



INDIAN INSTITUTE OF
INFORMATION
TECHNOLOGY

Logic and AI (First-order and Propositional logics)

Dr. Animesh Chaturvedi

Assistant Professor: IIIT Dharwad

Young Researcher: Heidelberg Laureate Forum

Postdoc: King's College London & The Alan Turing Institute

PhD: IIT Indore MTech: IIITDM Jabalpur



Indian Institute of Technology Indore
भारतीय प्रौद्योगिकी संस्थान इंदौर



PDPM

Indian Institute of Information Technology,
Design and Manufacturing, Jabalpur

The
Alan Turing
Institute

Logic and AI

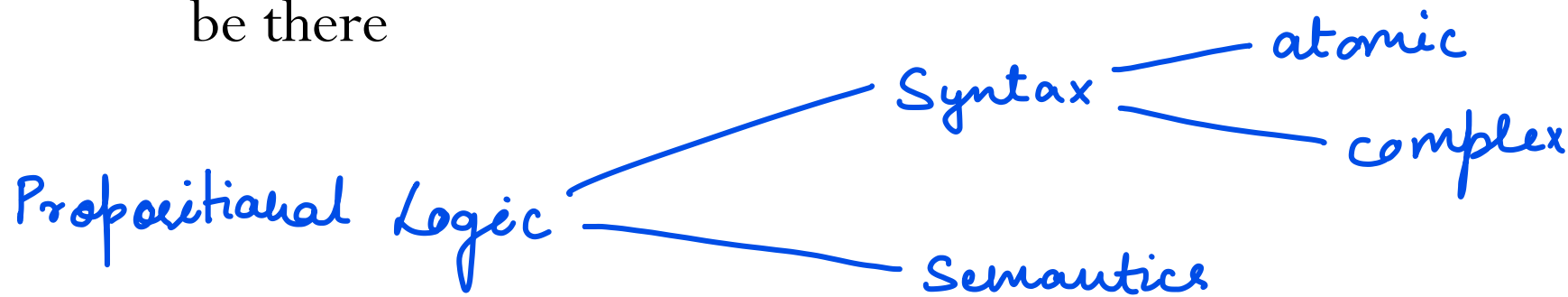
- Would like our AI to have **knowledge about the world**, and **logically draw conclusions** from it
- Search algorithms generate successors and evaluate them, but do not “understand” much about the setting
- Example question: is it possible for a chess player to have all her pawns and 2 queens?
 - Search algorithm could search through tons of states to see if this ever happens, but... *the algorithm doesn't understand the setting i.e game of chess has only 1 queen for each player.*

Syntax

We want our machines to be intelligent and aware of their setting thus we require some kind of

- What do well-formed sentences in the knowledge base look like?
- A BNF grammar: Backus Normal Form, an input language for
- $Symbol \rightarrow P, Q, R, \dots$, SH61 CLIs
- $Sentence \rightarrow \text{True} \mid \text{False} \mid Symbol \mid \text{NOT}(Sentence) \mid (Sentence \text{ AND } Sentence) \mid (Sentence \text{ OR } Sentence) \mid (Sentence \Rightarrow Sentence)$
- We will drop parentheses sometimes, but formally they really should always be there

knowledge Representation using PROPOSITIONAL LOGIC



*proposition is a statement that can be proved to be either true or false.

Semantics

- A model specifies which of the proposition symbols are true and which are false
- Given a model, I should be able to tell you whether a sentence is true or false
- Truth table defines semantics of operators:
- Given a model, can compute truth of sentence recursively with these

a	b	NOT(a)	a AND b	a OR b	a \Rightarrow b
false	false	true	false	false	true
false	true	true	false	true	true
true	false	false	false	true	false
true	true	false	true	true	true

Basic terms

- A **valid sentence** or **tautology** is a sentence that is True under all interpretations, no matter what the world is actually like or how the semantics are defined. Example: “It’s raining or it’s not raining.”
- An **inconsistent sentence** or **contradiction** is a sentence that is False under all interpretations. The world is never like what it describes, as in “It’s raining and it’s not raining.”
- **P entails Q**, written $P \models Q$, means that whenever P is True, so is Q. In other words, all models of P are also models of Q.
 → P implies Q

- TwoIsAnEvenNumber OR ThreeIsAnOddNumber

- is true (not exclusive OR)

- TwoIsAnOddNumber \Rightarrow ThreeIsAnEvenNumber
 S1 *S2*

- is true (if the left side is false it’s always true)

All of this is assuming those symbols are assigned their natural values...

