Knowledge Representation

and

Reasoning

Truth, Logic, and Provability

Deepak Khemani
Plaksha University

The Machinery of Deductive Reasoning

Logicians were mainly focused on reasoning much before the advent of computers

With digital computers the focus shifted to representation

We *begin* our study with logic and the algorithms for reasoning, before shifting our focus to representation

Leibniz: *Calculus Ratiocinator*

http://en.wikipedia.org/wiki/Gottfried_Leibniz

Leibniz believed that much of human reasoning could be reduced to calculations of a sort, and that such calculations could resolve many differences of opinion:

> "*The only way to rectify our reasonings is to make them as tangible as those of the Mathematicians, so that we can find our error at a glance, and when there are disputes among persons, we can simply say: Let us calculate [calculemus], without further ado, to see who is right.*"

Leibniz's goal was to find a foolproof mechanism which could be used to unambiguously decide who was right when people were arguing.

An approach to the principles of argument, known as *Tarka Shastra*, was developed independently in 17th Century India by Annambhatta in his treatise *Tarka-Sangraha*.

See https://en.wikipedia.org/wiki/Tarka-Sangraha

Leibniz: An Approach to Logic

The principles of Leibniz's logic and, arguably, of his whole philosophy, reduce to two:

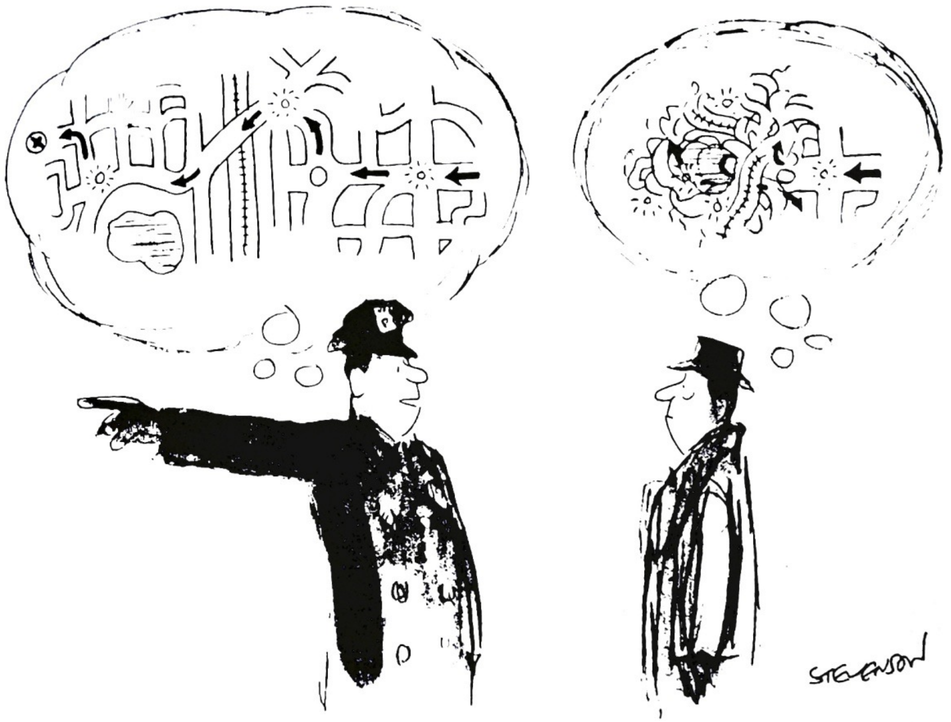
- All our ideas are compounded from a very small number of simple ideas, which form the alphabet of human thought.

- Complex ideas proceed from these simple ideas by a uniform and symmetrical combination, analogous to arithmetical multiplication.

- Leibniz's main aim in logic, however, was to extend the traditional syllogistic to a "*Universal Calculus*."

Leibniz: Logic. https://iep.utm.edu/leib-log/

Natural Languages



Drawing by Stevenson; © 1976 The New Yorker Magazine, Inc.

