *Part 1* (**Validity → Theoremhood**):
- 1) Let ***X*** be any **valid** *PL* sentence.
- 2) Then ***X*** is true on every *PL* interpretation for it. [From 1) by the definition of **valid sentence**]
- 3) Every *PL* **interpretation** satisfies ∅. [By the reasoning in step 3) of the preceding proof]
- 4) Since every *PL* **interpretation** at all for ***X*** makes ***X*** true, every interpretation for ***X*** that satisfies ∅ makes ***X*** true too, i.e. **∅** |= ***X***. [From 1) and 2) by the definition of **logical consequence**]
- 5) **∅** |– ***X.*** [From 4) by the **Completeness Theorem**]
- 6) ***X*** is a **logical theorem.** [Fom 5) by the definition of **logical theorem**]

Proof of **1.4.3d**

1.4.3d A *PL* sentence is **contradictory** if and only if a sentence and its negation are both **derivable** from it.

- *Part 1* (**Derivability → Contradictoriness**):
- 1) Let ***Y*** be a *PL* sentence such that **{*Y*}**|– ***X*** and **{*Y*}**|– ~***X***.
- 2 Then **{*Y*}**|= ***X*** and **{*Y*}**|= ~***X.*** [From 1) by the **Soundness Theorem**]
- 3) Assume for *reductio*: There is a *PL* **interpretation** $\boldsymbol{I_\Gamma}$ on which ***Y*** is true.
- 4) If I_Γ is not relevant to ***X***, expand I_Γ into $\boldsymbol{I_\Gamma}$ ***** such that I_Γ* interprets all of the vocabulary in ***X*** without altering any of the semantical assignments made by I_Γ. Since ***Y*** is true on I_Γ, it is true on I_Γ * as well.
- 5) Then ***X*** and ~***X*** are true on I_Γ *, which is a contradiction. [From 2) and 4) by the definition of **logical consequence**]
- 6) Hence, the *reductio* assumption is false: there is no *PL* **interpretation** on which ***Y*** is true. [From 3) through 5)]
- 7) ***Y*** is **contradictory** [From 6) by the definition of **contradictory sentence**]

Proof of **1.4.3d**

1.4.3d A *PL* sentence is **contradictory** if and only if a sentence and its negation are both **derivable** from it.

- *Part 1* (**Contradictoriness → Derivability**):
- 1) Let ***Y*** be a **contradictory** *PL* sentence and ***X*** any *PL* sentence.
- 2) Then there is <u>no</u> *PL* interpretation on which ***Y*** is true. [From 1) by the definition of **contradictory** sentence and the **Soundness Theorem**]
- 3) In order for ***X*** <u>not</u> to be a **logical consequence** of ***Y***, it must be <u>false</u> on a *PL* interpretation on which ***Y*** is <u>true</u>.
- 4) Thus, {***Y***}|= ***X*** <u>and</u> {***Y***}|= ~***X***. [From 2) and 3), applied to ***X*** & ~***X***]
- 5) So, {***Y***}|– ***X*** and {***Y***}|– ~***X***, as asserted. [From 4) by the **Completeness Theorem**]

Rules of Inference

- **1.4.4** There are three kinds of rules of inference on which we rely:
  - **Normal rules**
  - **Hypothetical rules**
  - **Replacement rules**
- The rules that we introduce include <u>all of the standard ones</u>. This makes it easier to <u>prove theorems **in** *PL*</u>, but harder to prove <u>metatheorems **about** *PL*</u>. So, later (in **1.4.7**) we describe a deduction system with a leaner set of rules, from which all the rules that we enumerate presently can be **derived**. This system will be important.

Proof Blocks

- Blocks are numbered according to the order in which they are opened. They are opened at two stages: the zero stage and a stage at which the assumption of a hypothetical rule occurs. The block opened at the zero stage is the ***0*-block** and encloses the **main derivation**. It closes at the conclusion of the derivation.
- A block B^* is a **subblock** of a block B just in case B^* is opened after B is opened and before B is closed. (So, all the blocks of a derivation, other than the *0*-block are subblocks of the *0*-block.) A block can be closed only after its subblocks are.
- **Nested blocks** are ordered in a series $B_1, B_2, B_3, \ldots, B_n$ such that B_{k+1} is a subblock of B_k, for all $k = 1, 2, 3, \ldots, n-1$. These are **stacks**: the last block to be opened is the first to be closed. Assumptions of a block are **discharged** outside of the block.

Example 1: Constructive Dilemma (*CS*)

Constructive Dilemma (CD)

[n
⋮
h $\mathbf{X} \vee \mathbf{Y}$
⋮
i $\mathbf{X} \to \mathbf{Z}_1$
⋮
j $\mathbf{Y} \to \mathbf{Z}_2$
⋮
k $\mathbf{Z}_1 \vee \mathbf{Z}_2$ h, i, j, CD
⋮
n]

Example 2: Disjunctive Syllogism (*DS*)

Disjunctive Syllogism (DS)

[n			
	⋮		
	h	$\mathbf{X} \vee \mathbf{Y}$	
	⋮		
	i	$\neg\mathbf{X}$ (or $\neg\mathbf{Y}$)	
	⋮		
	k	$\mathbf{Y}$ (or $\mathbf{X}$)	h, i, DS
	⋮		
n]			

Explanation of Rules

- In real derivations, *n, h, i, j,* and *k* get replaced with **numerals**. The lines *h, i,* and *j* are the inference's **antecedents** and the line *k* is the inference's **conclusion**. To the right of the conclusion is the **inferential license**.
- *Example*: The license, '*h, i, j,* (*CD*)', abbreviates, 'From lines *h*, *i*, and *j* by Constructive Dilemma,' while '*h*, *i*, (*DS*)' abbreviates, 'From lines *h* and *i* by Disjunctive Syllogism.'
- As before, the symbols $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z_1}$, and $\boldsymbol{Z_2}$ are metalinguistic variables that stand for *any PL sentences.*
- The **brackets** to the left represent the block within which the rule is applied. We will discover that normal rules apply only within open blocks.
  - *Foreshadow*: A left bracket followed by numeral *n* represents the first line of the *n*th block, and numeral *n* followed by a right bracket represents the last line of the *n*th block. Prior to '[n' the *n*th block has not been opened and after 'n]' the it is closed.

Concrete Illustration

- [*2*

…
- *6*) *P*s→(∃y)*B*y
- *7*) ~(∀x)~*D*x
- *8*) *P*s v (*R*ab & *Q*c)
- *9*) (*R*ab & *Q*c) → (∀x)~*D*x
- *10*) (∃y)*B*y v (∀x)~*D*x   *6, 8, 9*, *CD*
- *11*) (∃y)*B*y   *7, 10*, *DS*

…
- *2*]

Observations about Illustration

- In the previous example, the portion of the derivation displayed is part of the **second block**.
- The rules are applied **fully within an open block**. (Block *2* is opened at some point prior to the *6*th line and is closed after the *11*th line.)
- Any conclusion that we infer may be used as an antecedent for a later inference <u>if that conclusion occurs in the open block of the inference.</u>

Example 3: *DeM* and *MC*

<u>De Morgan's Laws</u> (***DeM***):

- ~(***X*** & ***Y***) **←→** (~***X*** v ~***Y***)
- ~(***X*** v ***Y***) **←→** (~***X*** & ~***Y***)

<u>Material Conditional</u> (***MC***):

- (***X*****→*****Y***) **←→** (~***X*** v ***Y***)

- *Note*: The bolded biconditional arrows mean that one can replace <u>either for the other</u>. One can also execute the replacements in a <u>proper subformula</u>.

Concrete Illustration

- [*1*

…

- *6*) *Ps* → (∀*x*)~(*Rab* & *Dx*)
- *7*) *Ps* → (∀x)(~*Rab* v ~*Dx*) *6*, *DeM*
- *8*) *Ps* → (∀*x*)(*Rab* → ~*Dx*) *7, MC*
- *9*) ~*Ps* v (∀x)(*Rab* → ~*Dx*) *8, MC*

…

- *1*]

- *N.B.*: Rules like *CD* and *DS* only apply to **entire lines**. By contrast, <u>replacement rules</u> like *DeM* and *MC* apply to **individual formulas**.

A Hypothetical Rule

Conditional Proof (CP): 'CPA' stands for 'Conditional Proof Assumption'.

[n			
	⋮		
[n+1	h	**X**	CPA
	⋮		
n+1]	k	**Y**	
	k+1	**X→Y**	h–k, CP
	⋮		
n]			

Concrete Illustration

- [*1*

…

- [*2* *i*) *Ps* (*CPA*)

…

- *2*] *k*) (∀*x*)~(*Rab* & *Dx*)
  k+1) *Ps* → (∀*x*)~(*Rab* & *Dx*) *i–k* (*CP*)

…

- *1*]

- *Note*: A block that is opened due to the application of some hypothetical rule closes at a line permitted by that specific rule. The inference's conclusion of a hypothetical rule is stated immediately after the line at which its block is closed.

Natural Deduction

- There are many *equivalent* **proof theories** for *PL*, i.e., systems that have the same theorems and valid inferences. But the most intuitive are **Natural Deduction** systems. We begin with the system, ***NDS***.
- ***NDS*** consists of *seventeen* **normal rules**, *three* **hypothetical rules**, and *thirteen* **replacement rules**, which we shall now enumerate.

- **1.4.5a Normal rules** can only be applied fully within an open block. The order of lines *h*, *j*, and *j* in the rule schemas to follow are irrelevant in their application. However, line *k* must succeed them.

Normal Rules: Reiteration

Reiteration (**Reit**):

```
[n
    ⋮
    h      X
    ⋮
    k      X      h, Reit
    ⋮
n]
```

Normal Rules: Conjunction

Conjunction (**Conj**):

[n
    ⋮
    h   **X**
    ⋮
    i   **Y**
    ⋮
    k   **X**∧**Y**   h, i, Conj
    ⋮
n]

Normal Rules: Simplification

Simplification (**Simp**): This is a two-part rule.

[n
    ⋮
    h   **X**∧**Y**
    ⋮
    k   **X** (or **Y**)   h, Simp
    ⋮
n]

Normal Rules: Addition

Addition (**Add**): This is a two-part rule.

[n
⋮
h $\mathbf{X}$
⋮
k $\mathbf{X} \vee \mathbf{Y}$ (or $\mathbf{Y} \vee \mathbf{X}$) h, Add
⋮
n]

Normal Rules: Disjunctive Syllogism

Disjunctive Syllogism (**DS**): This is a two-part rule.

[n
⋮
h $\mathbf{X} \vee \mathbf{Y}$
⋮
i $\neg\mathbf{X}$ (or $\neg\mathbf{Y}$)
⋮
k $\mathbf{Y}$ (or $\mathbf{X}$) h, i, DS
⋮
n]

Normal Rules: Modus Ponens

Modus Ponens (**MP**):

[n
⋮
h $\mathbf{X \to Y}$
⋮
i $\mathbf{X}$
⋮
k $\mathbf{Y}$ h, i, MP
⋮
n]

Normal Rules: Modus Tollens

Modus Tollens (**MT**)

[n
⋮
h $\mathbf{X \to Y}$
⋮
i $\neg\mathbf{Y}$
⋮
k $\neg\mathbf{X}$ h, i, MT
⋮
n]

Normal Rules: Biconditional Modus Ponens

Biconditional Modus Ponens (**BcMP**): This is a two-part rule.

[n
⋮
h $\mathbf{X} \leftrightarrow \mathbf{Y}$
⋮
i **X** (or **Y**)
⋮
k **Y** (or **X**) h, i, BcMP
⋮
n]

Normal Rules: Biconditional Modus Tollens

Biconditional Modus Tollens (**BcMT**): This is a two-part rule.

[n
⋮
h $\mathbf{X} \leftrightarrow \mathbf{Y}$
⋮
i $\neg\mathbf{X}$ (or $\neg\mathbf{Y}$)
⋮
k $\neg\mathbf{Y}$ (or $\neg\mathbf{X}$) h, i, BcMT
⋮
n]

Normal Rules: Hypothetical Syllogism

Hypothetical Syllogism (**HS**)

[n
⋮
h $\mathbf{X \rightarrow Y}$
⋮
i $\mathbf{Y \rightarrow Z}$
⋮
k $\mathbf{X \rightarrow Z}$ h, i, HS
⋮
n]

Normal Rules: Constructive Dilemma

Constructive Dilemma (**CD**)

[n
⋮
h $\mathbf{X \vee Y}$
⋮
i $\mathbf{X \rightarrow Z_1}$
⋮
j $\mathbf{Y \rightarrow Z_2}$
⋮
k $\mathbf{Z_1 \vee Z_2}$ h, i, j, CD
⋮
n]

Normal Rules: Explosion

Explosion (**Expl**): **Y** is any PL sentence.

```
[n
     ⋮
     h    X∧¬X
     ⋮
     k    Y          h, Expl
     ⋮
n]
```

Normal Rules: Universal Instantiation

Universal Instantiation (**UI**): **s** is any PL ***singular term***, **z** is a PL variable, and **X** is a PL formula that contains occurrences of **z** but no **z**-quantifiers. **X**[**s**] is the PL sentence formed by replacing ***all*** the occurrences of **z** in **X** by **s**.

```
[n
     ⋮
     h    (∀z)X
     ⋮
     k    X[s]       h, UI
     ⋮
n]
```

Normal Rules: Universal Generalization

Universal Generalization (**UG**): **X** is a PL sentence, **s** is a PL ***name*** that occurs in **X**, and **z** is a PL variable that does not occur in **X**. **s** is **arbitrary** at line h, that is, **s** does not occur in any premise or undischarged assumption listed on line h or prior to it. **X**[**z**] is the PL formula formed by replacing ***all*** the occurrences of **s** in **X** by **z**.

```
[n
      ⋮
      h      X
      ⋮
      k      (∀z)X[z]     h, UG
      ⋮
n]
```

Normal Rules: Existential Generalization

Existential Generalization (**EG**): **X** is a PL sentence, **s** is a PL ***singular term*** that occurs in **X**, and **z** is a PL variable that does not occur in **X**. **X**[**z**, **s**] is a PL formula formed by replacing ***one or more*** of the occurrences of **s** in **X** by **z**.

```
[n
      ⋮
      h      X
      ⋮
      k      (∃z)X[z, s]     h, EG
      ⋮
n]
```

Normal Rules: Identity

Identity (**Id**): **r** is any PL ***singular term***. Id does not require an antecedent.

[n
⋮
k **r = r** Id
⋮
n]

Normal Rules: Substitution

Substitution (**Sub**): **s** and **t** are PL ***singular terms*** and **X** is a PL sentence that contains occurrences of **s**. **X**[**t**, **s**] is a PL sentence formed by replacing ***one or more*** of the occurrences of **s** in **X** by **t**. This is a two-part rule.

[n
⋮
h **s** = **t** (or **t** = **s**)
⋮
i **X**
⋮
k **X**[**t**, **s**] h, i, Sub
⋮
n]

Hypothetical Rules

- Every hypothetical rule begins a new block and adds an assumption.
- It terminates with exiting the block and discharging the assumption.
- The order of the lines *h* and *k* is relevant in Hypothetical Rules.

Hypothetical Rules: Reductio Ad Absurdum

Reductio Ad Absurdum (**RAA**): 'RA' stands for 'Reductio Assumption'; this is a two-part rule.

[n			
	⋮		
[n+1	h	**X** (or ¬**X**)	RA
	⋮		
	k−1	**Y**	
n+1]	k	¬**Y**	
	k+1	¬**X** (or **X**)	h–k, RAA
	⋮		
n]			

Hypothetical Rules: Conditional Proof

Conditional Proof (**CP**): 'CPA' stands for 'Conditional Proof Assumption'.

[n			
	⋮		
[n+1	h	**X**	CPA
	⋮		
n+1]	k	**Y**	
	k+1	**X**→**Y**	h–k, CP
	⋮		
n]			

Hypothetical Rules: Existential Instantiation

Existential Instantiation (**EI**): **s** is a PL ***name***, **z** is a PL variable, and **X** is a PL formula that contains occurrences of **z** but no **z**-quantifiers. **s** satisfies three conditions: (1) it does not occur in any premise or undischarged assumption prior to line h, (2) it does not occur in (∃**z**)**X**, and (3) it does not occur in **Y**. **X**[**s**] is the PL sentence formed by replacing ***all*** the occurrences of **z** in **X** by **s**. 'EIA' stands for 'Existential Instantiation Assumption'.

[n			
	⋮		
	h–1	(∃**z**)**X**	
[n+1	h	**X**[**s**]	EIA, **s**
	⋮		
n+1]	k	**Y**	
	k+1	**Y**	h–1, h–k, EI
	⋮		
n]			

Replacement Rules

- Unlike other rules may be applied to **proper components** of a sentence. All replacements may be performed in the forward or reverse directions. ***X***, ***Y***, and ***Z*** are *PL* sentences or *PL* formulas.

Double Negation (**DN**):	$\neg\neg X$	$\Leftrightarrow$	X
Idempotence (**Idem**):	$X \wedge X$	$\Leftrightarrow$	X
	$X \vee X$	$\Leftrightarrow$	X
Commutation (**Com**):	$X \wedge Y$	$\Leftrightarrow$	$Y \wedge X$
	$X \vee Y$	$\Leftrightarrow$	$Y \vee X$

Replacement Rules

Association (**Assoc**):	$X \wedge (Y \wedge Z)$	$\Leftrightarrow$	$(X \wedge Y) \wedge Z$
	$X \vee (Y \vee Z)$	$\Leftrightarrow$	$(X \vee Y) \vee Z$
Distribution (**Dist**):	$X \wedge (Y \vee Z)$	$\Leftrightarrow$	$(X \wedge Y) \vee (X \wedge Z)$
	$X \vee (Y \wedge Z)$	$\Leftrightarrow$	$(X \vee Y) \wedge (X \vee Z)$
De Morgan's Laws (**DeM**):	$\neg(X \wedge Y)$	$\Leftrightarrow$	$\neg X \vee \neg Y$
	$\neg(X \vee Y)$	$\Leftrightarrow$	$\neg X \wedge \neg Y$
Material Conditional (**MC**):	$X \rightarrow Y$	$\Leftrightarrow$	$\neg X \vee Y$
Negated Conditional (**NC**):	$\neg(X \rightarrow Y)$	$\Leftrightarrow$	$X \wedge \neg Y$

Replacement Rules

Contraposition (**Cont**):	$X \rightarrow Y$	$\Leftrightarrow$	$\neg Y \rightarrow \neg X$
Exportation (**Expr**):	$X \rightarrow (Y \rightarrow Z)$	$\Leftrightarrow$	$(X \wedge Y) \rightarrow Z$
Biconditional (**Bc**):	$X \leftrightarrow Y$	$\Leftrightarrow$	$(X \rightarrow Y) \wedge (Y \rightarrow X)$
Negated Biconditional (**NBc**):	$\neg(X \leftrightarrow Y)$	$\Leftrightarrow$	$\neg X \leftrightarrow Y$
	$\neg(X \leftrightarrow Y)$	$\Leftrightarrow$	$X \leftrightarrow \neg Y$
Negated Quantifiers (**NQ**):	$\neg(\forall z)X$	$\Leftrightarrow$	$(\exists z)\neg X$
	$\neg(\exists z)X$	$\Leftrightarrow$	$(\forall z)\neg X$

Gentzen Deduction System (GDS)

- The system ***NDS*** is highly redundant. Most of the rules can be derived – assuming classical logic in the metatheory – from just a few.
- This facilitates reasoning **in** *PL*. We can use (almost) all the methods of reasoning that we unreflectively use in mathematics. But it is a pain for proving things **about** *PL*. We must deal with each rule separately!
- A pioneering middle ground is due to Gerhard Gentzen who invented this style of proof system (as opposed to, e.g., Hilbert's axiomatics).
- It is neither highly redundant, like ***NDS***, nor maximally lean, like systems we will discuss. It also inaugurated the so-called **conceptual role** approach to meaning in terms of **introduction and elimination rules**. We will call the system **Gentzen Deduction System** (***GDS***).

GDS Rules

- Here are the **seventeen** rules of ***GDS***:
- (1) *Reiteration*
- (2) *Conjunction Introduction – Conj* (^*I*)
- (3) *Conjunction Elimination – Simp* (^*E*)
- (4) *Conditional Introduction – CP* (→*I*)
- (5) *Conditional Elimination – MP* (→ *E*)
- (6) *Universal Introduction – UG* (∀*I*)
- (7) *Universal Elimination – UI* (∀*E*)

GDS Rules Continued.

- (8) *Existential Introduction – EG* (∃*I*)
- (9) *Existential Elimination – EI* (∃*E*)
- (10) *Identity Introduction – Id* (=*I*)
- (11) *Identity Elimination – Sub* (=*E*)
- (12) *Negation Introduction – RAA*, *Part 1* (~*I*) (with conclusion ~***X***)
- (13) *Negation Elimination – RAA, Part 2* (~*E*) (with conclusion ***X***).
- (14) *Disjunction Introduction – Add* (v*I*)
- (15) *Disjunction Elimination* – (v*E*) is the following <u>hypothetical</u> rule:

Disjunction Elimination

(j is an integer greater than 1)

[n			
	⋮		
	h	$X \vee Y$	
[n+1	h+1	X	$\vee$E Assumption
	⋮		
n+1]	m	Z	
[n+j	m+1	Y	$\vee$E Assumption
	⋮		
n+j]	k	Z	
	k+1	Z	h, (h+1)–m, (m+1)–k, $\vee$E
	⋮		
n]			

Biconditional Elimination

(16–17) **Biconditional Elimination** (↔E) is BcMP, and **Biconditional Introduction** (↔I) is the following hypothetical rule (j is an integer greater than 1):

[n			
	⋮		
[n+1	h	X	↔I Assumption
	⋮		
n+1]	m	Y	
[n+j	m+1	Y	↔I Assumption
	⋮		
n+j]	k	X	
	k+1	X↔Y	h–m, (m+1)–k, ↔I
	⋮		
n]			

Summing Up

- The rules of ***NDS*** and ***GDS*** are <u>mutually derivable</u> (against a classical metatheory). The former system is useful for ordinary reasoning. It is also useful in philosophical contexts in which we consider alternative logics. *DS*, for instance, is not a rule of ***GDS*** but plays a central role in the 'explosion' argument from a contradiction to an arbitrary claim.
- We will ultimately focus on a system with even fewer rules than ***GDS***. This will greatly expedite our proof of **metatheorems** <u>about</u> it.
- For purposes of appreciating what a *PL* system is and how to work in it, any such system serves. We now turn to the project of investigating such systems **from the outside** to discover their scope and limitations.

Symbolic Logic

Resources of the Metatheory

Ambient Background Assumptions

- We will ultimately be proving things about a fixed *PL* logic system, like that *Disjunction Introduction* is **sound** – that is, truth-preserving in all models.
- What rules of inference and mathematical assumptions are we allowed to use in proving such a thing? May we use the very rule, *Disjunction Introduction*?
- If we seek to justify *Disjunction Introduction*, to the satisfaction of a skeptic then we cannot assume it. But we aspire to something more modest: use finitely many applications of the rules (plus some mathematical principles) in order to justify the soundness of infinitely many derivations (constructed out of infinitely many possible combinations of the *NDS* rules of inference). Our arguments will be **rule circular**, but not **premise circular.** We are not assuming as a premise the soundness of *Disjunction Introduction* in arguing that this inference rule is sound!
- *Upshot*: Not only do mathematical theorems (like *Fermat's Last Theorem*) depend on the axioms that one assumes, but theorems about **what follows from what in a fixed logic** do as well. However, in the latter case the relevant axioms are logical.

Ambient Background Assumptions

- **2.1.3** Metalogic is all about proving **metatheorems**. These theorems require proofs, and proofs require logical resources, such as rules of inference. What are the rules of inference that are available at the meta-level? **All the rules of the Natural Deduction System (*NDS*)**.
- In addition to these inference rules, our metatheory assumes arithmetical and set-theoretic principles, such as the *Axiom of Mathematical Induction* and the *Axiom of Extensionality*.
- We begin by discussing arithmetical principles and their relevance.

Arithmetic Assumptions

- **2.2.1** Our metatheory assumes the existence of the structure, <ℕ, *s*, <, +, *>. ℕ is the set of natural numbers, *0, 1, 2…* The symbols **s, <, +, *** denote the *successor*, *less than*, *addition* and *multiplication* operations.
- We naively avail ourselves of the assumed properties of natural numbers in proving metatheorems. In particular, we assume:
- First, the relation, <, is **well-founded** on ℕ. In other words, every nonempty subset of ℕ has a minimal element with respect to <.
- Second, the following recursion equations based on $s(n) = n+1$ hold.

Recursion Equations

- **2.2.1a** For every natural number n, $n + 0 = n$.
- **2.2.1b** For all natural numbers k and n, $n + s(k) = s(n+k)$.
- **2.2.1c** For every natural number n, $n * 0 = 0$.
- **2.2.1d** For all natural numbers k and n, $n * s(k) = ((n * k) + n)$.
- **2.2.1e** For all natural numbers j, k, and m, $j – k = m$ iff $k + m = j$ and undefined otherwise.

Induction

- The most powerful arithmetical principle that we assume is *Induction*.
- *Induction* takes two equivalent (in a standard metatheory) forms:
  - **Principle of Mathematical Induction** (***PMI***): If $\boldsymbol{X}(\boldsymbol{n_0})$, and for every natural number $k \geq n_0$, $\boldsymbol{X}[\boldsymbol{S}(\boldsymbol{k})]$ if $\boldsymbol{X}(\boldsymbol{k})$, then for every natural number $n \geq n_0$, $\boldsymbol{X}(\boldsymbol{n})$.
  - **Principle of Complete Induction** (***PCI***): If $\boldsymbol{X}(\boldsymbol{n_0})$, and for every natural number $k \geq n_0$, $\boldsymbol{X}(\boldsymbol{k})$ when $\boldsymbol{X}(\boldsymbol{m})$ for each m such that $n_0 \leq m < k$, then for every natural number $n \geq n_0$, $\boldsymbol{X}(\boldsymbol{n})$.

Application of *PMI*

- ***Theorem***: For all $n \in \mathbb{N}$, $n \geq 1$, $1 + 2 + 3 + \ldots + n = n(n + 1) / 2$.
- ***Proof***: For the **Base Step**, let $n = 1$. Then $1 = 1(1 + 1) / 2 = 2 / 2 = 1$.
- For the **Inductive Step**, Let $k \geq 1$, and suppose, for the **Inductive Hypothesis**, that $1 + 2 + 3 + \ldots + k = k(k + 1) / 2$.
- We argue that, given this, $1 + 2 + 3 + \ldots + k + (k + 1) = (k + 1)[(k+1) + 1] / 2$.
- By the **Inductive Hypothesis**, $1 + 2 + 3 + \ldots + k + (k + 1) = k(k + 1) / 2 + (k+1) = [k(k+1) + 2(k+1)] / 2 = (k + 1)(k + 2) / 2 = (k + 1)[(k+1) + 1] / 2$, as desired.

Application of *PCI*

- ***Metatheorem***: For <u>any</u> sentence of *PL*, ***X***, if ***X*** is **quantifier-free** and one of its <u>sentential component occurrences</u> is of the form (***Y*** → ***Z***), then the *PL* sentence obtained by replacing that occurrence of (***Y*** → ***Z***) in ***X*** with an occurrence of (~***Y*** v ***Z***) is <u>logically equivalent</u> to ***X***.
- <u>Note</u>: This statement has nothing to do, on its face, with the natural numbers! The trick of *Mathematical Induction* is to see how to **transpose** statements explicitly about the likes of sentences into statements <u>about numbers</u> whose <u>predicates</u> concern sentences.

Application of *PCI*

- **Proof**: Let us write ***X***[***Y*** → ***Z***] to denote a *PL* sentence of which an occurrence of (***Y*** → ***Z***) is a sentential component, and ***X***[~***Y*** v ***Z***] for the result of replacing <u>that occurrence</u> with an occurrence of (~***Y*** v ***Z***).
- We define the **complexity** of ***X***[***Y***→ ***Z***] to be the <u>number of connective occurrences</u> that appear in ***X***[***Y***→ ***Z***] *other than* the occurrence of → in the aforementioned occurrence of (***Y*** → ***Z***).
- For the **Base Step**, let the complexity of ***X***[***Y***→ ***Z***] be *0*. Since, (***Y*** → ***Z***) is logically equivalent to ~(***Y*** v ***Z***), the **Base Case** is trivial.

Application of *PCI*

- For the **Inductive Step**, let us <u>suppose</u> as the **Inductive Hypothesis** that for every *m* < *k*, where *k* is some <u>non-zero</u> natural number, any <u>quantifier-free</u> sentence ***X***[***Y***→ ***Z***] whose complexity is *m* is logically equivalent to ***X***[~***Y*** v ***Z***] . Then we show that the theorem holds for a quantifier-free sentence ***W***[***Y***→ ***Z***] whose complexity is *k*.
- Since *k* is not zero, ***W***[***Y***→ ***Z***] contains at least one connective occurrence other than the → of the relevant occurrence of (***Y***→ ***Z***).
- ***W***[***Y***→ ***Z***] could, therefore, be a **negation**, **conjunction**, **disjunction**, **conditional**, or **biconditional** (with an occurrence of (***Y***→ ***Z***)).
- Let us consider each case in turn.

Application of *PCI*

- ***(a) NEGATION***
- Suppose that ***W***[***Y*** → ***Z***] is a **negation**, i.e., of the form ~***V***. Hence, the relevant occurrence of (***Y*** → ***Z***) must be a sentential component occurrence of ***V***. We can write ***V***[***Y*** → ***Z***]. But ***V***[***Y*** → ***Z***] has a complexity <u>less than *k*</u>, so the **Induction Hypothesis** applies to it.
- That is, ***V***[***Y*** → ***Z***] is **logically equivalent** to ***V***[~***Y*** v ***Z***].
- But in that case ~***V***[***Y*** → ***Z***] must be logically equivalent to ~***V***[~***Y*** v ***Z***] as well. Since ***W***[***Y*** → ***Z***] *is* ~***V***[***Y*** → ***Z***] and ***W***[~***Y*** v ***Z***] *is* ~***V***[~***Y*** v ***Z***], we have that ***W***[***Y*** → ***Z***] is logically equivalent to ***W***[~***Y*** v ***Z***].

Application of *PCI*

- ***(b) CONJUNCTION***
- Now we suppose that ***W***[***Y*** → ***Z***] is a **conjunction**. Then it has the form (***V*** & ***U***). Since (***Y*** → ***Z***) is assumed to be a sentential component of (***V*** & ***U***), it must be either a sentential component of ***V***, or of ***U***, or both.
- Without loss of generality, assume it is a sentential component of only ***V***. Then we can write ***V*** as ***V***[***Y*** → ***Z***]. Since the complexity of (***V*** & ***U***) is *k*, the complexity of ***V***, i.e. ***V***[***Y*** → ***Z***], must be less than *k*. So, the **Induction Hypothesis** applies to ***V***[***Y*** → ***Z***], and ***V***[***Y*** → ***Z***] is logically equivalent to ~***V***[~***Y*** v ***Z***]. But, then, ***V***[***Y*** → ***Z***] & ***U*** must be logically equivalent to ~***V***[~***Y*** v ***Z***] & ***U***, that is, ***W***[~***Y*** v ***Z***]. The case of ***U*** is identical.

Application of *PCI*

- ***(c) REMAINING CASES***
- Exactly parallel reasoning applies to disjunctions, conditionals and biconditionals. The **Inductive Step** is, thus, established.
- We may now conclude, *PCI*, that every quantifier-free *PL* sentence (of any complexity) that contains a sentential component occurrence of the form (***Y*** → ***Z***) is logically equivalent to the *PL* sentence that is obtained from the original sentence by replacing that occurrence of (***Y*** → ***Z***) with an occurrence of (~***Y*** v ***Z***).

Set Theoretic Assumptions

- **2.3.1** Set theoretic assumptions will allow us to prove theorems about functions, relations, collections, sizes and structures of objects.
- We will be as liberal about sets as we are in ordinary mathematical contexts. At first pass, will assume the following natural principle:
- **Naïve Comprehension**: *For every predicate (property), there is a set of things that satisfy that predicate (have the corresponding property)*.
  - *Note*: This applies to inconsistent predicates as well. Consider the predicate '$x \neq x$'. By Naïve Comprehension, there is a set of things that satisfy it. It is ∅!
- Why is this a naive? Because it turns out to be inconsistent!

Russell's Paradox

- Consider the predicate '$x \notin x$'. By Naïve Comprehension, there is a set, $R = \{x : x \notin x\}$. Since R is a set, either $R \in R$ or $R \notin R$.
- Assume for *reductio* that $R \in R$. Then R satisfies the predicate '$x \notin x$'. But R satisfies this predicate just in case $R \notin R$. This is a contradiction.
- Hence, suppose for *reductio*, that $R \notin R$. Then R satisfies the predicate for membership in R. So, $R \in R$. This is also a contradiction!
- *Upshot*: **Naïve Comprehension** (which was thought by Frege and Dedekind to be a principle of logic!) is contradictory, so must be false.

Diagonal Arguments

- Russell argument is called a **diagonal argument**, and the method, due to Cantor, pervades mathematical logic and set theory. The proofs of the existence of different sizes of infinity, the undecidability of the Halting problem and first-order logic, the undefinability of truth, the incompleteness of arithmetic, and the unprovability of mathematics' consistency (if it is consistent) all make use of diagonal arguments.

	M_a	M_b	M_c	M_d	M_e	M_f	M_g	M_h ...
M_a	0	1 ...						
M_b	1	1	1 ...					
M_c	0	1	0	0 ...				
M_d	1	1	0	0	0 ...			
...	...							

Definitions

- Before we outline the way in which we will (try!) to circumvent <u>Russell's Paradox</u>, we introduce the following ideas and definitions from naive set theory.
- **Individuation of *n*-tuples**: For all *n*-tuples, with $n \geq 1$, $<\boldsymbol{a_1, a_2, a_3, \ldots, a_n}>$ and $<\boldsymbol{b_1, b_2, b_3, \ldots, b_n}>$, are <u>identical</u> just in case $\boldsymbol{a_i} = \boldsymbol{b_i}$, for all $i \leq n$.
- **Subset**: *A* is a <u>subset</u> of *B*, written $\boldsymbol{A \subseteq B}$ just in case, for every $x \in A$, $x \in B$.
- **Proper Subset**: *A* is a <u>proper subset</u> of *B*, written $\boldsymbol{A \subset B}$ just in case, *A* is a <u>subset</u> of *B*, but *A* is not <u>identical</u> to *B*. **Note**: $[A \subseteq B \ \& \ B \subseteq A] \leftrightarrow (A = B)$.
- **Union**: If *F* is a **family** (set of sets), then the union of *F*, written $\cup F$ is the set of <u>members of members</u> of *F*, i.e., $F = \{x : \exists y \ \& \ y \in F \ \& \ x \in y\}$.
- **Partition**: A partition of a set, *A*, is a family, *F*, that is **exhaustive** -- i.e., such that $\cup F = A$ -- and such that all of its members are **disjoint** -- i.e., for <u>all</u> $A \in F$ and $B \in F$, $\{x : x \in A \ \& \ x \in B\} = \emptyset$. The last condition is written: $\cap \boldsymbol{F} = \emptyset$.

Definitions Continued

- **Cartesian Product**: If $A_1, A_2, \ldots A_n$ are nonempty sets, then their Cartesian Product, written $A_1 \times A_2 \times \ldots A_n$, is the set $\{<x_1, x_2, x_3, \ldots, x_n> : x_1 \in A_1 \;\&\; x_2 \in A_2 \;\&\ldots\; x_n \in A_n\}$.
  - The *n*-times *Cartesian Product* of *A* with itself is written $\boldsymbol{A^n}$.
- *n*-place relations and functions (which, we saw, are just *n+1*-place relations that assign to every *n*-tuple exactly one individual) are subsets of the *Cartesian Product* of the sets of related items. Hence, for any binary relation, *R*, on a set, *A*, $R \subseteq A^2$. There are a variety of important features that any such binary relation may possess.

Kinds of Relation

- *R* is **reflexive** (on a set, *A*) iff for all *x* in *A*, *x R x*.
- *R* is **irreflexive** iff for all *x* in *A*, it is not the case that *x R x*.
- *R* is **symmetric** iff for all *x* and *y* in *A*, if *x R y*, then *y R x*.
- *R* is **asymmetric** iff for all *x* and *y* in *A*, if *x R y*, it is not the case that *y R x*.
- *R* is **antisymmetric** iff for all *x* and *y* in *A*, if *x R y* and *y R x*, then $x = y$.
- *R* is **transitive** iff for all *x, y*, and *z* in *A*, if *x R y* and *y R z*, then *x R z*.
- *R* is **extendible** iff for all *x* in *A*, there is some *y* in *A* such that *x R y*.
- *R* is **total** (or **dichotomous**) iff for all *x* and *y* in *A*, either *x R y* or *y R x*.

Kinds of Relation Continued

- *R* is **connex** (**trichotomous**) iff for all *x* and *y* in *A*, either *x R y*, *y R x*, or *x* = *y*.
- *R* is **injective** (**one-to-one**) iff for all *x, y*, and *z* in *A*, if *x R z* and *y R z*, then *x* = *y*.
- *R* has a **minimal element** in $D \subseteq A$ iff there is $x \in D$, such that for every $y \in D$, it is not the case that (*y R x*). [x is called an ***R*-minimal element in *D***]
- *R* has a *maximal element* in *D* $D \subseteq A$ iff there is $x \in$, such that for every $y \in D$, it is not the case that (*x R y*). [x is called an ***R*-maximal element in *D***]

Properties of Functions

- A **function**, written $f : A \rightarrow B$, where *A* is the domain of *f* and *B* is the *f*'s range, may be **total** or **partial**. If $f : A \rightarrow B$ is total, then, for every $x \in A$, there exists a *y*, such that $\langle x, y \rangle \in f$. (We say that *f* is 'defined' for all members of the domain, *A*.) If it is partial, then this is not this case.
- A function, $f : A \rightarrow B$, is said to be **surjective** or **onto** *B* just in case, for all $y \in B$, there exists an *x*, such that $\langle x, y \rangle \in f$. *f* is merely **into** otherwise.
- $f : A \rightarrow B$, is **injective** or **one-to-one** whenever, if $f(x) = f(y)$, then $x = y$.
- $f : A \rightarrow B$, is **bijection** or a **one-to-one correspondence**, written $\boldsymbol{A \approx B}$, when it is total, one-to-one, and onto.
- Given $f : A \rightarrow B$ and $g : B \rightarrow C$, the **composition** of *f* and *g*, written $\boldsymbol{g \circ f : A \rightarrow C}$ is from *A* into *C* and assigns each argument $x \in A$, the value $g(f(x))$.
- Finally, if *f* is a **bijection**, then its **inverse**, written f^{1}, is defined: $f^{1} \circ f(x) = Id(x) = x$. So, f^{1} is a bijection 'reversing' the action of the bijection, *f*.

ZFC Axioms

- Given these definitions for set-theoretic entities, and given that Naïve Comprehension is inconsistent, what sets can we assume exist?
- Our answer is given by disconcertingly gerrymandered axioms which have become 'the axioms of mathematics'. Unfortunately, we will eventually find that *their consistency is not provable* in any useful sense. They are:
- **Extensionality**: Sets are identical if they have the same members.
- **Pairing**: For any sets, z and w, there is a set containing exactly z and w.
- **Union**: For any set, z, there is a set, $\cup z$, containing exactly the members of members of z.
- **Powerset**: For any set, z, there is a set containing just the subsets of z, ***P*(*z*)**. $P(z)$ is called the **powerset** of z.

ZFC Axioms Continued

- **Subsets** (**Restricted Comprehension**) **Schema**: For any set, z, and any predicate, Φ, there is a set that contains exactly those members of z which satisfy Φ.
  - **Corollary**: There is not **universal set**, i.e., $\{x : x = x\}$.
- **Infinity**: There is a set containing $\emptyset$, and containing the successor of z (i.e., $z \cup \{z\}$) whenever it contains z.
- **Foundation** (**Regularity**) **Schema**: For any predicate, Φ, if there is something that satisfies Φ, then there is a minimal z (with respect to the $\in$ relation) that does -- i.e., a z such that Φ and no $y \in z$ such that Φ.
- **Replacement Schema**: For any set, z, and any predicate Φ such that, for every $t \in z$, there is exactly one x with $\Phi(t, x)$, there is a set which contains just those things, x, for which $\Phi(t, x)$ holds for some $t \in z$.
- **Choice**: If z is a disjointed set not containing $\emptyset$, then there is a subset of $\cup z$ whose intersection with each member of z is a singleton.

Cardinality

- **Cardinality** opposes **ordinality**. The cardinality of a set answers 'how many?'. Its ordinality answers 'in what order?' In *ZFC*, cardinals are soc-called **initial ordinals**, i.e., the first ordinals with that many elements.
- The *Axiom of Choice* ensures that *every set, A,* finite or infinite*, has a cardinality,* written ***card*(*A*)** (and, so, an associated ordinality). The *ZFC* axioms are insufficient to tell us what cardinality some sets – like $\mathbb{R}$ – have. But *ZFC* certainly proves the following elementary constraint:
  - **Hume's Principle**: For all sets *A* and *B*, *card*(A) = *card*(*B*) just in case there is a **bijection** between *A* and *B*, i.e., just in case $A \approx B$.
- We say that *A* and *B* are **equinumerous** when *card*(*A*) = *card*(*B*).