S1 \Rightarrow S2 Truth Table

<i>S1</i>	<i>S2</i>	<i>S1 \Rightarrow S2</i>
T	T	T
T	F	F
F	T	T
F	F	T

→ our case

Tautologies

- A sentence is a **tautology** if it is true for any setting of its propositional symbols
- $(P \text{ OR } Q) \text{ OR } (\text{NOT}(P) \text{ AND } \text{NOT}(Q))$ is a tautology

P	Q	P OR Q	NOT(P) AND NOT(Q)	(P OR Q) OR (NOT(P) AND NOT(Q))
false	false	false	true	true
false	true	true	false	true
true	false	true	false	true
true	true	true	false	true

Logical equivalences

- Two sentences are **logically equivalent** if they have the same truth value for every setting of their propositional variables
- $P \text{ OR } Q$ and $\text{NOT}(\text{NOT}(P) \text{ AND } \text{NOT}(Q))$ are logically equivalent
- Tautology = logically equivalent to True

P	Q	P OR Q	NOT(NOT(P) AND NOT(Q))
false	false	false	false
false	true	true	true
true	false	true	true
true	true	true	true

Famous logical equivalences

- $(a \text{ OR } b) \equiv (b \text{ OR } a)$ *commutativity*
- $(a \text{ AND } b) \equiv (b \text{ AND } a)$ *commutativity*
- $((a \text{ AND } b) \text{ AND } c) \equiv (a \text{ AND } (b \text{ AND } c))$ *associativity*
- $((a \text{ OR } b) \text{ OR } c) \equiv (a \text{ OR } (b \text{ OR } c))$ *associativity*
- $\text{NOT}(\text{NOT}(a)) \equiv a$ *double-negation elimination*
- $(a \Rightarrow b) \equiv (\text{NOT}(b) \Rightarrow \text{NOT}(a))$ *contraposition*
- $(a \Rightarrow b) \equiv (\text{NOT}(a) \text{ OR } b)$ *implication elimination*
- $\text{NOT}(a \text{ AND } b) \equiv (\text{NOT}(a) \text{ OR } \text{NOT}(b))$ *De Morgan*
- $\text{NOT}(a \text{ OR } b) \equiv (\text{NOT}(a) \text{ AND } \text{NOT}(b))$ *De Morgan*
- $(a \text{ AND } (b \text{ OR } c)) \equiv ((a \text{ AND } b) \text{ OR } (a \text{ AND } c))$ *Distributivity*
- $(a \text{ OR } (b \text{ AND } c)) \equiv ((a \text{ OR } b) \text{ AND } (a \text{ OR } c))$ *Distributivity*

Reasoning patterns

- Obtain new sentences directly from some other sentences in knowledge base according to **reasoning patterns**
- If we have sentences a and $a \Rightarrow b$, we can correctly conclude the new sentence b

- This is called **modus ponens**

if $a \equiv T$ & $a \Rightarrow b \equiv T$ then b must be T .

a	b	$a \Rightarrow b$
T	T	T
T	F	F
F	T	T
F	F	T

→ our case

- A **Horn sentence** or **Horn clause** has the form:

$$P_1 \wedge P_2 \wedge P_3 \dots \wedge P_n \rightarrow Q_m \text{ where } n \geq 0, m \in \{0, 1\}$$

- If we have a AND b , we can correctly conclude a

- All of the logical equivalences from before also give reasoning patterns

$$\neg P_1 \vee \neg P_2 \dots \vee \neg P_n \vee Q_m \quad \underline{\text{OR}} \quad P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q_m \quad (P \rightarrow Q) = (\neg P \vee Q)$$

a Horn
Clause is
a clause
with at most
1 unnegated
literal,

Propositional logic (zeroth-order logic)

- **Logical constants:** true, false
- **Propositional symbols:** P, Q, S, ... (**atomic sentences**)
- Wrapping **parentheses:** (...)
- Sentences are combined by **connectives**:
 - \wedge ...and [conjunction]
 - \vee ...or [disjunction]
 - \Rightarrow ...implies [implication / conditional]
 - \Leftrightarrow ..is equivalent [biconditional]
 - \neg ...not [negation]
- **Literal:** atomic sentence or negated atomic sentence

Examples of PL sentences

- User defines a set of propositional symbols, like P and Q.
- User defines the **semantics** of each propositional symbol:
 - P means “It is hot”
 - Q means “It is humid”
 - R means “It is raining”
- $(P \wedge Q) \rightarrow R$
“If it is hot and humid, then it is raining”
- $Q \rightarrow P$
“If it is humid, then it is hot”
- A better way:
 - Hot = “It is hot”
 - Humid = “It is humid”
 - Raining = “It is raining”