Fig. 1.5 Using language to express a mental model

John Sowa: Conceptual Structures

Richness
Ambiguity
Verbosity
Impreciseness

The Language of Logic

- Unlike natural languages like English and Hindi, languages in logic are defined as formal languages
- A language for logic has a well defined alphabet
  - There are many logic languages
- A sentence or a statement in classical logic is a symbol system that can can be assigned a truth value *true* or *false*
  - A three valued logic may have an additional category *indeterminate*
  - Fuzzy logic has a continuum of truth values
- A sentence in classical logic is something that (usually) has a meaning
  - the sentence (usually) *denotes* something
- Both *truth values* and *meaning* are in the human mind
- The machine is an entirely syntactic entity

Formal Logics

Logics are formal languages with well defined rules for manipulation of representations.

A *knowledge base* (KB) is a set of sentences in a given logic language.

The family of logics varies on expressivity.

More expressivity comes at the cost of increasing computational complexity.

Formal Logic: A Syntactic Machine

- The machine is an entirely syntactic entity
  - It is only concerned with form, not content.
- Even when meaning is defined formally, it refers to another syntactic structure
- An open and interesting question – can a computer *understand* what it is processing?
  - So far our way of ascertaining this has been by some variation of the Turing test – can it answer a question “intelligently and meaningfully”?
  - But then comes the Chinese room argument.
- Logic has more modest goals – can we build machines that do something that is *meaningful for us*?
  - Even when it operates syntactically or mechanically (by applying rules)
- Logics are meant to capture *valid forms of reasoning*.

The Syllogism

The Greek syllogism embodies the notion of **reasoning** in **formal logic**.

An argument is valid **if** it conforms to a valid **form**

All men are mortal	All cities are congested	All politicians are honest
Socrates is a man	Chennai is a city	Sambit is a politician
Socrates is mortal	Chennai is congested	Sambit is honest

The Socratic argument

In a **valid** argument
IF the premises are true
THEN the conclusions
are **necessarily** true

Formal Logic

Logic is a formal system

Logical reasoning is concerned ONLY with the FORM of the argument, and not with CONTENT.

If the form is valid AND If the antecedents are *true*
THEN the conclusion is true.

Thus, the conclusion holds only if the antecedents are *true*.

Logic does not concern itself with the *truth* of the antecedents

OR

what the sentences are talking about (content).

Logic: A Branch of Mathematics

- Logic is a branch of mathematics
- Truth values are just members of a set of arity 2 to which sentences are mapped
  - {true, false}
  - {T, F}
  - {1, 0}
- When we begin by mapping a set of sentences to *true* we are interested in knowing which other sentences map to *true*
  - For humans a true sentence generally has some validation from the domain.
- Different logic languages have different levels of expressivity which has a tradeoff with complexity
- Logic is concerned with valid arguments
  – whatever the language chosen