Relative Size and Infinity

- For all sets *A* and *B*:
- **2.3.4a** *card*(*A*) ≤ *card*(*B*) if/f there is a set *C*, such that $C \subset B$ and $A \approx C$.
- **2.3.4b** *card*(*A*) < *card*(*B*) if/f *card*(*A*) ≤ *card*(*B*) but *card*(A) ≠ *card*(*B*).
- **2.3.4c** *card*(A) ≥ *card*(*B*) if/f *card*(*B*) ≤ *card*(*A*).
- **2.3.4d** *card*(*A*) > *card*(*B*) if/f *card*(*A*) ≥ *card*(*B*) but *card*(*A*) ≠ *card*(*B*).
- **2.3.4e** *A* is **infinite** if/f there is a set *B*, such that $B \subset A$ and $A \approx B$.
- **2.3.4f** *A* is **finite** iff it is not *infinite*.
- If an infinite set $A \approx \mathbb{N}$, then *A* is **countable**, and **uncountable** if not.

Existence of Different Sizes of Infinity

- **Cantor's Theorem**: For every set, *A*, *card*(*P*(*A*)) > *card*(*A*).
- *Proof* (brisk style): It is clear that *card*(*P*(*A*)) ≥ *A*, since *A* ≈{{x} : x ∈ *A*}⊂ *P*(*A*). It remains to show that *card*(A) ≠ *card*(*B*), i.e., for <u>any</u> *f* : *A* → *P*(*A*), ∃y ∈ *P*(*A*) such that for <u>no</u> *x* ∈ *A* does *f*(x) = *y*. So, let *f* be arbitrary and *y* = {*z* : *z* ∉ *f*(*z*)}. Suppose ∃x ∈ *A* with *f*(*x*) = *y*. Then *x* ∈ *y* or *x* ∉ *y*. If *x* ∈ *y*, then *x* ∉ *f*(*x*) = *y*. If *x* ∉ *y* = *f*(*x*), and, hence, *x* ∈ *y*.
- *Note*: Cantor's Theorem applies to <u>any</u> set. In particular, it applies to ℕ and *P*(ℕ). Hence, *card*(*P*(ℕ)) > *card*(ℕ), but <u>also</u> *card*(*P*(*P*(ℕ))) > *card*(*P*(ℕ)), *card*(*P*(*P*(*P*(ℕ)))) > *card*(*P*(*P*(ℕ))), and so on *ad infinitum*.
- *Upshot*: There are <u>infinitely-many sizes of infinity</u>!

Commentary

- It is not hard to see that ℝ ≈ *P*(ℕ). Therefore, the *real numbers* are a familiar infinite set whose cardinality is *greater than* the first infinite cardinality.
- How much greater? Let us represent the different <u>cardinalities</u>, as follows: $\aleph_1$, $\aleph_2$,$\aleph_3$,…, and the cardinalities corresponding to the hierarchy *card*(ℕ), *card*(*P*(ℕ)), *card*(*P*(*P*(ℕ)))… as: $\beth_0, \beth_1, \beth_2, \beth_3, \ldots,$
- Then the **Generalized Continuum Hypothesis** is the following:

$$\beth_\alpha = \aleph_\alpha \text{ for all ordinals } \alpha.$$

- The (restricted) **Continuum Hypothesis** simply says that: $\beth_1 = \aleph_1$.
- The *Continuum Hypothesis* was the **first** on Hilbert's agenda-setting list of mathematical problems to solve in the 20th century. But it turns out to fall victim to the <u>incompleteness phenomenon</u> that we will discuss! (So, don't let anyone tell you that incompleteness is limited to paradoxical sentences!)

Expressive Completeness

- We said that *NDS* and even *GDS* are **redundant.** What does that mean?
- **Expressive Completeness**: A set *S* of truth-functional connectives, such as {~, v, &, →, ↔}, is called **expressively complete** just in case every unary and binary truth-functional connective is expressible in terms of the set *S*.
  - *Example*: It is clear that *if* {~, v, &, →, ↔, ∀} is expressively complete, *then* so is {~, v, &, →, ∀}. We may **define (*P* ↔ *Q*)** as an **abbreviation** for **(*P* → *Q*) & (*Q* → *P*)** since we know that it has the same **truth-table**.
- *Recall*: The truth-value of sentence involving **truth-functional connectives** is **fully determined** by the truth-values of its sentential components.
- Although *NDS* and *GDS* are expressively complete, so is {~, →}. We will find that there are even sets of single connectives that are thus complete.

Unary and Binary Truth-Functions

- What are all the unary and binary truth-functions. We can list them.

Unary connective: **tautology**.

X	⊤ (standard symbol)
T	T
F	T

Unary connective: **contradiction**.

X	⊥ (standard symbol)
T	F
F	F

Unary and Binary Truth-Functions

Unary connective: **logical identity**; 'it is the case that', 'it is true that'.

X	X
T	T
F	F

Unary connective: **negation** or **denial**; 'no', 'not', 'it is not the case that', 'it is false that'.

X	¬X, ~X, –X (standard symbols)
T	F
F	T

Unary and Binary Truth-Functions

Binary connective: **conjunction**; 'and', 'as well as', 'but', 'although', 'in spite of'.

X	Y	(X∧Y), (X&Y), (X•Y) (standard symbols)
T	T	T
T	F	F
F	T	F
F	F	F

Binary connective: the **Sheffer Stroke** (or NAND); 'not both'.

X	Y	(X↑Y), (X|Y) (standard symbols)
T	T	F
T	F	T
F	T	T
F	F	T

Unary and Binary Truth-Functions

Binary connective: **Peirce's Arrow** (or NOR); 'neither-nor'.

X	**Y**	**(X↓Y)** (standard symbol)
T	T	F
T	F	F
F	T	F
F	F	T

Binary connective: **inclusive disjunction**; 'or', 'either-or', 'and/or'

X	**Y**	**(X∨Y)** (standard symbol)
T	T	T
T	F	T
F	T	T
F	F	F

Unary and Binary Truth-Functions

Binary connective: **exclusive disjunction**; 'either-or but not both'.

X	**Y**	**(X⊕Y)**, **(X+Y)** (standard symbols)
T	T	F
T	F	T
F	T	T
F	F	F

Binary connective: **material conditional** or **material implication**; 'if-then', 'only if'.

X	**Y**	**(X→Y)**, **(X⊃Y)** (standard symbols)
T	T	T
T	F	F
F	T	T
F	F	T

Unary and Binary Truth-Functions

Binary connective: **converse implication**.

X	Y	(X←Y), (X⊂Y) (standard symbols)
T	T	T
T	F	T
F	T	F
F	F	T

Binary connective: **material nonimplication**.

X	Y	(X ↛ Y), (X⊅Y) (standard symbols)
T	T	F
T	F	T
F	T	F
F	F	F

Unary and Binary Truth-Functions

Binary connective: **converse nonimplication**.

X	Y	(X↚Y), (X⊄Y) (standard symbols)
T	T	F
T	F	F
F	T	T
F	F	F

Binary connective: **biconditional**; 'if and only if', 'just in case', 'when and only when'.

X	Y	(X↔Y), (X≡Y) (standard symbols)
T	T	T
T	F	F
F	T	F
F	F	T

Unary and Binary Truth-Functions

The final six truth-functions are artificial.

The names of these connectives are, respectively, 'tautology', 'contradiction', '**X**-negation', '**Y**-negation', '**X**-projection', and '**Y**-projection'.

X	**Y**	(**X** ⊤ **Y**)	(**X** ⊥ **Y**)	(¬**XY**)	(**X**¬**Y**)	(ρ**XY**)	(**X**ρ**Y**)
T	T	T	F	F	F	T	T
T	F	T	F	F	T	T	F
F	T	T	F	T	F	F	T
F	F	T	F	T	T	F	F

Mini-Deduction System (*MDS*)

- We want to target an object language with an expressively complete set of connectives whose inference rules are **sound and complete**. The set of truth-functional connectives that we choose is Ω = {~, →}.
- Let Ω be a set of truth-functional connectives. Then a *PL* sentence, ***X***, or set of *PL* sentences, *Γ*, all of whose logical operators belong to Ω is written $\boldsymbol{X^{\Omega}}$ or $\boldsymbol{\Gamma^{\Omega}}$, respectively. We, correspondingly, call some **translation** of a *PL* sentence, ***X***, into one including only truth-functions from Ω, an **Ω expansion of *X***, and again write $\boldsymbol{X^{\Omega}}$. (In the case of ⊤ and ⊥ we pretend that a sentence/truth-value must be plugged in.)
- We now list the **Ω expansions** of every truth-function.

Ω Expansions of All Truth-Functions

Tautology:	⊤	=	$(X \to X)$
Contradiction:	⊥	=	$\neg(X \to X)$
Logical identity:	X	=	X
Negation:	$\neg X$	=	$\neg X$
Conjunction:	$(X \wedge Y)$	=	$\neg(X \to \neg Y)$
The Sheffer Stroke:	$(X \uparrow Y)$	=	$(X \to \neg Y)$
Peirce's Arrow:	$(X \downarrow Y)$	=	$\neg(\neg X \to Y)$
Inclusive Disjunction:	$(X \vee Y)$	=	$(\neg X \to Y)$
Exclusive Disjunction:	$(X \oplus Y)$	=	$\neg((\neg X \to Y) \to \neg(X \to \neg Y))$
Material conditional:	$(X \to Y)$	=	$(X \to Y)$

Ω Expansions of All Truth-Functions

Converse implication:	$(X \leftarrow Y)$	=	$(Y \to X)$				
Material nonimplication:	$(X \not\to Y)$	=	$\neg(X \to Y)$				
Converse nonimplication:	$(X \not\leftarrow Y)$	=	$\neg(Y \to X)$				
Biconditional:	$(X \leftrightarrow Y)$	=	$\neg((X \to Y) \to \neg(Y \to X))$				
Tautology and contradiction: as above.							
X-negation and **Y**-negation:	$(\neg XY)$	=	$\neg X$	and	$(X\neg Y)$	=	$\neg Y$
X-projection and **Y**-projection:	(ρXY)	=	X	and	$(X\rho Y)$	=	Y

Example: Pierce’s Arrow & Exclusive Disjunction

X	Y	$X \downarrow Y$	$\neg(\neg X \rightarrow Y)$		$X \oplus Y$	$\neg((\neg X \rightarrow Y) \rightarrow \neg(X \rightarrow \neg Y))$
T	T	T **F** T	**F** F T T T		T **F** T	**F** F T T T T T T F F T
T	F	T **F** F	**F** F T T F		T **T** F	**T** F T T F F F T T T F
F	T	F **F** T	**F** T F T T		F **T** T	**T** T F T T F F F T F T
F	F	F **T** F	**T** T F F F		F **F** F	**F** T F F F T F F T T F

Expressive Completeness of Sheffer Stroke

- Finally, we exhibit the expressive completeness of the single connective, the **Sheffer Stroke**, as follows. (Unlike the 2nd edition of Russell’s and Whitehead’s *Principia Mathematica*, we will not use it!)
- Since {~, →} is expressively complete, it suffices to give translations of negations and conditionals in terms of the Sheffer Stroke.

Negation: $\neg X$ = $(X \uparrow X)$

Material conditional: $(X \rightarrow Y)$ = $(X \uparrow (Y \uparrow Y))$

X	Y	$\neg X$	$X \uparrow X$		$X \rightarrow Y$	$X \uparrow (Y \uparrow Y)$
T	T	**F** T	T **F** T		T **T** T	T **T** T F T
T	F	**F** T	T **F** T		T **F** F	T **F** F T F
F	T	**T** F	F **T** F		F **T** T	F **T** T F T
F	F	**T** F	F **T** F		F **T** F	F **T** F T F

Mini Deduction System (*MDS*)

- Now that we have an **expressively complete** set of truth-functional connectives, it remains to find a **sound and complete** set of inference rules. (Note the difference between the two kinds of completeness!)
- Here they are (compare **1.4.5**):

Reiteration (Reit)
Modus Ponens (MP)
Conditional Proof (CP)
Universal Instantiation (UI)
Universal Generalization (UG)
Identity (Id)
Substitution (Sub)
Both parts of Reductio Ad Absurdum (RAA)

Deriving *MDS* from *GDS*

- How do we show that *MDS* is sound and complete? By using the Soundness and Completeness Theorems for *GDS,* which we remember to be *equivalent* to *NDS*. Recall that the *GDS* rules are the following:
- (1) *Reiteration*
- (2) *Conjunction Introduction – Conj* (^*I*)
- (3) *Conjunction Elimination – Simp* (^*E*)
- (4) *Conditional Introduction – CP* (→*I*)
- (5) *Conditional Elimination – MP* (→ *E*)
- (6) *Universal Introduction – UG* (∀*I*)
- (7) *Universal Elimination – UI* (∀*E*)

Deriving *MDS* from *GDS*

- (8) *Existential Introduction – EG* (∃*I*)
- (9) *Existential Elimination – EI* (∃*E*)
- (10) *Identity Introduction – Id* (=*I*)
- (11) *Identity Elimination – Sub* (=*E*)
- (12) *Negation Introduction – RAA*, *Part 1* (~*I*) (with conclusion ~***X***)
- (13) *Negation Elimination – RAA, Part 2* (~*E*) (with conclusion ***X***).
- (14) *Disjunction Introduction – Add* (v*I*)
- (15) *Disjunction Elimination* – (v*E*) is the following hypothetical rule:

Deriving *MDS* from *GDS*

- Since *MDS* already has Conditional Proof (*CP*), Modus Ponens (*MP*), the two parts of Reductio ad Absurdum (*RAA*), Universal Generalization (*UG*), Universal Instantiation (*UI*), Identity (*Id*), and Substitution (Sub), we only need to derive the introduction and elimination rules for the conjunction, disjunction, biconditional, and the existential quantifier.
- Letting Ω = {~, →, ∀}, we first prove the following:

Theorem 2.4.1: For every PL sentence **X** and for every set Γ of PL sentences, **X** is derivable from Γ in NDS iff $\mathbf{X}^{\Omega}$ is derivable from Γ^{Ω} in MDS—that is, $\Gamma \vdash_{NDS} \mathbf{X}$ iff $\Gamma^{\Omega} \vdash_{MDS} \mathbf{X}^{\Omega}$, where $\vdash_{NDS}$ and $\vdash_{MDS}$ denote the relation of derivability in NDS and MDS, respectively.

Deriving *MT*

- In order to prove this, it will be useful to begin by deriving Modus Tollens (*MT*).

Deriving MT

[n			
	⋮		
	h	$X \to Y$	Given
	⋮		
	i	$\neg Y$	Given
[n+1	i+1	X	RA
	i+2	Y	h, (i+1), MP
n+1]	i+3	$\neg Y$	i, Reit
	i+4	$\neg X$	(i+1)–(i+3) RAA
	⋮		
n]			

Deriving ^*I*

$\wedge$I and $\wedge$E: $X \wedge Y = \neg(X \to \neg Y)$

Deriving $\wedge$I:

[n			
	⋮		
	h	X	Given
	⋮		
	i	Y	Given
[n+1	i+1	$X \to \neg Y$	RA
	i+2	$\neg Y$	h, (i+1), MP
n+1]	i+3	Y	i, Reit
	i+4	$\neg(X \to \neg Y)$	(i+1)–(i+3), RAA
	⋮		
n]			

Deriving ^*E*

Deriving ∧E (two parts):

[n			
	⋮		
	h	$\neg(X \rightarrow \neg Y)$	Given
[n+1	h+1	$\neg X$	RA
[n+2	h+2	X	CPA
[n+3	h+3	Y	RA
	h+4	X	h+2, Reit
n+3]	h+5	$\neg X$	h+1, Reit
n+2]	h+6	$\neg Y$	(h+3)–(h+5), RAA
	h+7	$X \rightarrow \neg Y$	(h+2)–(h+6), CP
n+1]	h+8	$\neg(X \rightarrow \neg Y)$	h, Reit
	h+9	X	(h+1)–(h+8), RAA
	⋮		
n]			
[n			

Deriving ^*E* Continued

	⋮		
	h	$\neg(X \rightarrow \neg Y)$	Given
[n+1	h+1	$\neg Y$	RA
[n+2	h+2	X	CPA
n+2]	h+3	$\neg Y$	h+1, Reit
	h+4	$X \rightarrow \neg Y$	(h+2)–(h+3), CP
n+1]	h+5	$\neg(X \rightarrow \neg Y)$	h, Reit
	h+6	Y	(h+1)–(h+5), RAA
	⋮		
n]			

Deriving *vI*

$\vee$I and $\vee$E: $\mathbf{X} \vee \mathbf{Y} = \neg\mathbf{X} \rightarrow \mathbf{Y}$

Deriving $\vee$I (two parts):

[n			
	⋮		
	h	$\mathbf{X}$	Given
[n+1	h+1	$\neg\mathbf{X}$	CP
[n+2	h+2	$\neg\mathbf{Y}$	RA
	h+3	$\mathbf{X}$	h, Reit
n+2]	h+4	$\neg\mathbf{X}$	h+1, Reit
n+1]	h+5	$\mathbf{Y}$	(h+2)–(h+4), RAA
	h+6	$\neg\mathbf{X} \rightarrow \mathbf{Y}$	(h+1)–(h+5), CP
	⋮		
n]			

Deriving *vI* Continued

[n			
	⋮		
	h	$\mathbf{Y}$	Given
[n+1	h+1	$\neg\mathbf{X}$	CP
n+1]	h+2	$\mathbf{Y}$	h, Reit
	h+3	$\neg\mathbf{X} \rightarrow \mathbf{Y}$	(h+1)–(h+2), CP
	⋮		
n]			

Deriving *vE*

Deriving ∨E:

[n			
	⋮		
	h	$\neg X \to Y$	Given
[n+1	h+1	X	Assumption
	⋮		
n+1]	m	Z	(the subderivation (h+1)–m is given)
[n+j	m+1	Y	Assumption
	⋮		
n+j]	k	Z	(the subderivation (m+1)–k is given)
	k+1	$X \to Z$	(h+1)–m, CP
	k+2	$Y \to Z$	(m+1)–k, CP
[n+j+1	k+3	$\neg Z$	RA
	k+4	$\neg X$	k+1, k+3, MT
	k+5	$\neg Y$	k+2, k+3, MT
n+j+1]	k+6	Y	h, k+4, MP
	k+7	Z	(k+3)–(k+6), RAA
	⋮		
n]			

Deriving ↔*I*

↔I and ↔IE: $X \leftrightarrow Y = \neg((X \to Y) \to \neg(Y \to X))$

Deriving ↔I:

[n			
	⋮		
[n+1	h	X	Assumption
	⋮		
n+1]	m	Y	(subderivation h–m is given)
[n+j	m+1	Y	Assumption
	⋮		
n+j]	k	X	(subderivation (m+1)–k is given)
	k+1	$X \to Y$	h–m, CP
	k+2	$Y \to X$	(m+1)–k, CP
[n+j+1	k+3	$(X \to Y) \to \neg(Y \to X)$	RA
	k+4	$\neg(Y \to X)$	k+1, k+3, MP
n+j+1]	k+5	$Y \to X$	k+2, Reit
	k+6	$\neg((X \to Y) \to \neg(Y \to X))$	(k+3)–(k+5), RAA
	⋮		
n]			

Deriving ←→*E*

Deriving ↔E (two parts)

[n			
	⋮		
	h	$\neg((\mathbf{X}\rightarrow\mathbf{Y})\rightarrow\neg(\mathbf{Y}\rightarrow\mathbf{X}))$	Given
	⋮		
	i	$\mathbf{X}$	Given
[n+1	i+1	$\neg\mathbf{Y}$	RA
[n+2	i+2	$\mathbf{X}\rightarrow\mathbf{Y}$	CPA
[n+3	i+3	$\mathbf{Y}\rightarrow\mathbf{X}$	RA
	i+4	$\mathbf{Y}$	i, i+2, MP
n+3]	i+5	$\neg\mathbf{Y}$	i+1, Reit
n+2]	i+6	$\neg(\mathbf{Y}\rightarrow\mathbf{X})$	(i+3)–(i+5), RAA
	i+7	$(\mathbf{X}\rightarrow\mathbf{Y})\rightarrow\neg(\mathbf{Y}\rightarrow\mathbf{X})$	(i+2)–(i+6), CP
n+1]	i+8	$\neg((\mathbf{X}\rightarrow\mathbf{Y})\rightarrow\neg(\mathbf{Y}\rightarrow\mathbf{X}))$	h, Reit
	i+9	$\mathbf{Y}$	(i+1)–(i+8), RAA
	⋮		
n]			

Deriving ←→ *Continued*

[n			
	⋮		
	h	$\neg((\mathbf{X}\rightarrow\mathbf{Y})\rightarrow\neg(\mathbf{Y}\rightarrow\mathbf{X}))$	Given
	⋮		
	i	$\mathbf{Y}$	Given
[n+1	i+1	$\neg\mathbf{X}$	RA
[n+2	i+2	$\mathbf{X}\rightarrow\mathbf{Y}$	CPA
[n+3	i+3	$\mathbf{Y}\rightarrow\mathbf{X}$	RA
	i+4	$\mathbf{X}$	i, i+2, MP
n+3]	i+5	$\neg\mathbf{X}$	i+1, Reit
n+2]	i+6	$\neg(\mathbf{Y}\rightarrow\mathbf{X})$	(i+3)–(i+5), RAA
	i+7	$(\mathbf{X}\rightarrow\mathbf{Y})\rightarrow\neg(\mathbf{Y}\rightarrow\mathbf{X})$	(i+2)–(i+6), CP
n+1]	i+8	$\neg((\mathbf{X}\rightarrow\mathbf{Y})\rightarrow\neg(\mathbf{Y}\rightarrow\mathbf{X}))$	h, Reit
	i+9	$\mathbf{X}$	(i+1)–(i+8), RAA
	⋮		
n]			

Deriving ∃I

∃I and ∃E: $(\exists z)Y = \neg(\forall z)\neg Y$

Deriving ∃I: **X** is a PL sentence, **s** is a PL ***singular term*** that occurs in **X**, and **z** is a PL variable that does not occur in **X**. **X[z, s]** is a PL ***formula*** formed by replacing ***one or more*** of the occurrences of **s** in **X** by **z**.

[n				
	⋮			
	h	**X**	Given	
[n+1	h+1	$(\forall z)\neg X[z, s]$	RA	
	h+2	$\neg X[s, s]$	h+1, UI	(**X[s, s]** is obtained from **X[z, s]** by replacing all the occurrences of the variable **z** in **X[z, s]** by the singular term **s**. Since **X[z, s]** is formed by replacing ***one or more*** of the occurrences of **s** in **X** by **z**, **X[s, s]** is simply **X**.)
n+1]	h+3	**X**	h, Reit	
	h+4	$\neg(\forall z)\neg X[z, s]$	(h+1)–(h+3), RAA	
	⋮			
n]				

Deriving Deriving ∃E

Deriving ∃E: **s** is a PL ***name***, **z** is a PL variable, and **X** is a PL formula that contains occurrences of **z** but no **z**-quantifiers. **s** satisfies three conditions: (1) it does not occur in any premise or undischarged assumption prior to line h, (2) it does not occur in $\neg(\forall z)\neg X$, and (3) it does not occur in **Y**. **X[s]** is the PL sentence formed by replacing ***all*** the occurrences of **z** in X

[n				
	⋮			
	h–1	$\neg(\forall z)\neg X$	Given	
[n+1	h	**X[s]**	Assumption	
	⋮			
n+1]	k	**Y**	(the subderivation h–k is given)	
	k+1	$X[s] \rightarrow Y$	h–k, CP	
[n+2	k+2	$\neg Y$	RA	
	k+3	$\neg X[s]$	k+1, k+2, MT	
	k+4	$(\forall z)\neg X[z]$	k+3, UG	(**X[z]** is obtained from **X[s]** by replacing all the

Deriving ∃*E Continued*

	k+4	$(\forall \mathbf{z})\neg\mathbf{X}[\mathbf{z}]$	k+3, UG	($\mathbf{X}[\mathbf{z}]$ is obtained from $\mathbf{X}[\mathbf{s}]$ by replacing all the occurrences of the name **s** with the variable **z**. Given that **s** does not occur in **X** (second condition) and that $\mathbf{X}[\mathbf{s}]$ is formed by replacing ***all*** the occurrences of **z** in **X** by **s**, the formula $\mathbf{X}[\mathbf{z}]$ is simply **X**. **s** is arbitrary at line k+3 since **s** does not occur in any premise or undischarged assumption prior to line h (the first condition), **s** occurs in the assumption at line h but this assumption is discharged after line k, and **s** does not occur in the assumption at line k+2 (third condition).)
n+2]	k+5	$\neg(\forall \mathbf{z})\neg\mathbf{X}$	h–1, Reit	
	k+6	**Y**	(k+2)–(k+5), RAA	
⋮				
n]				

Taking Stock

- We have shown (against a classical metatheory) that all the rules of *DGS*, and, hence, *NDS* are derivable from the rules of *MDS*. But all the rules of *MDS* are included among those of *NDS*. Hence, *MDS* and *NDS* are **equivalent systems**: they validate just the same inferences.
- Moreover, by the Soundness of the *NDS* rules, every *PL* sentence is logically equivalent with its Ω expansion. Consequently, we have:

Theorem 2.4.2: For every PL sentence **X** and for every set Γ of PL sentences, **X** is a logical consequence of Γ iff $\mathbf{X}^{\Omega}$ is a logical consequence of Γ^{Ω}—that is, $\Gamma \models \mathbf{X}$ iff $\Gamma^{\Omega} \models \mathbf{X}^{\Omega}$.

Soundness and Completeness

- It follows from Theorems **2.4.1** and **2.4.2** that *MDS* is Sound and Complete if and only if *NDS* is Sound and Complete.

Proof

1. First, suppose that NDS is sound and complete.
2. From 1: for every PL sentence $\mathbf{X}$ and every PL set Γ, $\Gamma \models \mathbf{X}$ iff $\Gamma \vdash_{NDS} \mathbf{X}$.
3. Our goal now is to establish that MDS is sound and complete. In other words, we want to show that for every PL sentence $\mathbf{Y}^{\Omega}$ and every PL set Σ^{Ω}, $\Sigma^{\Omega} \models \mathbf{Y}^{\Omega}$ iff $\Sigma^{\Omega} \vdash_{MDS} \mathbf{Y}^{\Omega}$. We begin by assuming that $\Sigma^{\Omega} \models \mathbf{Y}^{\Omega}$.
4. From 2 and 3: $\Sigma^{\Omega} \vdash_{NDS} \mathbf{Y}^{\Omega}$.
5. From 4 by Theorem 2.4.1: $\Sigma^{\Omega} \vdash_{MDS} \mathbf{Y}^{\Omega}$.
6. From 3 through 5: if $\Sigma^{\Omega} \models \mathbf{Y}^{\Omega}$, then $\Sigma^{\Omega} \vdash_{MDS} \mathbf{Y}^{\Omega}$. Hence MDS is complete.
7. Now we assume that $\Sigma^{\Omega} \vdash_{MDS} \mathbf{Y}^{\Omega}$.
8. From 7 by Theorem 2.4.1: $\Sigma^{\Omega} \vdash_{NDS} \mathbf{Y}^{\Omega}$.

Soundness and Completeness

9. From 2 and 8: $\Sigma^{\Omega} \models \mathbf{Y}^{\Omega}$.
10. From 7 through 9: if $\Sigma^{\Omega} \vdash_{MDS} \mathbf{Y}^{\Omega}$, then $\Sigma^{\Omega} \models \mathbf{Y}^{\Omega}$. Hence MDS is sound.
11. From 1 through 10: if NDS is sound and complete, then MDS is sound and complete.
12. Second, suppose that MDS is sound and complete.
13. From 12: for every PL sentence $\mathbf{Y}^{\Omega}$ and every PL set Σ^{Ω}, $\Sigma^{\Omega} \models \mathbf{Y}^{\Omega}$ iff $\Sigma^{\Omega} \vdash_{MDS} \mathbf{Y}^{\Omega}$.
14. We want to prove now that NDS is sound and complete; that is, for every PL sentence $\mathbf{X}$ and every PL set Γ, $\Gamma \models \mathbf{X}$ iff $\Gamma \vdash_{NDS} \mathbf{X}$. We show first the completeness of NDS. Thus assume that $\Gamma \models \mathbf{X}$.
15. From 14 by Theorem 2.4.2: $\Gamma^{\Omega} \models \mathbf{X}^{\Omega}$.
16. From 13 and 15: $\Gamma^{\Omega} \vdash_{MDS} \mathbf{X}^{\Omega}$.
17. From 16 by Theorem 2.4.1: $\Gamma \vdash_{NDS} \mathbf{X}$.
18. From 14 through 17: if $\Gamma \models \mathbf{X}$, then $\Gamma \vdash_{NDS} \mathbf{X}$. This means that NDS is complete.

Soundness and Completeness

19. To show that NDS is sound, we assume that $\Gamma \vdash_{NDS} \mathbf{X}$.
20. From 19 by Theorem 2.4.1: $\Gamma^{\Omega} \vdash_{MDS} \mathbf{X}^{\Omega}$.
21. From 13 and 20: $\Gamma^{\Omega} \models \mathbf{X}^{\Omega}$.
22. From 21 by Theorem 2.4.2: $\Gamma \models \mathbf{X}$.
23. From 19 through 22: if $\Gamma \vdash_{NDS} \mathbf{X}$, then $\Gamma \models \mathbf{X}$. Hence NDS is sound.
24. From 12 through 23: if MDS is sound and complete, then NDS is sound and complete.
25. From 11 and 24: NDS is sound and complete iff MDS is sound and complete.

- **Upshot**: It suffices to prove the metatheorems to follow about *MDS*. All references to *PL* and its deductive system refer henceforth to *MDS* unless otherwise stated.

Symbolic Logic

Soundness and Completeness

The Soundness Theorem

- We have introduced the language of *PL* (with whatever non-logical vocabulary we choose) and a corresponding proof system, *MDS*, which we have shown to be sound and complete *if NDS is*.
- We have also enumerated the assumptions that we make in the **metalanguage**, when reasoning about *PL* + *MDS*. We assume *ZFC* (which proves the **Peano Axioms** of arithmetic and more), and *NDS*.
- We now investigate the scope and limitations of *PL* + *MDS*. Our first major metatheorem is that *MDS* is **sound** for the *PL* semantics. That is:

  Soundness Theorem for *PL*: For every set Γ of *PL* sentences and every sentence ***X*** of *PL*, ***if* Γ |– *X*, *then* Γ |= *X*** (that is, if ***X*** is a theorem of Γ, then ***X*** is also a logical consequence of Γ).

The Soundness Theorem

- Let Γ be any arbitrary set of *PL* sentences, of any cardinality, and ***X*** any *PL* sentence that is **derivable** from Γ.
- To say that ***X*** is **derivable** from Γ is to say that there is a *PL* **derivation**, ***D*** (metavariable!), of ***X*** from Γ. We will write $\boldsymbol{\Sigma_D}$ to designate the set of the members of Γ that are invoked in the derivation of ***X, D***. That is:
- Σ_D = {***Y***: ***Y*** ∈ Γ and ***Y*** appears in the derivation, ***D***} ≈ {premises of ***D***}
- Any *PL* derivation, ***D,*** of ***X*** is **finite sequence** of *PL* sentences, the last line of which is ***X*** itself. So, $\boldsymbol{\Sigma_D \subseteq \Gamma}$ and Σ_D is finite and includes ***X***.

The Soundness Theorem

- *Observation*: If Σ_D |= ***X***, then Γ |= ***X***.

- Suppose that Σ_D |= ***X*** and let ***M*** be <u>any model</u> of Γ that is *relevant* to ***X*** (i.e., any <u>interpretation</u> of Γ that makes its members true and *also interprets* ***X***).
- A model of Γ is also a model of any subset of Γ. Hence, ***M*** is a model of Σ_D.
- However, by assumption Σ_D |= ***X***. Therefore, ***X*** must be true on ***M*** as well.
- Since ***M*** was an <u>arbitrary</u> model of Γ, it follows that *every* model of Γ that is relevant to ***X*** is a model of ***X*** as well. That is, If Σ_D |= ***X***, then Γ |= ***X***.
- *Upshot*: To show that Γ |= ***X*** (when Γ |- ***X***) , it suffices to show that Σ_D |= ***X***.

The Soundness Theorem

- We prove the **Soundness Theorem** by *Principle of Complete Induction* (*PCI*) applied to the <u>number of a line from an arbitrary derivation</u>, ***D***. Recall:
  - **Principle of Complete Induction** (***PCI***): If ***X***($\boldsymbol{n_0}$), and for every natural number $k \geq n_0$, ***X***(***k***) when ***X***(***m***) for each m such that $n_0 \leq m < k$, then for every natural number $n \geq n_0$, ***X***(***n***).
- Let n be the number of <u>some line</u> of ***D***. If ***D*** consists of <u>*j* lines</u>, then $1 \leq n \leq j$.
- Let us write $\boldsymbol{Z_n}$ to designate the <u>sentence</u> that appears in derivation, ***D***, at line ***n***.
- Let us write $\boldsymbol{\Sigma_n}$ for the <u>set</u> of all the **premises and undischarged assumptions** that occur in ***D*** <u>at or prior to line *n*</u>.
- The <u>length</u> of ***D***, *j*, can be <u>any (finite!) number</u>. So, we will argue:
  - For <u>every</u> natural number $n \geq 1$, $\boldsymbol{\Sigma_n}$ |= $\boldsymbol{Z_n}$.

The Soundness Theorem

- Why does the fact that $\forall n \geq 1$, $\mathbf{\Sigma}_{\boldsymbol{n}} \models \boldsymbol{Z_n}$ show that $\mathbf{\Sigma}_{\boldsymbol{D}} \models \boldsymbol{X}$ (where $\mathbf{\Sigma}_{\boldsymbol{D}} \vdash \boldsymbol{X}$)?
- By the definition of a **derivation** of ***X*** of length *j*, $\boldsymbol{X} = \boldsymbol{Z_j}$ (***X*** is the last line of ***D***).
- By the proof rules, all assumptions introduced in ***D*** by hypothetical rules must be *discharged* before the conclusion of ***D***, ***X***, appears. So, the subblocks initiated in ***D*** must be *closed* before the main block can be.
- So, the set $\mathbf{\Sigma}_{\boldsymbol{j}}$ contains only premises (no undischarged assumptions) that occur in ***D***—i.e., $\mathbf{\Sigma}_{\boldsymbol{j}} = \mathbf{\Sigma}_{\boldsymbol{D}}$.
- Hence, if, $\forall n \geq 1$, $\mathbf{\Sigma}_{\boldsymbol{n}} \models \boldsymbol{Z_n}$, then, indeed, $\mathbf{\Sigma}_{\boldsymbol{D}} \models \boldsymbol{X}$.

Base Step

- Let $n = 1$. The **first line** of any derivation has no antecedents. So the sentence $\boldsymbol{Z_1}$ is either a premise, an assumption of a hypothetical rule, or an identity statement of the form $\boldsymbol{s} = \boldsymbol{s}$, introduced by rule, *Identity*.
- If $\boldsymbol{Z_1}$ is a premise or an assumption, then $\mathbf{\Sigma}_{\boldsymbol{1}} = \{\boldsymbol{Z_1}\}$. But $\mathbf{\Sigma}_{\boldsymbol{1}} = \{\boldsymbol{Z_1}\} \models \boldsymbol{Z_1}$, since any interpretation making a claim true makes that claim true!
- If $\boldsymbol{Z_1}$ is of the form $\boldsymbol{s} = \boldsymbol{s}$ (where ***s*** is any *PL* singular term), then $\mathbf{\Sigma}_{\boldsymbol{1}}$ (the set of all the **premises and undischarged assumptions** that occur in ***D*** at or prior to line 1) is empty. But $\boldsymbol{s} = \boldsymbol{s}$ is a **logical truth** (**validity**).
- So, $\emptyset \models \boldsymbol{Z_1}$ vacuously. There is no way to make $\boldsymbol{Z_1}$ false period.

Inductive Step

- For the Inductive Hypothesis, let $k > 1$, and **suppose** that $\forall m$, $1 \leq m < k$ (*Complete Induction*!), $\boldsymbol{\Sigma_m} \models \boldsymbol{Z_m}$. Since $k > 1$, there are three cases:
- (1) $\boldsymbol{Z_k}$ is a premise or an assumption introduced by some hypothetical rule
- (2) $\boldsymbol{Z_k}$ is an identity statement of the form '$\boldsymbol{s} = \boldsymbol{s}$', introduced by the rule *Identity.*
- (3) $\boldsymbol{Z_k}$ is the conclusion of one of eight *MDS* rules (other than Identity).
- There are, thus, ten cases in all.

First Three Cases

- (1) If $\boldsymbol{Z_k}$ is a premise or an assumption introduced by a hypothetical rule, then $\boldsymbol{Z_k} \in \boldsymbol{\Sigma_k}$, so certainly $\boldsymbol{\Sigma_k} \models \boldsymbol{Z_k}$.
- (2) If $\boldsymbol{Z_k}$ is an identity statement of the form '$\boldsymbol{s} = \boldsymbol{s}$', then $\boldsymbol{Z_k}$ is a logical truth, and so consequences of anything. In particular, $\boldsymbol{\Sigma_k} \models \boldsymbol{Z_k}$.
- (3) Reiteration (**Reit**):

```
[n
      ⋮
      h      X
      ⋮
      k      X      h, Reit
      ⋮
n]
```

Reiteration

- (3) If $\boldsymbol{Z_k}$ is introduced by **Reiteration** (*Reit*), then it also occurred on the p_{th} line, $p < k$. Since $p < k$, the Inductive Hypothesis applies to p. That is, $\boldsymbol{\Sigma_p} \models \boldsymbol{Z_p}$, where $\boldsymbol{Z_p}$ is the same sentence as $\boldsymbol{Z_k}$. Since *Reit* must be applied in an open block, *if* $\boldsymbol{\Sigma_p}$ contains undischarged assumptions (introduced by hypothetical rules) *then* these cannot be discharged at line k or prior to it. (If they could, then $\boldsymbol{Z_p}$ would occur in a closed block, and could not be reiterated at line k.)
- Thus, $\boldsymbol{\Sigma_p} \subseteq \boldsymbol{\Sigma_k}$. Since $\boldsymbol{\Sigma_p} \models \boldsymbol{Z_k}$, $\boldsymbol{\Sigma_k} \models \boldsymbol{Z_k}$ as well.

Case Four

- (4) Modus Ponens (**MP**):

```
[n
    ⋮
    h    X→Y
    ⋮
    i    X
    ⋮
    k    Y              h, i, MP
    ⋮
n]
```

Modus Ponens

- If $\boldsymbol{Z_k}$ is the conclusion of **Modus Ponens** (*MP*), then the antecedents are of the forms: $\boldsymbol{Y}$ and $(\boldsymbol{Y} \rightarrow \boldsymbol{Z_k})$, with $\boldsymbol{Y}$ and $(\boldsymbol{Y} \rightarrow \boldsymbol{Z_k})$ each occurring in the derivation, ***D***, prior to line *k*.
- Suppose, then, that $\boldsymbol{Y}$ and $(\boldsymbol{Y} \rightarrow \boldsymbol{Z_k})$ occur on lines *p* and *q*, respectively, where $p < q$. Since $p, q < k$, the Inductive Hypothesis applies. That is: $\boldsymbol{\Sigma_p} \models \boldsymbol{Z_p}$ and $\boldsymbol{\Sigma_q} \models \boldsymbol{Z_q}$, where $\boldsymbol{Z_p}$ is $\boldsymbol{Y}$ and $\boldsymbol{Z_q}$ is $(\boldsymbol{Y} \rightarrow \boldsymbol{Z_k})$. In other words, $\boldsymbol{\Sigma_p} \models \boldsymbol{Y}$ and $\boldsymbol{\Sigma_q} \models (\boldsymbol{Y} \rightarrow \boldsymbol{Z_k})$.
- Because *MP* must be applied in an open block, we again have that $\boldsymbol{\Sigma_p} \subseteq \boldsymbol{\Sigma_k}$ and $\boldsymbol{\Sigma_q} \subseteq \boldsymbol{\Sigma_k}$, so that $\boldsymbol{\Sigma_k} \models \boldsymbol{Y}$ and $\boldsymbol{\Sigma_k} \models (\boldsymbol{Y} \rightarrow \boldsymbol{Z_k})$.

Modus Ponens Continued

- Let ***M*** be any ***PL*** model (interpretation of $\boldsymbol{\Sigma_k}$ under which all its members are true) that is relevant to $\boldsymbol{Z_k}$. If ***M*** is not also relevant to ***Y***, expand ***M*** into a model, ***M****, which is just like ***M*** except that it interprets the additional *PL* vocabulary in ***Y***.
- ***M**** is still a model of $\boldsymbol{\Sigma_k}$ since it agrees with ***M*** on the interpretation of the vocabulary of $\boldsymbol{\Sigma_k}$. Moreover, since $\boldsymbol{Y}$ and $(\boldsymbol{Y} \rightarrow \boldsymbol{Z_k})$ are consequences of $\boldsymbol{\Sigma_k}$ (as we just argued), they must be true on ***M**** as well.
- Consequently, using *modus ponens* in the metatheory, $\boldsymbol{Z_k}$ is true on ***M****.
- But ***M*** and ***M**** agree on their interpretations of the *PL* vocabulary in $\boldsymbol{Z_k}$. Hence, $\boldsymbol{Z_k}$ must be true on ***M***. Since ***M*** was arbitrary, $\boldsymbol{\Sigma_k} \models \boldsymbol{Z_k}$.

Case Five

- (5) Conditional Proof (**CP**): 'CPA' stands for 'Conditional Proof Assumption'.