Propositional logic (PL)

- A simple language useful for showing key ideas and definitions
- A sentence (well formed formula) is defined as follows:
 - A symbol is a sentence
 - If S is a sentence, then $\neg S$ is a sentence
 - If S is a sentence, then (S) is a sentence
 - If S and T are sentences, then $(S \vee T)$, $(S \wedge T)$, $(S \rightarrow T)$, and $(S \leftrightarrow T)$ are sentences
 - A sentence results from a finite number of applications of the above rules

Definition of Sentence : Same as the previous slide

A BNF grammar of sentences in propositional logic

$S := \langle \text{Sentence} \rangle ;$

$\langle \text{Sentence} \rangle := \langle \text{AtomicSentence} \rangle \mid \langle \text{ComplexSentence} \rangle ;$

$\langle \text{AtomicSentence} \rangle := \text{"TRUE"} \mid \text{"FALSE"} \mid \text{"P"} \mid \text{"Q"} \mid \text{"S"} ;$

\rightarrow A symbol is a sentence
 \rightarrow If S is sentence then (S) is a sentence

$\langle \text{ComplexSentence} \rangle := \text{"("} \langle \text{Sentence} \rangle \text{"}")} \mid$
 $\langle \text{Sentence} \rangle \langle \text{Connective} \rangle \langle \text{Sentence} \rangle \mid$

$\text{"NOT"} \langle \text{Sentence} \rangle ;$

\rightarrow If S is sentence then $\neg S$ is a sentence

$\langle \text{Connective} \rangle := \text{"NOT"} \mid \text{"AND"} \mid \text{"OR"} \mid \text{"IMPLIES"} \mid \text{"EQUIVALENT"} ;$

\rightarrow for sentences S and T, $(S \wedge T), (S \vee T), (S \Rightarrow T), (S \leftrightarrow T)$
are sentences

Modus ponens

- The form of a modus ponens argument resembles a syllogism, with two premises and a conclusion:
 - If P, then Q.
 - P.
 - Therefore, Q.
- It is a deductive argument form and rule of inference. It can be summarized as "P implies Q. P is true. Therefore Q must also be true."

first ←
premise

second
↓
premise

→ conclusion

Modus ponens Example

- for all x : Loves(John, x)
 - John loves every thing
- for all y : (Loves(y , Jane) \Rightarrow FeelsAppreciatedBy(Jane, y))
 - Jane feels appreciated by every thing that loves her
- Can infer from this:
 - FeelsAppreciatedBy(Jane, John)
- Here, we used the substitution $\{x/\text{Jane}, y/\text{John}\}$
 - Note we used different variables for the different sentences
- General UNIFY algorithms for finding a good substitution

means Loves(John, Jane)

Sound rules of inference

- Here are some examples of sound rules of inference
 - *A rule is sound if its conclusion is true whenever the premise is true*
- Each can be shown to be sound using a truth table

<u>RULE</u>	<u>PREMISE</u>	<u>CONCLUSION</u>
Modus Ponens	$A, A \rightarrow B$	B
And Introduction	A, B	$A \wedge B$
And Elimination	$A \wedge B$	A, B
Double Negation	$\neg\neg A$	A
Unit Resolution	$A \vee B, \neg B$	A
Resolution	$A \vee B, \neg B \vee C, A \vee C$	

Soundness of modus ponens

A	B	$A \rightarrow B$	OK?
True	True	True	✓
True	False	False	✓
False	True	True	✓
False	False	True	✓

Example

Resolution — Modus ponens
— chaining
— Contradiction Detection

- Consider the problem of representing the following information:
 - Every person is mortal.
 - Confucius is a person.
 - Confucius is mortal.
- How can these sentences be represented so that we can infer the third sentence from the first two?
- In PL we have to create propositional symbols to stand for all or part of each sentence.
 - $P = \text{"person"}; Q = \text{"mortal"}; R = \text{"Confucius"}$
 - so the above 3 sentences are represented as:
 - $P \rightarrow Q; R \rightarrow P; R \rightarrow Q \rightarrow \text{chaining}$
- Although the third sentence is entailed by the first two, we needed an explicit symbol, R, to represent an individual, Confucius, who is a member of the classes "person" and "mortal"
- To represent other individuals we must introduce separate symbols for each one, with some way to represent the fact that all individuals who are "people" are also "mortal"