Logics

Epistemic Logics

Temporal Logics

Modal Logics

Classical two valued logics

Second Order Logic

First Order Logic

Propositional Logic

Event Calculus / Situation Calculus

Default Logics

Probabilistic reasoning

Horn Clauses

Description Logics

Fuzzy logics

Constraint logic programming

Rough sets

Qualitative reasoning

Entailment

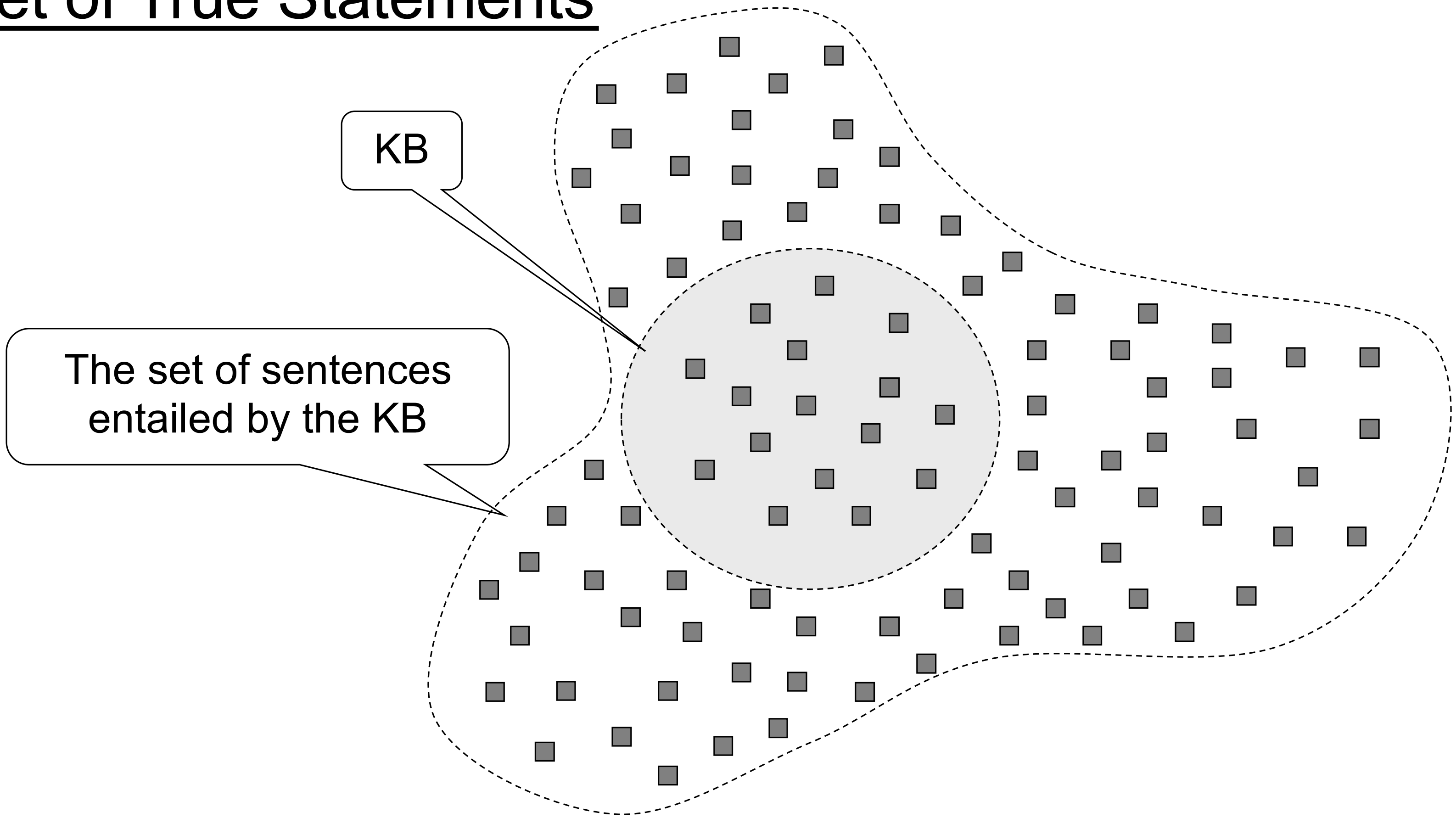
- A set of *premises* or *axioms* are initially given to be *true*.
  - We will refer to this set as S or KB the knowledge base.
  - Axioms are *known* to be true.
  - Premises are *given and assumed* to be true.
  - A KB is true iff all its sentences are true.
- Entailment – given the KB, a subset of all the sentences in the language are true or entailed.

A sentence α is said to be entailed by a set of sentences *S*/KB if the sentence is necessarily *true* whenever S/KB is *true*

$$KB \vDash \alpha$$

We also say that α is *true* (given the KB)