[n			
	⋮		
[n+1	h	**X**	CPA
	⋮		
n+1]	k	**Y**	
	k+1	**X→Y**	h–k, CP
	⋮		
n]			

Conditional Proof

- If $\boldsymbol{Z_k}$ is the conclusion of a **Conditional Proof** (*CP*), then a *CP* block precedes line *k*. This is initiated by an assumption ***Y*** at a line *p* prior to line *k–1* and exited at line *k*. (Line *k–1* is the last line of the *CP* block.)
- Let ***W*** be the sentence that appears on line *k–1*. Then $\boldsymbol{Z_k}$ is of the form (***Y*** → ***W***), and the *CP Assumption,* ***Y***, is discharged at line *k*.
- Since any subblock that is initiated after the *CP* block is opened must be exited before the *CP* block is exited, all the assumptions that are introduced after line *p* must be discharged at line *k–1* or before.

Conditional Proof Continued

- The *CP Assumption,* $\boldsymbol{Y}$, is introduced at line p and is discharged at line k. So, $\boldsymbol{\Sigma}_{k-1} = \boldsymbol{\Sigma}_k \cup \{\boldsymbol{Y}\}$.
- Since the Inductive Hypothesis applies to $k–1$, $\boldsymbol{\Sigma}_{k-1} \models \boldsymbol{Z}_{k-1}$.
- But $\boldsymbol{Z}_{k-1}$ is the sentence, $\boldsymbol{W}$. Thus, $\boldsymbol{\Sigma}_k \cup \{\boldsymbol{Y}\} = \boldsymbol{\Sigma}_{k-1} \models \boldsymbol{W}$.
- Let $\boldsymbol{M}$ be any model of $\boldsymbol{\Sigma}_k$ that is relevant to $\boldsymbol{Y}$ and $\boldsymbol{W}$. If $\boldsymbol{Y}$ is false in $\boldsymbol{M}$, then $(\boldsymbol{Y} \rightarrow \boldsymbol{W})$ is true in $\boldsymbol{M}$. If $\boldsymbol{Y}$ is true in $\boldsymbol{M}$, then so is $\boldsymbol{\Sigma}_k \cup \{\boldsymbol{Y}\}$. Thus, $\boldsymbol{W}$ is true in $\boldsymbol{M}$ too, in which case $(\boldsymbol{Y} \rightarrow \boldsymbol{W})$ is again true in $\boldsymbol{M}$.
- So, $\boldsymbol{Z}_k = (\boldsymbol{Y} \rightarrow \boldsymbol{W})$ is true in every model of $\boldsymbol{\Sigma}_k$ that is relevant to $\boldsymbol{Z}_k$, i.e., $\boldsymbol{\Sigma}_k \models \boldsymbol{Z}_k$.

Cases Six & Seven

- (6) & (7)

Reductio Ad Absurdum (**RAA**): 'RA' stands for 'Reductio Assumption'; this is a two-part rule.

[n			
	⋮		
[n+1	h	**X** (or ¬**X**)	RA
	⋮		
	k–1	**Y**	
n+1]	k	¬**Y**	
	k+1	¬**X** (or **X**)	h–k, RAA
	⋮		
n]			

Reductio Ad Absurdum

- If Z_k is the conclusion of the rule **Reductio Ad Absurdum** (*RAA*), then a *RAA* block is initiated at some line prior to line *k–1* with the introduction of the *Reductio Assumption* $\sim Z_k$ (or Z_k) and is exited at line *k* with the discharge of the *Reductio Assumption*.
- The last line of the *RAA* block is line *k–1*, and Z_{k-1} is $\sim Y$ (or Y), where Y ($\sim Y$) is some *PL* sentence, and Y ($\sim Y$) appears in the same *RAA* block as Z_{k-2}.
- Since Σ_k is just like Σ_{k-1}, except that it lacks the *Reductio Assumption*, we have that $\Sigma_{k-1} = \Sigma_k \cup \{\sim Z_k\}$ (alternatively: $\Sigma_{k-1} = \Sigma_k \cup \{Z_k\}$).
- Y and $\sim Y$ must occur in an open block, and as Z_{k-1} may be a premise, $\Sigma_{k-2} \subseteq \Sigma_{k-1}$.
- The Inductive Hypothesis applies to *k–1* and *k–2*. So, $\Sigma_{k-1} \models Z_{k-1} = \sim Y$ (or Y), and $\Sigma_{k-2} \models Z_{k-2} = Y$ (or $\sim Y$).

Reductio Ad Absurdum Continued

- So, $\Sigma_{k-1} \models Y$ and $\Sigma_{k-1} \models \sim Y$ (since $\Sigma_{k-2} \subseteq \Sigma_{k-1}$). As $\Sigma_{k-1} = \Sigma_k \cup \{\sim Z_k\}$ (alternatively: $\Sigma_{k-1} = \Sigma_k \cup \{Z_k\}$), $\Sigma_k \cup \{\sim Z_k\} \models Y$ and $\Sigma_k \cup \{\sim Z_k\} \models \sim Y$.
- Now let M be any *PL model* of Σ_k that is relevant to Z_k. If M is not also relevant to Y, expand M into a model, M^*, which is just like M except M^* interprets the non-logical vocabulary of Y. Then M^* is also a model of Σ_k since it agrees with M on the vocabulary of Σ_k.
- Suppose now that $\sim Z_k$ is true in M^*. Then M^* is a model $\Sigma_k \cup \{\sim Z_k\}$ (or $\Sigma_k \cup \{Z_k\}$), and, hence, both Y and $\sim Y$, which is impossible.
- Hence, Z_k ($\sim Z_k$) must be true in M^*. Since M interprets Z_k ($\sim Z_k$) as M^* does, Z_k ($\sim Z_k$) must be true on M as well. So, $\Sigma_k \models Z_k$ ($\Sigma_k \models \sim Z_k$).

Case 8

- (8) Universal Instantiation (**UI**): **s** is any PL ***singular term***, **z** is a PL variable, and **X** is a PL formula that contains occurrences of **z** but no **z**-quantifiers. **X**[**s**] is the PL sentence formed by replacing ***all*** the occurrences of **z** in **X** by **s**.

```
[n
    ⋮
    h      (∀z)X
    ⋮
    k      X[s]      h, UI
    ⋮
n]
```

Universal Instantiation

- Suppose that $\boldsymbol{Z_k}$ is the conclusion of the rule **Universal Instantiation** (*UI*).
- Then the antecedent of the rule is a sentence of the form $(\forall z)\boldsymbol{Y}$, occurring on some line *p* that is prior to line *k*, with conclusion, $\boldsymbol{Y}[\boldsymbol{t}]$, where $\boldsymbol{Y}[\boldsymbol{t}]$ is obtained from $\boldsymbol{Y}$ by replacing every occurrence of the variable $\boldsymbol{z}$ with the singular term $\boldsymbol{t}$. Hence, $\boldsymbol{Z_k}$ is $\boldsymbol{Y}[\boldsymbol{t}]$, and the Inductive Hypothesis applies to *p*. That is, $\boldsymbol{\Sigma_p} \models \boldsymbol{Z_p}$.
- *UI* must be applied in an open block. Again, $\boldsymbol{\Sigma_p} \subseteq \boldsymbol{\Sigma_k} \models \boldsymbol{Z_p} = (\forall z)\boldsymbol{Y}$.
- Let $\boldsymbol{M}$ be any *PL* model of $\boldsymbol{\Sigma_k}$ that is relevant to $\boldsymbol{Y}[\boldsymbol{t}]$. Since $\boldsymbol{\Sigma_k} \models (\forall z)\boldsymbol{Y}$, $(\forall z)\boldsymbol{Y}$ is true in $\boldsymbol{M}$. So, all substitution instances of $(\forall z)\boldsymbol{Y}$, including $\boldsymbol{Y}[\boldsymbol{t}] = \boldsymbol{Z_k}$, are too.
- Therefore, $\boldsymbol{\Sigma_k} \models \boldsymbol{Z_k}$.

Case 9

- (9) Universal Generalization (**UG**): **X** is a PL sentence, **s** is a PL ***name*** that occurs in **X**, and **z** is a PL variable that does not occur in **X**. **s** is **arbitrary** at line h, that is, **s** does not occur in any premise or undischarged assumption listed on line h or prior to it. **X**[**z**] is the PL formula formed by replacing ***all*** the occurrences of **s** in **X** by **z**.

```
[n
    ⋮
    h    X
    ⋮
    k    (∀z)X[z]    h, UG
    ⋮
n]
```

Universal Generalization

- If $\boldsymbol{Z_k}$ is the conclusion of the rule **Universal Generalization** (*UG*), then:

- $\boldsymbol{Z_k}$ is of the form (∀*z*)***Y***[*z*] (where the formula ***Y***[*z*] is obtained from the sentence ***Y*** by replacing every occurrence of ***s*** with an occurrence of *z*)
- on some line *p*, prior to line *k*, ***Y*** appears
- a name ***s*** occurs in ***Y***
- the variable *z* does not occur in ***Y***
- ***s*** does not occur in any member of $\boldsymbol{\Sigma_p}$.

Universal Generalization Continued

- Since $p < k$, the Inductive Hypothesis applies to p; i.e., $\boldsymbol{\Sigma_p} \models \boldsymbol{Z_p} = \boldsymbol{Y}$.
- Let ***M*** be any *PL* model of $\boldsymbol{\Sigma_p}$ that is relevant to (∀*z*)***Y***[*z*]. The name ***s*** does not occur in the members of $\boldsymbol{\Sigma_p}$ or in (∀*z*)***Y***[*z*]. However, we may ensure that ***s*** is in *LN*, and assign it to a member of ***M***, so that ***M*** interprets ***Y***.
- So, suppose that ***s*** is in *LN* for ***M***. Writing ***Y***[***s***] for the sentence ***Y*** to emphasize its occurrences of ***s***, note that since ***Y***[***s***] is a consequence of $\boldsymbol{\Sigma_p}$ and ***M*** is a model of $\boldsymbol{\Sigma_p}$ that is relevant to ***Y***[***s***], ***Y***[***s***] is true in ***M*** as well.
- Now let ***t*** be any name in *LN* of ***M***, and suppose that the referent of ***s*** in ***M*** is σ, and that the referent of ***t*** in ***M*** is τ. If σ and τ are the same object, then the interpretation of ***Y***[***t***] is actually identical to that of ***Y***[***s***].
- Therefore, in this case, ***Y***[***t***] is true in ***M*** because ***Y***[***s***] is.

Universal Generalization Continued

- If σ and τ are distinct, then we construct another model ***M**** that is just like ***M*** except that the referent of both ***s*** and ***t*** in ***M**** is τ. If σ now lacks a name, we introduce a new name in *LN* of ***M**** and assign it to τ.
- The name ***s*** does not appear in any sentence in $\boldsymbol{\Sigma_p}$, and the name ***s*** is the only part of the vocabulary interpreted in ***M*** that ***M**** disagrees on.
- So, the truth values of the members of $\boldsymbol{\Sigma_p}$ are the same in ***M*** and ***M****.
- Since ***M*** is a model of $\boldsymbol{\Sigma_p}$ that is relevant to ***Y***[***s***], ***M**** is too. But ***Y***[***s***] is a consequence of $\boldsymbol{\Sigma_p}$. So, ***Y***[***s***] is true in ***M****. Since ***M**** assigns τ both ***s*** and ***t***, ***M**** gives the same interpretation to ***Y***[***s***] and ***Y***[***t***] .

Universal Generalization Continued

- Thus, $\boldsymbol{Y}[\boldsymbol{t}]$ is true in $\boldsymbol{M^*}$, and since $\boldsymbol{M}$ and $\boldsymbol{M^*}$ agree on the interpretations of the vocabulary in $\boldsymbol{Y}[\boldsymbol{t}]$ (since $\boldsymbol{s}$ does not occur in $\boldsymbol{Y}[\boldsymbol{t}]$), $\boldsymbol{Y}[\boldsymbol{t}]$ is also true in $\boldsymbol{M}$.
- That is, <u>for any name</u> $\boldsymbol{t}$ in *LN* of $\boldsymbol{M}$, $\boldsymbol{Y}[\boldsymbol{t}]$ is true in $\boldsymbol{M}$. So, $(\forall z)\boldsymbol{Y}$ is true in $\boldsymbol{M}$. So, $\boldsymbol{\Sigma_p} \models (\forall z)\boldsymbol{Y}[z]$.
- Since *UG* must be applied in an <u>open block</u> $\boldsymbol{\Sigma_p} \subseteq \boldsymbol{\Sigma_k} \models (\forall z)\boldsymbol{Y}[z] = \boldsymbol{Z_k}$.

Case 10

- (10) Substitution (**Sub**): **s** and **t** are PL ***singular terms*** and **X** is a PL sentence that contains occurrences of **s**. **X**[**t**, **s**] is a PL sentence formed by replacing ***one or more*** of the occurrences of **s** in **X** by **t**. This is a two-part rule.

```
[n
   ⋮
   h      s = t (or t = s)
   ⋮
   i      X
   ⋮
   k      X[t, s]      h, i, Sub
   ⋮
n]
```

Substitution

- Finally, suppose that $\boldsymbol{Z_k}$ is the conclusion of the rule **Substitution** (*Sub*) Then sentences of the form $\boldsymbol{s} = \boldsymbol{t}$ (or $\boldsymbol{t} = \boldsymbol{s}$) and $\boldsymbol{Y}[\boldsymbol{s}]$ (or $\boldsymbol{Y}[\boldsymbol{t}]$) must proceed it, where $\boldsymbol{s}$ and $\boldsymbol{t}$ are any *PL* singular terms.
- $\boldsymbol{Z_k}$ is $\boldsymbol{Y}[\boldsymbol{s}, \boldsymbol{t}]$ ($\boldsymbol{Y}[\boldsymbol{t}, \boldsymbol{s}]$) where $\boldsymbol{Y}[\boldsymbol{s}, \boldsymbol{t}]$ is obtained from $\boldsymbol{Y}[\boldsymbol{s}]$ by replacing one or more occurrences of $\boldsymbol{s}$ with occurrences of $\boldsymbol{t}$.
- The two antecedents occur on different lines, p and q, prior to line k.
- Suppose that $p < q$. Then $\boldsymbol{Z_p}$ is $\boldsymbol{s} = \boldsymbol{t}$ and $\boldsymbol{Z_q}$ is $\boldsymbol{Y}[\boldsymbol{s}]$ (or $\boldsymbol{Y}[\boldsymbol{t}]$), writing $\boldsymbol{Y}[\boldsymbol{s}]$ to remind that $\boldsymbol{Y}$ has instances of $\boldsymbol{s}$. Since $p, q < k$, the Inductive Hypothesis applies to them; i.e., $\boldsymbol{\Sigma_p} \models \boldsymbol{s} = \boldsymbol{t}$ (or $\boldsymbol{\Sigma_p} \models \boldsymbol{t} = \boldsymbol{s}$) and $\boldsymbol{\Sigma_q} \models \boldsymbol{Y}[\boldsymbol{s}]$ (or $\boldsymbol{\Sigma_q} \models \boldsymbol{Y}[\boldsymbol{t}]$).
- As before, *Sub* must be applied in an open block, $\boldsymbol{\Sigma_p} \subseteq \boldsymbol{\Sigma_k}$ and $\boldsymbol{\Sigma_q} \subseteq \boldsymbol{\Sigma_k}$, so $\boldsymbol{\Sigma_k} \models \boldsymbol{s} = \boldsymbol{t}$ (or $\boldsymbol{\Sigma_k} \models \boldsymbol{t} = \boldsymbol{s}$) and $\boldsymbol{\Sigma_k} \models \boldsymbol{Y}[\boldsymbol{s}]$ (or $\boldsymbol{\Sigma_k} \models \boldsymbol{Y}[\boldsymbol{t}]$).

Substitution Continued

- To prove that $\boldsymbol{\Sigma_k} \models \boldsymbol{Y}[\boldsymbol{s}, \boldsymbol{t}] = \boldsymbol{Z_k}$ (or $\boldsymbol{Y}[\boldsymbol{t}, \boldsymbol{s}]$), consider a model $\boldsymbol{M}$ to be any model of $\boldsymbol{\Sigma_k}$ that is relevant to $\boldsymbol{Y}[\boldsymbol{s}, \boldsymbol{t}]$ (or $\boldsymbol{Y}[\boldsymbol{t}, \boldsymbol{s}]$).
- If $\boldsymbol{s}$ does not occur in $\boldsymbol{\Sigma_k}$ or in $\boldsymbol{Y}[\boldsymbol{s}, \boldsymbol{t}]$, then add vocabulary to $\boldsymbol{M}$ so that it interprets $\boldsymbol{s}$.
- Since $\boldsymbol{s} = \boldsymbol{t}$ and $\boldsymbol{Y}[\boldsymbol{s}]$ are consequences of $\boldsymbol{\Sigma_k}$, they are also true in $\boldsymbol{M}$, and $\boldsymbol{Y}[\boldsymbol{s}]$ and $\boldsymbol{Y}[\boldsymbol{s}, \boldsymbol{t}]$ have the same truth value in $\boldsymbol{M}$.
- Hence, $\boldsymbol{Y}[\boldsymbol{s}, \boldsymbol{t}]$ is true on $\boldsymbol{M}$, and $\boldsymbol{\Sigma_k} \models \boldsymbol{Y}[\boldsymbol{s}, \boldsymbol{t}] = \boldsymbol{Z_k}$, as desired.

Summing Up

- The ten cases that we have considered include all the ways that $\boldsymbol{Z_k}$ could be introduced at line *k* according to the rules of *MDS*.
- In all cases, we showed that $\boldsymbol{\Sigma_k} \models \boldsymbol{Z_k}$,
- We have thereby established the Inductive Step of our proof.
- This completes our proof of the Soundness Theorem for *PL*.

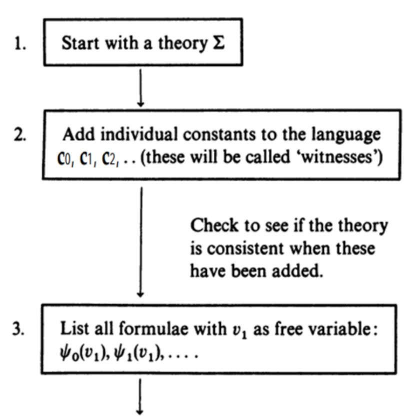
The Completeness Theorem

- The proof of the Soundness Theorem is tedious, but straightforward.
- The proof of the Completeness Theorem is more challenging and interesting. Recall that the Completeness Theorem is the *converse* of the Soundness Theorem. That is:
  - **The Completeness Theorem for *PL***: For every set Γ of *PL* sentences and every sentence ***X*** of *PL*, ***if*** **Γ |= *X*, *then* Γ |– *X*** (that is, if ***X*** is a logical consequence of Γ, then ***X*** is also a **theorem** of Γ).
- Because the proof is involved, it is useful to begin by outlining the main steps of the proof. They will serve as a roadmap for what follows.

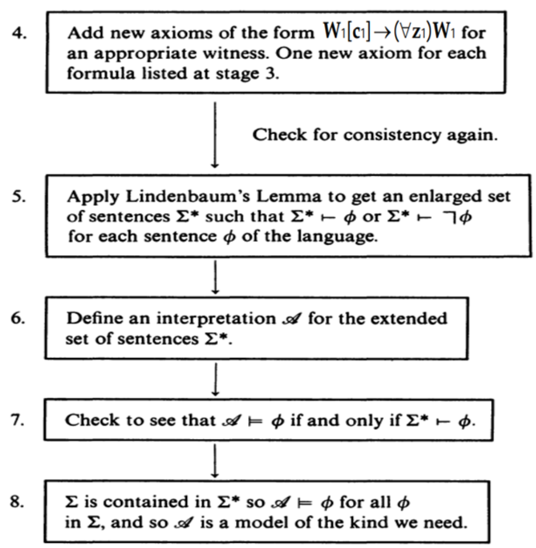
Preliminary Steps

- The proof of the Completeness Theorem begins with a simple argument that the Completeness Theorem is equivalent to the following:
  - **Model Existence Theorem for *PL***: For every set Γ of *PL* sentences, if Γ is (syntactically) *consistent*, then Γ has a model. (That is, if for no ***X*** is it the case that $\Gamma \vdash$ ***Y*** and $\Gamma \vdash$ ~***Y***, then there is a *PL* interpretation on which all of the members of Γ are true.)
- Since the Completeness Theorem and the Model Existence Theorem are *equivalent*, it suffice to prove the Model Existence Theorem.
- Here are the main steps that we take in order to do that (figure amended from that of Crossley et al.):

Main Steps of the Proof



Main Steps of the Proof



Big Idea

- We will not precisely follow this order. But the big idea of our proof will be the same: 'conflate' names with their referents; then enrich the theory, Γ (Σ in the figure) with new names, adding axioms to Γ saying that, whenever something is true a newly named object, it is true of everything; finally, interpret names as referring to equivalence classes of themselves.
- Let us first show that it suffices to prove the Model Existence Theorem.
- **Lemma 3.2.1a**: Γ ∪ (~***X***) is inconsistent if/f Γ |– ***X***. Likewise, Γ ∪ (***X***) is inconsistent if/f Γ |– ~***X***.
- **Proof**: Since the claims are relevantly identical, we prove the first.

Preliminary Lemmas

- **Inconsistency → Provability**
- 1) Suppose that Γ ∪ (~***X***) is <u>inconsistent</u>, i.e., there is an ***Y*** such that Γ |– ***Y*** and Γ |– ~***Y***.
- 2) We can combine the derivations of ***Y*** and ~***Y*** to obtain a derivation of anything, including ***X***, by Reductio Ad Absurdum. Consider:

[0	0	All the members of Γ that are invoked in D_1 and D_2	
[1	1	¬**X**	RA
	⋮	D_1	
	h	**Y**	
	⋮	D_2	
	i	¬**Y**	
1]	i+1	**Y**	h, Reit
0]	i+2	**X**	1–(i+1), RAA

Preliminary Lemmas

- 3) So, by 2), Γ |– ***X***.
- 4) Hence, by Conditional Proof (in the metatheory!), if Γ ∪ (~***X***) is <u>inconsistent</u>, then Γ |– ***X***.

- **Provability → Inconsistency**
- 1) Suppose that that Γ |– ***X***.
- 2) Then certainly Γ ∪ (~***X***) |– ***X***.
- 3) But also: Γ ∪ (~***X***) |– ~***X***.
- 4) Hence, by Conditional Proof, if Γ |– ***X***, then Γ ∪ (~***X***) is inconsistent (i.e., there is an ***Y*** – namely, ***X*** – such that Γ |– ***Y*** and Γ |– ~***Y***).

Preliminary Lemmas

- **Lemma 3.2.1b**: Γ |– **~(***W* **→** ***Z***) if/f Γ |– *W* and Γ |– **~*Z***.
- **Proof** that **If** Γ |– **~(***W* **→** ***Z***), **then** Γ |– *W* and Γ |– **~*Z***:
- Assume Γ |– **~(***W* **→** ***Z***). Then there is a *PL* derivation, ***D***, from Γ of **~(***W* **→** ***Z***). Given *D*, we construct derivations of *W* and ~***Z*** as follows:

[0	0	All the members of Γ that are invoked in D	
	⋮	D	
	i	¬(W→Z)	
[1	i+1	¬**W**	RA
[2	i+2	**W**	CPA
[3	i+3	¬**Z**	RA
	i+4	¬**W**	i+1, Reit
3]	i+5	**W**	i+2, Reit
2]	i+6	**Z**	(i+3)–(i+5), RAA
	i+7	W→**Z**	(i+2)–(i+6), CP
1]	i+8	¬(W→**Z**)	i, Reit
0]	i+9	**W**	(i+1)–(i+9), RAA

Preliminary Lemmas

[0	0	All the members of Γ that are invoked in D	
	⋮	D	
	i	¬(W→**Z**)	
[1	i+1	**Z**	RA
[2	i+2	**W**	CPA
2]	i+3	**Z**	i+1, Reit
	i+4	W→**Z**	(i+2)–(i+3), CP
1]	i+5	¬(W→**Z**)	i, Reit
0]	i+6	¬**Z**	(i+1)–(i+5), RAA

- By Conditional Proof (in the metatheory), **if** Γ |– **~(***W* **→** ***Z***), **then** Γ |– *W* and Γ |– **~*Z***. (This is a claim about what Γ *proves*!) Now to the converse:

Preliminary Lemmas

- **Proof** that **If** Γ |– *W* and Γ |– **~*Z***, **then** Γ |– **~(***W* → ***Z***).
- Assume that Γ |– *W* and Γ |– **~*Z***. Then there are corresponding derivations of *W* and **~*Z*** from Γ, D_1 and D_2, respectively. We can combine them into a single derivation of **~(***W* → ***Z***) from Γ thus:

[0	0	All the members of Γ that are invoked in D_1 and D_2.	
	⋮	D_1	
	h	**W**	
	⋮	D_2	
	i	¬**Z**	
[1	i+1	W→**Z**	RA
	i+2	**Z**	h, (i+1), MP
1]	i+3	¬**Z**	i, Reit
0]	i+4	¬(**W**→**Z**)	(i+1)–(i+3), RAA

9. From 8: Γ ⊢ ¬(**W**→**Z**).
10. From 6 through 9: if Γ ⊢ **W** and Γ ⊢ ¬**Z**, then Γ ⊢ ¬(**W**→**Z**).
11. From 5 and 10: Γ ⊢ ¬(**W**→**Z**) iff Γ ⊢ **W** and Γ ⊢ ¬**Z**.

Preliminary Lemmas

- **Lemma 3.2.1c**: **If** Γ |– ***Y***, the name ***s*** occurs in ***Y***, the variable ***z*** does not occur in ***Y***, and ***s*** does not occur in any member of Γ, **then** Γ |– (∀*z*)***Y***[*z*], where, as usual, ***Y***[*z*] is obtained from ***Y*** by replacing all the occurrences of ***s*** with occurrences of ***z***.
- **Proof**: Assume that Γ |– ***Y***, the name ***s*** occurs in ***Y***, the variable ***z*** does not occur in ***Y***, and ***s*** does not occur in any member of Γ.
- Then there is a *PL* derivation, ***D***, of ***Y*** from Γ.
- We can now use ***D*** to construct a derivation of (∀*z*)***Y***[*z*] from Γ as follows.

Preliminary Lemmas

0]	0	All the members of Γ that are invoked in D.	
	⋮	D	
	H	**Y**	**Y** is not a premise or an assumption. It is not a premise because it contains **s** and none of the premises contains **s**. It is not an assumption because it is the conclusion of a PL subderivation and the block rules do not allow a derivation to terminate with an assumption. Furthermore, all assumptions that might be introduced prior to line h must be discharged by the time h is reached. Otherwise, **Y** cannot be the conclusion of a PL subderivation. Hence **s** is arbitrary at line h. This entails that the rule Universal Generalization can be applied to line h.
0]	h+1	(∀**z**)**Y**[**z**]	h, UG

Preliminary Lemmas

- In light of **Lemma 3.2.1a**, **3.2.1b**, and **3.2.1c**, we are in a position to prove the following:
  - **Theorem 3.2.1**: The **Completeness Theorem** and the **Model Existence Theorem for *PL*** are equivalent – i.e., they imply (in the metatheory) one another.
- **Proof**: Let us introduce some notation that will come in handy down the road. We will write ***Con*(Γ)** for the claim that Γ is consistent, and **∃*M* |= Γ** for the claim that there is a model of Γ (i.e., a *PL interpretation* on which every member of Γ comes out true).

Preliminary Lemmas

- 1) Assume the **Completeness Theorem** and that *Con*(Γ).
- 2) Assume for *reductio* that ~∃*M* |= Γ.
- 3) Then, for any *PL* sentence, ***Y***, vacuously, Γ |= ***Y*** and Γ |= ~***Y***.
- 4) By 1) – namely, the *Completeness Theorem* – Γ |- ***Y*** and Γ |- ~***Y***.
- 5) So, ~*Con*(Γ), contra 1) – namely, that *Con*(Γ).
- 6) Hence, the *reductio* assumption is false, i.e., ∃*M* |= Γ.
- 7) Conversely, assume the **Model Existence Theorem** and that, for any *PL* sentence, ***Y***, and set of *PL* sentences, Γ, Γ |= ***Y***.
- 8) Then ~∃*M* |= Γ such that ~(*M* |= ***Y***). That is, ~∃*M* |= (Γ ∪ ~ ***Y***).

Preliminary Lemmas

- 9) By the **Model Existence Theorem**, *Con*(Γ ∪ ~ ***Y***) → ∃*M* |= (Γ ∪ ~ ***Y***).
- 10) Since ~∃*M* |= (Γ ∪ ~ ***Y***), we have that ~*Con*(Γ ∪ ~ ***Y***).
- 11) But, then, by **Lemma 3.2.1a** (that Γ ∪ (~***X***) is inconsistent if/f Γ |– ***X***), Γ |– ***Y***.
- 12) Hence, by *Conditional Proof* (in the metatheory), **if** Γ |= ***Y***, **then** Γ |– ***Y***, which is just the **Completeness Theorem**.

- Upshot: We can speak ambiguously of the **Completeness Theorem** *per se* and the **Model Existence Theorem** with 'The Completeness Theorem'.

Maximal Consistent sets of Sentences

- **Definition 3.2.1**: Let Δ be any set of *PL* sentences. We say:
- **3.2.1a** Δ is **maximal** if/f for every *PL* sentence ***X***, Δ includes it or its negation – i.e., either ***X*** ∈ Δ or ~***X*** ∈ Δ .
- **3.2.1b** Δ is **deductively closed** if/f Δ contains all its theorems – i.e., for every *PL* sentence ***X*** such that Δ |– ***X***, ***X*** ∈ Δ.
- **3.2.1c** Δ is **semantically closed** if/f Δ contains all its logical consequences – i.e., for every *PL* sentence ***X*** such that Δ |= ***X***, ***X*** ∈ Δ.
  - *Note*: The **Soundness Theorem** and the **Completeness Theorem** will ensure that deductive closure and semantic closure coincide.

Maximal Consistent sets of Sentences

- **Lemma 3.2.2a**: Every maximal **consistent** set is deductively closed.
- **Proof**:
- 1) Let Δ be maximal consistent.
- 2) Let Δ |- ***X*** (for some *PL* sentence, ***X***).
- 3) Suppose for *reductio* that ***X*** ∉ Δ.
- 4) Then ~***X*** ∈ Δ, since Δ is maximal.
- 5) So certainly Δ |- ~***X*** (since {***Y***} ∪ Γ |- ***Y*** for any ***Y*** and Γ).
- 6) Hence, Δ is inconsistent.
- 7) By *reductio ad absurdum*, ***X*** ∈ Δ – i.e., Δ is deductively closed.

Maximal Consistent sets of Sentences

- **3.2.2b**: A set Δ is **maximal consistent** if/f it is **consistent** and for every **proper extension** of it, Δ' ⊃ Δ, Δ' is inconsistent. (When Δ' is a mere **extension** of Δ, we will write Δ' ⊇ Δ, or equivalently Δ ⊆ Δ'.)
- **Proof** (right-to-left):
- 1) Assume that Δ is **consistent** and for every set, Δ', such that Δ ⊂ Δ', Δ' is inconsistent.
- 2) Suppose for *reductio* that there ∃***X*** such that ***X*** ∉ Δ and ~***X*** ∉ Δ.
- 3) Hence, Δ ⊂ Δ ∪ {***X***} and Δ ⊂ Δ ∪ {~***X***}.
- 4) Then, by 1), Δ ∪ {***X***} and Δ ∪ {~***X***} are both inconsistent.
- 5) So, again by **Lemma 3.2.1a**, Δ |– ***X*** and Δ |– **~*X***, i.e., Δ is inconsistent.
- 6) Therefore, the *reductio* assumption is false, and Δ is **maximal**.

Maximal Consistent sets of Sentences

- **3.2.2b**: A set Δ is **maximal consistent** if/f it is **consistent** and for every set, Δ', such that Δ ⊂ Δ', Δ' is inconsistent.
- **Proof** (left-to-right):
- 1) Assume that Δ is **maximal consistent** and that ∃Δ' with Δ ⊂ Δ'.
- 2) Consider an ***X*** such that ***X*** ∈ Δ' but ***X*** ∉ Δ.
- 3) Since Δ is maximal, and ***X*** ∉ Δ, ~***X*** ∈ Δ.
- 4) Since Δ ⊂ Δ', ~***X*** ∈ Δ'.
- 5) But, then, ***X*** ∈ Δ' and ~***X*** ∈ Δ'. So, Δ' is inconsistent, as desired.

Lindenbaum's Lemma

- With **Lemma 3.2.2a** and **3.2.2b** in hand, we can now proceed to the first major step of the proof of the **Model Existence Theorem**.
- **Lindenbaum's Lemma**: Every consistent set of *PL* sentences can be extended into a maximal consistent *PL* set.
- **Proof**:
- 1) Let Γ be a consistent set of *PL* sentences.
- 2) Fix an **enumeration** of all sentences in the language of *PL*, $\boldsymbol{X_1}$, $\boldsymbol{X_2}, \ldots \boldsymbol{X_n}, \ldots$ (This is a countable set of finite strings.)
- 3) Inductively define the following **extension** of the set Γ:

Lindenbaum's Lemma

- $\Delta_0 = \Gamma$
- $\Delta_{k+1} = \Delta_k \cup \{\boldsymbol{X_k}\}$ if $\Delta_k \cup \{\boldsymbol{X_k}\}$ is **consistent**
- $\Delta_{k+1} = \Delta_k$ if $\Delta_k \cup \{\boldsymbol{X_k}\}$ is **inconsistent**
- *Idea*: Begin with a theory, Γ, and, for every sentence in our enumeration, $\boldsymbol{X}$, add it if this is consistent, and leave the construction alone if it is not.
- *Note*: By construction, the sets so constructed are **nested**, $\Delta_0 \subseteq \Delta_1 \subseteq \Delta_2$...
- We also define:
- $\boldsymbol{F} = \{\Delta_k = k \in \mathbb{N}\}$
- ***Lindenbaum Set*** $= \boldsymbol{\Delta} = \cup \boldsymbol{F} = \{\boldsymbol{X} : \exists \Delta_{\boldsymbol{i}} \in \boldsymbol{F}\ \&\ \boldsymbol{X} \in \Delta_{\boldsymbol{i}}\}$
- *Note*: $\Delta_k \subseteq \boldsymbol{\Delta}$, $\forall k \in \mathbb{N}$, i.e., $\boldsymbol{\Delta}$ is an **extension** of every Δ_k, including Γ.

Lindenbaum's Lemma

- It is intuitively clear that each set, Δ_k, is consistent. However, we can 'prove' this fact inductively, since the sets were defined by induction.
- $\Delta_0 = \Gamma$ is consistent by assumption. Now suppose that Δ_k is consistent. Then $\Delta_{k+1} = \Delta_k \cup \{\boldsymbol{X_k}\}$ or $\Delta_{k+1} = \Delta_k$ depending on whether this is consistent. So, Δ_{k+1} is also consistent. Hence, $\forall k \in \mathbb{N}$, Δ_k is consistent.
- Why is the full ***Lindenbaum Set*** = **Δ** consistent?
- 1) Suppose for *reductio* that it is not.
- 2) Then ∃***Y*** such that **Δ** |- ***Y*** and **Δ** |- ~***Y***. Let $\Sigma_{\boldsymbol{Y}}$ and $\Sigma_{\sim\boldsymbol{Y}}$ be the sets of premises from **Δ** that occur in the derivations of ***Y*** and ~***Y***, respectively. Writing $\Sigma = \Sigma_{\boldsymbol{Y}} \cup \Sigma_{\sim\boldsymbol{Y}}$, we have that Σ |- ***Y*** and Σ |- ~***Y***.

Lindenbaum's Lemma

- 3) Σ ⊆ **Δ**. Since **Δ** = ∪***F***, ***Z*** ∈ Σ → ***Z*** ∈ Δ_k for some $k \in \mathbb{N}$ -- and, indeed, ***Z*** ∈ Δ_j for all $j \geq k$. We will write $\Delta_{\boldsymbol{Z}}$ for the **first** Δ_m such that ***Z*** ∈ Δ_m.
- 4) Let ***K*** ={$\Delta_{\boldsymbol{Z}}$: ***Z*** is a member Σ}. (That is, for each ***Z*** in Σ, ***K*** collects the first Δ_m in which it occurs.) ***K*** is **finite** (since Σ is), and its members form a **nested chain** ordered by ⊆ relation (since the Δ_ms of ***F*** do). Thus, ***K*** has a top element Δ* of which all other elements of ***K*** are subsets.
- 5) Σ ⊆ Δ*. But, then, Δ* |- ***Y*** and Δ*|- ~***Y***. Since $\Delta^* = \Delta_m$ for some $m \in \mathbb{N}$ (each of which is consistent), the *reductio* assumption must be false.
- 5) Hence, by Reductio ad Absurdum (in the metatheory) **Δ** is consistent.

Lindenbaum's Lemma

- Why is the full ***Lindenbaum Set*** = **Δ** maximal?

- 1) Let Δ' be a set such that **Δ** ⊂ Δ'.
- 2) Then there is an X_i such that $X_i \in$ Δ' and $X_i \notin$ Δ.
- 3) Since $X_i \notin$ Δ, it was excluded from Δ_{i+1}.
- 4) By the construction of **Δ**, $\Delta_i \cup X_i$ is inconsistent.
- 5) So, by **3.2.2b**, **Δ** is maximal, if consistent.
- 6) Since we just proved that **Δ** is consistent, **Δ** is indeed maximal.

Henkin Sets

- We have now proved **Lindenbaum's Lemma**, that every consistent set of *PL* sentences can be extended into a maximal consistent *PL* set.
- We next use Lindenbaum's Lemma to enlarge a given consistent set, Γ, into a maximal consistent set that, intuitively, **captures the truth conditions** of all *PL* sentences. A set with this feature is a **Henkin set**.
- **Conditional** (→): We know that (***X*** → ***Y***) is true in a model, ***M***, just when either (inclusive) ***X*** is false in ***M*** or ***Y*** is true in ***M***. So, if **Δ** 'captures the truth conditions' of (***X*** → ***Y***), then we should have that (***X*** → ***Y***) ∈ **Δ** just in case ~***X*** ∈ **Δ** or ***Y*** ∈ **Δ** (or both). This is the case:

Henkin Sets

- First, we can show that ~***X*** ∈ **Δ** → (***X*** → ***Y***) ∈ **Δ** as follows.

[0	0	¬**X**	P
[1	1	**X**	CPA
[2	2	¬**Y**	RA
	3	¬**X**	1, Reit
2]	4	**X**	2, Reit
1]	5	**Y**	2–4, RAA
0]	6	**X**→**Y**	1–5, CP

- This shows that if ~***X*** ∈ **Δ** → (***X*** → ***Y***) ∈ **Δ** by **Lemma 3.2.2a**, that every maximal **consistent** set (including **Δ**) is deductively closed.

Henkin Sets

- There is also derivation demonstrating that if ***Y*** ∈ **Δ**, then (***X*** → ***Y***) ∈ **Δ**.

[0	0	**Y**	P
[1	1	**X**	CPA
1]	2	**Y**	0, Reit
0]	3	**X**→**Y**	1–2, CP

- What about the other direction? Let (***X*** → ***Y***) ∈ **Δ**. For each of ***X*** and ***Y***, either it or its negation, but not both (!), is included in **Δ** by the maximal consistency of **Δ**. But if ***X*** ∈ **Δ** and ~***Y*** ∈ **Δ**, then **Δ** is inconsistent. So, we must have that ~***X*** ∈ **Δ** or (inclusive) ~***Y*** ∈ **Δ**.

Henkin Sets

- There are two remaining connectives in the language of *MDS,* **Ω =** {~, →, ∀}, whose truth-conditions we may hope are mirrored within the ***Lindenbaum Set*** = **Δ**. One of the connectives is straightforward.
- **Negation** (~): ~***X*** is true in a model, ***M***, just when ***X*** is false in ***M***. Thus, if **Δ** 'captures the truth conditions' of ~***X***, then we should have that ~***X*** ∈ **Δ** just in case ***X*** ∉ **Δ**. Indeed, either ***X*** ∈ **Δ** or ~***X*** ∈ **Δ** by **Δ**'s maximality. And by **Δ**'s consistency, if ~***X*** ∈ **Δ** (so **Δ** |- ~***X***), then ***X*** ∉ **Δ** and conversely.
- The subtle case is the universal quantifier, ∀. By the deductive closure of **Δ**, if (∀***z***)***Y*** ∈ **Δ**, then ***Y***[***s***] ∈ **Δ**, for every *PL* name, ***s***. What about the converse?

Henkin Sets & Universal Quantification

- Consider a *1*-place predicate ***Y*** and let Ψ be the set of all *PL* sentences that result from appending a *PL* name ***s*** to the predicate, ***Y***. That is:
- Ψ ={***Ys***: ***s*** is a *PL* name}.
- Now consider the following interpretation, *J:*
- *UD* = {*–1*, *0*, *1*, *2*, *3*, …}
- *LN* = {*x* : *x* is a *PL* name} ∪ {$\mathbf{c}_{-1}$}
- Interpretation, *J,* assigns the appended **constant**, $\mathbf{c}_{-1}$, to *–1*, *PL* **names** to non-negative integers, and ***Y*** to the set of non-negative integers.

Henkin Sets & Universal Quantification

- By design, ***Ys*** is true for every *PL* name, ***s***. But the universal claim, $(\forall z)Y$, is not because it has a false substitution instance in *LN*: $\Psi \nVdash Y[\mathbf{c}_{-1}]$.
- By the **Soundness Theorem**, $\Psi \nvdash Y[\mathbf{c}_{-1}]$. So, by **Lemma 3.2.1a** (i.e., that $\Gamma \cup \sim X$ is inconsistent if/f $\Gamma \vdash X$), $Con(\Delta \cup \sim Y[\mathbf{c}_{-1}])$. By ***Lindenbaum's Lemma***, $\Delta \cup \sim Y[\mathbf{c}_{-1}]$ may be extended to a maximal consistent set.
- *Note*: $Con(\Delta \cup \sim Y[\mathbf{c}_{-1}])$ concerns sentences in the extended Vocabulary of the Interpretation (*Voc(J)*). We exploit the substitutional interpretation the language of *PL* to show that $\Delta \nvdash (\forall z)Y$ in the language of *PL*.
- *Upshot*: Even if a maximal consistent set contains every basic substitutional instance of $(\forall z)Y$, it need not contain $(\forall z)Y$ itself.