Propositional logic is a weak language

- Hard to identify “individuals” (e.g., Mary, 3)
- Can’t directly talk about properties of individuals or relations between individuals (e.g., “Bill is tall”)
- Generalizations, patterns, regularities can’t easily be represented (e.g., “all triangles have 3 sides”)
- First-Order Logic (abbreviated FOL or FOPC) is expressive enough to concisely represent this kind of information

FOL adds relations, variables, and quantifiers, e.g.,

- “Every elephant is gray”: $\forall x (\text{elephant}(x) \rightarrow \text{gray}(x))$
- “There is a white alligator”: $\exists x (\text{alligator}(x) \wedge \text{white}(x))$

Limitations and Problems of propositional logic

- No notion of objects
- No notion of relations among objects
- Some English statements are hard to model in propositional logic.
- Zeroth-order logic (Propositional Logic) is
 - first-order logic without variables or quantifiers.

First-order logic

- First-order logic (FOL) models the world in terms of
 - **Objects**, which are things with individual identities
 - **Properties** of objects that distinguish them from other objects
 - **Relations** that hold among sets of objects
 - **Functions**, which are a subset of relations where there is only one “value” for any given “input”
- Examples:
 - Objects: Students, lectures, companies, cars ...
 - Relations: Brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes, ...
 - Properties: blue, oval, even, large, ...
 - Functions: father-of, best-friend, second-half, one-more-than ...

Reasoning about many objects at once

- **Variables**: x, y, z, \dots can refer to multiple objects
- New operators “for all” and “there exists”
 - **Universal quantifier** and **existential quantifier**
- for all x : $\text{CompletelyWhite}(x) \Rightarrow \text{NOT}(\text{PartiallyBlack}(x))$
 - Completely white objects are never partially black
- there exists x : $\text{PartiallyWhite}(x) \text{ AND } \text{PartiallyBlack}(x)$
 - There exists some object in the world that is partially white and partially black

KB \equiv Knowledge Base

Inference rules

- **Logical inference** is used to create new sentences that logically follow from a given set of predicate calculus sentences (KB).
- An inference rule is **sound** if every sentence X produced by an inference rule operating on a KB logically follows from the KB. (That is, the inference rule does not create any contradictions)
- An inference rule is **complete** if it is able to produce every expression that logically follows from (is entailed by) the KB. (Note the analogy to complete search algorithms.)

Limitations of First-Order Logic

- We are **not** allowed to reason in general about relations and functions
- We need **higher-order logic** (which is more powerful).
- “If a property is inherited by children, then for any thing, if that property is true of it, it must also be true for any child of it” → *Not applicable in FOL*
- **Axioms**: basic facts about the domain, our “initial” knowledge base
- **Theorems**: statements that are logically derived from axioms

Thinking Rationally

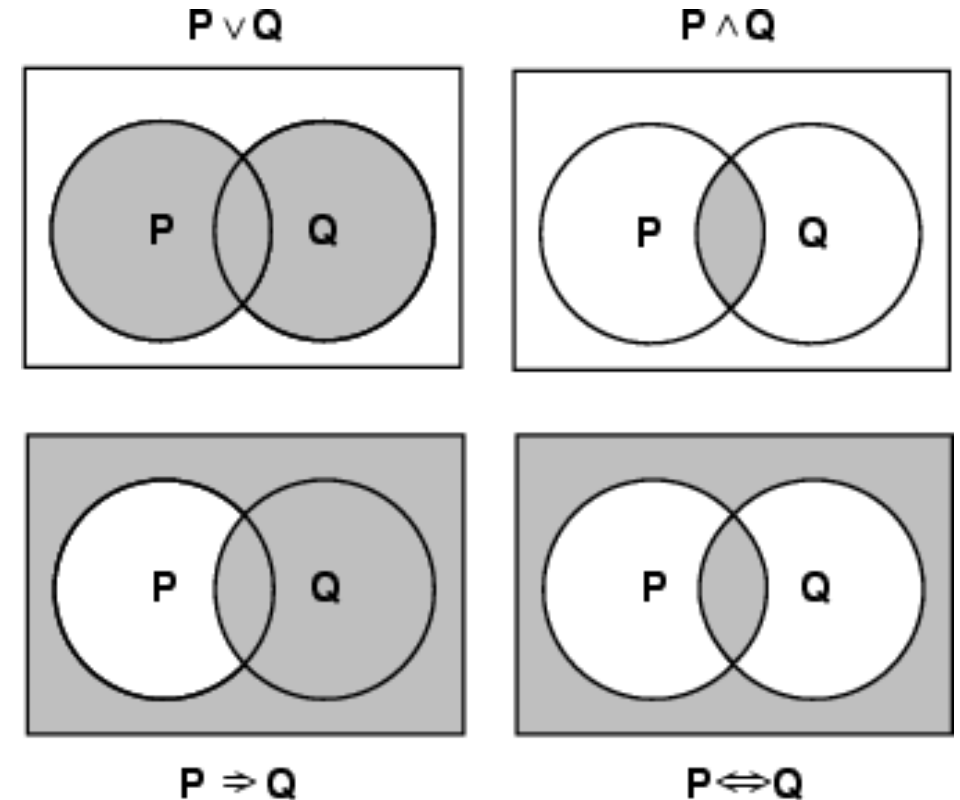
- **Right Thinking:** “arguments structures that always gave correct conclusions given correct premises”
- **Syllogism:** “Socrates is a man; all men are mortal therefore Socrates is mortal.”
- **Field of Logic:** Laws of thought were supposed to govern the operation of the mind.
 - **Logical notation** to find solution to a problem.
 - Finding solution to all kinds of things in the world and the relations between them.
- **Issue 1:** Not easy to take informal knowledge and state it in the formal terms required by Logical notation, particularly when the knowledge is less than 100% certain.
- **Issue 2:** There is a big difference between being able to solve a problem “in principle” and doing so in practice. Means proposing algorithm and it’s coding are different problems.
- Power of the representation and reasoning systems