The Set of True Statements



Provability

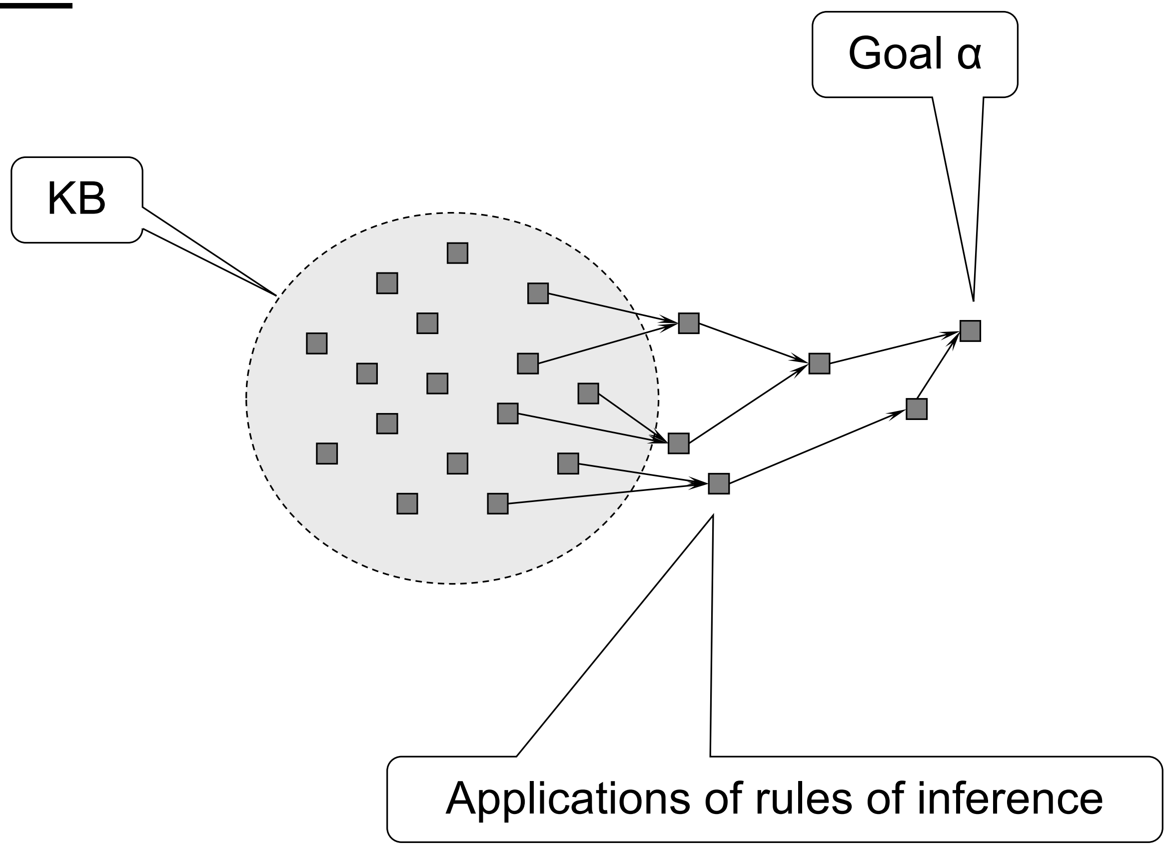
- The Logic Machine has Rules of Inference
  - A rule of inference is a rule which specifies when another sentence can be added to the KB.
- A Logic Machine has algorithms for deriving new sentences from old by repeated application of the rules of inference
  - The derivation is a proof.
- Given a KB can a new sentence α be added to the KB by repeated application of some rules of inference?

  If yes, then we say that α is provable.