Henkin Sets

- How do we ensure that the maximal consistent set we end up with contains $(\forall z)Y$ when it contains every basic substitution instance? We can demand that it includes $Y[c]$, where ***c*** is **arbitrary**. Then Universal Generalization applies, and $(\forall z)Y$ belongs to the set by closure.
- In order to guarantee the existence of an arbitrary name for each universally quantified sentence, we add countably-many new names to the language of *PL*: $\alpha_1, \alpha_2, \alpha_3, \ldots \alpha_n, \ldots$ These are called **α names**, and the system of predicate logic with the new vocabulary is called ***PL+***.
- *Note*: Since our initial set, Γ, is in the language of *PL*, and the language of *PL+* includes that of *PL*, the set Γ is also in the language of *PL+*.

Henkin Sets

- As with all sentences of *PL*, $X_1, X_2, \ldots X_n, \ldots,$ we fix an enumeration of all universally quantified sentences of *PL+*: $(\forall z_0)W_0$, $(\forall z_1)W_1$, $(\forall z_2)W_2, \ldots$ $(\forall z_n)W_n, \ldots$ Since any finite set of such sentences uses only finitely-many **α names**, we always have infinitely-many other **α names** to choose from.
- We can, therefore, construct the following sequence of special **α names.**
- Let us define $\mathbf{c}_0$, $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \ldots \mathbf{c}_n, \ldots$ as a sequence of **α names** such that:
- $\mathbf{c}_0$ is the first **α name** (in the above list) that does not occur in $(\forall z_0)W_0$
- $\mathbf{c}_1$ is the first **α name** that does not occur in $(\forall z_1)W_1$
- …
- In general, $\mathbf{c}_n$ is the first **α name** that does not occur in $(\forall z_n)W_n$
- *Note*: Remember that the $\mathbf{c}_i$s are metalinguistic variables, not *PL+* names.

Henkin Sets

- Given our stock of **α names**, and enumeration, $\mathbf{c}_0$, $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \ldots \mathbf{c}_n, \ldots$, we now define, for every universally quantified sentence, $(\forall z_n)W_n$, the W_n-**Conditional**, θ_n:
- θ_0: $W_0[\mathbf{c}_0] \rightarrow (\forall z_0)W_0$
- θ_1: $W_1[\mathbf{c}_1] \rightarrow (\forall z_1)W_1$
- …
- θ_n: $W_n[\mathbf{c}_n] \rightarrow (\forall z_n)W_n$
- (where $W_n[\mathbf{c}_n]$ is obtained from W_n by replacing all occurrences of z_n with occurrences of $\mathbf{c}_n$)
- *Intuition*: Each θ_n promises an arbitrary basic substitutional instance justifying the universally quantified sentence, $(\forall z_n)W_n$. Alternatively, one can think of the θ_ns as promising a **witness**, $\mathbf{c}_n$, whenever $(\exists z_n)W_n$ holds.

Henkin Sets

- **Theorem 3.2.2**: If $\Theta = \{\boldsymbol{\theta_n} : n \in \mathbb{N}\}$, then *Con*(Γ ∪ Θ).
  - *Note*: Remember that Γ is an arbitrary consistent set of ***PL*** sentences.
- **Proof**:
- 1) Suppose for *reductio* that ~*Con*(Γ ∪ Θ), given that *Con*(Γ).
- 2) Let ***Y*** be a sentence such that (Γ ∪ Θ) |- ***Y*** and (Γ ∪ Θ) |- ~***Y***, and let $\Sigma_{\boldsymbol{Y}}$ and $\Sigma{\sim}_{\boldsymbol{Y}}$ be the sets of premises from (Γ ∪ Θ) invoked in some fixed derivations of ***Y*** and ~***Y***, respectively.
- 3) Since $\Sigma_{\boldsymbol{Y}} \subseteq (\Gamma \cup \Theta)$ and $\Sigma{\sim}_{\boldsymbol{Y}} \subseteq (\Gamma \cup \Theta)$, $(\Sigma_{\boldsymbol{Y}} \cup \Sigma{\sim}_{\boldsymbol{Y}}) = \boldsymbol{\Sigma} \subseteq (\Gamma \cup \Theta)$.
- 4) Thus, **Σ** |- ***Y*** and **Σ** |- ~***Y***, and **Σ** ⊆ (Γ ∪ Θ) is inconsistent.

Henkin Sets

- 5) Since *Con*(Γ), **Σ** cannot be a subset of Γ.
- 6) Hence, (**Σ** ∩ Θ) ≠ ∅, and we may designate (**Σ** ∩ Θ) = Φ.
- 7) We likewise designate (**Σ** ∩ Γ) = Ψ.
- 8) Given our definitions:
- (a) (Φ ∪ Ψ) = (**Σ** ∩ Θ) ∪ (**Σ** ∩ Γ) = **Σ**.
- (b) Ψ is a finite subset of Γ (because **Σ** is finite).
- (c) Φ is a nonempty but finite subset of Θ (it is nonempty by 6).
- *Observation*: Because **Σ** = (Φ ∪ Ψ) is inconsistent, and each of Φ and Ψ is a finite set (because each is formed by intersecting a set with the finite set, **Σ**), the union of Γ and **finitely many** *W*-Conditionals must be inconsistent.

Henkin Sets

- 5) Define **Λ** to be the set of the <u>first *q* *W*-conditionals</u>, $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{q-1}, \boldsymbol{\theta}_q$ such that $Con(\Gamma \cup \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{q-1}\})$ but $\sim Con(\Gamma \cup \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{q-1}, \boldsymbol{\theta}_q\})$. That is, $\sim Con(\Gamma \cup \Lambda)$, but $Con(\Gamma \cup \boldsymbol{\Lambda} - \{\boldsymbol{\theta}_q\})$.
  - *Note*: There must be such a set, **Λ**, even if it is only the singleton, $\{\boldsymbol{\theta}_0\}$, since $Con(\Gamma)$ but the union of Γ and finitely-many *W*-conditionals is inconsistent.
- 6) By **Lemma 3.2.1a** (that $\Gamma \cup (\sim \boldsymbol{X})$ is <u>inconsistent</u> if/f $\Gamma \vdash \boldsymbol{X}$), $\Gamma \cup \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{q-1}\} \vdash \sim\boldsymbol{\theta}_q$.
- 7) From 6), by the definition of $\boldsymbol{\theta}_q$, $\Gamma \cup \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{q-1}\} \vdash \sim(\boldsymbol{W}_q[\mathbf{c}_q] \rightarrow (\forall z_q)\boldsymbol{W}_q)$
- 8) By **Lemma 3.2.1b** (that $\Gamma \vdash \sim(W \rightarrow \boldsymbol{Z})$ if/f $\Gamma \vdash W$ <u>and</u> $\Gamma \vdash \sim\boldsymbol{Z}$), $\Gamma \cup \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{q-1}\} \vdash \boldsymbol{W}_q[\mathbf{c}_q]$ <u>and</u> $\Gamma \cup \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{q-1}\} \vdash \sim(\forall z_q)\boldsymbol{W}_q$.

Henkin Sets

- 9) Since Γ is a set of ***PL*** sentences, no member of Γ contains any **α name**, much less $\mathbf{c}_q$. Moreover, as $\mathbf{c}_q$ is <u>*q*th in the list</u> of **α names**, it does not occur in any of the first $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{q-1}$ ***W*-Conditionals**.
- <u>Upshot</u>: *Relative to* $\Gamma \cup \boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{q-1}$, $\mathbf{c}_q$ occurs **arbitrarily** in $\boldsymbol{W}_q[\mathbf{c}_q]$.
- 10) We now recall **Lemma 3.2.1c**: **If** $\Gamma \vdash \boldsymbol{Y}$, the <u>name</u> ***s*** occurs in ***Y***, the <u>variable</u> ***z*** does <u>not</u> occur in ***Y***, and ***s*** does <u>not</u> occur in <u>any member</u> of Γ, **then** $\Gamma \vdash (\forall z)\boldsymbol{Y}[z]$. Consequently, $\Gamma \cup \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{q-1}\} \vdash (\forall z_q)\boldsymbol{W}_q$.
- 11) Since 10) contradicts 8), we conclude that $Con(\Gamma \cup \Theta)$, given that $Con(\Gamma)$, as desired.

Lindenbaum's Lemma Again

- ***Upshot***: *Con*(T) → *Con*(Γ ∪ Θ). Hence, so long as we begin with a consistent set, Γ (i.e., Con(T)), we can make use of Lindenbaum's Lemma and extend (Γ ∪ Θ) into a maximal consistent set of *PL*+ sentences, **Π**, that *also* captures the truth-conditions of *every* sentence in the *PL*+ language.
- **Π** is called a **Henkin Set**. We already know, by **Π**'s maximal consistency, that that ($\boldsymbol{X} \rightarrow \boldsymbol{Y}$) ∈ **Π** just in case ~$\boldsymbol{X}$ ∈ **Π** or $\boldsymbol{Y}$ ∈ **Π** (or both), that ~$\boldsymbol{X}$ ∈ **Π** just in case $\boldsymbol{X}$ ∉ **Π**, and that if $(\forall z)\boldsymbol{Y}$ ∈ **Π** then $\boldsymbol{Y}[\boldsymbol{s}]$ ∈ **Π**, for every *PL*+ name, $\boldsymbol{s}$.
- However, we are finally in a position to show the following:

Lindenbaum's Lemma Again

- ∀ **Lemma**: If $\boldsymbol{Y}[\boldsymbol{s}]$ ∈ **Π**, for every *PL*+ name, $\boldsymbol{s}$, then $(\forall z)\boldsymbol{Y}$ ∈ **Π**.
- **Proof**: Suppose that $\boldsymbol{Y}[\boldsymbol{s}]$ ∈ **Π**, for every *PL*+ **name**, $\boldsymbol{s}$. The sentence, $(\forall z)\boldsymbol{Y}$, must occur as some $(\forall z_k)\boldsymbol{W}_k$, since, by construction, every universally quantified sentence of *PL*+ appears in our enumeration.
- Since Θ ⊆ **Π**, the conditional, $\boldsymbol{W}_k[\mathbf{c}_k] \rightarrow (\forall z_k)\boldsymbol{W}_k$, which is just $\boldsymbol{\theta}_k$, is a member of **Π**. But, by assumption, $\boldsymbol{Y}[\boldsymbol{s}] = \boldsymbol{W}_k[\mathbf{c}_n]$ ∈ **Π**. Hence:
- $\{\boldsymbol{W}_k[\mathbf{c}_n], \boldsymbol{W}_k[\mathbf{c}_k] \rightarrow (\forall z_k)\boldsymbol{W}_k\} \subseteq$ **Π**. Since **Π** is deductively closed (because it is maximally consistent), $(\forall z_k)\boldsymbol{W}_k \subseteq$ **Π** by modus ponens.
- *Upshot*: The maximal consistent **Henkin Set**, **Π**, captures the truth-conditions of universally quantified sentences as it does the others. We have that, indeed, $(\forall z)\boldsymbol{Y}$ ∈ **Π** just in case $\boldsymbol{Y}[\boldsymbol{s}]$ ∈ **Π** for every *PL*+ **name**, $\boldsymbol{s}$.

Terms

- Actually, the same holds for substitution instances involving complex terms, not names. (∀*z*)***Y*** ∈ **Π** just in case ***Y***[***t***] ∈ **Π** for every *PL*+ **singular term**, ***t***.
- **Proof**:
- Let **Π** be a Henkin Set and suppose that ***Y***[***t***] ∈ **Π** for every *PL*+ **singular term**, ***t***. Every *PL*+ name *is* a *PL*+ singular term. So, ***Y***[***s***] ∈ **Π** for every *PL*+ ***name***, ***s***. So, by ∀ **Lemma**, (∀*z*)***Y*** ∈ **Π**. Conversely, suppose that (∀*z*)***Y*** ∈ **Π**. Then, since any Henkin Set is maximal consistent, and, again, every such set is deductively closed (**Lemma 3.2.2a**), ***Y***[***t***] ∈ **Π**, for every *PL*+ **term**, ***t***, since this is deducible using the rule, Universal Instantiation.

Summary

- **Rehash**: We have shown that, given a consistent set of sentences, Γ, it can always be extended to a maximally consistent one that captures the truth-conditions of all sentences in the extended set's language.
- **Details**: We extended the language of Γ, *PL*, into *PL*+ by adding infinitely many new names. We then added a *W*-conditional for every universally quantified *PL*+ sentence and proved that the resulting set was still consistent. Finally, we used Lindenbaum's Lemma to extend the result into a maximal consistent set **Π**, called a Henkin Set.
- However, our aim (recall!) was to prove the **Model Existence Theorem**. How do we get from a **Henkin Set** to a model of Γ?

Henkin Models

- The key idea is to construct a model of our Henkin Set, called a **Henkin Model**, $\boldsymbol{H^{\Pi}}$, with the feature that $\boldsymbol{H^{\Pi}} \models \boldsymbol{Y}$ just in case $\boldsymbol{Y} \in \boldsymbol{\Pi}$. That is:
    - $\boldsymbol{Y}$ is true in $\boldsymbol{H^{\Pi}}$ when $\boldsymbol{Y} \in \boldsymbol{\Pi}$, and $\boldsymbol{Y}$ is false in $\boldsymbol{H^{\Pi}}$ when $\boldsymbol{Y} \notin \boldsymbol{\Pi}$.
- *Upshot*: $\boldsymbol{H^{\Pi}}$ will be a model of $\Gamma \subseteq \boldsymbol{\Pi}$, as desired.
- We may specify a **Henkin Model of Γ** (an interpretation of all of the members of **Γ** under which each comes out true) as follows.
- For simplicity, we first assume that *PL*+ lacks the identity predicate, =.

Henkin Model of Γ

- **Universe of Discourse** (*UD*): The set of all the *PL*+ singular terms
- **List of Names** (*LN*): All the members of *UD* (i.e., *LN* = *UD*).
- **Semantical Assignments** (*SA*):
- For every name, $\boldsymbol{s}$, in *LN*, $\boldsymbol{H}^{\Gamma}(\boldsymbol{s}) = \boldsymbol{s}$.
- For every *1*-place predicate, $\boldsymbol{P}$, in *Voc*(Γ), $\boldsymbol{H}^{\Gamma}(\boldsymbol{P}) = \{\boldsymbol{s} : \boldsymbol{Ps} \in \Gamma\}$
- For every *n*-place predicate, $\boldsymbol{R^n}$, in *Voc*(Γ), $\boldsymbol{H}^{\Gamma}(\boldsymbol{R^n}) = \{<\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3}, \ldots, \boldsymbol{t_n}> : \boldsymbol{R^n t_1 t_2 t_3 \ldots t_n} \in \Gamma\}$.
- For every *n*-place function symbol, $\boldsymbol{f^n}$, in *Voc*(Γ), $\{<\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3}, \ldots, \boldsymbol{t_n} \boldsymbol{f^n t_1 t_2 t_3 \ldots t_n}> : \boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3}, \ldots, \boldsymbol{t_n}$ are *PL*+ singular terms$\}$

Henkin Model of Γ

- *Note*: By construction, $\boldsymbol{H}^{\Gamma}$ is a model of all atomic members of Γ. In order to ensure that $\boldsymbol{H}^{\Gamma}$ is also a model of all complex members of Γ, we must expand Γ into a **Henkin Set**, as before, and trade $\boldsymbol{H}^{\Gamma}$ for $\boldsymbol{H}^{\Pi}$.
- **Truth-Membership Theorem**: Let Γ be a consistent set of sentences, and let $\mathbf{\Pi} \supseteq \Gamma$ be a Henkin Set. Then, for every *PL*+ sentence, $\boldsymbol{Y}$, $\boldsymbol{H}^{\Pi} \models \boldsymbol{Y}$ if/f $\boldsymbol{Y} \in \mathbf{\Pi}$.
- **Proof**: We will prove this by the Principle of Complete Induction (*PCI*) on the number of connective (and quantifier) occurrences in a given sentence.
- **Base Step**:
- 1) Let the complexity of $\boldsymbol{X}$ be $\boldsymbol{0}$. Then $\boldsymbol{X}$ is an atomic sentence of the form $\boldsymbol{R^n a_1, a_2, a_3, \ldots, a_n}$, where $\boldsymbol{R^n}$ is an *n*-place *PL* predicate ($n > 1$) and $\boldsymbol{t_1, t_2, \ldots t_n}$ are (perhaps not distinct) *PL* singular terms. (Remember that there is no identity predicate.)

Henkin Model of Γ

- 2) $\boldsymbol{R^n a_1, a_2, a_3, \ldots, a_n}$ is true on $\boldsymbol{H}^{\Pi}$ just in case $<(\boldsymbol{H}^{\Pi}(\boldsymbol{t_1}), \boldsymbol{H}^{\Pi}(\boldsymbol{t_2}), \ldots \boldsymbol{H}^{\Pi}(\boldsymbol{t_n}))> \in \boldsymbol{H}^{\Pi}(\boldsymbol{R^n})$.
- 3) By the definition of $\boldsymbol{H}^{\Pi}$, $<(\boldsymbol{H}^{\Pi}(\boldsymbol{t_1}), \boldsymbol{H}^{\Pi}(\boldsymbol{t_2}), \ldots \boldsymbol{H}^{\Pi}(\boldsymbol{t_n}))> \in \boldsymbol{H}^{\Pi}(\boldsymbol{R^n})$ just in case $<\boldsymbol{t_1, t_2, \ldots t_n}> \in \boldsymbol{H}^{\Pi}(\boldsymbol{R^n})$, since $\boldsymbol{H}^{\Pi}$ assigns terms to themselves. And $<\boldsymbol{t_1, t_2, \ldots t_n}> \in \boldsymbol{H}^{\Pi}(\boldsymbol{R^n})$ just in case $\boldsymbol{R^n t_1, t_2, \ldots t_n} \in \mathbf{\Pi}$, as desired.
- **Inductive Step**:
- Assume that the **Truth-Membership Theorem** is true for any sentence, $\boldsymbol{X}$, with complexity m ($0 \leq m < k$). We show that, under this assumption, it also holds for any sentence of complexity k as well.

Henkin Model of Γ

- 4) Since $k > 0$, ***X*** must be a complex sentence, i.e., a negation, conditional, or a universally quantified sentence. Therefore, to begin, let ***X*** be a **negation**, i.e., of the form ~***Y***, for some *PL+* sentence ***Y***.
- 5) Since ***Y*** has a complexity less than *k*, the Induction Hypothesis applies to ***Y***, meaning that $\boldsymbol{H^{\Pi}} \models \boldsymbol{Y}$ if/f $\boldsymbol{Y} \in \boldsymbol{\Pi}$.
- 6) But **Π** is Henkin set. So, it is maximal consistent. Thus, $\boldsymbol{Y} \in \boldsymbol{\Pi}$ if/f $\sim\boldsymbol{Y} \notin \boldsymbol{\Pi}$, and either $\boldsymbol{Y} \in \boldsymbol{\Pi}$ or $\sim\boldsymbol{Y} \in \boldsymbol{\Pi}$.
- 7) By 5) & 6), ***Y*** is not true, and so (by **bivalence**) false, on $\boldsymbol{H^{\Pi}}$ just in case $\sim\boldsymbol{Y} \in \boldsymbol{\Pi}$. That is, $\boldsymbol{H^{\Pi}} \models \boldsymbol{X}$ if/f $\boldsymbol{X} \in \boldsymbol{\Pi}$, as desired.

Henkin Model of Γ

- 8) So, let ***X*** be a **conditional**, i.e., of the form $(\boldsymbol{Y} \rightarrow \boldsymbol{Z})$, for some *PL+* sentences ***Y*** and ***Z***, where ***Y*** and ***Z*** have complexities $< k$.
- 9) Then, by the Inductive Hypothesis, $\boldsymbol{H^{\Pi}} \models \boldsymbol{Y}$ if/f $\boldsymbol{Y} \in \boldsymbol{\Pi}$, and $\boldsymbol{H^{\Pi}} \models \boldsymbol{Z}$ if/f $\boldsymbol{Z} \in \boldsymbol{\Pi}$.
- 10) Either $\boldsymbol{H^{\Pi}} \models \boldsymbol{X}$ or not. So, suppose first that $\boldsymbol{H^{\Pi}} \models \boldsymbol{X} = (\boldsymbol{Y} \rightarrow \boldsymbol{Z})$.
- 11) Then, by the truth-conditions for →, either $\boldsymbol{H^{\Pi}} \models \sim\boldsymbol{Y}$ or $\boldsymbol{H^{\Pi}} \models \boldsymbol{Z}$.
- 12) Since the Inductive Hypothesis applies to ***Y***. Then ***Y*** is false on $\boldsymbol{H^{\Pi}}$ just in case $\boldsymbol{H^{\Pi}} \models \sim\boldsymbol{Y}$ just in case $\boldsymbol{Y} \notin \boldsymbol{\Pi}$.
- 13) Since **Π** maximal, $\sim\boldsymbol{Y} \in \boldsymbol{\Pi}$, in which case $(\boldsymbol{Y} \rightarrow \boldsymbol{Z}) \in \boldsymbol{\Pi}$, since **Π** is deductively closed (**Lemma 3.2.2a**).

Henkin Model of Γ

- 14) So, suppose that $\boldsymbol{H^{\Pi}}$ |= ***Z***.
- 15) Since the <u>Inductive Hypothesis</u> applies to ***Z***, $\boldsymbol{H^{\Pi}}$ |= ***Z*** just in case ***Z*** ∈ **Π**. Since $\boldsymbol{H^{\Pi}}$ |= ***Z***, (***Y*** → ***Z***) ∈ **Π**, again by the <u>deductive closure</u> of **Π**.
- 16) Thus, from 10) – 15), if $\boldsymbol{H^{\Pi}}$ |= (***Y*** → ***Z***), then (***Y*** → ***Z***) ∈ **Π**.
- 17) Now suppose (***Y*** → ***Z***) is <u>false</u> on $\boldsymbol{H^{\Pi}}$, i.e., $\boldsymbol{H^{\Pi}}$ |= ~(***Y*** → ***Z***).
- 18) Then $\boldsymbol{H^{\Pi}}$ |= ***Y***, and ***Z*** is <u>false</u> on $\boldsymbol{H^{\Pi}}$, i.e., $\boldsymbol{H^{\Pi}}$ |= ~***Z***.
- 19) Since the <u>Inductive Hypothesis</u> applies to ***Y*** and ***Z***, ***Y*** ∈ **Π** and ***Z*** ∉ **Π**.
- 20) By the <u>maximality</u> of **Π**, ~***Z*** ∈ **Π**.

Henkin Model of Γ

- 21) As both ***Y*** ∈ **Π** and ~***Z*** ∈ **Π**, ~(***Y*** → ***Z***) ∈ **Π** by its <u>deductive closure</u>, and (***Y*** → ***Z***) ∉ **Π** by its <u>consistency</u>.
- 22) Thus, from 10) – 21), $\boldsymbol{H^{\Pi}}$ |= (***Y*** → ***Z***) just in case (***Y*** → ***Z***) ∈ **Π**.
- 23) Finally, let ***X*** be the **universally quantified** sentence, (∀***z***)***Y***, where the <u>inductive hypothesis</u> applies to ***Y***[***s***], for every name in *LN* – i.e., ***Y***[***s***] ∈ **Π** just in case $\boldsymbol{H^{\Pi}}$ |= ***Y***[***s***], for every *LN* name, ***s***.
- 24) Either $\boldsymbol{H^{\Pi}}$ |= (∀***z***)***Y*** or not. So, suppose first that $\boldsymbol{H^{\Pi}}$ |= (∀***z***)***Y***.
- 25) By the truth-conditions of the <u>universal quantifier</u>, $\boldsymbol{H^{\Pi}}$ |= ***Y***[***s***], for every *LN* name, ***s***.

Henkin Model of Γ

- 26) Since the <u>Inductive Hypothesis</u> applies to each such instance, ***Y***[***s***] ∈ **Π** for every ***s*** in *LN*.
- 27) But **Π** is a <u>Henkin Set</u>. So, (∀***z***)***Y*** ∈ **Π** just in case ***Y***[***t***] ∈ **Π** for every *PL+* **singular term**, ***t***.
- 28) Moreover, on the <u>Henkin Interpretation</u>, *LN* = *UD* = {*x* : *x* is a *PL+* <u>singular term</u>}.
- 29), So, by 26) – 28), (∀***z***)***Y*** ∈ **Π**, as desired.
- 30) What if (∀***z***)***Y*** is <u>false</u> on $\boldsymbol{H^{\Pi}}$? Then, by the truth-conditions for ∀, ***Y***[***s***] is false on $\boldsymbol{H^{\Pi}}$ for some name, ***s****, in *LN*.

Henkin Model of Γ

- 30) Since the <u>Inductive Hypothesis</u> applies to each instance, ***Y***[***s***], for every ***s*** in *LN*, ***Y***[***s****] ∉ **Π**.
- 31) By the <u>maximality</u> of **Π**, ~***Y***[***s****] ∈ **Π**, for every ***s**** such that ***Y***[***s****] ∉ **Π**.
- 32) So, by the <u>deductive closure</u> of **Π**, ~(∀***z***)***Y*** ∈ **Π**.
- 33) Finally, by the <u>consistency</u> of **Π**, (∀***z***)***Y*** ∉ **Π**, as desired.
- 34) By 4), 8) & 23), $\boldsymbol{H^{\Pi}}$ |= ***Y*** if/f ***Y*** ∈ **Π** (i.e., the **Truth-Membership Theorem** is true) for any sentence, ***Y***, with <u>complexity</u> *k* whenever it is true of any sentence, ***X***, with complexity *m*, where $0 \leq m < k$.
- 35) Hence, by *PCI*, the **Truth-Membership Theorem** is established.

Summary

- We have shown that a sentence of *PL+*, which includes the language of *PL*, is true in the **Henkin Model**, $\boldsymbol{H^{\Pi}}$, just in case it is a member of a **Henkin Set Π**. So, the Henkin Model is a model of a Henkin Set.
- But we also showed that every consistent set of sentences, Γ, can be extended to a Henkin Set, **Π**. Since every Henkin set has a model, and a model of a set is also a model of all its subsets, $\boldsymbol{Con(\Gamma) \rightarrow \exists M \models \Gamma}$.
- This is just the **Model Existence Theorem** which we proved was equivalent to the **Completeness Theorem**, i.e., that for every set Γ of *PL* sentences and every sentence ***X*** of *PL*, ***if*** **Γ |= *X*, *then*** **Γ |– *X***.
- We have, therefore, proved the **Completeness Theorem**.

Loose End: Identity

- We proved the Completeness Theorem for *PL* without an identity predicate. There is no difficulty with **logical truths**, like (***a*** = ***a***). Our assignment of names to themselves poses no problem in the case of such logical truths.
- The problem is with identity statements that are not logical truths, like ***a*** = ***b***. The Henkin Interpretation that we constructed assigns ***a*** and ***b*** to *themselves*. But ***a*** and ***b*** are distinct names! So, (***a*** = ***b***) is false on such an interpretation.
- A natural fix recommends itself: we should **partition** the set of *PL+* singular terms into **equivalence classes** of terms that the theory regards as 'equal'.
- *Recall*: A **partition** of a set, *A*, is a family, *F*, that is **exhaustive** -- i.e., such that $\cup F = A$ -- and such that all of its members are **disjoint** -- i.e., for all $A \in F$ and $B \in F$, $\{x : x \in A \;\&\; x \in B\} = \emptyset$. The last condition is written: $\cap \boldsymbol{F} = \boldsymbol{\emptyset}$.

Loose End: Identity

- So, let us **partition** the universe of discourse (*UD*) of our original *Henkin Interpretation* thus: $\boldsymbol{E}[\boldsymbol{t}] = \{\boldsymbol{r} : \boldsymbol{r}$ is a *PL+* singular term & $(\boldsymbol{t} = \boldsymbol{r}) \in \mathbf{\Pi^*}\}$.
- *Note*: We use * to designate the new theory, universe of discourse, and so on.
- Given our *partition*, we may collect the equivalence classes, $\boldsymbol{E}[\boldsymbol{t}]$ where $\boldsymbol{t}$ is a *PL+* singular term, to form: $\mathcal{U} = \{\boldsymbol{E}[\boldsymbol{t}] : \boldsymbol{t}$ is a *PL+* singular term$\}$.
- Why is this a **partition**? Each $\boldsymbol{E}[\boldsymbol{t}]$ is nonempty, since it includes $\boldsymbol{t}$. Now suppose that $\boldsymbol{E}[\boldsymbol{t}] \cap \boldsymbol{E}[\boldsymbol{r}] \neq \emptyset$. Then $\exists \boldsymbol{q}$ with $\boldsymbol{q} \in \boldsymbol{E}[\boldsymbol{t}]$ and $\boldsymbol{q} \in \boldsymbol{E}[\boldsymbol{r}]$. So, $\boldsymbol{q} = \boldsymbol{t} \in \mathbf{\Pi^*}$ and $\boldsymbol{q} = \boldsymbol{r} \in \mathbf{\Pi^*}$. By the deductive closure of Π^* $\boldsymbol{t} = \boldsymbol{r} \in \mathbf{\Pi^*}$ and $\boldsymbol{E}[\boldsymbol{t}] = \boldsymbol{E}[\boldsymbol{r}]$. Finally, for each $\boldsymbol{t}$, $\boldsymbol{E}[\boldsymbol{t}] \subseteq UD$, so $\cup\mathcal{U} \subseteq UD$. To show that $UD \subseteq \cup\mathcal{U}$, let $\boldsymbol{t}$ be any *PL+* singular term in *UD*. Then $\boldsymbol{t} = \boldsymbol{t} \in \mathbf{\Pi^*}$, so $\boldsymbol{t} \in \boldsymbol{E}[\boldsymbol{t}]$, and, thus, $\boldsymbol{t} \in \cup\mathcal{U}$.
- To get a model, $\boldsymbol{H}^{\boldsymbol{\Pi}^*}$, of any sentence in the Henkin Set, $\mathbf{\Pi^*}$, of a set, Γ^*, language of *PL* that includes the identity predicate, =, we modify $\boldsymbol{H}^{\boldsymbol{\Pi}}$ thus:

$\boldsymbol{H}^{\boldsymbol{\Pi}^*}$ Interpretation

- A) $UD^* = \mathcal{U} = \{\boldsymbol{E}[\boldsymbol{t}] : \boldsymbol{t}$ is a *PL+* singular term$\}$. (That is, we replace *UD* with the family, $\mathcal{U}$, i.e., the set of equivalence classes of *PL+* singular terms.)
- B) $LN^* = LN$ = The set of all the *PL+* singular terms.
- C) Assign every name, q, in *LN* to the $\boldsymbol{E}[\boldsymbol{t}]$ such that $q \in \boldsymbol{E}[\boldsymbol{t}]$.
- D) Assign every n-place predicate, $n \geq 1$, $\boldsymbol{R}^n$ to the set of n-tuples $\{<\boldsymbol{E}[\boldsymbol{t}_1], \boldsymbol{E}[\boldsymbol{t}_2], \boldsymbol{E}[\boldsymbol{t}_3], \ldots, \boldsymbol{E}[\boldsymbol{t}_n]> : \boldsymbol{R}^n\boldsymbol{t}_1\boldsymbol{t}_2\boldsymbol{t}_3\ldots\boldsymbol{t}_n \in \mathbf{\Pi^*}\}$
- E) Assign every n-place function symbol, $\boldsymbol{f}^n$, the set of $n+1$ tuples $\{<\boldsymbol{E}[\boldsymbol{t}_1], \boldsymbol{E}[\boldsymbol{t}_2], \boldsymbol{E}[\boldsymbol{t}_3], \ldots, \boldsymbol{E}[\boldsymbol{t}_n], \boldsymbol{E}[\boldsymbol{f}^n\boldsymbol{t}_1\, \boldsymbol{t}_2\boldsymbol{t}_3\ldots\boldsymbol{t}_n]> : \boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \ldots, \boldsymbol{t}_n$ are *PL+* singular terms$\}$.

Revised Proof

- Using our revised **Henkin Interpretation***, H^{Π^*}, we can obtain a model of any *PL* theory, Γ*, incorporating the equality symbol, **=**, where Γ* ⊆ **Π*** and **Π*** is a *PL*+ **Henkin Set*** with the equality symbol. The only change to the proof of the **Truth-Membership Theorem** that is required concerns the **Base Case**.
- 1) First, let X be of the form $r = t$ where r and t are any *PL*+ terms.
- 2) Suppose that $r = t \models H^{\Pi^*}$.
- 3) Then $E[r] = E[t]$, where $E[r] = \{q: q = r \in \Pi^*\}$ and $E[t] = \{q: q = t \in \Pi^*\}$.
- 4) So, $r = t \in \Pi^*$, as desired.
- 5) Conversely, let $r = t \in \Pi^*$.
- 6) Then $r \in E[t] = \{q: q = t \in \Pi^*\}$.
- 7) Since, $r \in E[r]$, $E[r] = E[t]$, by our partition argument.

Revised Proof

- 8) Second, let X be of the form $R^n t_1, t_2, t_3, \ldots, t_n$.
- 9) Suppose that $R^n t_1, t_2, t_3, \ldots, t_n$ is true on H^{Π^*}.
- 10) Then $<E[t_1], E[t_2], E[t_3], \ldots, E[t_n]> \in \{<E[t_1], E[t_2], E[t_3], \ldots, E[t_n]> : R^n t_1 t_2 t_3 \ldots t_n \in \Pi^*\}$.
- 11) Hence, $R^n t_1 t_2 t_3 \ldots t_n \in \Pi^*$.
- 12) Conversely, suppose that $R^n t_1 t_2 t_3 \ldots t_n \in \Pi^*$.
- 13) Then $E[t_1], E[t_2], E[t_3], \ldots, E[t_n]> \in \{<E[t_1], E[t_2], E[t_3], \ldots, E[t_n]> : R^n t_1 t_2 t_3 \ldots t_n \in \Pi^*\}$.
- 14) So, X, which is of the form $R^n t_1, t_2, t_3, \ldots, t_n$, is true on H^{Π^*}.

Compactness Theorem

- We have proved the **Truth-Membership Theorem***. So, given a consistent *PL* set in a language with or without the equality symbol, Γ* , we may expand it to a maximal consistent **Henkin Set***, **Π***, and specify a **Henkin Interpretation***, $H^{\Pi*}$, such that $H^{\Pi*} \models Y$ just in case $Y \in \Pi*$.
- Since the **Henkin Model*** is a model of a **Henkin Set***, along with all of its subsets, we have proved that every consistent set has a model, i.e., the **Model Existence Theorem** which (we proved) is equivalent to the **Completeness Theorem**. This has two important corollaries.
- **3.3.1 The Compactness Theorem**: For every *PL* set Γ (in a language with or without equality) and every *PL* sentence ***X***, if ***X*** is a logical consequence of Γ, then ***X*** is a logical consequence of a **finite subset** of Γ.

Compactness Theorem

- *Note*: We already know that if ***X*** is derivable from Γ, then ***X*** is a derivable of a **finite subset** of Γ. The **Completeness Theorem** is now telling us that a corresponding fact holds of (semantic) validity as well.
- **Proof**:
- 1) Suppose that Γ is any set of *PL* sentences and ***X*** is any *PL* sentence such that $\Gamma \models X$.
- 2) By the **Completeness Theorem**, $\Gamma \vdash X$.
- 3) Then there is a (finite) derivation, ***D***, of ***X*** from Γ and a finite set, Σ_D, containing all of the members of Γ that occur in ***D***.

Finite Satisfiability Theorem

- 4) So, $\boldsymbol{\Sigma_D}$ |- ***X***.
- 5) So, by the **Soundness Theorem**, $\boldsymbol{\Sigma_D}$ |= ***X***.
- *Note*: All that mattered for this proof was that the *PL* provability relation, |-, was Sound and Complete for the semantic consequence relation, |=. So, the same argument works for any other Sound and Complete formal system, such as propositional or modal logical ones.
- Another theorem that is equivalent to the **Compactness Theorem** is:
- **Finite Satisfiability Theorem**: If every finite subset of a *PL* set, Γ, has a model, then Γ itself has a model. We will call a set, Γ, with the property that every finite subset of Γ has a model, **finitely satisfiable**.

Compactness → Finite Satisfiability

- **Metatheorem**: The **Compactness Theorem** and the **Finite Satisfiability Theorem** are equivalent (in a classical metatheory).
- **Proof** (Compactness → Finite Satisfiability):
- 1) Suppose for *reductio* that Γ is finitely satisfiable, but not satisfiable.
- 2) Then, vacuously, for any *PL* sentence, ***X***, Γ |= ***X*** and Γ |= ~***X***.
- 3) By **Compactness**, these implications are witnessed by finite subsets of Γ, $\boldsymbol{\Sigma_X}$ and $\boldsymbol{\Sigma_{\sim X}}$, respectively. If we let $\boldsymbol{\Sigma} = \boldsymbol{\Sigma_X} \cup \boldsymbol{\Sigma_{\sim X}}$, then **Σ** is a finite subset of Γ such that **Σ** |= ***X*** and **Σ** |= ~***X***.
- 4) But, by 1), Γ is finitely satisfiable – i.e., ∃***M*** **|= Σ**.

Finite Satisfiability → Compactness

- 5) If ***M*** fails to interpret the vocabulary of ***X***, expand ***M*** to a model, ***M****, that is just like ***M*** but does interpret this vocabulary.
- 6) Then $\boldsymbol{M^*} \models \boldsymbol{\Sigma} \cup \{\boldsymbol{X}\}$ and $\boldsymbol{M^*} \models \boldsymbol{\Sigma} \cup \{\sim \boldsymbol{X}\}$.
- 7) But no model can satisfy a sentence and its negation.
- 8) So, if Γ is finitely satisfiable, then it must be satisfiable – i.e., the **Finite Satisfiability Theorem** is true.
- **Proof** (Finite Satisfiability → Compactness):
- 9) For the converse, suppose for *reductio* that $\Gamma \models \boldsymbol{X}$, but that $\boldsymbol{\Sigma_{fin}} \not\models \boldsymbol{X}$ for every finite subset $\boldsymbol{\Sigma_{fin}}$ of Γ.

Finite Satisfiability → Compactness

- 10) So, $\forall \boldsymbol{\Sigma_{fin}} \subseteq \Gamma, \exists \boldsymbol{M} \models \boldsymbol{\Sigma_{fin}} \cup \{\sim \boldsymbol{X}\}$.
- 11) Hence, for every finite subset of $\Gamma \cup \{\sim \boldsymbol{X}\}$, $\boldsymbol{\Sigma}_{\sim X \boldsymbol{fin}}$, $\exists \boldsymbol{M} \models \boldsymbol{\Sigma}_{\sim X \boldsymbol{fin}}$.
- 12) By the **Finite Satisfiability Theorem**, $\Gamma \cup \{\sim \boldsymbol{X}\}$ is satisfiable, i.e., $\exists \boldsymbol{M} \models \Gamma \cup \{\sim \boldsymbol{X}\}$
- 13) So, by the definition of logical consequence, $\Gamma \not\models \boldsymbol{X}$, contrary to 1).

Illustration: Undefinability of Finiteness

- The **Compactness / Finite Satisfiability Theorem** betray an expressive limitation of first-order logic. In particular, **finiteness** is not **definable**.
- *Note*: We will talk later about second-order logic, PL^2, for which this is not the case. But second-order logic does not solve the philosophical problem that is raised, that of explaining the determinacy of our concept of *finite*. PL^2 takes it for granted that *finite*, and even $P(\mathbb{N})$, is determinate.
- The identity predicate, =, understood as a **logical constant** allows us express many things about (finite) size. For example, we can say that there are exactly two things as follows: $(\exists x)(\exists y)[x \neq y \;\&\; \forall z(z = x \vee z = y)]$.

Undefinability of Finiteness

- However, while we can force the Universe of Discourse (*UD*) to be finite, *and* we can force it to be infinite, we cannot concoct a sentence that is true in all and only finite (or infinite) models. In other words, (in)finiteness is (first-order) **undefinable** – *inexpressible* or *ineffable*.
- This follows from the **Compactness** / **Finite Satisfiability Theorem**:
- A) Suppose that Φ is a sentence true in every model with a finite *UD*.
- B) Then each finite subset of the following set, Γ, of sentences is consistent because it has a model (this follows from **Soundness**).

Undefinability of Finiteness

- **0**. Φ
- **1**. ~(∃*x*)[∀*z*(*z* = *x*)] ['It's not the case that there is at most one thing.']
- **2**. ~(∃*x*)(∃*y*)[(∀*z*(*z* = *x* v *z* = *y*)] ['It's not the case that there are at most two things.']
- …
- ***n***. ~(∃*x*)(∃*y*)…v *z* = *y*)] ['It's not the case that there are at most *n* things.']
- …
- Since Φ is true in every model with a finite *UD*, in order to generate a model of sentences 0. – *n*. we just require that *UD* have *n+1* elements.

Undefinability of Finiteness

- C) Since each finite subset of Γ is consistent, Γ itself must be consistent, by the **Compactness** / **Finite Satisfiability Theorem**.
- D) But, then, every sentence, Φ, that is true is all finite models fails to rule out that there are more than *n* things, for every natural number, *n*!
- E) Hence, every sentence, Φ, that is true is all finite models must be true in some infinite models as well.
- F) So, finiteness is not (first-order) definable.
- *Observation*: If there were a sentence, Φ*, true in all and only the infinite models, then ~Φ* would be true in all and only the finite models. Since there is no such ~Φ*, infinitude is not definable either!