Machine Acting V/S Thinking?

- Weak AI Hypothesis vs. Strong AI Hypothesis
 - Weak Hyp: machines could act as if they are intelligent
 - Strong Hyp: machines that act intelligent have to think intelligently too
- Hypothesis:
 - “a supposition or **proposed** explanation made on the basis of limited evidence as a starting point for further investigation”
 - “a **proposition** made as a basis for reasoning, without any assumption of its truth”
- Machine learning:
 - “algorithms to build model using data for training, which makes machine capable enough to make **predictions** or decisions without being explicitly programmed”
- What is Machine learning Hypothesis?— *a predicted function that explains one past data, closely resembling the actual function but not equivalent to it.*


Models of complex sentences

- The meaning or **semantics** of a sentence determines its **interpretation**. *meaning*
- Given the truth values of all symbols in a sentence, it can be “evaluated” to determine its **truth value** (True or False).
- A **model** for a KB is a “possible world” (assignment of truth values to propositional symbols) in which each sentence in the KB is True.

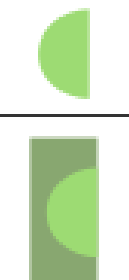


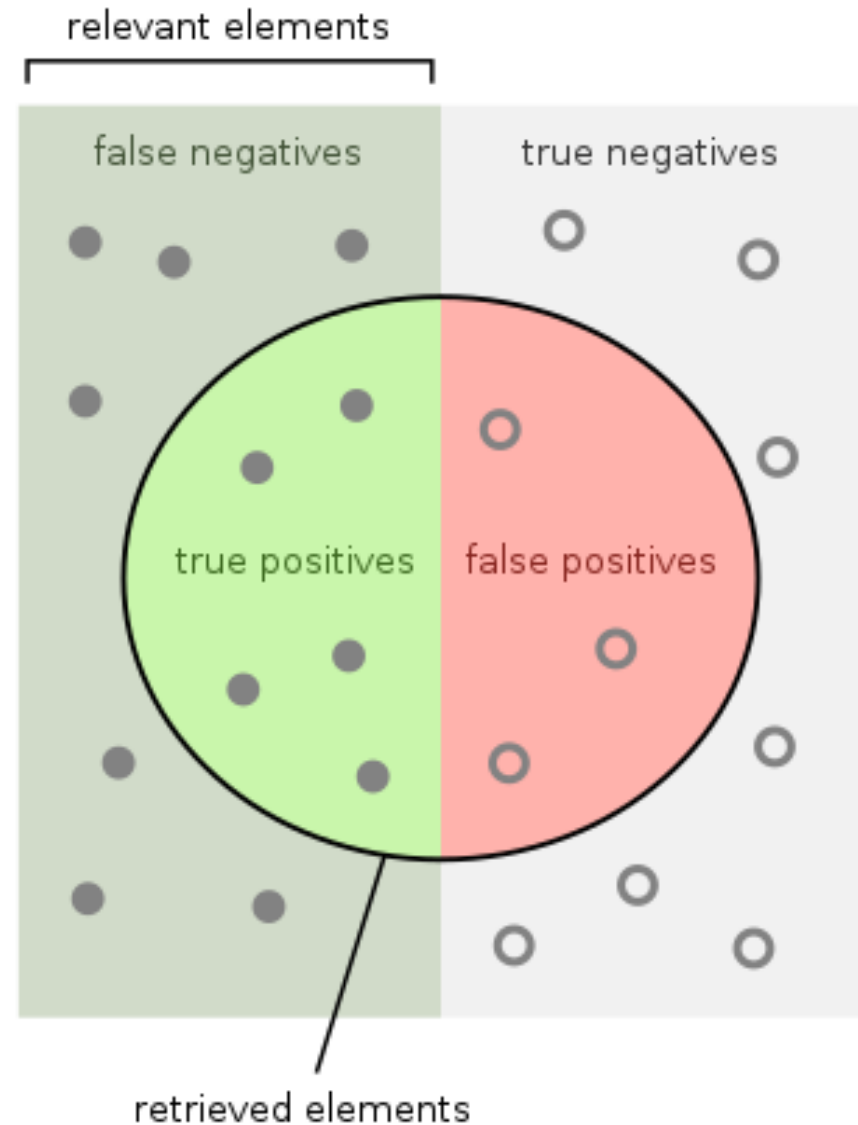
Precision and Recall

How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$


How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$




Precision and Recall

- A set of retrieved documents
 - (e.g. the list of documents produced by a web search engine for a query)
- A set of relevant documents
 - (e.g. the list of all documents on the internet that are relevant for a certain topic)

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Logic and Artificial Intelligence

1. Logic and Artificial Intelligence
2. John McCarthy and Common Sense Logicism
3. Nonmonotonic Reasoning and Nonmonotonic Logics
4. Reasoning about Action and Change
5. Causal Reasoning
6. Spatial Reasoning
7. Reasoning about Knowledge
8. Towards a Formalization of Common Sense
9. Logical Approaches to Natural Language and Communication
10. Taxonomic Representation and Reasoning
11. Contextual Reasoning