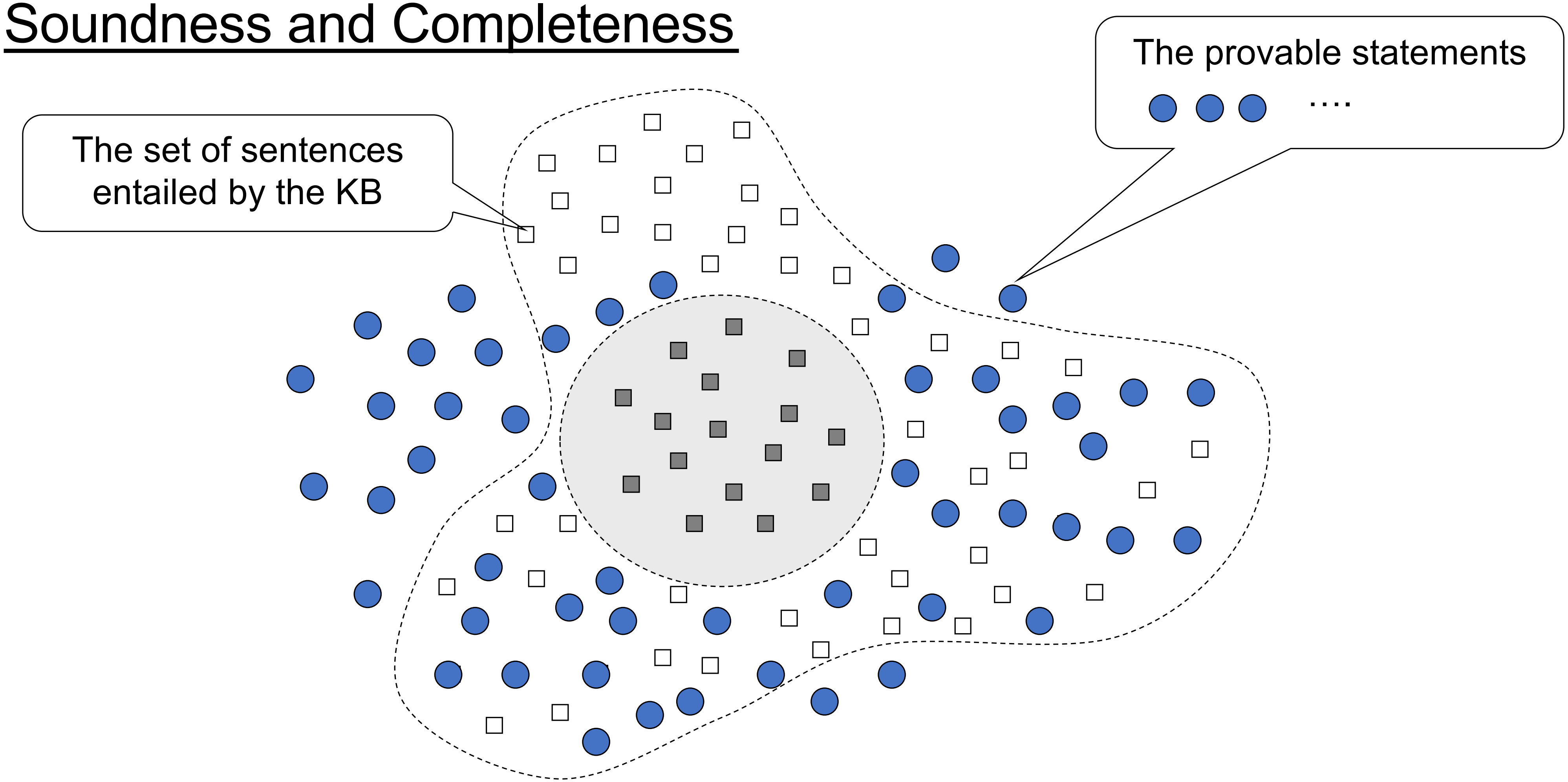
$$KB \vdash \alpha$$

- The goal of the logic machinery is ideally to mechanically produce *all and only* the sentences that are entailed.

A Proof



Soundness and Completeness



Soundness and Completeness

Given a knowledge base and a reasoning algorithm –

Entailment: which other sentences in the language are necessarily true?

Proof: which other sentences in the language can one produce by the reasoning algorithm?

Soundness (of the reasoning algorithm):

A logic is sound if only true statements in the language can be proved

if KB $\vdash \alpha$ **then** KB $\models \alpha$

Completeness (of the reasoning algorithm):

A logic is complete if all true statements in the language can be proved

if KB $\models \alpha$ **then** KB $\vdash \alpha$

The set of sentences in a logic language

The set of sentences that constitute the formal language of a logic is constructed as follows

1. A set of atomic or primitive sentences or formulas is defined

2. A set of logical connectives are used to construct compound sentences or *well formed formulas*

The set of sentences in any logic is unbounded or infinite

Common logical connectives

- A logical connective takes one or more sentences as input and constructs a new sentence.
- All logic languages have a common set of connectives – more expressive languages may have more
- Unary connective. Common symbol $\neg$ or $\sim$. If $\boldsymbol{\alpha}$ is a sentence then so is $(\neg\boldsymbol{\alpha})$ or $(\sim\boldsymbol{\alpha})$, generally read as *not alpha*. We will use the former, also written without the brackets as $\neg\boldsymbol{\alpha}$.
- Binary connectives
  - $\wedge$ usually read as AND. If $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are sentences then so is $(\boldsymbol{\alpha} \wedge \boldsymbol{\beta})$
  - $\vee$ usually read as OR. If $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are sentences then so is $(\boldsymbol{\alpha} \vee \boldsymbol{\beta})$
  - $\supset$ or $\rightarrow$ or $\Rightarrow$ usually read as IMPLIES. We will use the first one. If $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are sentences then so is $(\boldsymbol{\alpha} \supset \boldsymbol{\beta})$
  - … there are 16 binary connectives
- Ternary connectives would connect three sentences.
- But whatever we can express with ternary or higher order connectives can also be expressed with binary connectives