Elementary Equivalence & Isomorphism

- We now introduce a few additional important concepts going forward.
- **Elementary Equivalence**: If Γ is a set of *PL* sentences, and I_Γ and J_Γ are two interpretations of Γ, then I_Γ and J_Γ are **elementary equivalent** with respect to *Voc*(Γ) if/f for every *PL* sentence, ***X***, whose vocabulary is limited to that of *Voc*(Γ): $\boldsymbol{I} \models \boldsymbol{X} \leftrightarrow \boldsymbol{J} \models \boldsymbol{X}$. (Recall that a *model* of ***X*** is merely an *interpretation*, like *I* or *J*, under which ***X*** is true.)
- When $\boldsymbol{I}_\Gamma$ and $\boldsymbol{J}_\Gamma$ are **elementary equivalent**, we will write: $\boldsymbol{I}_\Gamma =_\Gamma \boldsymbol{J}_\Gamma$.
- **Isomorphism**: A function *h* is an **isomorphism** between $\boldsymbol{I}_\Gamma$ and $\boldsymbol{J}_\Gamma$ if/f *h* is a bijection between UD_I and UD_J such that the following holds:

Isomorphism

- (1) For every name ***c*** in *Voc*(Γ), $h(I_\Gamma(\boldsymbol{c})) = J_\Gamma(\boldsymbol{c})$.
- (2) For each *1*-place predicate $\boldsymbol{P^1}$ in *Voc*(Γ), and for each individual, β, in UD_I, $\beta \in I_\Gamma(\boldsymbol{P^1})$ iff $h(\beta) \in J_\Gamma(\boldsymbol{P^1})$.
- (3) For every *n*-place predicate $\boldsymbol{P^n}$ in *Voc*(Γ), where $n > 1$, and for each *n*-tuple $<\beta_1, \beta_2, \beta_3, \ldots, \beta_n>$ of individuals in UD_I, $<\beta_1, \beta_2, \beta_3, \ldots, \beta_n> \in I_\Gamma(\boldsymbol{P^n})$ iff $<h(\beta_1), h(\beta_2), h(\beta_3), \ldots, h(\beta_n)> \in J_\Gamma(\boldsymbol{P^n})$.
- (4) For every *n*-place function symbol $\boldsymbol{g^n}$ in *Voc*(Γ), and for each *n*-tuple $<\beta_1, \beta_2, \beta_3, \ldots, \beta_n>$ of individuals in UD_I, $h(I_\Gamma(\boldsymbol{g^n})(\beta_1, \beta_2, \beta_3, \ldots, \beta_n)) = J_\Gamma(\boldsymbol{g^n})(h(\beta_1), h(\beta_2), h(\beta_3), \ldots, h(\beta_n))$.

Isomorphism

- Two interpretations, I_Γ and J_Γ, of *Voc*(Γ) are **isomorphic** when there is an **isomorphism** between them. We write $\boldsymbol{I_\Gamma \cong_\Gamma J_\Gamma}$ or just $\boldsymbol{I \cong_\Gamma J}$.
- **Isomorphism** is strictly stronger than **elementary equivalence**. We will see that *there can be elementary equivalent interpretations that are not isomorphic. But there cannot be isomorphic interpretations that are not elementary equivalent*. Isomorphic interpretations do not merely make the same sentences true. If $\boldsymbol{I_\Gamma \cong_\Gamma J_\Gamma}$, then I_Γ and J_Γ are 'structurally identical', at least in the way that they interpret *Voc*(Γ).
- *Note*: While constituents of *I* and *J* that interpret Γ are *structurally identical*, those interpretations need not be so identical in general!

Properties of *PL* Sets

- In light of the **Soundness** and **Completeness** theorems, deductive and logical closure come to the same thing. Therefore, we can say:
- A *PL* set, **Σ**, is (semantically or syntactically) **complete** if/f for every sentence, ***X***, such that *Voc*(***X***) ⊆ *Voc*(**Σ**), either **Σ |- *X*** or **Σ |- ~*X***.
- *Note*: The **Completeness Theorem** concerns a different kind of completeness, the relationship between provability and implication.
- A *PL* set **Σ** with the following feature is called a PL **theory**:
- For every sentence, ***X***, such that *Voc*(***X***) ⊆ *Voc*(**Σ**), *if* **Σ |= *X***, *then* **Σ ∈ *X***.
- *Note*: This is just semantic closure *limited to the vocabulary of **Σ***.

Corollaries 3.5.1a - 3.5.1d

- The following corollaries are immediate consequences of the **Soundness** and **Completeness** theorems, in tandem with the definition of a **theory**:
- **Corollary 3.5.1a**: For every *PL* set **Σ**, **Σ** is a **theory** if/f for every *PL* sentence ***X*** such that *Voc*(***X***) ⊆ *Voc*(**Σ**), if **Σ** |– ***X***, then **Σ** ∈ ***X***.
- **Corollary 3.5.1b**: For every sentence, ***X***, such that *Voc*(***X***) ⊆ *Voc*(**Σ**), either **Σ** |= ***X*** or **Σ** |= ~***X***.
- **Corollary 3.5.1c**: For every *PL* set **Σ**, the set of all the logical consequences of **Σ**, ***X***, such that *Voc*(***X***) ⊆ *Voc*(**Σ**) is a **theory**, written ***Th***(**Σ**).
- **Corollary 3.5.1d**: For every *PL* set **Σ**, *Th*(**Σ**) = {***X***: *Voc*(***X***) ⊆ *Voc*(**Σ**) & **Σ** |- ***X***}.

Corollary 3.5.1e

- **Corollary 3.5.1e**: If *Voc*(**Σ**) is a *PL* vocabulary containing the logical vocabulary of *PL* and some (maybe not all) of the non-logical vocabulary interpreted by interpretation *J*, then the following set is *consistent and complete*:
  - $\boldsymbol{Th_{\Sigma}(J)}$ = {***X***: ***X*** is a sentence in *Voc*(**Σ**) that is true on *J*} = the set all the sentences composed of *Voc*(**Σ**) that are true on interpretation, *J*.
- *Note*: We simply write ***Th*(*J*)** if *Voc*(**Σ**) is the all of *Voc*(*J*), i.e., the logical vocabulary of *PL* plus all the extra-logical vocabulary interpreted by *J*.
- **Proof**:
- 1) Let *J* be a *PL* interpretation and *Voc*(**Σ**) and $Th_{\Sigma}(J)$ as above.

Corollary 3.5.1e

- 2) Then $Th_{\Sigma}(J)$ is a *PL* **theory** because models are closed under logical consequence.
  - Suppose that ***X*** is sentence in the language of *Voc*(**Σ**) and $Th_{\Sigma}(J)$ |= ***X***.
  - Since *J* is a model of $Th_{\Sigma}(J)$, ***X*** is true on *J* (by the definition of logical implication).
  - Hence, by the definition of $Th_{\Sigma}(J)$, ***X*** ∈ $Th_{\Sigma}(J)$, as desired.
- 3) $Th_{\Sigma}(J)$ is **complete** because models are **bivalent**.
  - Either ***X*** or ~***X*** is true on *J*.
  - Since $Th_{\Sigma}(J)$ is a theory, either ***X*** ∈ $Th_{\Sigma}(J)$ or ~***X*** ∈ $Th_{\Sigma}(J)$.
  - So, $Th_{\Sigma}(J)$ |- ***X*** or $Th_{\Sigma}(J)$ |- ~***X***, by Reiteration.
- 4) Finally, $Th_{\Sigma}(J)$ is consistent because it is (by definition) satisfiable. (This is just an application of the **Soundness Theorem**.)

Corollary 3.5.1f

- **3.5.1f** Every deductively or semantically closed *PL* set is a *PL* **theory**. (The converse is *not* true.)
- **Proof**: If a *PL* set, **Σ**, is semantically closed, then it must contain all of its logical consequences, including the ones expressed in *Voc*(**Σ**). So, it is a *PL* **theory**. By Completeness, the same is true of deductive closure.

Axiomatic Theories

- We mentioned that one motive for formal logic was to supply an ultimate court of appeals for questions about the validity of a proof.
- We prove things in mathematics. But we also prove things in physics, economics, computer science, linguistics, and ordinary philosophy.
- How do we decide whether an alleged theorem of, say, number theory is a **theorem** in fact? In practice, we ask other number theorists to check! But what if there is recalcitrant dispute among them (as there is with the *ABC Conjecture*)? Then we **translate** the (informal) proof into a sequence of *PL* sentences (in the vocabulary of number theory) and check that every line of the result is either an **axiom** or follows from the previous lines by a rule of *MDS* (equivalently: *GDS* or *MDS*).

Axiomatic Theories

- *Note*: By the **Completeness Theorem**, this is the same as checking that every line is either an axiom or provable from the previous lines.
- What counts as an axiom of a mathematical, physical, or other theory, **Σ**? Not all the members of **Σ**. It is trivial that every member of a theory, **Σ**, follows from itself! Maybe any **proper subset** of **Σ** from which all other members of **Σ** follow? This is too lax. Maybe any **finite subset** of **Σ** from which all other members of **Σ** follow? This is too demanding. Few axioms systems of interest, including those of arithmetic and set theory, are finite when regimented in (first-order) *PL*.

Axiomatic Theories

- A set of sentences in *Voc*(**Σ**), Γ, counts as a set of axioms for the set, **Σ**, when it is a **subset** of **Σ** from which all other members of **Σ** follow *and* there is an **effective decision procedure** to check membership in Γ – an **algorithm** which delivers the verdict **'yes'** or **'no'** after a **finite number of deterministic steps** depending on whether the string **is** or **is not** in Γ.
- *Example*: Although *ZFC* has infinitely-many axioms (e.g., one for each formula instance of the **Subsets Scheme**), there *is* an effective decision procedure to check whether a string of *PL* symbols is a *ZFC* axiom.
- *Note*: This procedure need not be 'feasible'! Many effective decision procedures are not. Measures of feasibility are studied in **computational complexity theory**, where questions like the famous '*P = NP?*' arise.

Axiomatic Theories

- We mentioned at the start of the semester that **effective decision procedures** are different from (mere) effective **'yes'** or **'no'** procedures.
  - Any **effective** procedure is, by definition, **mechanical** in a sense that we will make precise in two ultimately extensionally equivalent ways.
- However, an effective 'yes' procedure merely promises confirmation of membership in a set. It may fail to confirm lack of membership!
- *Example*: Let us write *Th*(Γ) for the set {***X*** : Γ |= ***X***}. Then *Th*(∅) = {***X*** : ∅ |= ***X***} = {*P* : ∅ |- ***X***} is just the set of of *PL* validities, i.e., logical truths.
- The **Church-Turning Theorem**, which we will prove later, says that, while there is an effective 'yes' procedure for testing membership in *Th*(∅), there is **no effective decision procedure** for testing this.

Axiomatic Theories

- Sets membership in which admit of effective decision procedures are called **decidable** or **recursive**. Concepts are labeled analogously.
- Sets (concepts) membership in (application of) which merely admit of effective 'yes' procedures are called **semidecidable** or **recursively enumerable**. The latter phrase is apt because there is an effective procedure for listing the members of any semidecidable set.
- *Observation**: *Th*(Γ) is semidecidable whenever membership in Γ is decidable, since membership in *Th*(∅) is semidecidable. Therefore, we can **effectively enumerate** all the theorems of any axiomatized theory.

Axiomatic Theories

- Summing Up:
- **3.5.1a** For any *PL* theory **Σ** with *Voc*(**Σ**), Γ is a **set of axioms** for **Σ** just in case Γ is a **decidable** set of sentences in *Voc*(**Σ**) and **Σ** = *Th*(Γ) ={***X*** : ***X*** is in *Voc*(Σ) and Γ |= ***X***} ={***X*** : ***X*** is in *Voc*(**Σ**) and Γ |– ***X***}.
- **3.5.1b** A *PL* theory is **axiomatizable** just when it has a set of axioms.
- **3.5.1c** A *PL* theory is **finitely axiomatizable** just in case it has a finite set of axioms.
- Given these definitions, we have the following theorem:

Completeness & Decidability

- **Theorem 3.5.1**: Every complete axiomatizable *PL* **theory** is decidable.
- **Proof**: Let **Σ** be a complete axiomatizable *PL* theory.
- 1) By **Corollary 3.5.1a** (for every *PL* set **Σ**, **Σ** is a **theory** if/f for every *PL* sentence ***X*** such that *Voc*(***X***) ⊆ *Voc*(**Σ**), if **Σ** |– ***X***, then **Σ** ∈ ***X***), Σ |- ***X*** if/f Σ ∈ ***X***.
- 2) Either **Σ** is consistent or not.
- 3) If **Σ** is not consistent, then **Σ** is decidable simply because **Σ** is the set of all *PL* sentences, and the set of all *PL* sentences is decidable.
- 4) So, suppose that **Σ** is consistent. Then *if* **Σ** |- ~***X***, *then* ***X*** ∉ Σ and vice versa (since, in that case, **Σ** ∈ ~***X***, by 1).

Completeness & Decidability

- 5) But **Σ** is also complete. So, **Σ** |- ***X*** or **Σ** |- ~***X*** for every *PL* sentence in *Voc*(**Σ**).
- 6) Moreover, **Σ** is axiomatizable. So, there is a decidable set of axioms, T, such that **Σ** = *Th*(Γ) ={***X*** : ***X*** is in *Voc*(**Σ**) and Γ |= ***X***} ={***X*** : ***X*** is in *Voc*(**Σ**) and Γ |– ***X***}.
- 7) Hence, there is an effective procedure for enumerating the derivations from T, and, hence, the members of **Σ**, by *Observation**.
- 8) This gives the following decision procedure for checking membership in **Σ** :

Completeness & Decidability

- It is decidable which *PL* sentences are in *Voc*(**Σ**). So, first check whether ***X*** is in *Voc*(**Σ**). If it is not, conclude that ***X*** does not belong to **Σ**.
- If ***X*** is in *Voc*(**Σ**), check whether ***X*** is the last line (i.e., the conclusion) of the first derivation in our enumeration of derivations from T.
- If ***X*** is the last line of the first derivation, conclude that that ***X*** ∈ **Σ**.
- If ~***X*** is the last line instead, conclude that ***X*** ∉ **Σ**.
- Continue in this way. After a finite number of steps, we must see ***X*** or ~***X*** as the last line of a derivation, since **Σ** is complete. And since **Σ** is consistent, we know that if we see ~***X***, then ***X*** ∉ **Σ** and vice versa.

Axiomatizability, Completeness, Decidability

- **Theorem 3.5.1** illustrates an important fact to which we will return:
- If a theory is axiomatizable and undecidable, then it is incomplete.
- If a theory is complete and undecidable, then it is not axiomatizable.
- Upshot: The triad of axiomatizability, completeness and *un*decidability is inconsistent.

Completeness & Elementary Equivalence

- Before clarifying what we mean by 'mechanical' (and 'effective'), we conclude with some concepts from **model theory**. First, a theorem:
- **Theorem 3.5.2**: A set of *PL* sentence in *Voc*(**Σ**) is **complete** just in case all of its models are **elementary equivalent** with respect to *Voc*(**Σ**).
- **Proof**:
- (Completeness → Elementary Equivalence)
- 1) Let **Σ** be a **complete** *PL* set in *Voc*(**Σ**), let $\boldsymbol{I}_{\Sigma}$ and $\boldsymbol{J}_{\Sigma}$ be models of **Σ**, and assume that *Con*(**Σ**) (if ~*Con*(**Σ**), then **Σ** has no models).
- Consider a sentence, ***X***, in *Voc*(**Σ**) that is true in $\boldsymbol{I}_{\Sigma}$.

Completeness & Elementary Equivalence

- 2) As **Σ** is complete, either **Σ** |- ***X*** or **Σ** |- ~***X***, and, hence, by **Soundness**, either **Σ** |= **X** or **Σ** |= ~**X.**
- 3) But it cannot be that **Σ** |= ~***X***, since then ~***X*** would be true on $\boldsymbol{I}_{\Sigma}$ (since it would true on all models of **Σ**), contradicting the assumption that ***X*** is true on $\boldsymbol{I}_{\Sigma}$**.**
- 4) So, it must be that **Σ** |= **X**.
- 5) So, **X** is true on every model in which **Σ** is true, including $\boldsymbol{J}_{\Sigma}$.
- 6) Since $\boldsymbol{I}_{\Sigma}$ and $\boldsymbol{J}_{\Sigma}$ are arbitrary and the reasoning is symmetric, $\boldsymbol{I}_{\Sigma} =_{\Sigma} \boldsymbol{J}_{\Sigma}$ – i.e., all models of Σ are **elementary equivalent** with respect to *Voc*(**Σ**).

Completeness & Elementary Equivalence

- (Elementary Equivalence → Completeness):
- 7) Now suppose that all models of **Σ** are **elementary equivalent**, and let ***X*** be any sentence in *Voc*(**Σ**).
- 8) Since models are <u>bivalent</u>, either *M* |= ***X*** or *M* |= ~***X***, and not both, for any model, *M*.
- 9) So, suppose that *M* |= ***X***. Then for <u>every</u> other model of **Σ**, *N*, *N* |= ***X***, since all models of **Σ** are <u>elementary equivalent</u>.
- 10) Hence, by the definition of <u>logical consequence</u>, if *M* |= ***X***, then **Σ** |= ***X***.
- 11) A symmetric argument holds in the case that *M* |= ~***X***.
- 12) Hence, by the <u>Completeness Theorem</u>, if all models of **Σ** are **elementary equivalent**, then **Σ** is **complete** (i.e., Σ |- ***X*** or Σ |- ~***X***), as desired.

Completeness & Elementary Equivalence

- Since <u>isomorphism</u> implies <u>elementary equivalence</u>, and elementary equivalence implies <u>completeness</u> -- and *vice versa* -- isomorphism implies completeness. However, completeness does <u>not</u> imply isomorphism!
- A set of *PL* sentences, **Σ**, all of whose models are **isomorphic** with respect to *Voc*(**Σ**) is very special. We call such a set **categorical**. We also say:
  - For any cardinal, κ, the *PL* set, **Σ**, is **κ-categorical** if/f all its models <u>whose cardinality is κ</u> are isomorphic with respect to *Voc*(**Σ**).
- A <u>categorical set</u>, **Σ**, that is consistent **defines**, or '**captures**', a certain **structure**, relative to *Voc*(**Σ**), since, intuitively, any of its models can be obtained from any of the others by simply 'relabeling' the elements.
- Unfortunately, categoricity is **very** hard to come by, as we will discover.

The Löwenheim-Skolem Theorem

- We are now in position to state a fundamental result of (first-order) logic concerning the sizes of models of a *PL* sets in a countable language.
- **The (downward) Löwenheim-Skolem Theorem**: If **Σ** is a *PL* set with a model, and the language of *PL* is countably infinite, then **Σ** has a countable model.
- **Proof**: Every set with a model is consistent (by the **Soundness Theorem**), and every consistent set of sentences has a countable model – namely, a **Henkin model** – by our proof of the Completeness Theorem.
- So, every set with a model has a countable model.
- **Skolem's Paradox**: Some sets of *PL* sentences (e.g., the *ZFC* axioms) imply the existence of uncountable sets. By the **Löwenheim-Skolem Theorem**, these sets must have countable models. Hence, some countable models contain uncountably-many things. But this is a contradiction!

Skolem's Paradox

- What is wrong with the argument of **Skolem's Paradox**?
- It confuses the perspective of the model with the perspective of our metatheory.
- Think about what it means for 'the real numbers are uncountable' to be true in a model, *M*. It means that there is no bijection in *M* between *M*'s natural numbers and *M*'s real numbers. But this is consistent with *M*'s being **countable**. Just because *M* is countable does not mean that it contains a bijection between its natural numbers and its real numbers. It could be that, in our metatheory, we have access to bijections that inhabitants of *M* do not! We understand a countable model, *M*, in which 'the real numbers are uncountable' is true to be missing some functions between *M*'s natural numbers and *M*'s reals.

Skolem's Paradox

- Although Skolem's Paradox is not literally a paradox, it does make salient an additional (quite dramatic!) expressive limitation of (first-order) logic.
- We already observed that 'finite' (correlatively, 'infinite') is not (first-order) **definable**. There is no sentence in the (first-order) language of *PL* whose models are all and only the finite (infinite) ones.
- We now see that the situation with 'uncountable' ('has cardinality $\aleph_2$', etc.) is much worse. There is not even a sentence, or a set of them (even if it is not recursive or not recursively enumerable!) that is true in only uncountable (or of cardinality $\aleph_2$, etc.) models. Set aside the 'all'!
- **Remaining Philosophical Problem**: How could we determinately mean finite by 'finite' -- *a fortiori* uncountable by 'uncountable' -- given that there is nothing that we can say in a formal language than pins this down?

Symbolic Logic

Computability

Effective Procedures

- We have been discussing **effective** procedures, which operate 'mechanically'. But what do these words really mean?
- The problem is one of **conceptual analysis**, which is a characteristic activity of philosophers since antiquity. Plato's dialogues famously analyze normative concepts like *goodness*, *knowledge*, and *justice*.
- The general method of conceptual analysis is to propose a **definition** and then look for counterexamples. A correct analysis takes the form:
  - For any object, *a,* ***a is C*** just in case ***a is F***, for some independently specifiable property, *F*. However, we also require that '*C*' has the **same meaning** as '*F*' (where *C* is the *analysandum*, the concept to be analyzed).

Effective Procedures

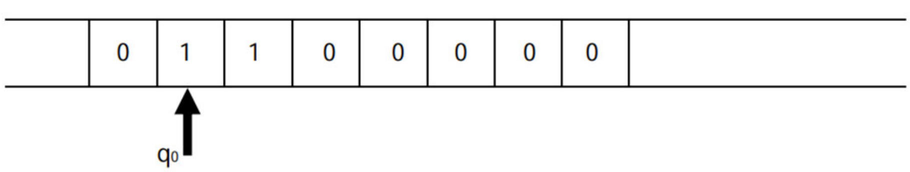
- *Example*: Plato proposed the following analysis of the concept of knowledge: a belief is *knowledge* just in case it is justified and true.
- Amazingly, three philosopher-logicians, Gödel, Turing, and Church, proposed superficially very different analyses of the concept of an *effective procedure*, and they all turned out to be **extensionally equivalent**!
- If they are all correct analyses, and sameness of meaning is **transitive**, then they must also mean the same thing – which is doubtful.
- Even if they are not all correct as conceptual analyses, however, they agree on what procedures count as effective. Hence, for typical mathematical purposes, they are equivalent. We will focus on Gödel's and Turing's.

Effective Procedures

- Both Gödel's and Turing's analyses treat procedures as **partial mathematical functions**, variously called **computable** or **recursive**.
- Moreover, strictly speaking, such functions just act on **natural numbers**. They take *n*-tuples of numbers (which we will see can themselves be coded, if need be, as single natural numbers) and output at most one natural number.
- This is no real limitation since it turns out that every non-number can be coded as a number, an idea that we will illustrate with Gödel's Theorems.
- There is no known function that is intuitively computable but *not* computable according to Gödel *et al.*'s criteria. It is more doubtful that every Turing computable function is intuitively computable – as with Plato's analysis of knowledge.
- The philosophical claim that *all and only* intuitively computable functions are Church-Gödel-Turing computable is the **Church-Turning Thesis.**

Turing Machines

- The first definition of computable functions, $f: \mathbb{N}^n \rightarrow \mathbb{N}$, is Turing's. It is the most natural, and has the strongest claim to giving the correct conceptual analysis of *effective procedure*, as even Gödel conceded.
- It makes use of a theoretical device, called a **Turing Machine**, which is stipulated to have unlimited memory, hardware material, and time.
- It represents inputs and outputs (*n*-tuples of numbers and individual numbers, respectively) as tallies on an infinite tape. Here is a picture:



Turing Machines

- A **Turing Machine**, $\boldsymbol{T_M}$, then consists of the following components:
- (1) an infinite tape that is divided into identical squares. Each of these squares contains the numeral *0* or a tally, i.e., the numeral, *1*.
- (2) a pointer that points at one square at a time.
  - This pointer can do any of the following (but nothing else):
    - read the numeral on the square, erase what is on it, write *0* or *1*, move one square to the left, or move one square to the right.
- (3) a register that keeps track of the internal states of the machine.
- (4) a set of instructions that represents the program of the machine.

Turing Machines

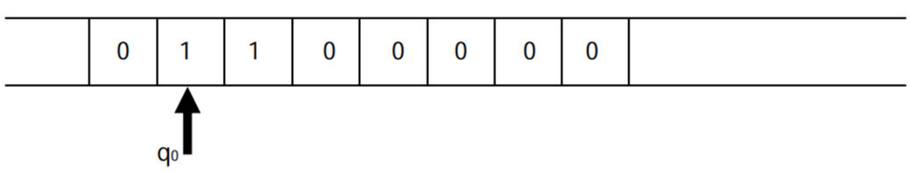
- A Turing Machine, T_M, is uniquely defined, and so often simply *identified* with, its program – i.e., by its set of instructions.
- The set of instructions for a *Turning Machine* consists of an even number of instruction lines, $\{l_1, l_2, l_3, l_4, \ldots l_m\}$, each of the form $\boldsymbol{q_iXYq_k}$.
- If $\boldsymbol{l_s}$ is $\boldsymbol{q_iXYq_k}$, then:
- $\boldsymbol{X}$ is the **input** of line $\boldsymbol{l_s}$, and is either ***0*** or ***1***
- $\boldsymbol{Y}$ is **output** of line $\boldsymbol{l_s}$, and is either ***0***, ***1***, ***R*** (for 'right'), or ***L*** (for 'left')
- $\boldsymbol{q_i}$ is the **initial state** of $\boldsymbol{l_s}$
- $\boldsymbol{q_k}$ is the **terminal state** of $\boldsymbol{l_s}$

Turing Machines

- If T_M is the *Turing Machine* with the set of instruction lines, $\{l_1, l_2, l_3, l_4, \ldots l_m\}$, then ***$T_M$*'s initial state** (as opposed to the initial state of a line) is just the initial state of the first instruction line, written: $\boldsymbol{q_0}$.
- Similarly, ***T_M*'s terminal state** is the terminal state of at least one of the instruction lines (not necessarily the last in the list of instructions), but it cannot be the initial state of any of them. This state is written: $\boldsymbol{q_e}$.
- The **set of internal states** of T_M, $\{q_0, q_1, q_2, \ldots q_e\}$, is then the set of all initial and terminal states of the instruction lines: $\{l_1, l_2, l_3, l_4, \ldots l_m\}$.
- *Note*: No two *instruction lines* share the same first two symbols.

Turing Machines

- Any **Turing Machine**, T_M, *begins* in internal state, q_0, with its pointer at the leftmost square with a *1* in it if its first input is *1*, as illustrated.
- If its first input is instead *0*, then its pointer is at some square with a *0*.
- If T_M **halts**, it always halts in the internal state $\boldsymbol{q_e}$, with its pointer at the leftmost square with a *1*, or at a square with a *0* in it if the output is *0*.
- However, T_M may **not halt**. If it does not, then its output is undefined.



Four Kinds of Instruction

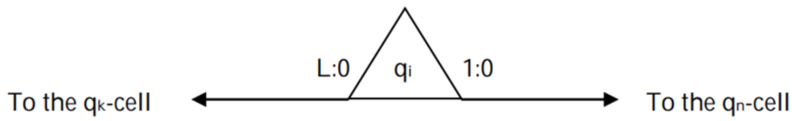
- There are **four kinds of instruction** that a Turing Machine can follow:
- q_iX0q_k : If T_M is in state q_i and the pointer reads X (where X is either *1* or *0*), then the pointer writes *0* and T_M enters state q_k.
- q_iX1q_k : If T_M is in state q_i and the pointer reads X, then the pointer writes *1* and T_M enters state q_k.
- q_iXRq_k : If T_M is in state q_i and the pointer reads X, then the pointer moves one square to the right and T_M enters state q_k.
- q_iXLq_k : If T_M is in state q_i and the pointer reads X, then the pointer moves one square to the left and T_M enters state q_k.

Diagram Cell

- Turing Machines can be represented by **diagrams** consisting of a number of **cells**. With the exception of the q_e cell, every **diagram cell** is a **triangle** with two instruction lines exiting that begin with the same initial state.
- The inside of the triangle specifies the **initial state** of the two lines.
- Existing arrows represent the inputs, outputs, and terminal states of one of the two instruction lines. These arrows, in turn, connect to other cells.
- *Note*: The instructions of the **left-hand arrows** are arranged in the **reverse order** of the instructions of the lines that they represent.

Diagram Cell

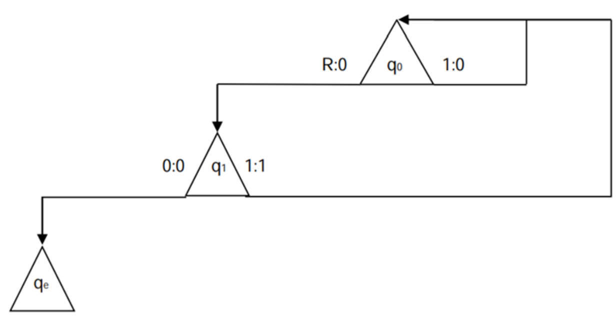
- Here is an example of a *diagram cell*:



- If l_S is the instruction line, $q_i 0 L q_k$, and l_T is the instruction line, $q_i 1 0 q_n$, then the above *diagram cell* represents the following two commands:
- **Right-Hand Arrow**: If the internal state is q_i and the pointer reads *1*, write *0* in place of *1* and enter state q_n.
- **Left-Hand Arrow**: If the internal state is q_i and the pointer reads *0*, move the pointer one square to the left and enter state q_k.

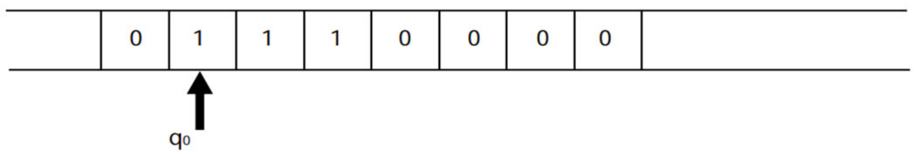
Example: The Zero Function

- The **zero function**, $\forall n, f(n) = 0: \mathbb{N} \rightarrow \mathbb{N}$, is intuitively computable. So, on the strength of the **Church-Turing Thesis**, it must be **Turing Computable**. What is a *Turing Machine*, T_0, that computes it?
- Here is one:
  - $l_1 : q_0 1 0 q_0$
  - $l_2 : q_0 0 R q_1$
  - $l_3 : q_1 1 1 q_0$
  - $l_4 : q_1 0 0 q_e$



Example: The Zero Function

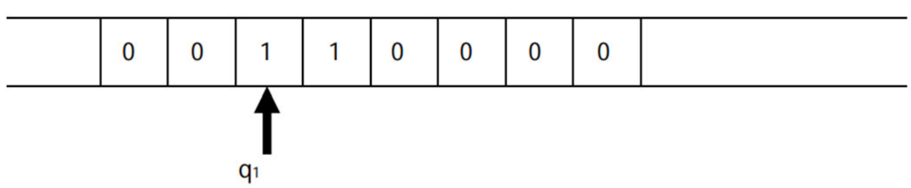
- Let us illustrate $\boldsymbol{T_0}$'s action on the input, *3*.
- a) The machine is in the state $\boldsymbol{q_0}$. There are three consecutive squares (encoding the input, the number *3*) containing a stroke, *1*. The rest have *0*'s in them. The pointer is positioned at the leftmost square that contains *1*.



- The state above is $\boldsymbol{q_0}$ and the input is *1*. So, the right-hand arrow of the **$\boldsymbol{q_0}$-cell** applies. It tells the machine to replace *1* with *0* and stay at state $\boldsymbol{q_0}$.

Example: The Zero Function

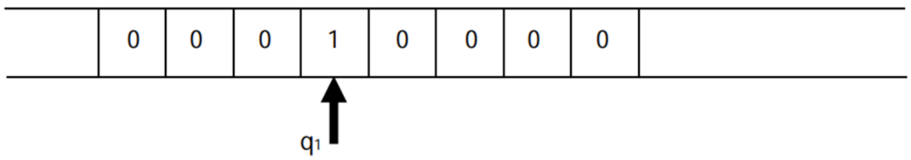
- b) Now the state is $\boldsymbol{q_0}$ and the input is *0*. Hence, the left-hand arrow of the **$\boldsymbol{q_0}$-cell** tells $\boldsymbol{T_0}$ to move one square to the right and enter state $\boldsymbol{q_1}$.



- c) After executing b), the state is $\boldsymbol{q_1}$ and the input is *1*. Consequently, the applicable arrow is the right-hand arrow of the **$\boldsymbol{q_1}$-cell**. This arrows tells $\boldsymbol{T_0}$ to leave the tally, *1*, that it reads as it is but change the state back to $\boldsymbol{q_0}$.

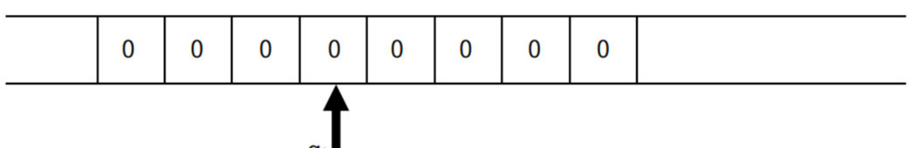
Example: The Zero Function

- d) Since T_0 is back at the q_0**-cell**, step a) is repeated. The input is again *1*. Hence, the instruction of the right-hand arrow again applies. This arrow tells T_0 to replace the tally, *1,* with *0* and to remain in state q_0.
- e) Step b) repeats. The input is *0*, so the left-hand arrow of the q_0**-cell** tells T_0 to move one square to the right and change the state to q_1.



Example: The Zero Function

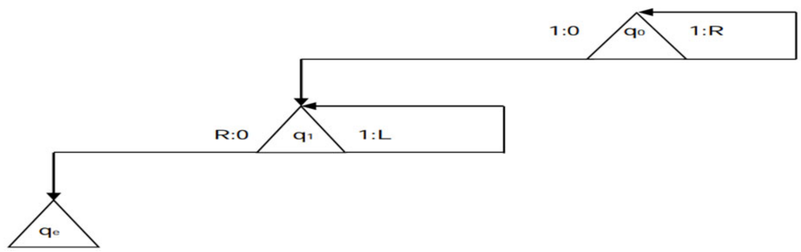
- d) T_0 is back at the q_1**-cell**, and step c) repeats. The pointer stays at the same square, leaving it unchanged. But T_0 changes its state to q_0, as before.
- e) T_0 repeats step a). It replaces *1* with *0* and remains in state q_0. We have:



- d) Step b) repeats: T_0 moves a square to the right and changes to state q_1
- e) Finally T_0 is in state q_1 with an input of *0* instead of *1*. The left arrow applies. T_0 leaves the *0* alone and enters the terminal state, q_e, thus **halting**.

Fancier Machines

- Turing Machines are primitive. But once you have one, you can build more sophisticated machines by incorporating it into the construction. For example, the following machine, T_S, computes the **Successor Function**, $\forall n, f(n) = n+1: \mathbb{N} \rightarrow \mathbb{N}$:



- Given T_S, we can construct a machine that computes **addition**, since addition is just repeatedly taking the successor. Given a machine for addition, we can construct a machine for **multiplication**, since multiplication is repeated addition. And so on.

Notation

- With the idea of a **Turing Machine**, T_M, we can define what it is for a function (from *n*-tuples of natural numbers to natural numbers) to be **Turing Computable**. This is important in order to show that certain functions are not computable, not to show that they are. The direction of the **Church-Turing Thesis** that is hard to deny is that *if* a function is intuitively computable, *then* it is Church-Gödel-Turing computable.
- We write $\rightarrow m$ for the *n*-tuple, $<m_1, m_2, m_3, \ldots, m_n>$, and treat the expression, 'T_M' as a function symbol. $T_M(<m_1, m_2, m_3, \ldots, m_n>) = T_M(\rightarrow m) = \downarrow$ means that T_M **halts** on input $<m_1, m_2, m_3, \ldots, m_n>$, and $T_M(<m_1, m_2, m_3, \ldots, m_n>) = T_M(\rightarrow m) = \uparrow$ means that it **fails to so halt**.
- As with functions, $T_M(<m_1, m_2, m_3, \ldots, m_n>)$ denotes the output of T_M when its input is $<m_1, m_2, m_3, \ldots, m_n>$, *assuming that* $T_M(\rightarrow m) = \uparrow$.

Turing Computability

- **Turing-Computable Function**: A (partial) function $\boldsymbol{f}$: $\mathbb{N}^n \rightarrow \mathbb{N}$ is **Turing-Computable** if/f there is a *Turing Machine*, $\boldsymbol{T_f}$, that computes f. Formally:
  - $\forall \vec{\boldsymbol{m}} \in \mathbb{N}^n$, $\boldsymbol{T_f}(\vec{\boldsymbol{m}}) = \boldsymbol{f}(\vec{\boldsymbol{m}}) = \boldsymbol{f}(<m_1, m_2, m_3, \ldots, m_n>)$, if $\vec{\boldsymbol{m}} \in dom(\boldsymbol{f})$.
  - $\forall \vec{\boldsymbol{m}} \in \mathbb{N}^n$, $\boldsymbol{T_f}(\vec{\boldsymbol{m}}) = \uparrow$ = undefined, if $\vec{\boldsymbol{m}} \notin dom(\boldsymbol{f})$.
- What about **decidable** and **semidecidable** sets? These concepts are defined in terms of **characteristic functions** and **listing functions**, respectively. Suppose that $K \subseteq \mathbb{N}^n$. Then $\boldsymbol{\chi_K}$ is the **characteristic function** of K just when $\boldsymbol{\chi_K}$ is the **total** function, $\mathbb{N}^n \rightarrow \mathbb{N}$, such that:
  - $\forall \vec{\boldsymbol{m}} \in \mathbb{N}^n$, $\boldsymbol{\chi_K}(\vec{\boldsymbol{m}}) = 1$ if $\vec{\boldsymbol{m}} \in K$
  - $\forall \vec{\boldsymbol{m}} \in \mathbb{N}^n$, $\boldsymbol{\chi_K}(\vec{\boldsymbol{m}}) = 0$ if $\vec{\boldsymbol{m}} \notin K$

Turing Computability

- $\boldsymbol{\lambda_K}$ is a **listing function** of K if/f $\boldsymbol{\lambda_K}$ is the **partial** function, $\mathbb{N}^n \rightarrow \mathbb{N}$, such that:
  - $\forall \vec{\boldsymbol{m}} \in \mathbb{N}^n$, $\boldsymbol{\lambda_K}(\vec{\boldsymbol{m}}) = 1$ if $\vec{\boldsymbol{m}} \in K$
  - $\forall \vec{\boldsymbol{m}} \in \mathbb{N}^n$, $\boldsymbol{\lambda_K}(\vec{\boldsymbol{m}}) = \uparrow$ if $\vec{\boldsymbol{m}} \notin K$
- A set K is **decidable** just when its **characteristic function** is (Turing) computable. It is **semidecidable** just when its **listing function** computable.
- We say that K is **effectively enumerable** when there is a total computable function, $\boldsymbol{f}$: $\mathbb{N}^n \rightarrow \mathbb{N}$, with $ran(\boldsymbol{f}) = K$. (**Semidecidable** sets and **effectively enumerable** sets are the same thing.) But how can we allow Turing Machines to output *n-tuples* of numbers, not just individual ones? Instead of altering the definition of a Turing Machine, we can simply use the **Fundamental Theorem of Arithmetic** to encode n-tuples as numbers. The scheme is: $<m_1, m_2, m_3, \ldots, m_n> \rightarrow 2^{m_1} * 3^{m_2} * 5^{m_3} * \ldots p_n^{m_n}$ (where p_n is the nth prime number).

Kleene's Theorem

- There is a connection between decidability and effective enumerability.
- **Kleene's Theorem**: For any $K \subseteq \mathbb{N}^n$, K is decidable just in case both K and its *complement*, $\mathbb{N}^n - K$, are effectively enumerable (equivalently, semidecidable).
- (**Effective Enumerability → Decidability**)
- 1) Suppose that K and $\mathbb{N}^n - K$ are both effectively enumerable.
- 2) Then there are Turing Machines, $\boldsymbol{T_{M1}}$ and $\boldsymbol{T_{M2}}$, that enumerate each, respectively.
- 3) Using $\boldsymbol{T_{M1}}$ and $\boldsymbol{T_{M2}}$, define another machine $\boldsymbol{D_M}$, which decides, for any n-tuple, $<m_1, m_2, m_3, \ldots, m_n>$, whether $<m_1, m_2, m_3, \ldots, m_n> \in K$ or $<m_1, m_2, m_3, \ldots, m_n> \notin K$ as follows.
- 4) Run $\boldsymbol{T_{M1}}$ and $\boldsymbol{T_{M2}}$ in parallel. For any input, $<m_1, m_2, m_3, \ldots, m_n> \in \mathbb{N}^n$, wait some finite number of steps to see which of $\boldsymbol{T_{M1}}$ and $\boldsymbol{T_{M2}}$ outputs $<m_1, m_2, m_3, \ldots, m_n>$. (One of them must because $<m_1, m_2, m_3, \ldots, m_n>$ is either in K or not.)
- 5) If $\boldsymbol{T_{M1}}$ outputs $<m_1, m_2, m_3, \ldots, m_n>$, then $<m_1, m_2, m_3, \ldots, m_n> \in K$. If instead $\boldsymbol{T_{M2}}$ outputs $<m_1, m_2, m_3, \ldots, m_n>$, then $<m_1, m_2, m_3, \ldots, m_n> \notin K$.
- *6*) So, K is decidable.