12. Prospects for a Logical Theory of Practical Reason

Stanford Encyclopedia of Philosophy

Logic and Artificial Intelligence

Summary

- The process of deriving new sentences from old one is called **inference**.
 - **Sound** inference processes derives true conclusions given true premises
 - **Complete** inference processes derive all true conclusions from a set of premises
- A **valid sentence** is true in all worlds under all interpretations
- If an implication sentence can be shown to be valid, then—given its premise—its consequent can be derived
- Different logics make different **commitments** about what the world is made of and what kind of beliefs we can have regarding the facts
 - Logics are useful for the commitments they do not make because lack of commitment gives the knowledge base engineer more freedom
- **Propositional logic** commits only to the existence of facts that may or may not be the case in the world being represented
 - It has a simple syntax and simple semantics. It suffices to illustrate the process of inference
 - Propositional logic quickly becomes impractical, even for very small worlds

References

- Vincent Conitzer, Logic CPS 270: Artificial Intelligence
<http://www.cs.duke.edu/courses/fall08/cps270/>
- Vincent Conitzer, First-Order Logic CPS 270: Artificial Intelligence
<http://www.cs.duke.edu/courses/fall08/cps270/>
- Some material adopted from notes and slides by Tim Finin, Marie desJardins, Andreas Geyer-Schulz and Chuck Dyer
- Wikipedia contents
 - https://en.wikipedia.org/wiki/Precision_and_recall
 - https://en.wikipedia.org/wiki/Confusion_matrix
- Stuart Russel, and Peter Norvig. "Artificial intelligence: A modern approach. Third edit." Upper Saddle River, New Jersey 7458 (2015). <http://aima.cs.berkeley.edu/>

ขอบคุณ

Thai

Grazie
Italian

תודה רבה
Hebrew

धन्यवादः
Sanskrit

ಧನ್ಯವಾದಗಳು
Kannada

Ευχαριστώ
Greek

Thank You
English

Gracias
Spanish

Спасибо
Russian

Obrigado
Portuguese

شكراً
Arabic

<https://sites.google.com/site/animeshchaturvedi07>

Merci
French

多謝
Traditional
Chinese

धन्यवाद
Hindi

Danke
German

多谢
Simplified
Chinese

நன்றி
Tamil

ありがとうございました
Japanese

감사합니다
Korean