Semantics of logical connectives

- The semantics of compound sentences is derived from the definition of the logical connective.
- The connectives are defined using truth tables.
- This is also known as *truth functional semantics*.
- The truth table on the left is for the logical *not*. Not surprisingly $\neg\alpha$ is *true* when α is *false,* $\neg\alpha$ is *false* when α is *true.*
- The truth table on the right is for the logical *AND.* $(\alpha \wedge \beta)$ is *true* only when both α and β are true. Else it is *false.*

α	$\neg\alpha$
true	false
false	true

α	β	$(\alpha \wedge \beta)$
true	true	true
false	true	false
true	false	false
false	false	false

Some common binary connectives

α Implies β

α	β	(α ⊃ β)
true	true	true
false	true	true
true	false	false
false	false	true

xor: either α or β

α	β	(α ⊕ β)
true	true	false
false	true	true
true	false	true
false	false	false

equivalence of α and β

α	β	(α ≡ β)
true	true	true
false	true	false
true	false	false
false	false	true

α or β

α	β	(α ∨ β)
true	true	true
false	true	true
true	false	true
false	false	false

nor: neither α nor β

α	β	(α ↓ β)
true	true	false
false	true	false
true	false	false
false	false	true

Semantics of compound sentences

- The semantics of compound sentences is derived from the definition of the logical connectives and the constituent sentences
- This is again given by a truth table
- The number of rows in the truth table is 2^N for N input sentences.
- The following truth table shows that the sentence $((\alpha \supset \beta) \equiv (\neg\alpha \vee \beta))$ is always true.

α	β	$(\alpha \supset \beta)$	$\neg\alpha$	$(\neg\alpha \vee \beta)$	$((\alpha \supset \beta) \equiv (\neg\alpha \vee \beta))$
true	true	true	false	true	true
false	true	true	true	true	true
true	false	false	false	false	true
false	false	true	true	true	true

Semantics of compound sentences

- The number of rows in the truth table is 2^N for N input sentences.
- A compound sentence with three constituents

α	β	$\boldsymbol{\delta}$	$(\alpha \supset \beta)$	$\neg\boldsymbol{\delta}$	$((\alpha \supset \beta) \vee \neg\boldsymbol{\delta})$
true	true	true	true	false	true
false	true	true	true	false	true
true	false	true	false	false	false
false	false	true	true	false	true
true	true	false	true	true	true
false	true	false	true	true	true
true	false	false	false	true	true
false	false	false	true	true	true

Entailment

Given that (α ⊃ β) is *true*

α	β	(α ⊃ β)
true	true	true
false	true	true
true	false	false
false	false	true

…and that α is *true*

α	β	(α ⊃ β)
true	true	true
false	true	true
false	false	true

β is necessarily *true*

α	β	(α ⊃ β)
true	true	true

$\{(\alpha \supset \beta), \alpha\} \vDash \beta$

…and that β is *true*

α	β	(α ⊃ β)
true	true	true
false	true	true
false	false	true

α can be *true or* α can be *false*

α	β	(α ⊃ β)
true	true	true
false	true	true

$\{(\alpha \supset \beta), \beta\} \nvDash \alpha$

α Implies β

α	β	(α ⊃ β)
true	true	true
false	true	true
true	false	false
false	false	true

Patterns in Arguments

Given $(\alpha \supset \beta)$
and α

Infer β

Given $(\alpha \supset \beta)$
and β

Infer α

Note that α and β are *variables* and can be substituted by *any* sentence.

From $(\alpha \supset \beta)$
and α
Infer β

$$\frac{\begin{array}{c}(\alpha \supset \beta)\\ \alpha\end{array}}{\beta}$$

Sound rule of inference
Valid argument

From $(\alpha \supset \beta)$
and β
Infer α

$$\frac{\begin{array}{c}(\alpha \supset \beta)\\ \beta\end{array}}{\alpha}$$

Not sound
Invalid argument

$\{(\alpha \supset \beta), \alpha\} \models \beta$

$\{(\alpha \supset \beta), \beta\} \not\models \alpha$

Deduction

$$\begin{array}{l} (\alpha \supset \beta) \\ \underline{\alpha \qquad\qquad} \\ \beta \end{array}$$

Modus Ponens

Most students of logic begin with *Modus Ponens.*

We say that *Modus Ponens* is a *sound rule of inference.*

It is a valid pattern of *deductive reasoning*.