Kleene's Theorem

- (**Decidability → Effective Enumerability**)
- 7) Suppose that there exists a Turing Machine, $\boldsymbol{D_M}$, which decides, for any n-tuple, $<m_1, m_2, m_3, \ldots, m_n>$, whether $<m_1, m_2, m_3, \ldots, m_n> \in K$ or $<m_1, m_2, m_3, \ldots, m_n> \notin K$.
- 8) Let $\boldsymbol{T^*}$ be a Turing Machine that enumerates all n-tuples whatever, $<m_1, m_2, m_3, \ldots, m_n>$.
- 9) Apply $\boldsymbol{D_M}$ to each $<m_1, m_2, m_3, \ldots, m_n>$.
- 10) If $\boldsymbol{D_M}$ decides that $<m_1, m_2, m_3, \ldots, m_n> \in K$, let $\boldsymbol{T_{M1}}$ outputs $<m_1, m_2, m_3, \ldots, m_n>$. If $\boldsymbol{D_M}$ decides that $<m_1, m_2, m_3, \ldots, m_n> \notin K$, let $\boldsymbol{T_{M2}}$ outputs $<m_1, m_2, m_3, \ldots, m_n>$.
- 11) Thus, K and $\mathbb{N}^n - K$ are both effectively enumerable.

Halting Problem

- The basic limitative results of theoretical computer science is another diagonal argument, like Cantor's or Russell's. At first approximation, the **Halting Problem** is to decide whether, for any *Turing Machine*, ***T*** (henceforth dropping the subscript on ***T*** for readability), and any input, *m*, whether or not ***T***(*m*) halts -- where *m* is the *n*-tuple of only *m*s.
- We will prove that this problem is not Turing Computable. So, assuming the Church-Turing Thesis, not computable by any machine.
- *Note*: The problem amounts to computing a precisely defined function! So, the Halting Problem shows that merely precisely defining a function is insufficient for establishing its computability.

Coding Turing Machines

- Turing Machines take numbers as input. So, in order to assess the Turing Computability of a function that takes Turing Machines themselves as arguments, we first must **code** the latter as numbers.
- Since every Turing Machine is uniquely characterized by its **program**, i.e., **set of instructions**, we need merely to settle on codes for **those**.
- Recall that every line of such a set takes the form q_iXYq_j, where i is any natural number, j is any natural number or the letter, 'e' (q_e is the terminal state of a Turing Machine), X is either *0* or *1*, and Y is either *0, 1, R*, or *L*. So, we first associate basic codes with these symbols.

Coding Turing Machines

- Let us write, [α] for the **code numeral** (not number!) of the object, α. Then:

- [$\boldsymbol{q}_{\mathbf{m}}$] = *1*…(insert *m 0*-numerals)…*0*
- [***0***] = 2
- [***1***] = 3
- [***R***] = 4
- [***L***] = 5
- [$\boldsymbol{q}_{\mathbf{e}}$] = *7*

Coding Turing Machines

- If {l_1, l_2, l_3, l_4,…l_j}, be the instruction lines of *Turing Machine*, ***T*** and l_i is $\boldsymbol{q_k XY q_s}$, then the **code numeral**, [l_i], is the sequence, [$\boldsymbol{q}_k$][***X***][***Y***][$\boldsymbol{q}_s$], and the **code numeral**, [$\boldsymbol{T}_m$], is the sequence, [l_1][l_2][l_3]… [l_j].
- [$\boldsymbol{q}_k$][***X***][***Y***][$\boldsymbol{q}_s$] is not [$\boldsymbol{q}_k$] * [***X***] * [***Y***] * [$\boldsymbol{q}_s$]. Nor is [l_1][l_2][l_3]… [l_j] the product of the [l_i]s. The objects, [$\boldsymbol{q}_k$], [***X***], and so on are ***numerals***.
- *Illustration*: Recall $\boldsymbol{T_0}$ given by the following instruction lines:
  - l_1 : $\boldsymbol{q_0 1 0 q_0}$
  - l_2 : $\boldsymbol{q_0 0 R q_1}$
  - l_3 : $\boldsymbol{q_1 1 1 q_0}$
  - l_4 : $\boldsymbol{q_1 0 0 q_e}$

Coding Turing Machines

- The entire machine program, $\boldsymbol{T_0}$, can then be coded as follows:

- $[l_1] = [\boldsymbol{q_0 10 q_0}] = [\boldsymbol{q_0}][\boldsymbol{1}][\boldsymbol{0}][\boldsymbol{q_0}] = 1321$
- $[l_2] = [\boldsymbol{q_0 0 R q_1}] = 12410$
- $[l_3] = [\boldsymbol{q_1 11 q_0}] = 10331$
- $[l_4] = [\boldsymbol{q_1 00 q_e}] = 10227$

- $[\boldsymbol{T_0}] = [l_1][l_2][l_3][l_4] = 1321124101033110227$

Coding Turing Machines

- Why does this result in an **effective decoding** as well as **coding** scheme?
- Because, given any numeral, we can effectively check if its digits represent basic codes and whether the set of instructions obtained from those codes is a Turing Machine program. If it is such a program, then we can go on to determine what Turing Machine program the set is. Consider again:
- $[\boldsymbol{T_0}] = [l_1][l_2][l_3][l_4] = 1321124101033110227$
- This code is the following numerals in sequence: *1321 12410 10331 10227*
- But *1321* codes $\boldsymbol{q_0 10 q_0}$; *12410* codes $\boldsymbol{q_0 0 R q_1}$; *10331* codes $\boldsymbol{q_1 11 q_0}$; and *10227* codes *10227*. These are just the codes of l_1, l_2, l_3, and l_4 of $\boldsymbol{T_0}$.

Halting Function

- We can now specify a function, ***f***, from the set of all Turing Machines, ***TM***, into the set of natural numbers (*not* numerals!), such that for every ***T*** ∈ ***TM***, ***f***(***T***) = the number picked out by the numeral, [T].
- This function is well defined, since every Turing Machine *has* a unique numerical code, and every such code picks out a unique number. It is also a *one-to-one* function, as no distinct Turing Machines have the same numerical code, and no two numerals pick out the same number.
- This function is, however, not **onto**, since infinitely-many natural numbers fail to correspond to any Turing Machine in the mapping.
- Given this coding of Turing Machines, we specify the **Halting Function**:

Halting Function

- The **Halting Function**, ***H***(*m*, *n*), is a total function from the set of pairs of natural numbers, $\mathbb{N}^2$, into the set of natural numbers, $\mathbb{N}$, such that for all pairs of natural numbers, <*n*, *m*> ∈ $\mathbb{N}^2$, ***H***(*n*, *m*) =
  - *1* if ∃***T*** ∈ ***TM*** such that *n* = the number picked out by the numeral [***T***] and ***T***(*m*) halts (where *m* is a *k*-tuple of *m*s, for *k* = ***T***'s number of inputs)
  - *2* if not
- *Summary*:
- If *n* is not the number picked out by the numerical code of any Turing Machine, ***T***, ***H***(*m*, *n*) = *2*.
- If *n* is the number picked out by the numerical code of a Turing Machine, ***T***, then either ***T***(*m*) halts or not. If it does not, then, again, *H*(*m*, *n*) = *2*.
- If ***T***(*m*) does halt, then ***H***(*m*, *n*) = *1*.

Halting Problem

- **Turing's Theorem**: The *Halting Function*, $\boldsymbol{H}(m, n)$, is <u>not</u> *computable*.
- **Proof**:
- 1) On the strength of the **Church-Turing Thesis**, it suffices to prove that $\boldsymbol{H}(m, n)$ is not <u>Turing Computable</u>.
- 2) Suppose for *reductio* that $\boldsymbol{H}(m, n)$ *is* <u>Turing Computable</u>.
- 3) Then $\exists \boldsymbol{T_H} \in \boldsymbol{TM}$ that computes it, i.e.
  - $\boldsymbol{T_H}(m, n) = 1$ just in case $\boldsymbol{H}(m, n) = 1$
  - $\boldsymbol{T_H}(m, n) = 2$ just in case $\boldsymbol{H}(m, n) = 2$
- 4) We may now use $\boldsymbol{T_H}$ to construct another <u>Turing Machine</u>, $\boldsymbol{T_H}^*$:

Halting Problem

- $\forall n \in \mathbb{N}$:
  - $\boldsymbol{T_H}^*(n, n) = \uparrow$ (<u>fails to halt</u>) if/f $\boldsymbol{T_H}(n, n) = 1$ if/f $\boldsymbol{H}(n, n) = 1$
  - $\boldsymbol{T_H}^*(n, n) = 2$ if/f $\boldsymbol{T_H}(n, n) = 2$ if/f $\boldsymbol{H}(m, n) = 2$
- *Idea*:
- $\boldsymbol{T_H}^*$ <u>reverses the action</u> of $\boldsymbol{T_H}$. $\boldsymbol{T_H}$ halts with an output *1*, for the input, $\langle n, n \rangle$, where *n* is the code of a *Turing Machine* $\boldsymbol{T}$ that <u>halts</u> for the input <u>*n*</u>. So, $\boldsymbol{T_H}(n, n)$ halts with output *1* when $\boldsymbol{T}(\underline{n})$ does.
- By contrast, $\boldsymbol{T_H}^*(n, n)$, <u>fails to halt</u> when $\boldsymbol{T}(\underline{n})$ halts. $\boldsymbol{T_H}^*(n, n)$ <u>halts</u> when and only when *n* is <u>not the numerical code</u> of any Turing Machine, $\boldsymbol{T}$, <u>or</u>, alternatively, when *n is* the code of a Turing Machine, $\boldsymbol{T}$, but $\boldsymbol{T}$ does <u>not</u> halt on the input <u>*n*</u>.

Halting Problem

- 5) Since $\boldsymbol{T_H}$* is a Turing Machine, it has a numerical code $r = [\boldsymbol{T_H}*]$.
- 6) By stipulation, either $\boldsymbol{H}(r; r) = 1$ or $\boldsymbol{H}(r; r) = 2$. So, suppose first that $\boldsymbol{H}(r; r) = 1$.
- 7) Then, since *r* is the numerical code of $\boldsymbol{T_H}$*, $\boldsymbol{T_H}$* must halt when applied to <*r; r*>.
- 8) But $\boldsymbol{T_H}$* only halts with the output *2*.
- 9) By the definition of $\boldsymbol{T_H}$*, $\boldsymbol{H}(r; r) = 2$, which contradicts our assumption that $\boldsymbol{H}(r; r) = 1$.
- 10) So, suppose instead that $\boldsymbol{H}(r; r) = 2$.

Halting Problem

- 11) As *r* is the numerical code of $\boldsymbol{T_H}$*, $\boldsymbol{T_H}$* fails to halt when for the input <*r; r*>.
- 12) So, by the definition of $\boldsymbol{T_H}$*, $\boldsymbol{H}(r; r) = 1 \neq 2$, contrary to our assumption.
- 13) The *Reductio Assumption* is false; the Halting Function, $\boldsymbol{H}(m, n)$, is *not* Turing Computable.
- 14) On the strength of the **Church-Turing Thesis**, $\boldsymbol{H}(m, n)$ is not computable.

Diagonal Arguments Again

- We noted that Cantor's Theorem, and even Russell's Paradox, are kinds of **diagonal argument**. But Turing's Theorem is a more vivid example.
- By the definition of T_H*, T_H* acts on the diagonal of the function ***H***(*n, m*).
- If *n* is the numerical code of a Turing Machine, ***T***, ***H*** applies ***T*** to input *m*.
- If ***T***(*m*) halts, ***H*** outputs *1*, and if ***T***(*m*) does not halt, ***H*** outputs *2*.
- ***H***(*n*, *n*) is called the diagonal value of ***H***. If *n* is the numerical code of a Turing Machine, ***T***, then ***H*** takes ***T*** and applies it to its own code.
- T_H* then applies to these diagonal values: for any Turing Machine with code, *n*, if ***T*** halts on *n*, T_H* does not halt; and if ***T*** does not halt on *n*, T_H* halts.
- The trick is, as before, to ask about the diagonal value of T_H* itself.

Partial Recursive Functions

- We have been discussing one analysis of the notion of an **effective procedure**, in terms of Turing Machines. There are several others, like Church's. But, historically, the first was actually due to Gödel.
- Gödel defined a class of functions (which, amazingly, turn out to be exactly the Turing & Church computable functions) as follows.
- First, he specified basic functions: the **zero function**, the **successor function**, and the **projection functions** (there are infinitely-many).
- The **zero function**, ***Z***, is a total function, $\mathbb{N}$ into $\mathbb{N}$, such that $\forall n \in \mathbb{N}$, ***Z***(*n*) = *0*.

Partial Recursive Functions

- The **successor function**, $\boldsymbol{S}$, is a total function, $\mathbb{N}$ into $\mathbb{N}$, such that $\forall n \in \mathbb{N}$, $\boldsymbol{S}(n)$ = the successor of n (where 'successor of' is conceptually primitive).
- The **projection functions**, $\boldsymbol{J}_{\mathbf{i}}^{\mathbf{n}}$, are total functions, $\mathbb{N}^n$ into $\mathbb{N}$, such that $\boldsymbol{J}_{\mathbf{i}}^{\mathbf{n}}(\langle m_1, m_2, m_3, \ldots, m_n \rangle) = m_i$.
- Next, Gödel specified some **operations**. The first is **composition**:
  - If $\boldsymbol{G}$ is an m-place function, and $\boldsymbol{H_1}, \boldsymbol{H_2}, \ldots$, and $\boldsymbol{H_m}$ are at most m distinct n-place functions, then the **composition** of $\boldsymbol{G}$ and $\boldsymbol{H_1}, \boldsymbol{H_2}, \ldots$, and $\boldsymbol{H_m}$ is the n-place function, $\boldsymbol{F}$, defined as follows:
  - $\boldsymbol{F}(m_1, m_2, m_3, \ldots, m_n) = \boldsymbol{G}(\boldsymbol{H_1}(m_1, m_2, m_3, \ldots, m_n), \boldsymbol{H_2}(\boldsymbol{G}(\boldsymbol{H_1}(m_1, m_2, m_3, \ldots, m_n), \ldots, \boldsymbol{H_m}(m_1, m_2, m_3, \ldots, m_n))$.

Primitive Recursion

- The second **operation** on recursive functions is **primitive recursion**.
- Let $\boldsymbol{G}$ be an n-place function and $\boldsymbol{H}$ an $n+2$-place function. Then we may define an $n+1$-place function, $\boldsymbol{F}$, from $\boldsymbol{G}$ and $\boldsymbol{H}$ as follows:
  - $\boldsymbol{F}(m_1, m_2, m_3, \ldots, m_n, 0) = \boldsymbol{G}(m_1, m_2, m_3, \ldots, m_n)$, and
  - $\boldsymbol{F}(m_1, m_2, m_3, \ldots, m_n, \boldsymbol{S}(k)) = \boldsymbol{H}(m_1, m_2, m_3, \ldots, m_n, k, \boldsymbol{F}(m_1, m_2, m_3, \ldots, m_n, k))$
- If $n = 0$, $\boldsymbol{F}$ is defined from the function $\boldsymbol{H}$ as follows:
  - $\boldsymbol{F}(0) = p$ (where p is some constant, $p \in \mathbb{N}$), and
  - $\boldsymbol{F}(\boldsymbol{S}(k)) = \boldsymbol{H}(k, \boldsymbol{F}(k))$
- In this case, $\boldsymbol{F}$ is said to be defined by **primitive recursion**.

Primitive Recursion

- Primitive Recursion can appear more technical than Composition. However, it is actually familiar. It is just generalized induction. The value of a function, ***F***, is defined for the argument *0* and then its value is defined for the argument ***S***(*q*) in terms of its value for *q*; ***F*** is thereby defined for every natural number.
- The only sense in which Primitive Recursion *generalizes* the inductive manner of definition is that it concerns functions with any number of arguments.
  - *Note*: In set theory, one generalizes recursion (and hence induction) in another way – namely, from natural numbers to well-orderings, including the class of all ordinals.
- When we only consider functions generated by composition and primitive recursion the resulting functions are called **primitive recursive functions**.
- All primitive recursive functions are total, since all the basic recursive functions are total and these two operations yield total functions when applied to total functions. But Gödel's final operation can yield partial functions.

Minimization

- The final operation is called **minimization**. Let ***F*** be an *n+1*-place function, where $n > 0$, and suppose that for some $k_1, k_2, k_3, \ldots, k_n$, there exists a natural number *m* such that $\boldsymbol{F}(k_1, k_2, k_3, \ldots, k_n, m) = 0$.
- Moreover, for every natural number $t < m$, $\boldsymbol{F}(k_1, k_2, k_3, \ldots, k_n, t) > 0$.
  - We say that *m* is the **least zero** for ***F***.
- Then we may define an *n*-place function **μ*F*** by minimization from the *n+1*-place function ***F*** such that $\boldsymbol{\mu F}(k_1, k_2, k_3, \ldots, k_n) = m$. That is:
- $\boldsymbol{\mu F}(k_1, k_2, k_3, \ldots, k_n) =$
  - *m* if/f $\boldsymbol{F}(k_1, k_2, k_3, \ldots, k_n, m) = 0$ and for $\forall t < m$, $\boldsymbol{F}(k_1, k_2, k_3, \ldots, k_n, t) > 0$
  - ↑ (undefined) otherwise

Partial Recursive Functions

- We may finally define the class of Gödel's Partial Recursive Functions:
- **Partial Recursive Functions**: A function ***F*** from $\mathbb{N}^n$ into $\mathbb{N}$ is a partial recursive function just in case ***F*** is a **basic recursive function** (a **zero**, **successor**, or **projection** function), or ***F*** is obtained from the basic recursive functions by finitely many applications of one or more of the **operations** (**composition**, **primitive recursion**, and **minimization**).
- *Terminology*: The **primitive recursive functions** are those obtainable from the basic ones *via* composition and primitive recursion alone.
- And instead of speaking of Turing computable or enumerable sets and relations, we will (up to extensional equivalence) speak of **recursive sets** and relations, and **recursively enumerable** sets and relations, respectively.

Symbolic Logic

Incompleteness

Complete Theories

- We have been discussing the system *PL*, its metalogical properties, and the notion of decidable and semidecidable sets, relations, and so on.
- In the final section of this class, all of these concepts come together in the context of some further concepts which we have yet to introduce.
- Recall that a *PL* set **Σ** is a **theory** just in case **Σ** contains all its logical consequences that are in *Voc*(**Σ**). Likewise, **Σ** is **complete** just when, for <u>every</u> *PL* sentence, ***X***, in *Voc*(**Σ**), either **Σ** |- ***X*** or **Σ** |- ~***X***.
- *Reminder*: This is distinct from the completeness of the <u>Completeness Theorem</u>, which says that, for any set, **Σ**, and sentence, ***X***, **Σ** |- ***X*** if **Σ** |= ***X***.

Arithmetic

- By the **Soundness** and **Completeness** theorems, we can speak ambiguously between <u>syntactic</u> and <u>semantic</u> consequences.
- **Σ** is a **theory** just in case **Σ** = {***X*** : ***X*** is in *Voc*(**Σ**) & **Σ** |- ***X***} just in case {***X*** : ***X*** is in *Voc*(**Σ**) & **Σ** |= ***X***}.
- Perhaps the most important theory in all of mathematics is the theory of <u>natural number arithmetic</u>. It underlies the most rudimentary mathematical thought in which we engage concerning cardinalities and ordinalities of finite things. But it is also in the background of nearly all <u>formal theorizing</u>. We pervasively applied the <u>arithmetic</u> principle of *Mathematical Induction* when proving **Soundness**. And we have hinted that the theory of <u>*PL* proofs</u> and even <u>Turing Machines</u> is arithmetic in disguise (insofar as we can <u>code</u> symbols and instructions as numbers).

Peano Arithmetic

- What is the canonical theory of natural number arithmetic? It is is *Th*(*PA*) = {***X*** : ***X*** is in *Voc*(*PA*) & *PA* |- ***X***} ={***X*** : ***X*** is in *Voc*(*PA*) & *PA* |= ***X***}. That is, it is the set of consequence of the Peano Axioms.
- We rehearsed the Peano Axioms when discussing the Resources of the Metatheory. But we did not write them in the language of (first-order) *PL*. Even now, we will not be quite so particular. Officially, we said that the non-logical predicates in a *PL* language take the form: $A_1, B_1, C_1, \ldots, X_1, Y_1, Z_1; A_2, B_2, C_2, \ldots, X_2, Y_2, Z_2; A_3, B_3, C_3, \ldots, X_3, Y_3, Z_3;$ …. Being humans (!), we will not write the predicates for addition and multiplication in this robotic way. We will write them as we do ordinarily, regarding this as an abbreviation of the official expressions.

Peano Arithmetic

- Before specifying the (first-order) Peano Axioms (*PA*), we should carefully distinguish *Th*(*PA*) from *Th*(ℕ). Recall that $Th_{\boldsymbol{\Sigma}}(J)$ = {***X***: ***X*** is a sentence in *Voc*(**Σ**) that is true on *J*}. So, *Th*(ℕ), known as **True Arithmetic**, is the set, {***X***: ***X*** is a sentence in *Voc*(*PA*) that is true on ℕ}.
- If *Con*(*PA*), then certainly *Th*(*PA*) ⊆ *Th*(ℕ). But whether the converse holds, or holds for any (recursively) axiomatizable extension of *Th*(*PA*), is a matter which will occupy us repeatedly throughout this section.
- ℕ is called the **Standard Model**, the assumption being that it is a model of *PA*, and, therefore, that *Con*(*PA*). Let us assume as much.

Peano Arithmetic

- *Voc*(*PA*) includes the constant, ***0*** (‘*zero*’), the monadic function symbol, ***s***(*x*) (‘the *successor* of *x*’), and the binary function symbols, + (‘plus’) and * (‘times’), along with the logical vocabulary, including =.
- We can now describe the interpretation, i.e., (we assume) model, ℕ.
- *UD*: {*0*, *1*, *2*, *3*, …}
- *LN*: $\boldsymbol{0}, \boldsymbol{c_1}, \boldsymbol{c_2}, \ldots \boldsymbol{c_n} \ldots$
- **Semantical assignments**:
- $\mathbb{N}(\boldsymbol{0})$: *0*; $\mathbb{N}(\boldsymbol{c_1})$: *1*; $\mathbb{N}(\boldsymbol{c_2})$: *2*; …; $\mathbb{N}(\boldsymbol{c_n})$: *n*; …
- ℕ(***s***(*x*)): the successor of *x*: *S*(*x*)
- ℕ(*x* + *y*): the sum of x and y: *x* + *y* (*N.B*. metalanguage vs. object language!)
- ℕ(*x* * *y*): the product of *x* and *y*: *x* * *y*

Peano Arithmetic

- *Th*(ℕ), i.e., **True Arithmetic**, is then simply the set of sentences in *Voc*(*PA*) that are true under the aforementioned interpretation.
- What about *PA*? Like the *ZFC* axioms, it has infinitely-many members:
- $\mathbf{Ax_1}$ $(\forall x)\ \boldsymbol{0} \neq \boldsymbol{s}(x)$
- $\mathbf{Ax_2}$ $(\forall x)(\forall y)(\boldsymbol{s}(x) = \boldsymbol{s}(y) \rightarrow x = y)$
- $\mathbf{Ax_3}$ $(\forall x)(x + \boldsymbol{0}) = x$
- $\mathbf{Ax_4}$ $(\forall x)(\forall y)(x + \boldsymbol{s}(y)) = \boldsymbol{s}(x + y)$
- $\mathbf{Ax_5}$ $(\forall x)(x * \boldsymbol{0}) = \boldsymbol{0}$
- $\mathbf{Ax_6}$ $(\forall x)(\forall y)(x * \boldsymbol{s}(y)) = ((x * y) + x)$
- **IS** If ***X***[*z*] is formula in *Voc*(*PA*) that contains exactly one variable with all and only free occurrences, *z*, and it does not contain occurrences of the variables *v* or *y*, then the following is an axiom: $\boldsymbol{X}[\boldsymbol{0}]\ \&\ ((\forall v)(\boldsymbol{X}[v] \rightarrow \boldsymbol{X}[\boldsymbol{s}(v)]) \rightarrow (\forall y)\boldsymbol{X}[y])$.

Peano Arithmetic

- The axioms, *PA*, form an infinite set because **IS** is a metalinguistic schema giving one axiom for each *PL* formula, ***X***. Since there are infinitely-many formulas in *Voc*(*PA*), there are so many **IS** axioms.
- Crucially, however, *PA* is a set of axioms in our sense. Recall that Γ is a **set of axioms** for **Σ** just when Γ is a **decidable** set of sentences in *Voc*(**Σ**) and **Σ** = *Th*(Γ) ={***X*** : ***X*** is in *Voc*(Σ) and Γ \|= ***X***} ={***X*** : ***X*** is in *Voc*(**Σ**) and Γ \|– ***X***}. *PA* is a set of axioms because it is decidable whether a string is any of $\mathbf{Ax_1}$ – $\mathbf{Ax_6}$ and whether it is an instance of **IS**.
- *Note*: To say that it is decidable whether a string is a member of *PA* is not to say that it is decidable whether it is a member of *Th*(*PA*)!

Peano Arithmetic

- We have assumed *Con*(*PA*), and, hence, by the **Soundness Theorem**, *Con*(*Th*(*PA*)). *Th*(*PA*) is by definition **axiomatizable**. But is it **complete**? Is it the case that for every ***X*** in *Voc*(*PA*), *PA* \|- ***X*** or *PA* \|- ~***X***?
- Ordinary mathematical practice suggests that completeness is at least presupposed. When attacking a problem, number theorists expect there to be an answer, however difficult it may be to deduce. Strictly speaking, they may allow that the answer only follows from stronger axioms than the *PA* axioms, perhaps even all of *ZFC* and a bit more.
- We will find that **every** consistent recursively axiomatized theory extending *PA* is **incomplete**, even as it concerns natural number arithmetic.

Peano Arithmetic

- **Lemma 5.1.1**: Suppose that, for every $\boldsymbol{X} \in \Sigma$, $\Gamma \mid\!- \boldsymbol{X}$, and $\Sigma \mid\!- \boldsymbol{Z}$. Then $\Gamma \mid\!- \boldsymbol{Z}$.
- **Proof**:
- 1) Since $\Sigma \mid\!- \boldsymbol{Z}$, there is a <u>finite</u> subset $\Sigma_{\text{fin}} \subseteq \Sigma$ such that $\Sigma_{\text{fin}} \mid\!- \boldsymbol{Z}$.
- 2) Since for every $\boldsymbol{X} \in \Sigma$, $\Gamma \mid\!- \boldsymbol{X}$, and $\Sigma_{\text{fin}} \subseteq \Sigma$, for every $\boldsymbol{X} \in \Sigma_{\text{fin}}$, $\Gamma \mid\!- \boldsymbol{X}$.
- 3) From 2), for every $\boldsymbol{X} \in \Sigma_{\text{fin}}$, there is a <u>derivation</u> $\boldsymbol{D}_X$ of $\boldsymbol{X}$ from Γ.
- 4) Since Σ_{fin} is a <u>finite</u> set, there is a finite set consisting, for each $\boldsymbol{X} \in \Sigma_{\text{fin}}$, of a derivation of $\boldsymbol{X}$ from Γ.

Peano Arithmetic

- 5) Similarly, there is also a derivation <u>derivation</u> $\boldsymbol{D}_z$ of $\boldsymbol{Z}$ from Σ_{fin}.
- 6) We may now <u>combine</u> all of the aforementioned into <u>one</u> derivation of $\boldsymbol{Z}$ from Γ.
- 7) So, by **Soundness**, $\Gamma \mid\!- \boldsymbol{Z}$.
- *Upshot*: *PA* is **complete** if *Th*(*PA*) is. So, we can focus on *PA*.
- Let us henceforth avail ourselves of additional abbreviations. We use of the full vocabulary of and inference machinery of *NDS*, recalling that its connectives are <u>truth-functionally equivalent</u> to {∀, ~, →}, and its inference rules our <u>proof-theoretically equivalent</u> to those of *MDS*.

Representability in *PA*

- We will also abbreviate **numerals** (not numbers!).
- We let $\boldsymbol{s}^0\boldsymbol{0}$ abbreviate ***0***,
- $\boldsymbol{s}^1\boldsymbol{0}$ abbreviate ***s0***,
- $\boldsymbol{s}^2\boldsymbol{0}$ abbreviate ***ss0***;
- …
- In general, $\boldsymbol{s}^n\boldsymbol{0}$ abbreviates ***sss***…(*n* times)…***0***, and write ***n*** for an arbitrary numerical term whose **referent** is *n* in the standard model of arithmetic, ℕ.
- We also rely on the following conventions:
- **5.2a** For all terms in *Voc*(*PA*), ***t*** and ***s***, ***t*** < ***s*** abbreviates (∃z)(z ≠ ***0*** & (***t*** + ***z***) = ***s***).
- **5.2b** For all terms in *Voc*(*PA*), ***t*** and ***s***, ***t*** ≤ ***s*** abbreviates ***t*** < ***s*** v ***t*** = ***s***.

Representability in *PA*

- In light of these conventions, we can state the following theorems:
- **Theorem 5.2.1**: For all natural numbers *n* and *m*, *n* = *m* just in case *PA* |– $\boldsymbol{s}^n\boldsymbol{0} = \boldsymbol{s}^m\boldsymbol{0}$, and *n* ≠ *m* just in case *PA* |– $\boldsymbol{s}^n\boldsymbol{0} \neq \boldsymbol{s}^m\boldsymbol{0}$.
- **Theorem 5.2.2**: ℕ($\boldsymbol{s}^n\boldsymbol{0}$) = *n*, for every natural number *n* (i.e., for every numeral $\boldsymbol{s}^n\boldsymbol{0}$ in *Voc*(*PA*), its *referent* on ℕ is the natural number, *n*).
- **5.2c** For all natural numbers *n* and *m*, if *n* < m, then *PA* |– ***n*** < ***m***.
- **5.2d** For all natural numbers *n* and *m*, if *n* ≤ *m*, then *PA* |– ***n*** ≤ ***m***.
- **5.2e** For every natural number *n*, if *n* is even, then *PA* |– (∃z)(z ≤ ***n*** & ***n*** = (*z* + *z*)).
- **5.2f** For every natural number *n*, if *n* is odd, then *PA* |– (∃z)(z ≤ ***n*** & ***n*** = (*z* + *z*)).
- **5.2g** For all natural numbers *n*, *m*, and *k*, if (*n* + *m*) = *k*, then *PA* |– (***n*** + ***m***) = ***k***.
- **5.2h** For all natural numbers *n*, *m*, and *k*, if (*n* * *m*) = *k*, then *PA* |– (***n*** * ***m***) = ***k***.

Representability in *PA*

- All of these theorems are proved by routine applications of Mathematical Induction. But let us look at the proof of one half of **Theorem 5.2.1**.
- **Theorem 5.2.1a** For all natural numbers *n* and *m*, $n = m$ just in case *PA* |– $s^n 0 = s^m 0$.
- **Proof**:
- 1) For the base case, suppose that $n = 0$. Then if $n = m$, $m = 0$ too, by the transitivity of equality.
- 2) But *PA* |– $0 = 0 = s^0 0 = s^0 0$, since ∅ |- $s^0 0 = s^0 0$.
- 3) Conversely, if *PA* |–$s^0 0 = s^m 0$, then $m = 0$ (since otherwise *PA* would prove that *0* is the successor of some number, contrary to $\mathbf{Ax_1}$ $(\forall x)\ 0 \neq s(x)$).

Representability in *PA*

- 4) For the inductive case, suppose that for every *m*, $k = m$ just in case *PA* |- $s^k 0 = s^m 0$, and assume that $s^{k+1} 0 = s^m 0$.
- 5) Then $k = m - 1$, and, by the Induction Hypothesis, *PA* |- $s^k 0 = s^{m-1} 0$.
- 6) So, also, *PA* |- $ss^k 0 = ss^{m-1} 0$.
- 7) But $ss^k 0$ is just $s^{k+1} 0$ and $ss^{m-1} 0$ is just $s^m 0$, i.e. *PA* |- $s^{k+1} 0 = s^m 0$.
- 8) Conversely, assume that *PA* |- $s^{k+1} 0 = s^m 0$, that is *PA* |- $ss^k 0 = ss^{m-1} 0$.
- 9) Then, by $\mathbf{Ax_2}$ $(\forall x)(\forall y)(s(x) = s(y) \rightarrow x = y)$, *PA* |- $s^k 0 = s^{m-1} 0$.
- 10), So, by the Induction Hypothesis, $k = m - 1$, and, hence, $k + 1 = m$.

Representability in *PA*

- The upshot of **5.2.1** and **5.2.2** is that simple arithmetic facts are '<u>mirrored</u>' in the theory *PA*. *PA* '<u>knows</u>' grade school arithmetic. More generally:
- **Definition 5.2.1**
- **5.2.1a** For every <u>set of natural numbers</u>, *B*, *B* is **representable** in *Th*(*PA*) just in case there is a <u>formula</u> in *Voc*(*PA*), ***X***[*z*] with <u>one free variable</u> such that, for <u>every natural number</u>, *k*, <u>if</u> $k \in B$, <u>then</u> *PA* |- ***X***[***k***], and <u>if</u> $k \notin B$, <u>then</u> *PA* |– ~***X***[*k*].
- **5.2.1b** For every <u>*n*-place relation</u>, *R*, on ℕ, *R* is **representable** in *Th*(*PA*) just in case there is a <u>formula</u> in *Voc*(*PA*), ***X***[$z_1, z_2, z_3, \ldots, z_n$] with <u>*n* free variables</u> such that <u>for each *n*-tuple</u> of natural numbers $<k_1, k_2, k_3, \ldots, k_n>$, <u>if</u> $<k_1, k_2, k_3, \ldots, k_n> \in R$, <u>then</u> *PA* |– ***X***[$\mathbf{k_1, k_2, k_3, \ldots, k_n}$], and <u>if</u> $<k_1, k_2, k_3, \ldots, k_n> \notin R$, <u>then</u> *PA* |– ~***X***[$\mathbf{k_1, k_2, k_3, \ldots, k_n}$].

Representability in *PA*

- **5.2.1c** For every <u>total *n*-place function</u>, *F*, $\mathbb{N}^n \rightarrow \mathbb{N}$, *F* is **representable** in *Th*(*PA*) just in case there is <u>formula</u> in *Voc*(*PA*), ***X***[$z_1, z_2, z_3, \ldots, z_n, z_{n+1}$] with <u>*n+1* free variables</u> such that for <u>each</u> *n*-tuple $<k_1, k_2, k_3, \ldots, k_n>$ of <u>natural numbers</u> and for <u>each</u> natural number, *k*, <u>if</u> $k_1, k_2, k_3, \ldots, k_n) = k$, <u>then</u> *PA* |– ***X***[$\mathbf{k_1, k_2, k_3, \ldots, k_n, k}$] <u>and</u> *PA* |– $(\forall x)$(***X***[$\mathbf{k_1, k_2, k_3, \ldots, k_n}$, *x*] $\rightarrow x = k$).
- **Representability Theorem**: <u>Every</u> (total) <u>recursive function</u> is <u>representable</u> in *Th*(*PA*) (and, moreover, it is representable by a so-called $\mathbf{\Sigma_1}$ formula).
- **Theorem 5.2.4**: Let $D \subseteq \mathbb{N}^n$, for $n \geq 1$. Then *D* is **representable** in *Th*(*PA*) just in case its <u>characteristic function</u> is representable in *Th*(*PA*).
- *Upshot*: Since <u>characteristic functions</u> of recursive <u>sets</u> and <u>relations</u> are recursive, <u>all recursive sets and relations are also representable</u> in *Th*(*PA*).

Representability in *PA*

- The Representability Theorem is one of the integral components to the proof of the Incompleteness Theorems. The proof of this theorem is tedious but routine. One argues that the basic recursive functions are representable, and that representability is closed under the three operations that we discussed.
- **Definition 5.2.1** makes precise the sense in which *PA* 'mirrors' or 'knows' grade school arithmetic and much more. Whenever an *n*-tuple of numbers belongs to a recursive set (which may be a relation or a function), *PA* proves that a corresponding formula holds of the *n*-tuple of numerals of those numbers.
- Conversely, whenever an *n*-tuple fails to belong to such a set, *PA* proves that the relevant formula fails to hold of the *n*-tuple of numerals of those numbers.
- In the function case, it also proves that the last member of the tuple is unique.

Arithmetization of the Metatheory

- In fact, a function is recursive just in case it is representable in a very weak fragment of *Th*(*PA*) known as **Robinson Arithmetic** *(RA*). This is basically *Peano Arithmetic* minus **IS**. *RA* is so weak that it does not even prove that addition is commutative! But it is already vulnerable to the Incompleteness Theorems, as it represents all recursive functions.
- Given the Representability Theorem, the next order of business is to establish a **Gödel Numbering**. This is an assignment of natural numbers to expressions in *Voc*(*PA*) meeting the following conditions.
  - Every grammatical expression and sequence of sentences in *Voc(PA)* has a unique Gödel number, and that no two different items have the same Gödel number. Finally, the encoding and decoding procedures are **effective**.

Arithmetization of the Metatheory

- Although Gödel's technique was revolutionary in 1931, it is now familiar with the advent of computers. These routinely code the expressions that we type as numbers. There are many different Gödel numberings the choice of which is immaterial. Therefore, we ignore the details of any particular coding.
- What is important is that facts about the syntax of *PA* – and, hence, *PL* – correspond to facts about the natural numbers, via Gödel numberings.
- Moreover, whenever those facts are recursive, *Th*(*PA*), 'knows' them.
- In other words, *Th*(*PA*) **proves many facts about its own syntax**.
- *Example*: Consider $\mathbf{Ax_1}$ $(\forall x)(\boldsymbol{0} \neq \boldsymbol{s}x)$. We will write $[(\forall x)(\boldsymbol{0} \neq \boldsymbol{s}x)]$ for the Gödel *number* (*GN*) of $\mathbf{Ax_1}$ and $\ulcorner (\forall x)(\boldsymbol{0} \neq \boldsymbol{s}x) \urcorner$ for the *numeral* in *Voc*(*PA*) of that Gödel number. If $\mathbf{SENT}_{PA}$ is the set of *GN*s of sentences in *Voc*(*PA*), and $[(\forall x)(\boldsymbol{0} \neq \boldsymbol{s}x)] = m$, then $m \in \mathbf{SENT}_{PA}$, and $PA \vdash \mathbf{sent}_{PA}(\boldsymbol{m})$, for some corresponding predicate in *Voc*(*PA*) $\mathbf{sent}_{PA}(x)$, because $\mathbf{SENT}_{PA}$ *is recursive*.

Arithmetization of the Metatheory

- There are three different concepts to keep in separated:
  - **Syntactic items**
  - **Numbers**
  - **Numerical terms for those numbers in *Voc*(*PA*)**
- Here is the picture:

Φ	→	Gödel number	→	$\ulcorner \Phi \urcorner$
↓		↓		↓
An expression in *Voc*(*PL*)		An element of $\mathbb{N}$		A numerical term in *Voc*(*PL*) for the Gödel number of Φ

Arithmetization of the Metatheory

- What syntactic functions, relations, and sets are recursive? They include:

- The set of **basic symbols** of *Voc(PA)*
- The set of **terms** of *Voc(PA)*
- The set of **atomic formulas** of *Voc(PA)*
- The set of **complex formulas** of *Voc(PA)*
- The set of **sentences** of *Voc(PA)* (just illustrated)
- The set of **proofs from *PA***

- *Upshot*: There are predicates in *Voc*(*PA*) representing each, as with $\mathbf{sent}_{PA}(x)$, namely:

Arithmetization of the Metatheory

- The set of **basic symbols** = $\mathbf{SYM}_{PA}$. Since this is recursive, there is a predicate in *Voc*(*PA*), $\mathbf{sym}_{PA}(x)$, such that *PA* |- **sym(*k*)** if *k* is the *GN* of a basic symbol, and *PA* |- ~$\mathbf{sym}_{PA}(\boldsymbol{k})$ if *k* is not the *GN* of a basic symbol.
- The set of **terms** = $\mathbf{TERM}_{PA}$….there is a predicate in *Voc*(*PA*), $\mathbf{term}_{PA}(x)$, such that *PA* |- $\mathbf{term}_{PA}(\boldsymbol{k})$ if *k* is the *GN* of a term, and *PA* |- ~$\mathbf{term}_{PA}(\boldsymbol{k})$ if not.
- The set of **atomic formulas = $\mathbf{AFORM}_{PA}$**….there is a predicate, $\mathbf{aform}_{PA}$ (*x*), such that *PA* |- $\mathbf{aform}_{PA}(\boldsymbol{k})$ if *k* is the *GN* of an atomic formula, and *PA* |- ~$\mathbf{aform}_{PA}(\boldsymbol{k})$ if not.

Arithmetization of the Metatheory

- The set of **formulas = FORM**$_{PA}$….there is a predicate **form**$_{PA}$(*x*), such that *PA* |- **form**$_{PA}$(***k***) if *k* is the *GN* of a formula, and *PA* |- ~**form**$_{PA}$(***k***) if not.
- The relation of **proof = PROOF**$_{PA}$ (alternatively: the set of proof-proved pairs)….there is a predicate **proof**$_{PA}$(*x, y*) such that *PA* |- **proof**$_{PA}$(***m, n***) if *m* is the *GN* of a proof whose conclusion has *GN n*, and *PA* |- ~**proof**$_{PA}$(***m, n***) if not.
- However, we will see that the property of being provable from *PA* (set of *GN*s of provable sentences, i.e., theorems of *PA*) is **not** recursive.

Arithmetization of the Metatheory

- Why is **PROOF**$_{PA}$ recursive? Because given any finite sequence, ***S***, of sentences *Voc*(*PA*) and any sentence ***X***, there is an effective decision procedure to check whether ***S*** is a *PL* derivation of ***X*** from *PA* or not.
- The set of *PA* axioms, and the *MDS* rules, are decidable by definition.
- So, we can examine every sentence in ***S***. First, check the terminal sentence X_n to see if it ***X***. If it is not, then ***S*** is not a derivation of ***X***. If it is, check the first sentence X_1 to see if it is a *PA* axiom or is introduced by one of the rules of *MDS*. If X_1 passes the test, examine X_2 to check whether it is a *PA* axiom or is introduced by one of the *MDS* rules. And so on. Since ***S*** is finite, we only have to apply this procedure finitely many times before we know whether ***S*** is a proof.

Arithmetization of the Metatheory

- **Terminology (Quantifier Complexity)**:
- A $\mathbf{\Sigma_1}$ formula is of the form $(\exists_1 x)(\exists_2 x)....(\exists_n x)\Delta_0$, where Δ_0 is a formula with only <u>bounded</u> quantifiers.
- A $\mathbf{\Pi_1}$ formula is of the form $(\forall_1 x)(\forall_2 x)....(\forall_n x)\Delta_0$.
- *Upshot*: The negation of a $\mathbf{\Sigma_1}$ formula is $\mathbf{\Pi_1}$, while the negation of a $\mathbf{\Pi_1}$ formula is $\mathbf{\Sigma_1}$.
- The recursive relation, $\mathbf{PROOF}_{PA}$, is of special interest. On the basis of it, we can **define** the set of <u>*Gödel numbers of a theorems of PA*</u>:
  - $\mathbf{THEOREM}_{PA}(n) \stackrel{def}{=} (\exists x)\mathbf{PROOF}_{PA}(x, n)$

Arithmetization of the Metatheory

- $\mathbf{THEOREM}_{PA}(n)$ can be <u>expressed</u> or <u>defined</u> (not <u>represented</u>!) using a $\mathbf{\Sigma_1}$ predicate, $(\exists x)\mathbf{proof}_{PA}(x, \boldsymbol{n})$, in *Voc*(*PA*):
  - $\mathbf{Thrm}_{PA}(\boldsymbol{n}) \stackrel{def}{=} (\exists x)\mathbf{proof}_{PA}(x, \boldsymbol{n})$
- $\mathbf{THEOREM}_{PA}(n)$ is <u>expressed</u> by $\mathbf{Thrm}_{PA}(\boldsymbol{n})$ in that if $n \in \mathbf{THEOREM}_{PA}$, then $\mathbb{N} \models \mathbf{Thrm}_{PA}(\boldsymbol{n})$, and if $n \notin \mathbf{THEOREM}_{PA}$, $\mathbb{N} \models {\sim}\mathbf{Thrm}_{PA}(\boldsymbol{n})$. (That is, the predicate is <u>true</u> of a <u>singular term</u> picking out the number *n* just in case that number is the *GN* of a theorem. Whether *PA* '<u>knows</u>' this, however, is a different matter.)
- Some authors distinguish **strong** and **weak** representability. In those terms, **THEOREM**(*n*) is <u>weakly</u>, but <u>not strongly</u> representable, in *Th*(*PA*). That is:
- If $n \in \mathbf{THEOREM}_{PA}$, then *PA* |- $\mathbf{Thrm}_{PA}(\boldsymbol{n})$
- However, it is **not** the case that: If $n \notin \mathbf{THEOREM}_{PA}$, *PA* |- ${\sim}\mathbf{Thrm}_{PA}(\boldsymbol{n})$!

Arithmetization of the Metatheory

- **Weakly representable** properties (sets), like $\textbf{THEOREM}_{PA}$, are **recursively enumerable**, but not **recursive.** Why is this the case?
- Consider the set, $\textbf{THEOREM}_{PA}$ to which *k* belongs just in case $(\exists x)\textbf{PROOF}_{PA}(x, k)$. The problem with this condition is its **unbounded** existential quantifier. Suppose we are given a number *k*, and we want to check whether $k \in \textbf{THEOREM}_{PA}$ or not. We look through ordered pairs of numbers, $<a, k>$, checking each *a* to see if it is the *GN* of a proof of the sentence whose *GN* is *k*. *If* there is such a number, *then* $<a, k> \in \textbf{PROOF}_{PA}$, and $k \in \textbf{THEOREM}_{PA}$. We will eventually find this *a*.
- But what if there is no such number? Then we will search forever.

Arithmetization of the Metatheory

- *If the existential quantifier in* $(\exists x)\textbf{PROOF}_{PA}(x, k)$ *were bounded*, then we would have a **decision procedure** for checking provability in *PA*.
- But proofs in *PA* have no (finite) bound. They can be of any length. So, there is no limit on the *GN*s of possible proofs of the sentence with *GN*, *k* – for arbitrary *k*.
- Even if we place a bound on possible proof lengths – say, *2*^(*100*)^(*100*), and call the resulting set, $\textbf{THEOREM}_{PA}$#, $\textbf{THEOREMHOOD}_{PA}$# may remain **effectively undecidable** by humans -- despite being **decidable** (i.e., **recursive**), and so representable in *PA* by some predicate, $\textbf{Thrm}_{PA}\#(\boldsymbol{n})$.

Arithmetization of the Metatheory

- *Illustration*: If the shortest proof of the *Twin Primes Conjecture* in *PA* has more lines than there are fundamental particles in the universe, then there is a sense in which this conjecture is not provable – by us, at least.
- This returns us to the controversial direction of the **Church-Turing Thesis**: is every Turing computable function *really* computable in any theoretically interesting sense?
- Setting aside this matter for another day (next Spring at the Ultrafinitism conference!), let us turn to the central idea of the proof of the First Incompleteness Theorem.

Diagonalization

- It is not clear exactly how exactly Gödel came upon the proof of his Incompleteness Theorems. He later cited his philosophical belief in Frege's and Russell's **platonism** and his repudiation of Hilbert's **formalism** (as well as Carnap's so-called 'conventionalism').
- Platonism is – roughly! -- the view that what we can prove in mathematics is one thing, and what is true is another. Moreover, mathematical truths obtain independent of human minds and languages (just like, most would say, facts about quarks or dinasaurs do). So, platonism allows (but does not require) that **truth outstrips provability** from any (recursive) set of axioms, like *PA*.
- The formalist says that truth and provability are the same thing. There is only truth-relative-to-*PA*, truth-relative-to-*ZFC*, and so forth for any set of axioms. And truth-relative-to-*PA* = provability-from-the-*PA*-axioms. Truth simpliciter, divorced from any **formal system**, makes no sense in mathematics, at least.

Diagonalization

- For the formalist, mathematics is like a game (e.g., chess or Go), and the only factual question is whether you followed the rules. There is no such question of whether the rules (of, say, *PA*) themselves are right. Some may be more useful for a purpose. But the axioms of, say, elliptic geometry and Euclidean geometry are equally legitimate, understood as pure mathematical theories.
  - *Compare*: relativism in ethics.
- Hence, according to the formalist, if *PA* fails to imply either ***X*** or ~***X***, then ***X*** has no truth-value (in the context of number theory).
- Gödel managed to show that key **claims 'about' formal systems that the formalist takes to be factual** are undecidable in those systems -- since they amount, via Gödel numbering, to undecidable number-theoretic claims!
- Therefore, the position that there are objective facts about formal systems (e.g., that they are consistent) but not objective facts about what those systems represent (e.g., numbers) is -- arguably -- incoherent.

Diagonalization

- It is also said that Gödel was thinking about the **liar paradox**. Consider the following sentence:
  - (1) Sentence (1) is false.
- If (1) is false, then what is says is false. But what (1) says is that (1) is false. Thus, if (1) is false, (1) is true, which is a contradiction.
- If (1) is true, then what it says is true. But, again, (1) says is that (1) is false. So, if (1) is true, then (1) is false, which is also a contradiction.
- We will return to this paradox in connection with **Tarski's Theorem**.

Diagonalization

- For now, Gödel's idea was to replace the paradoxical sentence, (1), with the unparadoxical sentence, '(1) Sentence (1) is unprovable in in *T*.', where '*T*' refers to the relevant formal system, like (first-order) *PA*.
- A simple argument for incompleteness stems from the **semantic** assumption that *PA* is **sound**, i.e., only proves truths.
- Construct a sentence, $\mathbf{G}_{PA}$, that codes its own unprovability in *PA*.
- Now suppose that $\mathbf{G}_{PA}$ is provable in *PA*. Then it is false, and *PA* **unsound**.
- So, if *PA* is **sound**, then $\mathbf{G}_{PA}$ is not provable in *PA*. But if $\mathbf{G}_{PA}$ is not provable in *PA*, then $\mathbf{G}_{PA}$ is a truth that is not provable in *PA*.
- Similarly, since *PA* is **sound**, $\sim\mathbf{G}_{PA}$ is a falsehood that is not provable in *PA*.
- Hence, if *PA* is **sound**, then it is **incomplete**, i.e., fails to prove ***S*** or ~***S*** for an ***S*** in *Voc*(*PA*).

Diagonalization

- Gödel did not rest content with this argument. It assumes the falsity of **formalism** and **conventionalism**, which were dominant conceptions of mathematics in the 1920s, under the influence of the Vienna Circle.
- Gödel's task was to prove that *PA* (and any recursively axiomatizable extension of it) was incomplete without relying on 'semantic' ideas, like soundness (which implies truth). We will see that, whether he actually accomplished this is open to dispute. However, J. Barkley Rosser fixed the vulnerability in Gödel's proof, resulting in a uncontroverisally **syntactic argument** for Incompleteness.
- If *Con*(*PA*), then there is a sentence, $\mathbf{G}_{PA}$, such that $PA \nvdash \mathbf{G}_{PA}$, and $PA \nvdash \sim\mathbf{G}_{PA}$. Moreover, we will discover that $PA \vdash \mathbf{G}_{PA} \leftrightarrow Con(PA)$!
- First, however, we will prove Gödel weaker result.

Diagonalization

- The general trick that Gödel exploited is due to philosopher, Rudolf Carnap, who discovered the crucial lemma, as we will see shortly.
- Let ***X***[***z***] be an open **formula** in *Voc*(*PA*) that contains only free variable, ***z***, but no ***z***-quantifiers. Then the **sentence**, ***X***[***k***], where *k* is the Gödel number of the open formula, ***X***[***z***], is called the **diagonalization** of ***X***[***z***].
- Why is ***X***[***k***] called the diagonalization of ***X***[***z***]? Consider the array:

	X_1	X_2	X_3	X_4	…	X_n	…
k_1	$X_1[k_1]$	$X_2[k_1]$	$X_3[k_1]$	$X_4[k_1]$	…	$X_n[k_1]$	…
k_2	$X_1[k_2]$	$X_2[k_2]$	$X_3[k_2]$	$X_4[k_2]$	…	$X_n[k_2]$	…
k_3	$X_1[k_3]$	$X_2[k_3]$	$X_3[k_3]$	$X_4[k_3]$	…	$X_n[k_3]$	…
k_4	$X_1[k_4]$	$X_2[k_4]$	$X_3[k_4]$	$X_4[k_4]$	…	$X_n[k_4]$	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
k_n	$X_1[k_n]$	$X_2[k_n]$	$X_3[k_n]$	$X_4[k_n]$	…	$X_n[k_n]$	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Diagonalization

- In the previous picture, $X_1, X_2, X_3, \ldots, X_n, \ldots$ is a complete list of formulas in *Voc*(*PA*) with only one free variable, and $k_1, k_2, k_3, \ldots, k_n, \ldots$ are the Gödel numbers of $X_1, X_2, X_3, \ldots, X_n, \ldots$, respectively.
- In other words, using our convention: $\forall n \geq 1$, $k_n = [X_n[z]]$.
- Now note that the left-to-right diagonal sequence, $X_1[k_1], X_2[k_2], X_3[k_3], \ldots, X_n[k_n]$ is the sequence of diagonalizations of $X_1, X_2, X_3, \ldots, X_n \ldots$.
- **Theorem 5.4.1**: There is a (total) recursive function, **DIAG**, ℕ into ℕ, such that for every natural number, *k*, if *k* is the Gödel number of a formula ***X***[***z***] in *Voc*(*PA*) with exactly one free variable, ***z***, then **DIAG**(*k*) is the Gödel number (*GN*) of the **diagonalization** of ***X***[***z***] (i.e., the *GN* of ***X***[***k***], where *k* is the *GN* of ***X***[***z***]).
- **If** *k* is not the *GN* of a formula ***X***[***z***] in *Voc*(*PA*) with only one free variable, then **DIAG**(*n*) = *0*.

Diagonalization: Picture

X[*z*]	→	***X***[***k***]
↓		↓
has Godel number *k*		has Godel number *m*
k	→ diag	*m*

Diagonalization: Illustration

- Suppose that the <u>Gödel number</u> of the formula, (***x*** + ***ss0***) = (***ss0*** * ***x***), is *k*.
  - In our notation: **[**(***x*** + ***ss0***) = (***ss0*** * ***x***)**]** = *k*.
- Then the <u>diagonalization</u> of this formula is the <u>sentence</u>: (***k*** + ***ss0***) = (***ss0*** * ***k***).
  - So, writing, **DIAG** for the <u>diagonalization function</u>, we have:
    - **DIAG**(**[**(***x*** + ***ss0***) = (***ss0*** * ***x***)**]**) = **DIAG**(*k*).

Diagonalization: Illustration

- Now let the Gödel number of the diagonalalization of the Gödel number of formula, (***x*** + ***ss0***) = (***ss0*** * ***x***) -- i.e., of (***k*** + ***ss0***) = (***ss0*** * ***k***) -- be *m*.
  - That is: **DIAG**(*k*) = **[**(***k*** + ***ss0***) = (***ss0*** * ***k***)**]** = *m*. So, <*k*, *m*> ∈ **DIAG**.
- Since **DIAG** is recursive, and representable in *Th*(*PA*), there is a formula **diag**(*x*, *y*), with two free variables such that, for all natural numbers *k* and *m*:
  - If **DIAG**(*k*) = *m*, then *PA* |– **diag**(***k***, ***m***), and
  - *PA* |– (∀x)[**diag**(***k***, *x*)→*x* = ***m***].

Carnap Lemma

- **The Diagonalization Lemma (Carnap Lemma)**: If ***W***[*z*] is a formula in *Voc*(*PA*) with only one free variable, ***z***, then there is an sentence, **G**, *Voc*(*PA*) with the property that *PA* |– **G** ←→ ***W***[***g***], where ***g*** = **[G]**.
- What is this theorem saying? *Every one-place formula has a **fixed point***, a point where you 'get out what you put in'. That is, for any formula with one free variable, ***W***[*z*], there is a sentence, **G**, such that **G** is provably equivalent in *PA* to a sentence obtained by applying ***W*** to the Gödel number of **G** itself.
- *Intuition*: Think of the Gödel number, *g*, of **G** as a name of **G**. Then the lemma is saying that *PA* proves that **G** is true just in case a sentence that applies ***W*** to **G** is true. If we imagine that provably equivalent sentences say the same thing, then the lemma says that *PA* proves that **G** says: ***I am W***.

Carnap Lemma: Proof

- 1) Let ***W***[***z***] be any formula in *Voc*(*PA*) with one free variable, ***z***, let **diag**(*x*, *y*) be some formula that represents **DIAG** in *Th*(*PA*), and let ***W***[***y***] be the formula that results from replacing all the occurrences of ***z*** in ***W***[***z***] with occurrences of ***y*** (if ***y*** occurs in ***W***[***z***], use a different variable). Then the formula, (∃y)(**diag**(*x*, *y*) & ***W***[*y*]), contains only ***x*** as a free variable.
- 2) Let **G** be the diagonalization of (∃y)(**diag**(*x*, *y*) & ***W***[*y*]), i.e., (∃y)(**diag(*n*,** *y*) & ***W***[*y*]), where *n* is the Gödel number of (∃y)(**diag**(*x*, *y*) & ***W***[*y*]).
- 3) Let **[G]** = *g*. Then since **[**(∃y)(**diag**(*x*, *y*) & ***W***[*y*])**]** = *n*, **DIAG**(*n*) = *g*.
- 4) From 1) – 3): *PA* |– **diag**(***n***, ***g***) and *PA* |– (∀***v***)(**diag**[***n***, ***v***] → ***v*** = ***g***).

Carnap Lemma: Proof

- 5) Hence, there are *PA* derivations $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$ of **diag**(***n***, ***g***) and of (∀***v***)(**diag**[***n***, ***v***] → ***v*** = ***g***), respectively, from *PA*. So, let Σ_1 and Σ_2 be the finite subsets of *PA* that occur in $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$, respectively, so that Σ_1 |– **diag**(***n***, ***g***) and Σ_2 |– (∀***v***)(**diag**[***n***, ***v***] → ***v*** = ***g***).
- 6) We can now union the two finite subsets of *PA* from which **diag**(***n***, ***g***) and (∀***v***)(**diag**[***n***, ***v***] → ***v*** = ***g***) follow, respectively, to get another finite subset of *PA* from which they both follow – i.e., $\Sigma_1 \cup \Sigma_2 = \Sigma$, and Σ |- **diag**(***n***, ***g***) and Σ |- (∀***v***)(**diag**[***n***, ***v***] → ***v*** = ***g***).
- 7) Using $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$, both from Σ, there exists a derivation of **G** ←→ ***W***[***g***] from Σ. Here is one:

Carnap Lemma: Proof

[0	0	Σ	Premises (recall that Σ is a finite subset of PA)
	D_1		This derivation is given
	i	**diag[n, g]**	Conclusion of D_1
	D_2		This derivation is given
	j	$(\forall v)$(**diag[n,** v**]** → v = **g**)	Conclusion of D_2
[1	j+1	$(\exists y)$(**diag[n,** y**]** ∧ **W[**y**]**)	CPA (this is G)
[2	j+2	**diag[n, t]**∧**W[t]**	EIA (**t** is a PL name that does not occur in any member of Σ nor does it occur in $(\exists y)$(**diag[n,** y**]**∧**W[**y**]**). Since Σ is finite, there is always such a name.)
	j+3	**diag[n, t]**	j+2, Simp
	j+4	**diag[n, t]** → **t** = **g**	j, UI
	j+5	**t** = **g**	j+3, j+4, MP
	j+6	**W[t]**	j+2, Simp
2]	j+7	**W[g]**	j+5, j+6, Sub (since **t** does not occur in **W[**y**]**, **W[g]** may be considered as obtained from **W[**y**]** by replacing y with **g**.)

Carnap Lemma: Proof

1]	j+8	**W[g]**	(j+2) – (j+7), EI
	j+9	$(\exists y)$(**diag[n,** y**]** ∧ **W[**y**]**) → **W[g]**	(j+1) – (j+8), CP
[3	j+10	**W[g]**	CPA
	j+11	**diag[n, g]**	i, Reit
	j+12	**diag[n, g]** ∧ **W[g]**	j+10, j+11, Conj
3]	j+13	$(\exists y)$(**diag[n,** y**]**∧**W[**y**]**)	j+12, EG
	j+14	**W[g]** → $(\exists y)$(**diag[n,** y**]**∧**W[**y**]**)	(j+10) – (j+13), CP
	j+15	(($\exists y$)(**diag[n,** y**]** ∧ **W[**y**]**) → **W[g]**)∧(**W[g]** → $(\exists y)$(**diag[n,** y**]**∧**W[**y**]**))	j+9, j+14, Conj
0]	j+16	$(\exists y)$(**diag[n,** y**]**∧**W[**y**]**) ↔ **W[g]**	j+15, Bc

9. From 2, 6 and 8: Since Σ is a finite subset of PA, PA ⊢ $(\exists y)$(**diag[n,** y**]**∧**W[**y**]**) ↔ **W[g]**, that is, PA ⊢ G ↔ **W[g]**, where g is the gödel number of G.

First Incompleteness Theorem

- **Definition**: A *PL* set Σ is **ω-consistent** just in case it is not the case that there is a formula ***X***[***z***] composed of *Voc*(Σ) with one free variable, ***z*** such that Σ |– ~***X***[***n***], ∀*n* ∈ ℕ, but also Σ |– (∃***z***)***X***[***z***].
- *Note*: If Σ is **ω-consistent** then Σ is consistent, but not conversely. Equivalently, if Σ is inconsistent, then it is **ω-inconsistent**, but not conversely.
- **Gödel's First Incompleteness Theorem**: If *Th*(*PA*) is ω-consistent, then it is incomplete.
- **Proof**:
- 1) Assume that *Th*(*PA*) is **ω-consistent**.
- 2) The relation **PROOF** is recursive, so it is representable in *Th*(*PA*), and there is a formula in *Voc*(*PA*) **proof**(*x, y*) with two free variables that represents **PROOF** in *Th*(*PA*).

First Incompleteness Theorem

- 3) Let **Thrm**(*y*) be the formula in *Voc*(*PA*), (∃*x*)**proof**(*x*, *y*), with one free variable, ***y***.
- 4) By the **Diagonalization (Carnap) Lemma**, there is a sentence in *Voc*(*PA*), $\mathbf{G}_{PA}$, such that *PA* |– $\mathbf{G}_{PA}$ ←→ ~**Thrm**(***g***)**,** where *g* is the Gödel number of $\mathbf{G}_{PA}$. ('*PA* proves: $\mathbf{G}_{PA}$ says that $\mathbf{G}_{PA}$ is not a theorem of *PA*.')
- 5) Assume for *reductio* that *PA* |- $\mathbf{G}_{PA}$.
- 6) Then there is a derivation, ***D***, of $\mathbf{G}_{PA}$ from *PA* with some Gödel number, *d*.
- 7) If *g* = **[**$\mathbf{G}_{PA}$**]**, then since *d* = **[*D*]**, the ordered pair <*d*, *g*> ∈ **PROOF**.
- 8) Since $\mathbf{PROOF}_{\mathrm{PA}}$ is representable in *Th*(*PA*), *PA* |– $\mathbf{proof}_{\mathrm{PA}}$**(*d, g*)** .

First Incompleteness Theorem

- 9) By 4) *PA* |– ~$\mathbf{Thrm}_{PA}(\boldsymbol{g})$.
- 10) From the definition of $\mathbf{Thrm}_{PA}(x)$, *PA* |– ~$(\exists x)\mathbf{proof}_{PA}(x, \boldsymbol{g})$.
- 11) Equivalently, *PA* |– $(\forall x)$~$\mathbf{proof}_{PA}(x, \boldsymbol{g})$.
- 12) So, by Universal Instantiation in *PA*, *PA* |– ~$\mathbf{proof}_{PA}(\boldsymbol{d}, \boldsymbol{g})$.
- 13) From 8) and 12), ~*Con*(*PA*), so, *a fortiori*, *PA* is not **ω-consistent**.
- 14) Hence, the first *reductio* assumption is false – i.e., *PA* ⊬ $\mathbf{G}_{PA}$.
- 15) Now assume, for the second *reductio* assumption, that *PA* |- ~$\mathbf{G}_{PA}$.
- 16) Since *PA* |– $\mathbf{G}_{PA} \leftrightarrow$ ~$\mathbf{Thrm}_{PA}(\boldsymbol{g})$, *PA* |- $\mathbf{Thrm}_{PA}(\boldsymbol{g})$.

First Incompleteness Theorem

- 17) By the definition of $\mathbf{Thrm}_{PA}(y)$, *PA* |- $(\exists x)\mathbf{proof}_{PA}(x, \boldsymbol{g})$.
- 18) Since *PA* ⊬ $\mathbf{G}_{PA}$, and $g = [\mathbf{G}_{PA}]$, $\langle m, g\rangle \notin \mathbf{PROOF}_{PA}$, $\forall m \in \mathbb{N}$.
- 19) Since $\mathbf{PROOF}_{PA}$ is representable in *PA*, *PA* |- ~$\mathbf{proof}_{PA}(\boldsymbol{m}, \boldsymbol{g})$, $\forall m \in \mathbb{N}$.
- 20) So, by 16) and 19), *PA* |- ~$\mathbf{proof}_{PA}(\boldsymbol{m}, \boldsymbol{g})$, $\forall m \in \mathbb{N}$ and *PA* |- $(\exists x)\mathbf{proof}_{PA}(x, \boldsymbol{g})$.
- 21) Since, $PA \subseteq Th(PA)$, *Th*(*PA*) is not **ω-consistent**, contrary to 1).
- 22) So, by *reductio ad absurdum*, it is also not the case that *PA* |- ~$\mathbf{G}_{PA}$.
- *Upshot*: If *Th*(*PA*) is ω-consistent, then *PA* ⊬ $\mathbf{G}_{PA}$ and *PA* ⊬ ~$\mathbf{G}_{PA}$.

First Incompleteness Theorem

- Assuming that PA is ω-consistent (and, hence, consistent), the **Completeness Theorem** (in the form of the **Model Existence Theorem**) promises the existence there are two models of *PA*.
- One of these is a model of *PA* + $\mathbf{G}_{PA}$ while the other is a model of *PA* + ~$\mathbf{G}_{PA}$.
- Hence these two models are <u>not</u> **elementarily equivalent**.
- *A fortiori*, the models are <u>not</u> **isomorphic**.
- Therefore, *Th*(*PA*) is <u>not</u> **categorical**.
- Moreover, by the **Lowenheim-Skolem Theorem**, every theory has a **countable** model. So, *PA* is <u>not</u> $\boldsymbol{\aleph_0}$**-categorical**.

Rosser's Improvement of the First Theorem

- We only needed to assume that *Con*(*PA*) to argue that $PA \nvdash \mathbf{G}_{PA}$. But we had to assume **ω-consistent** in order to argue that $PA \nvdash {\sim}\mathbf{G}_{PA}$.
- Is the assumption of **ω-consistency** avoidable? Yes, as Rosser showed.
- We first define a *1*-place function, **NEG**, from the Gödel number of ***X*** to the Gödel number of ~***X.***
- Second, we define the *2*-place relation **DISPROOF** as follows: $\forall n,m \in \mathbb{N}$, $\langle n, m\rangle \in$ **DISPROOF** just in case $\langle n,$ **NEG**$(m)\rangle \in$ **PROOF**.
- **DISPROOF** is recursive because both **NEG** and **PROOF** are.
- Let **proof**(*x*, *y*), **disproof**(*x*, *y*) and (*x* **<** *y*) <u>represent</u> **PROOF**, **DISPROOF**, and < in *Th*(*PA*), respectively.

Rosser's Improvement of the First Theorem

- Then the following formula contains only the <u>one</u> free variable, *y*:
  - $(\forall x)(\mathbf{proof}(x, y)) \rightarrow (\exists z)(z < x \;\&\; \mathbf{disproof}(z, y)))$
- By the **Diagonalization Lemma**, there is a <u>sentence</u>, $\mathbf{R}_{PA}$, called a **Rosser Sentence**, with Gödel number, *r*, such that:
  - $PA \vdash \mathbf{R}_{PA} \leftrightarrow (\forall x)(\mathbf{proof}(x, \boldsymbol{r})) \rightarrow (\exists z)(z < x \;\&\; \mathbf{disproof}(\mathrm{z}, \boldsymbol{r})))$
- What does $\mathbf{R}_{PA}$ 'say'? Roughly: that *if* there is a <u>derivation</u> of $\mathbf{R}_{PA}$ from *PA*, *then* there is an <u>earlier</u> derivation of $\sim\mathbf{R}_{PA}$ from *PA* (where the <u>order</u> in question concerns that <u>sizes of the Gödel numbers</u> of the derivations).
- One can now prove that if *Con*(*PA*), then $PA \nvdash \mathbf{R}_{PA}$ and $PA \nvdash \sim\mathbf{R}_{PA}$.

Rosser's Improvement of the First Theorem

- We will not give a rigorous proof of Rosser's improved version of the First Incompleteness Theorem. But the basic idea is straightforward.
- $\mathbf{R}_{PA}$ promises that it is not provable in *PA* <u>before</u> $\sim\mathbf{R}_{PA}$ (in the ordering).
- So, suppose that *PA* is (merely) <u>consistent</u>, and that $PA \vdash \mathbf{R}_{PA}$. Then there is a derivation of $\mathbf{R}_{PA}$, and no earlier derivation of $\sim\mathbf{R}_{PA}$ (on pain of inconsistency).
- However, a derivation of $\mathbf{R}_{PA}$ along with all those ordered before it would amount to a derivation of $\boldsymbol{R}_{PA}$ *is provable <u>before</u>* $\sim\boldsymbol{R}_{PA}$ -- and thus of $\sim\mathbf{R}_{PA}$ <u>itself</u>. This contradicts the assumption that *Con*(*PA*).
- Suppose, then, that $PA \vdash \sim\mathbf{R}_{PA}$. Then there is a derivation of $\sim\mathbf{R}_{PA}$, and no <u>earlier</u> derivation of $\mathbf{R}_{PA}$. A derivation of $\sim\mathbf{R}_{PA}$ along with all those ordered before it would amount to a derivation that $\boldsymbol{R}_{PA}$ *is not provable in PA before* $\sim\boldsymbol{R}_{PA}$, i.e., that $\mathbf{R}_{PA}$. This again contradicts the assumption that *Con*(*PA*).

Is $\mathbf{G}_{PA}$ *true*?

- It is routinely said (even by experts) that Gödel's theorems demonstrate that there are truths of arithmetic that are not provable in any (recursively) axiomatizable formal system, like *Peano Arithmetic* – or, more carefully, that there are such truths if the system is consistent.
- Every such system has a Gödel sentence, like $\mathbf{G}_{PA}$, and these seem true.
- But this is actually too quick for reasons to which we will return. What is hard to deny is that $\mathbf{G}_{PA}$ is true in *in the model,* $\mathbb{N}$, if *Con*(*PA*).
- **Theorem 5.4.2**: $\mathbf{G}_{PA}$ is *true* in $\mathbb{N}$.
- **Proof**:
- 1) Suppose that $\mathbb{N} \models PA$.
- 2) By the Carnap Lemma, $PA \vdash \mathbf{G}_{PA} \leftrightarrow \sim\mathbf{Thrm}_{\mathrm{PA}}(\boldsymbol{g})$.

Is $\mathbf{G}_{PA}$ *true*?

- 3) By the definition of $\mathbf{Thrm}_{\mathrm{PA}}(x)$, $PA \vdash \mathbf{G}_{PA} \leftrightarrow \sim(\exists x)\mathbf{proof}_{\mathrm{PA}}(x, \boldsymbol{g})$.
- 4) By the Soundness Theorem, $PA \models \mathbf{G}_{PA} \leftrightarrow \sim(\exists x)\mathbf{proof}_{\mathrm{PA}}(x, \boldsymbol{g})$.
- 5) So, in particular, $\mathbb{N} \models \mathbf{G}_{PA} \leftrightarrow \sim(\exists x)\mathbf{proof}_{\mathrm{PA}}(x, \boldsymbol{g})$.
- 6) By the First Incompleteness Theorem, $PA \nvdash \mathbf{G}_{PA}$, i.e. there is no derivation of $\mathbf{G}_{PA}$ from the Peano Axioms.
- 7) Hence, $\forall m \in \mathbb{N}$, $\langle m, g \rangle \notin \mathbf{PROOF}_{\mathrm{PA}}$.
- 8) Since $\mathbf{PROOF}_{\mathrm{PA}}$ is recursive, and represented in *PA* by $\mathbf{proof}(x, y)$, $\forall m \in \mathbb{N}$, $PA \vdash \sim\mathbf{proof}_{\mathrm{PA}}(\boldsymbol{m}, \boldsymbol{g})$.
- 9) So, again by Soundness, $\forall m \in \mathbb{N}$, $PA \models \sim\mathbf{proof}_{\mathrm{PA}}(\boldsymbol{m}, \boldsymbol{g})$, and, hence, $\mathbb{N} \models \sim\mathbf{proof}_{\mathrm{PA}}(\boldsymbol{m}, \boldsymbol{g})$.

Is $\mathbf{G}_{PA}$ *true*?

- 10) So, $\mathbb{N} \models \sim\mathbf{proof}_{\mathrm{PA}}(\boldsymbol{t}, \boldsymbol{g})$ for every name ***t*** in *LN* (since, on $\mathbb{N}$, we explicitly correlated *LN* with the natural numbers), i.e., $\mathbb{N} \models \sim(\exists x)\mathbf{proof}_{\mathrm{PA}}(x, \boldsymbol{g})$.
- 11) Since, $\mathbb{N} \models \mathbf{G}_{PA} \leftrightarrow \sim(\exists x)\mathbf{proof}_{\mathrm{PA}}(x, \boldsymbol{g})$, we have that $\mathbb{N} \models \mathbf{G}_{PA}$.

- What is tendentious about the claim that $\mathbf{G}_{PA}$ is *true simpliciter*?
- That $\mathbb{N} \vdash PA$! For all Gödel's Theorem says, we may have $\sim Con(PA)$!
- What if we assume that $Con(PA)$, or that *PA* is ω-consistent?
- Still, we only get something negative about what *PA* implies: that $PA \nvdash \mathbf{G}_{PA}$ and $PA \nvdash \sim\mathbf{G}_{PA}$.
- What if we add, not only that $Con(PA)$, but that $\mathbb{N} \models PA$? This only shows that $\mathbb{N} \models \mathbf{G}_{PA}$ assuming that our natural numbers are those of $\mathbb{N}$ – that is, that we are 'living' in the Standard Model. By **Skolem's Paradox**, this needs argument!

A Potpourri of Implications

- **Theorem 5.5.1**: $Th(PA) \subset Th_{PA}(\mathbb{N})$.

- **Proof**:
- 1) $Th_{PA}(\mathbb{N}) = \{\boldsymbol{X}$: is a sentence in $Voc(PA)$ such that $\mathbb{N} \models \boldsymbol{X}\}$.
- 2) Hence, by **Theorem 5.4.2**, $G_{PA} \in Th_{PA}(\mathbb{N})$.
- 3) But by the First Incompleteness Theorem, $\mathbf{G}_{PA} \notin Th(PA)$.
- 5) So, since $\mathbb{N} \models Th(PA)$, $Th(PA) \subset Th_{PA}(\mathbb{N})$.

A Potpourri of Implications

- **Theorem 5.5.2**: If Σ is a consistent *PL* theory in which all recursive functions are representable, then the set of the Gödel numbers of the sentences in Σ is **not** representable in Σ.
- **Proof**:
- 1) Suppose that the antecedent is true – i.e., that *Con*(Σ), that Σ is a theory, and that Σ represents all recursive functions.
- 2) Since Σ is a theory, Σ = {***X*** : ***X*** is in *Voc*(Σ) and Σ |- ***X***} = {***X*** : Σ |= ***X***}.
- 3) Let GN_{Σ} = {*n* : *n* is the *GN* of ***X*** such that ***X*** ∈ Σ} = {**[*X*]** : ***X*** ∈ Σ}.
- 4) Suppose for *reductio* that GN_{Σ} is representable in Σ.

A Potpourri of Implications

- 4) Then there is a formula ***X***[*z*] in *Voc*(Σ) with one free variable such that, for every natural number, *k*, if *k* ∈ GN_{Σ}, then Σ |– ***X***[***k***], and if *k* ∉ GN_{Σ}, then Σ |– ~***X***[***k***].
- 5) Since **DIAG** is recursive, it is representable in Σ (by 1).
- 6) By the proof of the Diagonalization Lemma, there is a sentence $\mathbf{G}_{\Sigma}$ in *Voc*(Σ) such that Σ |– $\mathbf{G}_{\Sigma}$ ←→ ~***X***[***g***], where *g* is the Gödel number of $\mathbf{G}_{\Sigma}$ (and ***X***[*z*] is the formula representing GN_{Σ} in Σ).
- 7) Now suppose for *reductio* that Σ ⊬ $\mathbf{G}_{\Sigma}$.
- 8) By the definitions of GN_{Σ} and $\mathbf{G}_{\Sigma}$, *g* ∉ GN_{Σ} (since *g* = [$\mathbf{G}_{\Sigma}$]).

A Potpourri of Implications

- 9) Since $\boldsymbol{X}[z]$ represents GN_Σ in Σ, Σ |- ~$\boldsymbol{X}[\boldsymbol{g}]$.
- 10) So, by 6), Σ |– $\mathbf{G}_\Sigma$, which contradicts 7).
- 11) Hence, by *reductio*, Σ |– $\mathbf{G}_\Sigma$, and $g \in GN_\Sigma$ (since g = **[**$\mathbf{G}_\Sigma$**]**).
- 12) Since $\boldsymbol{X}[z]$ represents GN_Σ in Σ, Σ |- $\boldsymbol{X}[\boldsymbol{g}]$, and, so, Σ |- ~~$\boldsymbol{X}[\boldsymbol{g}]$.
- 13) But, then, by 6), Σ |– ~$\mathbf{G}_\Sigma$, and ~*Con*(Σ), by 12), contrary to 1).
- 14) So, our first *reductio* assumption is false: GN_Σ is *not* representable in Σ.

A Potpourri of Implications

- *Note*: It follows from **Theorem 5.5.2** that $\mathbf{THEOREM}_{PA}$ (i.e., the set of *GN*s of theorems of *PA*) is not recursive, if *Th*(*PA*) is consistent. *Th*(*PA*) represents all recursive functions. So, the fact that it fails to represent $\mathbf{THEOREM}_{PA}$ shows that this set is not recursive.
- **Theorem 5.5.3**: If Σ is a consistent *PL* theory in which all recursive functions are representable, then Σ is undecidable.
- **Proof**:
- 1) Let Σ be a consistent theory in which all recursive functions are representable.
- 2) Again, let us write GN_Σ for the set, {*n* : *n* is the *GN* of $\boldsymbol{X}$ such that $\boldsymbol{X} \in \Sigma$} = {**[*X*]** : $\boldsymbol{X} \in \Sigma$}, and assume for *reductio* that Σ *is* decidable.

A Potpourri of Implications

- 3) Then there is an effective decision procedure, $\boldsymbol{D}_{\Sigma}$, for deciding membership in Σ.
- 4) Using $\boldsymbol{D}_{\Sigma}$, we can construct a decision procedure for membership in GN_{Σ}.
  - $\forall k \in \mathbb{N}$, check whether or not $k = [\boldsymbol{X}]$ for a sentence $\boldsymbol{X}$ in *Voc*(Σ).
  - If k is not the Gödel number of a sentence in *Voc*(Σ), $k \notin GN_{\Sigma}$.
  - If k is the Gödel number of a sentence in *Voc*(Σ), $k \in GN_{\Sigma}$, apply $\boldsymbol{D}_{\Sigma}$ to $\boldsymbol{X}$.
  - If $\boldsymbol{X} \in \Sigma$, $k \in GN_{\Sigma}$, and if $\boldsymbol{X} \notin \Sigma$, $k \notin GN_{\Sigma}$.
- 5) By Church's Thesis (invoked for simplicity only!), the characteristic function of GN_{Σ} is Turing computable, and, hence, recursive.

A Potpourri of Implications

- 6) Since all recursive functions are representable in Σ, all recursive sets and relations are also representable in Σ.
- 7) So, by 5), GN_{Σ} is representable in Σ, contradicting **Theorem 5.5.2**.
- 8) Thus, by *reductio ad absurdum*, Σ is undecidable.

- **Theorem 5.5.4**: If Σ is a consistent axiomatizable *PL* theory in which all recursive functions are representable, then Σ is incomplete.
- **Proof**: If Σ is a consistent *PL* theory that represents all recursive functions, then, by **Theorem 5.5.3**, Σ is undecidable. However, we learned from **Theorem 3.5.1** (that every complete axiomatizable *PL* theory is *decidable*) that if Σ is complete and axiomatizable, then it must be decidable. So, assuming that Σ *is* axiomatizable, it be incomplete (because it is undecidable).

A Potpourri of Implications

- **Theorem 5.5.5**: No consistent extension of *Th*(*PA*) is decidable (where a consistent extension of Γ is any set Γ* of sentences in *Voc*(Γ) such that *Con*(T*) and Γ ⊆ Γ*).
- **Proof**: All recursive functions are representable in *Th*(*PA*), and, hence, in any consistent extension of *Th*(*PA*), *Th*(*PA**).
- Therefore, all recursive functions are representable in *Th*(*PA*)*.
- So, by **Theorem 5.5.3**, *Th*(*PA**) is undecidable.
- **Corollary**: Both *Th*(*PA*) and $Th_{PA}(\mathbb{N})$ are undecidable, since each is a consistent extension of *Th*(*PA*).

A Potpourri of Implications

- **Theorem 5.5.6:** *Th*(*PA*) is semidecidable, i.e., recursively enumerable.
- **Proof**:
- 1) Recall that $\mathbf{PROOF}_{PA}$ is the set of Gödel numbers of all *PA* proofs, i.e., *m* ∈ $\mathbf{PROOF}_{PA}$ just in case there is a *PL* derivation from *PA,* ***D***, and *GN*(***D***) = *m*.
- 2) Since (by definition) membership in $\mathbf{PROOF}_{PA}$ is decidable, we can arrange all such ***D*** according to the magnitude of their Gödel numbers (as with Rosser's Theorem).
- 3) By the definition of theorem, for or any sentence in *Voc*(*PA*), ***Z***, ***Z*** ∈ *Th*(*PA*) just in case ∃***m*** ∈ $\mathbf{PROOF}_{PA}$ such that ***Z*** is the last sentence of the derivation that it numbers, ***D***.

A Potpourri of Implications

- 3) Hence, by sequentially checking the members of **PROOF**$_{PA}$, we can find a proof of ***Z*** whenever ***Z*** a theorem of *PA*, enumerating ***Z*** in turn.
- 4) By contrast, ~∃*m* ∈ **PROOF**$_{PA}$ such that *GN*(***D***) = *m* and the last line of ***D*** is ***Z***, then this sequential check will never terminate.
- 5) So, by the definition of semidecidability (recursive enumerability), *Th*(*PA*) is semidecidable, i.e., recursively enumerable.