It is entirely syntactic in nature – can be automated.

It does not refer to truth values at all!

Question:

If we can establish a valid conclusion (entailment) by constructing a truth table, why should we bother with derivation and proof?

- Because the number of rows in the truth table grows exponentially with the number of ”variables”.
- And the number of columns grows linearly with the number of connectives.
- Most importantly the truth table method does not apply to more expressive logics.

Abduction

In contrast, the rule

From	$(\alpha \supset \beta)$
and	β
Infer	α

is not sound.

$$\frac{\begin{array}{c}(\alpha \supset \beta)\\ \beta\end{array}}{\alpha}$$

That means that the conclusion is not necessarily true.

For example, consider the following argument

1. If you water the garden (α), the lawn becomes wet (β).
2. The lawn is wet (β).
3. Therefore, (α) you watered the garden.

However, *abductive inferences* are commonplace in our lives!

Not least in (medical) diagnosis.

1. If you have Covid (α), then you have cough and fever (β).
2. You have cough and fever (β).
3. Therefore, (α) you have Covid.

Probably

After making the inference a medic might advise a confirmatory test.

Valid Rules of Inference

How does one know whether a rule of inference is valid / sound?

A rule $\{\alpha, \beta\} \vdash \boldsymbol{\delta}$ is sound iff $((\alpha \wedge \beta) \supset \boldsymbol{\delta})$ is *true* for all values of α, β, and $\boldsymbol{\delta}$.

That is, if and only if $((\alpha \wedge \beta) \supset \boldsymbol{\delta})$ is a *tautology*.

For *Modus Ponens* $((\alpha \wedge (\alpha \supset \beta)) \supset \beta)$ is a tautology as shown below.

α	β	$(\alpha \supset \beta)$	$(\alpha \wedge (\alpha \supset \beta))$	$((\alpha \wedge (\alpha \supset \beta)) \supset \beta)$
true	true	true	true	true
false	true	true	false	true
true	false	false	false	true
false	false	true	false	true

Some common valid rules of inference

From $\alpha \supset \beta$
and $\underline{\alpha}$
Infer β
Modus Ponens (MP)

From $\alpha \supset \beta$
and $\underline{\neg\beta}$
Infer $\neg\alpha$
Modus Tollens (MT)

From α
and $\underline{\beta}$
Infer $\alpha \wedge \beta$
Conjunction (C)

From $\alpha \vee \beta$
and $\underline{\sim\alpha}$
Infer β
Disjuncuntive Syllogism (DS)

From $\underline{\alpha}$
Infer $\alpha \vee \beta$
Addition (A)

From $\underline{\alpha \wedge \beta}$
Infer α
Simplification (S)

From $\alpha \supset \beta$
and $\underline{\beta \supset \gamma}$
Infer $\alpha \supset \gamma$
Hypothetical Syllogism (HS)

From $(\alpha \supset \beta) \wedge (\gamma \supset \delta)$
and $\underline{\alpha \vee \gamma}$
Infer $\beta \vee \delta$
Constructive Dilemma (CD)

From $(\alpha \supset \beta) \wedge (\gamma \supset \delta)$
and $\underline{\neg\beta \vee \neg\delta}$
Infer $\neg\alpha \vee \neg\gamma$
Destructive Dilemma (DD)

Exercise: Show that the above rules of inference are sound

Rules of Substitution

A rule of substitution allows one to replace one sentence with another. This is possible when one sentence is logically equivalent to another. As an example let us look at the following equivalence.

$$((\alpha \supset \beta) \equiv (\neg\alpha \vee \beta))$$

If the above equivalence is a tautology, then the sentence $(\alpha \supset \beta)$ will always take the same truth value as the sentence $(\neg\alpha \vee \beta)$. Hence either of the two could be replaced by the other without any loss. We can verify that the equivalence is a tautology by constructing a truth table.

α	β	$(\alpha \supset \beta)$	$\neg\alpha$	$(\neg\alpha \vee \beta)$	$((\alpha \supset \beta) \equiv (\neg\alpha \vee \beta))$
true	true	true	false	true	true
false	true	true	true	true	true
true	false	false	false	false	true
false	false	true	true	true	true

Common rules of substitution

$\alpha \equiv (\alpha \vee \alpha)$	idempotence of $\vee$
$\alpha \equiv (\alpha \wedge \alpha)$	idempotence of $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of $\vee$
$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	DeMorgan's Law
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$	DeMorgan's Law
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$
$(\alpha \supset \beta) \equiv (\neg\beta \supset \neg\alpha)$	contrapositive
$(\alpha \supset \beta) \equiv (\neg\alpha \vee \beta)$	implication
$(\alpha \equiv \beta) \equiv ((\alpha \supset \beta) \wedge (\beta \supset \alpha))$	equivalence
$((\alpha \wedge \beta) \supset \gamma) \equiv (\alpha \supset (\beta \supset \gamma))$	exportation
$((\alpha \supset \beta) \wedge (\alpha \supset \neg\beta)) \equiv \neg\alpha$	absurdity

$(\alpha \vee \text{true}) \equiv \text{true}$

$(\alpha \vee \text{false}) \equiv \alpha$

$(\alpha \wedge \text{true}) \equiv \alpha$

$(\alpha \wedge \text{false}) \equiv \text{false}$

$(\alpha \wedge \neg\alpha) \equiv \text{false}$

$(\alpha \vee \neg\alpha) \equiv \text{true}$

$\alpha \equiv \neg(\neg\alpha)$

Expressivity: Are all connectives needed?

- A binary connective $\circ$ defines the semantics of the sentence $(\alpha \circ \beta)$.
- There are 16 ways in which the last column of the truth table for a binary connective.
- Exercise: Explore all the 16 connectives.
- Do we need all 16 connectives to express a compound sentence?

The answer is *no*.

Consider the equivalence $((\alpha \supset \beta) \equiv (\neg\alpha \vee \beta))$.

Wherever we have $(\alpha \supset \beta)$ in a compound sentence,

we can substitute it with $(\neg\alpha \vee \beta)$.

In this way the connective $\supset$ can be eliminated.

Expressivity: Complete Sets of Connectives

It can be shown that the following sets of connectives are sufficient to express whatever can be expressed with other connectives.

{∧, ¬}, {∨, ¬}

{⊃, ¬} Gottlob Frege used this set in his *Propositional Calculus*

In practice we often use the set {∧, ∨, ¬} in our implementations.

Interestingly there are two singleton sets that are complete. They are,

NAND or Not-AND

α	β	(α ↑ β)
true	true	false
false	true	true
true	false	true
false	false	true

Also known as *Sheffer Stroke*

NOR or Not-OR or Joint Denial

α	β	(α ↓ β)
true	true	false
false	true	false
true	false	false
false	false	true

Also known as *Peirce's arrow*

Frege's Propositional Calculus

- Given a set of axioms / premises which set of rules suffices to derive all entailments?
- Gottlob Frege showed that just one rule Modus Ponens with the set of connectives {$\supset$, $\neg$} is complete, given the axioms below.
- Frege's Propositional Calculus is an axiomatic system that can derive all true statements or tautologies given the following axioms (accepted as true statements):

1. *THEN-1* $\alpha \supset (\beta \supset \alpha)$ *Note that α and β are variables.*
2. *THEN-2* $(\alpha \supset (\beta \supset \gamma)) \supset ((\alpha \supset \beta) \supset (\alpha \supset \gamma))$
3. *THEN-3* $(\alpha \supset (\beta \supset \gamma)) \supset (\beta \supset (\alpha \supset \gamma))$ *Redundant*
4. *FRG-1* $(\alpha \supset \beta) \supset (\neg\beta \supset \neg\alpha)$
5. *FRG-2* $\neg\neg\alpha \rightarrow \alpha$
6. *FRG-3* $\alpha \rightarrow \neg\neg\alpha$

https://en.wikipedia.org/wiki/Frege_propositional_calculus

Next

PROPOSITIONAL LOGIC