- **Theorem 5.5.7** (**Incompletability Theorem**): Arithmetic, i.e., $Th_{PA}(\mathbb{N})$, is not axiomatizable.

A Potpourri of Implications

- **Proof**: Immediate from **Theorem 3.5.1** (that any *PL* theory that is complete and axiomatizable is decidable). By **Theorem 5.5.5** $Th_{PA}(\mathbb{N})$ is undecidable. So, since $Th_{PA}(\mathbb{N})$ is complete, it is not axiomatizable.
- *Note*: We have now proved what we earlier anticipated: the triad of (1) **completeness**, (2) **axiomatizability**, and (3) **undecidability** is inconsistent.
- **Theorem 5.5.8**: For any decidable set Ω of sentences in the full *Voc*(*PA*), if $\Omega \subseteq Th_{PA}(\mathbb{N})$, then *Th*(Ω) is incomplete.
- **Proof**: Suppose that *Th*(Ω) is complete. Then $Th(\Omega) = Th_{PA}(\mathbb{N})$. Since Ω is decidable, $Th_{PA}(\mathbb{N})$ would be axiomatizable. This is impossible, by **Theorem 5.5.7**.

Presburger Arithmetic

- *Note*: For the previous theorem, it is essential that the vocabulary of *Th*(Ω) encompasses the *full* vocabulary of *Th*(*PA*), including **+** and *****.
- It turns out that one can construct a complete and axiomatizable (and thus, we know, decidable) arithmetic theory without multiplication.
- One such theory has the standard axioms for successor and addition and induction for all its formulas. But it cannot define, *a fortiori* prove, its own consistency. The theory is **Presburger Arithmetic**.

Definability in ℕ

- We introduced the concept of **definability** when describing some implications of the Compactness Theorem (which itself, recall, is a consequence of the Soundness and Completeness Theorems).
- A property, **F** is **definable** simpliciter just in case there is a formula, Φ, such that, for every model, Φ is true of all and only the **F** things.
  - *Example*: Being 5-membered is definable, but being finite is not.
- It is more common to consider *definability in a fixed model*, *M***.**
- A property or relation (set of *D* of *n*-tuples) is definable in the Standard Model of arithmetic, ℕ, just in case there is an *n*-variable formula in *Voc*(*PA*), $\boldsymbol{D}[x_1, x_2, x_3, \ldots, x_n]$ such that for all natural numbers $n_1, n_2, n_3, \ldots, n_n$, $< n_1, n_2, n_3, \ldots, n_n > \in D$ just in case $\mathbb{N} \models \boldsymbol{D}[\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3, \ldots, \boldsymbol{n}_n]$.

Definability in ℕ

- We discovered that a great deal of arithmetic properties (sets and functions) are definable in ℕ, and recursive ones are even representable.
- *Example 1*: Consider $\mathbf{PROOF}_{PA}$, i.e., the set of ordered pairs of Gödel numbers <*m*, *n*>, such that *m* is the Gödel number of a proof from *PA* whose conclusion has Gödel number *n*. This is clearly definable in ℕ.
- There is a formula, $\mathbf{proof}_{PA}(x, y)$, such that $\mathbf{proof}_{PA}(\boldsymbol{m}, \boldsymbol{n})$ is true in ℕ of all and only the pairs, <*m*, *n*>, where *m* is the Gödel number of a proof from *PA* whose conclusion has Gödel number *n*.
- Moreover, if <*m*, *n*> ∈ $\mathbf{PROOF}_{PA}$, then *PA* |- $\mathbf{proof}_{PA}(\boldsymbol{m}, \boldsymbol{n})$, and if <*m*, *n*> ∉ $\mathbf{PROOF}_{PA}$, then *PA* |- ~$\mathbf{proof}_{PA}(\boldsymbol{m}, \boldsymbol{n})$. So, $\mathbf{PROOF}_{PA}$ is representable too.

Definability in ℕ

- *Example 2*: Now consider prov*ability*, i.e., theoremhood, in *PA*. We know that this is not representable in *PA*, so is not recursive. But it is definable in ℕ. There is a formula, $(\exists x)\mathbf{proof}_{PA}(x, y)$, such that $(\exists x)\mathbf{proof}_{PA}(x, \boldsymbol{n})$ is true in ℕ -- ℕ |= $(\exists x)\mathbf{proof}_{PA}(x, \boldsymbol{n})$ -- when there is a natural number that is the Gödel number of a proof of a sentence with Gödel number, *n*. Likewise, $(\exists x)\mathbf{proof}(x, \boldsymbol{n})$ is false in ℕ when there is no such number.
- We say that *PL* in *Voc*(*PA*) can define the property of theoremhood in ℕ.
- *Note*: Provability-in-*PA* is representable in **True Arithmetic**, Th_{PA}(ℕ). It is definable in ℕ, and representability and definability are the same in Th_{PA}(ℕ).
- In general, many **syntactic** properties (like being a sentence, proof, or term) are recursive, so are both representable and definable in ℕ. Other such properties, like provability, which are just recurisively enumerable and not representable, are still definable in ℕ. *Question*: What about **semantic** properties, like **truth**?

Definability in ℕ

- **Undefinability of Truth Theorem**: The set of Gödel numbers of sentences, ***X***, in *Voc*(*PA*) such that ℕ |= ***X***, written #Th_{PA}(ℕ), is not definable in ℕ ("arithmetical truth is not arithmetically definable").
- 1) Th_{PA}(ℕ) is a theory, so for every ***X***, in *Voc*(*PA*), ***X*** ∈ Th_{PA}(ℕ) if/f Th_{PA}(ℕ) |= ***X*** if/f Th_{PA}(ℕ) |- ***X*** (by the **Completeness Theorem**).
- 2) By the definition of Th_{PA}(ℕ), for every every ***X*** in *Voc*(*PA*), ℕ |= ***X*** if/f Th_{PA}(ℕ) |- ***X***.
- 3) A set, *D* ⊆ ℕ, is definable in ℕ if/f there is a *1*-variable formula in *Voc*(*PA*), ***D***, such that for all natural numbers *n*, *n* ∈ *D* if/f ℕ |= ***D***[***n***].

Definability in ℕ

- 4) So, *D* is definable in ℕ if/f there is a *1*-variable formula in *Voc*(*PA*), ***D***, such that for all natural numbers *n*, if *n* ∈ *D* then Th_{PA}(ℕ) |– ***D***[***n***], and if *n* ∉ *D* then Th_{PA}(ℕ) |– ~***D***[***n***].
- 5) That is, *D* is definable in ℕ if/f *D* is representable in Th_{PA}(ℕ).
- 6) Since *Th*(*PA*) ⊂ Th_{PA}(ℕ), and since all recursive functions are representable in *Th*(*PA*), all recursive functions are representable Th_{PA}(ℕ).
- 7) But Th_{PA}(ℕ) is also consistent (by definition), so **Theorem 5.5.2** precludes that #Th_{PA}(ℕ) is representable in Th_{PA}(ℕ).
- 8) So, by 5), #Th_{PA}(ℕ) is not definable in ℕ either.

Robinson Arithmetic, *Th*(***Q***)

- We have seen (**Theorem 5.5.3**) that any consistent *PL* theory representing all recursive functions is undecidable and (**Theorem 5.5.4**) incomplete. How weak can a theory be while still representing all recursive functions? *Very*! **Robinson Arithmetic** suffices.
- **Robinson Arithmetic** *Th*(***Q***) is the closure of the *Peano Axioms* minus all instances of **IS**, plus one axiom:

- $\mathbf{Ax_1}$ $(\forall x)\ \boldsymbol{0} \neq \boldsymbol{s}(x)$
- $\mathbf{Ax_2}$ $(\forall x)(\forall y)(\boldsymbol{s}(x) = \boldsymbol{s}(y) \rightarrow x = y)$
- $\mathbf{Ax_3}$ $(\forall x)(x + \boldsymbol{0}) = x$

...

Robinson Arithmetic, *Th*(***Q***)

- $\mathbf{Ax_4}$ $(\forall x)(\forall y)(x + \boldsymbol{s}(y)) = \boldsymbol{s}(x + y)$
- $\mathbf{Ax_5}$ $(\forall x)(x * \boldsymbol{0}) = \boldsymbol{0}$
- $\mathbf{Ax_6}$ $(\forall x)(\forall y)(x * \boldsymbol{s}(y)) = ((x * y) + x)$

- +

- $\mathbf{Ax_7}$ $(\forall x)(x \neq \boldsymbol{0} \rightarrow (\exists y)x = \boldsymbol{s}y)$

From *Q* to the Church-Turing Theorem

- Not only is *Q* exceptionally weak, it is **finitely-axiomatized**. Hence, we can conjoin its axioms into a single sentence, $\boldsymbol{Q}_{\mathbf{AX}}$. If the set, $\boldsymbol{T_Q}$, of conditionals, ($\boldsymbol{Q}_{\mathbf{AX}} \rightarrow \boldsymbol{P}$), where ***P*** is arbitrary, is undecidable, then the set of logical truths, *Th*(∅) must be too – since $\boldsymbol{T_Q} \subset Th(\emptyset)$.
- **Lemma 5.5.1** (**Deduction Theorem**): For any *PL* sentences ***Z*** and ***W***, ***Z*** |– ***W*** just in case |– ***Z*** → ***W***.
- **Proof** (left-to-right):
- 1) Assume that ***Z*** |– ***W***, i.e., that there is a *PL* derivation of ***W*** from ***Z***.

From *Q* to the Church-Turing Theorem

- 2) Then, using Conditional Proof (*CP*), we can obtain a derivation, ***D***, of (***Z*** → ***W***) from the empty set, ∅, i.e., |– ***Z*** → ***W***.
- 3) So, by Conditional Proof in the *metatheory*, for any *PL* sentences ***Z*** and ***W***, if ***Z*** |– ***W*** then |– ***Z*** → ***W***.
- **Proof** (right-to-left):
- 4) Now assume that |– ***Z*** → ***W***, i.e. that there is a *PL* derivation, ***D****, of ***W*** → ***Z*** from the empty set, ∅.
- 5) Now, with ***Z*** as the only premise, use ***D**** to derive the conclusion ***W***, by *modus ponens*. This establishes that ***Z*** |– ***W***.
- 6) So, again by Conditional Proof, if |– ***Z*** → ***W***, then ***Z*** |– ***W***.

From *Q* to the Church-Turing Theorem

- We are now in a position to prove another landmark of logic:

- **Church-Turing Theorem**: *Th*(∅) is undecidable.
- **Proof**:
- 1) By the Deduction Theorem, for every *PL* sentence, ***X***, |- ($\boldsymbol{Q}_{\mathbf{AX}}$ → ***X***) just in case $\boldsymbol{Q}_{\mathbf{AX}}$ |- ***X***.
- 2) So, for every sentence, ***X***, in *Voc*(*PL*), ***X*** ∈ *Th*(***Q***) just in case ($\boldsymbol{Q}_{\mathbf{AX}}$ → ***X***) ∈ *Th*(∅).
- 3) Now suppose for *reductio* that *Th*(∅) is **decidable**.
- 4) Then *Th*(***Q***) is decidable.
- 5) But *Th*(***Q***) is represents all recursive functions and its axiomatizable, so is undecidable by **Theorem 5.5.4**.
- 6) Hence, the *reductio* assumption is false: *Th*(∅) is **undecidable**.

Recursive Enumerability of *Th*(∅)

- While *Th*(∅) is undecidable, it must be **semidecidable**, i.e., **recursively enumerable**, because any theory is closed under logical consequence, so must **weakly represent** the set of logical truths.
- For any ***X*** ∈ *Th*(∅), there is a *PL* derivation, ***D***, of ***X*** from ∅. Let ***LD*** = {***D***: ***D*** is a *PL* derivation from ∅}. Then ***LD*** is a decidable set. So, to check whether ***X*** ∈ *Th*(∅), we wait to see ***X*** or ~***X*** as the last line of some ***D*** ∈ ***LD***, concluding that ***X*** ∈ *Th*(∅) or ***X*** ∉ *Th*(∅), respectively.
- But if ***X*** ∉ *Th*(∅), and not contradictory, we will wait forever. So, this procedure returns the answer **Yes** just in case ***X*** ∈ *Th*(∅). But it might not return an answer if the correct answer is **No**, i.e., if ***X*** ∉ *Th*(∅).

Summing Up

- We have found that first-order validity (= theoremhood by Soundness and Completeness) is not decidable. It follows that neither is inconsistency (i.e., contradictoriness). Moreover, while each concept is semidecidable (recursively enumerable), contingency is not even semidecidable.
- If there were a Yes-procedure, ***P***, for contingency, then we could combine it with the Yes-procedure, ***P****, for *Th*(∅) resulting in a decision procedure for *Th*(∅).
- For any sentence, ***X***, apply ***P*** and ***P**** to it concurrently. If ***P*** returns '**Yes**', then ***X*** is contingent, and so ***X*** ∉ *Th*(∅). If instead ***R*** returns '**Yes**', then ***X*** ∉ *Th*(∅), and if ***R*** returns '**Yes**' when applied ~***X***, not ***X***, then ***X*** ∉ *Th*(∅).
- *Note*: It follows that invalidity (i.e., not-validity) is not semidecidable either.

Summing Up

- *Upshot*: If the set of contingent sentences were even *semidecidable*, then since the set of (first-order) validities is semidecidable too, we could form a decision procedure for the latter – contravening the **Church-Turing Theorem**.
- What about **arguments**? An argument, Γ / ***X***, is **valid** just when the conditional, (Γ_{fin} → ***X***) ∉ *Th*(∅), for some finite subset of Γ, Γ_{fin}. So, a decision procedure for validity of arguments would induce a decision procedure for validity of conditionals – including those of the form ($\boldsymbol{Q}_{\mathbf{AX}}$ → ***X***). Since we saw that this is impossible, the question of whether an argument from zero or more premises to an arbitrary conclusion is valid is undecidable.
- *Note*: The set of arguments of special kinds (e.g., those only involving monadic predicates) is decidable. But the set of all arguments is not.

Second-Order Logic (PL^2)

- We have so far been discussing first-order logic. This lets us quantify over things, and ascribe predicates to them. But it does not let us quantify into the position of the predicates. It merely lets us quantify into the position of names, like 'Lebron James', or, in the case of arithmetic, ***0***, ***sss0 + s0***, etc.
- What happens if we allow for quantification into predicate position? Given a 'full' or 'standard' semantics, metalogic changes dramatically!
- Consider the following argument:
- 1) The Evening Star has every property that the Morning Star has.
- 2) The Morning Star is a planet in the solar system.
- 3) Therefore, the Evening Star is a planet in our solar system.

Second-Order Logic (PL^2)

- Argument 1) – 3) seems valid. But it quantifies into predicate position in premise 1). Hence, it is not readily formalizable in first-order *PL*.
- A first-order quantifier applies to a variable that occupies a slot that could be occupied by a name, like 'Lebron James', if the quantifier were not present. By contrast, a second-order quantifier applies to a variable that occupies a slot that could be occupied by a (first-order) predicate, like 'red', if the quantifier were not present.
- We use the uppercase variable letters as second-order variables, and use uppercase constant letters to serve as second-order constants.

Second-Order Logic (PL^2)

- *Example*: (∃***Z***)(***Z****a* & ***Z****b*) says, intuitively, that there is a property that is had by individuals *a* and *b*. If we removed the second-order quantifier, so that the second-order variables were no longer bound, we would obtain an open sentence – exactly as in the first-order case.
- A difference with the first-order case is that, in second-order logic, predicates can apply to other predicates, as well as to individuals.
- *Example*: We can say that there is a property that is a color and is had by the text on this slide. We could write: (∃***Z***)(*C****Z*** & ***Z****t*).
- Having outlined the basic idea of second-order logic (albeit not the recursive syntactic definitions), PL^2, how can we formalize 1) – 3)?

Second-Order Logic (PL^2)

- *First*, we specify a second-order interpretation, extending a first-order one:
- *e*: The Evening Star
- *m*: The Morning Star
- ***P****z*: *z* is a planet in the solar system
- Next, we formalize the argument:
- 1) (∀***Z***)(***Z****m* → ***Z****e*)
- 2) ***P****m*
- 3) ***P****e*

Second-Order Logic (PL^2)

- This argument is valid assuming second-order Universal Instantiation.
- The predicate ***P*** can (presumably) be substituted for the second-order variable, ***Z***, in the formula ***Z****m* → ***Z****e* to obtain ***P****m* → ***P****e*. The sentence ***P****m* can then be inferred from ***P****e* and ***P****m* → ***P****e* by *modus ponens*.
- How, though, should we think of properties for the purposes of semantic interpretation? In the first-order case, individual predicates corresponded to subsets of the Universe of Discourse (UD). The crucial choice point is *whether to interpret the second-order universal quantifier, ∀**Z**, as ranging over the full powerset of that universe*. It is only if we do that we get radically different metalogical properties.

Second-Order Logic (PL^2)

- We will adopt this so-called **full semantics** for the second-order quantifiers. Accordingly, a second-order sentence like (∀***Z***)(***Z****m* → ***Z****e*) is interpreted to say something about every subset of *UD*.
- It says that for every subset of *UD*, *S*, of *UD*, if the referent of the name, *m*, belongs to the set, *S*, then the referent of *e* belongs to *S* too. That is: every subset of *UD* that contains $\boldsymbol{J}^2(m)$ also contains $\boldsymbol{J}^2(e)$.
- *Note*: The vocabulary of a second-order theory is the first-order one but with second-order variables included among the logical vocabulary.

Second-Order Peano Arithmetic (PA^2)

- One interesting application of second-order logic suggests itself: replace the <u>schemas</u> occurring in first-order formulations of our mathematical theories with (single sentence) second-order counterparts.
- *Example*: Rather that adjoining <u>infinitely-many axioms</u> (given by the Induction <u>schema</u>) to the first six axioms of *PA* (one for each formula in the language), we may state directly: *for any property*, ***P***, if *0* has ***P*** and *n+1* has ***P*** whenever *n* has *P*, then all natural numbers have ***P***. In symbols:
- <u>Induction Axiom</u> (*IA*): $(\forall \boldsymbol{Z})(\boldsymbol{Z}0 \,\&\, (\forall v)(\boldsymbol{Z}v \rightarrow \boldsymbol{Zs}v)) \rightarrow (\forall y)\boldsymbol{Z}y)$

Second-Order Peano Arithmetic (PA^2)

- Whereas the <u>Induction Schema</u> (IS) of *PA* says that any <u>formula</u> in the language that is true of *0* and that is true of ***s****v* whenever it is true of *v* is true of every natural number, *IA* says that any <u>property</u> that is had by *0* and that is had by ***s****v* whenever it is had by *v* is had by every natural number.
- Interpreted in a model (where properties are sets), $\boldsymbol{M}^2$, *IA* says that any <u>subset</u>, *A*, of *UD* that contains $\boldsymbol{M}^2$'s *0*, and contains $\boldsymbol{M}^2$'s successor of a number whenever it contains the number, contains everything in $\boldsymbol{M}^2$.
- We will write $\mathbb{N}^2$ for the model of PA^2 that is just like the <u>Standard Model</u> of *Peano Arithmetic* (*PA*), except that it interprets quantification of <u>subsets</u> of (its) natural numbers as well.

Second-Order Peano Arithmetic (PA^2)

- PA^2 is thought to be special because it is **categorical**, all its models are isomorphic. Hence, unlike *PA*, its axioms 'pin down' what we mean.
- **Theorem 5.6.1**: PA^2 is categorical.
- Writing $Th(PA^2)$ for the set of all sentences that are semantically implied by PA^2, we now have the following (dramatic?) results:
- **Theorem 5.6.2a** $Th(PA^2)$ is semantically complete.
- **Theorem 5.6.2b** $Th(PA^2) = Th_{PA2}(\mathbb{N}^2)$.
- **Theorem 5.6.2c** $Th_{PA2}(\mathbb{N}^2)$ is finitely axiomatizable.
- Let us survey the proofs and explore their philosophical significance.

Second-Order Peano Arithmetic (PA^2)

- **Theorem 5.6.2a** $Th(PA^2)$ is semantically complete.
- **Proof**:
- 1) For any sentence in $Voc(PA^2)$, $\boldsymbol{X}$, either $\mathbb{N}^2 \models_2 \boldsymbol{X}$ or $\mathbb{N}^2 \models_2 \sim\boldsymbol{X}$, by the definition of a model.
- 2) By **Theorem 5.6.1**, any two models of $Th(PA^2)$ are isomorphic. That is, for any model of $Th(PA^2)$, $\boldsymbol{M^2}$, $\mathbb{N}^2$ is isomorphic with $\boldsymbol{M^2}$ with respect to $Voc(PA^2)$, written: $\mathbb{N}^2 \cong \boldsymbol{M^2}$.
- 3) By **Theorem 3.4.1**, isomorphism implies elementary equivalence. So, by 2), $\mathbb{N}^2$ and $\boldsymbol{M^2}$ are elementary equivalent, written: $\mathbb{N}^2 \equiv \boldsymbol{M^2}$.
- 4) So, if $\mathbb{N}^2 \models_2 \boldsymbol{X}$, then $\boldsymbol{M^2} \models \boldsymbol{X}$, and if $\mathbb{N}^2 \models_2 \sim\boldsymbol{X}$, then $\boldsymbol{M^2} \models_2 \sim\boldsymbol{X}$.

Second-Order Peano Arithmetic (PA^2)

- 5) By the definition of logical consequence, $Th(PA^2) \models_2 \boldsymbol{X}$ or $Th(PA^2) \models_2 {\sim}\boldsymbol{X}$ depending on whether $\mathbb{N}^2 \models_2 \boldsymbol{X}$ or $\mathbb{N}^2 \models_2 {\sim}\boldsymbol{X}$.
- 6) So, by Conditional Proof (in the metatheory), if $\mathbb{N}^2 \models \boldsymbol{X}$, then $Th(PA^2) \models_2 \boldsymbol{X}$, and if $\mathbb{N}^2 \models_2 {\sim}\boldsymbol{X}$, $Th(PA^2) \models_2 {\sim}\boldsymbol{X}$.
- 7) Hence, any sentence in $Voc(PA^2)$, $\boldsymbol{X}$, either $Th(PA^2) \models_2 \boldsymbol{X}$ or $Th(PA^2) \models {\sim}\boldsymbol{X}$, i.e., $Th(PA^2)$ is semantically complete.

- **Theorem 5.6.2b** $Th(PA^2) = Th_{PA2}(\mathbb{N}^2)$.

Second-Order Peano Arithmetic (PA^2)

- **Proof**:
- 1) $\mathbb{N}^2 \models_2 Th(PA^2)$, so $Th(PA^2) \subseteq Th_{PA2}(\mathbb{N}^2)$.
- 2) By **Theorem 5.6.1**, for every model of $Th(PA^2)$, $\boldsymbol{M^2}$, $\mathbb{N}^2 \cong \boldsymbol{M^2}$ with respect to $Voc(PA^2)$, and, hence, $\mathbb{N}^2 = \boldsymbol{M^2}$ (by **Theorem 3.4.1**).
- 3) So, if $\mathbb{N}^2 \models_2 \boldsymbol{X}$, then, for any model of $Th(PA^2)$, $\boldsymbol{M^2}$, $\boldsymbol{M^2} \models_2 \boldsymbol{X}$.
- 4) Thus, $Th(PA^2) \models_2 \boldsymbol{X}$, and, by the definition of a theory, $\boldsymbol{X} \in Th(PA^2)$.
- 5) So, if $\boldsymbol{X} \in Th_{PA2}(\mathbb{N}^2)$, then $\boldsymbol{X} \in Th(PA^2)$.
- 6) Hence, $Th(PA^2) \subseteq Th_{PA2}(\mathbb{N}^2)$ and $Th_{PA2}(\mathbb{N}^2) \subseteq Th(PA^2)$, i.e., $Th(PA^2) = Th_{PA2}(\mathbb{N}^2)$.

Second-Order Peano Arithmetic (PA^2)

- **Theorem 5.6.2c** $Th_{PA2}(\mathbb{N}^2)$ is finitely axiomatizable.
- **Proof:** We just saw that $Th(PA^2) = Th_{PA2}(\mathbb{N}^2)$. But PA^2 is a finite set. Since any finite set is decidable, PA^2 qualifies as a set of axioms. Consequently, there exists a finite set of axioms from which all the members of $Th_{PA2}(\mathbb{N}^2)$ logically (semantically) follow, i.e., $Th_{PA2}(\mathbb{N}^2)$ is finitely axiomatizable.
- **Theorem 5.6.3:** The Compactness Theorem holds for any (decidable) sound and complete deductive system for PL^2 whose derivations are finite.
- **Proof**: Let $|\!-_2$ be a (decidable) sound, complete, and finite proof relation on $Voc(PA^2)$, and let Σ be a set of sentences in $Voc(PL^2)$ and $\boldsymbol{X}$ any sentence in $Voc(PA^2)$ such that $\Sigma \models_2 \boldsymbol{X}$. Since $\Sigma \models_2 \boldsymbol{X}$, and $|\!-_2$ is complete and finite, Σ_{fin} $|\!-_2 \boldsymbol{X}$ for some finite subset of Σ, Σ_{fin}. And since $|\!-_2$ is sound, $\Sigma_{\text{fin}} \models_2 \boldsymbol{X}$.

Second-Order Peano Arithmetic (PA^2)

- We will now show that the Compactness Theorem does not hold for $\models_2$, from which it follows that there is no (decidable) sound and complete deductive system for PL^2 whose derivations are finite.
- **Theorem 5.6.4**: There exists a set of sentences in $Voc(PL^2)$ (with whatever non-logical vocabulary we wish), Σ, and sentence $\boldsymbol{X}$ such that $\Sigma \models_2 \boldsymbol{X}$ but for no finite subset of Σ, T, T $\models_2 \boldsymbol{X}$.
- **Proof Sketch**:
- 1) Expand $Voc(PA^2)$ with one additional name, $\boldsymbol{e}$. Now define $\Theta = \{\boldsymbol{e} \neq \boldsymbol{s}^{n}\boldsymbol{0} : n$ is a positive integer$\}$. Since PA^2 is finite, we can construct the conjunction of these axioms, PA^2_{AX}. So, let $\Sigma = \{PA^2_{\text{AX}}\} \cup \Theta$.

Second-Order Peano Arithmetic (PA^2)

- 2) Σ, and, thus, PA^2_{AX} and Θ, are satisfiable because an interpretation, $\boldsymbol{M^2}$, just like $\mathbb{N}^2$ except that ***e*** is interpreted as the number, *0*, is a model of Σ.
- 3) Since each numerical term, $\boldsymbol{s^n0}$, picks out the number, *n*, for any positive integer (we proved), *n*, $\boldsymbol{e \neq s^n0}$ is true on $\boldsymbol{M^2}$.
- 4) Since $\boldsymbol{M^2}$ is a model of PA^2, it is isomorphic to $\mathbb{N}^2$ with respect to $Voc(PA^2)$, by **Theorem 5.6.1**. Hence, $\boldsymbol{M^2}(\boldsymbol{s^n0}) = n$, $\forall n \in \mathbb{N}$, so for the additional vocabulary, ***e***, we must have $\boldsymbol{M^2}(\boldsymbol{e}) = 0$.
- 5) But no finite subset, T, of Θ implies this, by a variation on the Compactness argument that we gave in the first-order case.
- 6) So, $\Sigma \models_2 \boldsymbol{e} = \boldsymbol{0}$, but for no finite subset, T, do we have $T \models_2 \boldsymbol{e} = \boldsymbol{0}$, i.e., the Compactness Theorem fails for (full) second-order consequence, $\models_2$.

Second-Order Peano Arithmetic (PA^2)

- **Theorem 5.6.5**: There is no (decidable) sound and complete deductive system for PL^2 whose derivations are finite.
- **Proof**: Immediate from **Theorem 5.6.3** and **Theorem 5.6.4**.
- ***Note***: The same holds even of (decidable) sound and complete deductive systems for PL^2 whose derivations may be infinite.
- Second-order logic is sometimes thought to vindicate Russell's logicism, the view that math is just logic. Crispin Wright even showed that PA^2 is derivable from **Hume's Principle** in a standard (sound and incomplete) system of second-order logic. So, if *Hume's Principle* is analytic, then arithmetic is indeed provable from definitions alone.

Second-Order Peano Arithmetic (PA^2)

- The problem is that second order logic is not *logic* in the sense that has been central to philosophy since Aristotle. Since there is no (sound and) complete proof theory for PL^2, there are infinitely-many 'logical truths' without proofs – no matter which (decidable) proof theory we adopt.
- Moreover, the semantics of second-order logic is inseparable from the metatheoretic set theory. There is a sentence in $Voc(PL^2)$ (with no non-logical vocabulary) that has a model just in case *CH* holds, and another sentence that has a model just in case it does not hold! So, doubts about the clarity of set-theoretic concepts become doubts about the clarity of second-order quantification. Perhaps the categoricity argument is a mirage.

Second Incompleteness Theorem

- **Hilbert's Program** was an influential agenda in the philosophy of mathematics around the turn of the 20th century. The key idea was to vindicate (what we called) formalism, the view that *most* of mathematics is a meaningless game with symbols. So, there is no need to explain how we know it, or what it is about. The philosophical mysteries dissolve.
- Why 'most'? Because the theory of symbol manipulation itself had better *not* be meaningless! There must be a fact as to whether or not, e.g., one can produce the string '$0 = 1$' using just the rules of *NDS* from *PA*.
- Hilbert aimed to prove, in a 'finitary' (first-order) metatheory, which *Th*(*PA*) extends, that one cannot derive a contradiction from standard mathematics, *ZFC*. If this can be done, then *PA* |- *Con*(*PA*) *a fortiori*.

Second Incompleteness Theorem

- Gödel's Second Incompleteness Theorem shows that this is impossible, if *PA* is consistent. More carefully, PA ⊬ *Con*(*PA*), if *Con*(*PA*), where '*Con*(*PA*)' abbreviates, ~(∃*x*)**proof**(*x*, ⌜ ***0*** = ***s0*** ⌝), or, equivalently, (∀*x*)~**proof**(*x*, ⌜ ***0*** = ***s0*** ⌝), and **proof**(*x*, *y*) is a standard proof predicate, in a sense to be defined.
- One nonstandard proof predicate is due to Rosser. According to it, a sequence, ***D***, of sentences in *Voc*(*PA*) is only a *PA* proof if its last line does not contradict the last line of a *PA* proof whose Gödel number is smaller than the Gödel number of ***D***. The Second Incompleteness Theorem fails for Rosser's proof predicate.
- The **intensionality of the Second Incompleteness Theorem** refers to the sensitivity of the Second Incompleteness Theorem – as opposed to the First Incompleteness Theorem – to the choice of proof predicate in its formulation.

Second Incompleteness Theorem

- Let introduce a predicate in the metalanguage, $\mathbf{Thrm}_{PA}$*(*y*), that refers to any provability predicate for *PA* meeting the following conditions:
- **HB1** For every sentence, ***X***, in *Voc*(*PL*), if *PA* |– ***X***, then *PA* |- $\mathbf{Thrm}_{PA}$*(⌜ ***X*** ⌝).
  - If *PA* proves ***X***, then *PA* proves that it proves ***X***.
- **HB2** For every sentence, ***X***, in *Voc*(*PL*), if *PA* |- $\mathbf{Thrm}_{PA}$*(⌜ ***X*** → ***Y*** ⌝) → [$\mathbf{Thrm}_{PA}$*(⌜ ***X*** ⌝) → ($\mathbf{Thrm}_{PA}$*(⌜ ***Y*** ⌝)]
  - *PA* proves that: if *PA* proves both ***X*** → ***Y*** and ***X***, then it also proves ***Y***.
- **HB3** For every sentence, ***X***, in *Voc*(*PL*), *PA* |– $\mathbf{Thrm}_{PA}$*(⌜ ***X*** ⌝) → $\mathbf{Thrm}_{PA}$*(⌜ $\mathbf{Thrm}_{PA}$*(⌜ ***X*** ⌝) ⌝) .
  - If *PA* proves that it proves ***X***, then *PA* also proves the fact that it proves this.

Second Incompleteness Theorem

- Let us call any $\mathbf{Thrm}_{PA}$*(y) a standard provability predicate, and any proof predicate out of which it is built, $\mathbf{proof}_{PA}$*(x, y), a standard proof predicate. Finally, let us write *Con**(*PA*) for ~$\mathbf{Thrm}_{PA}$*(⌜ ***0*** = ***s0*** ⌝) = ~(∃x)$\mathbf{proof}_{PA}$*(x, ⌜ ***0*** = ***s0*** ⌝) = (∀x)~$\mathbf{proof}_{PA}$*(x, ⌜ ***0*** = ***s0*** ⌝).
- Gödel's proof, and, hence, provability predicate was standard. Using his provability predicate, or any other standard one, we have:
- **Gödel's Second Incompleteness Theorem**: If *Con**(*PA*), *PA* ⊬ *Con**(*PA*) (where *Con**(*PA*), and, hence, $\mathbf{Thrm}_{PA}$*(y) is standard, i.e., satisfies **HB1**, **HB2**, and **HB3**).

Second Incompleteness Theorem

- 1) Suppose that *Con**(*PA*), and, for *reductio*, that *PA* |– *Con**(*PA*).
- 2) Apply the **Diagonalization (Carnap) Lemma** to the formula, $\mathbf{Thrm}_{PA}$*(y) → (***0*** = ***s0***), to get a sentence, ***B***, such that *PA* |- ***B*** ←→ $\mathbf{Thrm}_{PA}$*(⌜ ***B*** ⌝) → (***0*** = ***s0***).
- 3). So, *PA* |– ~$\mathbf{Thrm}_{PA}$*(⌜ ***0*** = ***s0*** ⌝) and *PA* |- ***B*** ←→ $\mathbf{Thrm}_{PA}$*(⌜ ***B*** ⌝) → (***0*** = ***s0***).
- 4) We can now use the properties of standard proof predicates to combine derivations of ~$\mathbf{Thrm}_{PA}$*(⌜ ***0*** = ***s0*** ⌝) and of ***B*** ←→ $\mathbf{Thrm}_{PA}$*(⌜ ***B*** ⌝) → (***0*** = ***s0***) to get a derivation of ***0*** = ***s0***.

Second Incompleteness Theorem

- Yaqub illustrates this with the following lengthy but routine *PL* proof:

[0	0	PA^F	Premises
	⋮	D_1	This PA derivation is given
	i	$\neg\mathbf{prov}\langle\mathbf{0} = \mathbf{1}\rangle$	Conclusion of D_1
	⋮	D_2	This PA derivation is given
	j	$B \leftrightarrow (\mathbf{prov}\langle B\rangle \rightarrow \mathbf{0} = \mathbf{1})$	Conclusion of D_2
[1	j+1	$\mathbf{prov}\langle\mathbf{0} = \mathbf{1}\rangle$	CPA
[2	j+2	$\mathbf{0} \neq \mathbf{1}$	RA
	j+3	$\mathbf{prov}\langle\mathbf{0} = \mathbf{1}\rangle$	j+1, Reit
2]	j+4	$\neg\mathbf{prov}\langle\mathbf{0} = \mathbf{1}\rangle$	i, Reit
1]	j+5	$\mathbf{0} = \mathbf{1}$	(j+2)–(j+4), RAA
	j+6	$\mathbf{prov}\langle\mathbf{0} = \mathbf{1}\rangle \rightarrow \mathbf{0} = \mathbf{1}$	(j+1)–(j+5), CP
	j+7	$(B \rightarrow (\mathbf{prov}\langle B\rangle \rightarrow \mathbf{0} = \mathbf{1})) \wedge ((\mathbf{prov}\langle B\rangle \rightarrow \mathbf{0} = \mathbf{1}) \rightarrow B)$	j, Bc
	j+8	$B \rightarrow (\mathbf{prov}\langle B\rangle \rightarrow \mathbf{0} = \mathbf{1})$	j+7, Simp

Second Incompleteness Theorem

	j+9	$\mathbf{prov}\langle B \rightarrow (\mathbf{prov}\langle B\rangle \rightarrow \mathbf{0} = \mathbf{1})\rangle$	j+8, PC1 (observe that line j+8 is derived from PA axioms alone)
	j+10	$\mathbf{prov}\langle B \rightarrow (\mathbf{prov}\langle B\rangle \rightarrow \mathbf{0} = \mathbf{1})\rangle \rightarrow (\mathbf{prov}\langle B\rangle \rightarrow \mathbf{prov}\langle\mathbf{prov}\langle B\rangle \rightarrow \mathbf{0} = \mathbf{1}\rangle)$	PC2 (substitute B for $\mathbf{X}$ and $\mathbf{prov}\langle B\rangle \rightarrow \mathbf{0} = \mathbf{1}$ for $\mathbf{Y}$)
	j+11	$\mathbf{prov}\langle B\rangle \rightarrow \mathbf{prov}\langle\mathbf{prov}\langle B\rangle \rightarrow \mathbf{0} = \mathbf{1}\rangle$	j+9, j+10, MP
	j+12	$\mathbf{prov}\langle\mathbf{prov}\langle B\rangle \rightarrow \mathbf{0} = \mathbf{1}\rangle \rightarrow (\mathbf{prov}\langle\mathbf{prov}\langle B\rangle\rangle \rightarrow \mathbf{prov}\langle\mathbf{0} = \mathbf{1}\rangle)$	PC2 (substitute $\mathbf{prov}\langle B\rangle$ for $\mathbf{X}$ and $\mathbf{0} = \mathbf{1}$ for $\mathbf{Y}$)
	j+13	$\mathbf{prov}\langle B\rangle \rightarrow (\mathbf{prov}\langle\mathbf{prov}\langle B\rangle\rangle \rightarrow \mathbf{prov}\langle\mathbf{0} = \mathbf{1}\rangle)$	j+11, j+12, HS
	j+14	$\mathbf{prov}\langle B\rangle \rightarrow \mathbf{prov}\langle\mathbf{prov}\langle B\rangle\rangle$	PC3 (substitute B for $\mathbf{X}$)
[3	j+15	$\mathbf{prov}\langle B\rangle$	CPA
	j+16	$\mathbf{prov}\langle\mathbf{prov}\langle B\rangle\rangle \rightarrow \mathbf{prov}\langle\mathbf{0} = \mathbf{1}\rangle$	j+13, j+15, MP
	j+17	$\mathbf{prov}\langle\mathbf{prov}\langle B\rangle\rangle$	j+14, j+15, MP
3]	j+18	$\mathbf{prov}\langle\mathbf{0} = \mathbf{1}\rangle$	j+16, j+17, MP
	j+19	$\mathbf{prov}\langle B\rangle \rightarrow \mathbf{prov}\langle\mathbf{0} = \mathbf{1}\rangle$	j+15–j+18, CP

Second Incompleteness Theorem

	j+20	**prov**⟨B⟩→**0** = **1**	j+6, j+19, HS
	j+21	B	j, j+20, BcMP
	j+22	**prov**⟨B⟩	j+21, PC1 (again, line j+21 is derived from PA axioms alone)
0]	j+23	**0** = **1**	j+20, j+22, MP

- 5) So, *PA* |– ***0*** = ***s0***.
- 6) But Universally instantiating $\mathbf{Ax_1}$ (∀*x*) ***0*** ≠ ***s***(*x*) gives ***0*** ≠ ***s0***.
- 7) So, ~*Con**(*PA*), contradicting 1).
- 8) Hence, by *reductio ad absurdum*, *PA* ⊬ *Con**(*PA*) if *Con**(*PA*).

Henkin Sentence

- We have discovered that there is a sentence, **G**, intuitively expressing its own unprovability from *PA*, such that *PA* ⊬ **G** and *PA* ⊬ **~G**, if *Con*(*PA*). What about a sentence, **H**, expressing its own provability?
- By the Carnap Lemma, there must be fixed point for $\mathbf{Thrm}_{PA}$*(*x*), i.e.:
  - *PA* |- **H** ←→ $\mathbf{Thrm}_{PA}$*(⌜ **H** ⌝)
- Is **H** provable from *PA*? **H** intuitively 'says' that it is true just in case it is provable from *PA*. So, reflection on **H**'s meaning is of no use.

Löb's Theorem

- Martin Löb solved the problem by establishing a general result:
  - **Löb's Theorem**: If *PA* |- **Thrm**$_{PA}$*****(⌜ **P** ⌝) → **P**, then *PA* |- **P**.
- (You are asked to derive this theorem using **HB1** – **HB3** on your exam.)
- In particular, if *PA* |- Thrm$_{PA}$*(⌜ H ⌝) → *H*, then *PA* |- *H*.
- We can also use **Löb's Theorem** to give a different proof of the Second Incompleteness Theorem:

Löb's Theorem

- **Proof**:
- 1) By **Löb's Theorem**: If *PA* |- **Thrm**$_{PA}$*****(⌜ ***0 = s0*** ⌝) → ***0 = s0***, then *PA* |- ***0 = s0***.
- 2) So, suppose that *PA* |- ~**Thrm**$_{PA}$*****(⌜ ***0 = s0*** ⌝), i.,e, *PA* |- *Con**(*PA*).
- 3) Then *PA* |- **Thrm**$_{PA}$*****(⌜ ***0 = s0*** ⌝) → ***0 = s0***.
- 4) So, by **Löb's Theorem**, *PA* |- ***0 = s0***.
- 5) But *PA* |- ***0 ≠ s0***.
- 6) So, if Con*(*PA*), then *PA* ⊬ ~**Thrm**$_{PA}$*****(⌜ ***0 = s0*** ⌝), i.e. *PA* ⊬ *Con**(*PA*).