Artificial Intelligence

Module 4: Logic and Deduction

PART 4.1: Knowledge Representation
PART 4.2: Propositional logic

Dr. Chandra Prakash
Assistant Professor
Department of Computer Science and Engineering

(Slides adapted from StuartJ. Russell, B Ravindran, Mausam, Dan Klein and Pieter Abbeel, Partha P Chakrabarti, Saikishor Jangiti)

Module 4: Logic and Deduction

AI Problem Areas /Tasks

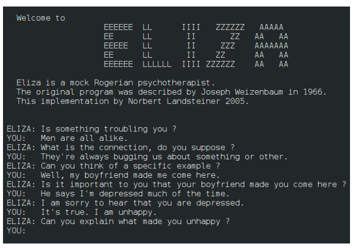
- **1st Generation of AI :** Fomal cognitive Tasks
  - Game
    - Tic-Tac-Toe
    - Chess
    - Checkers
    - Go
  - Mathematics
    - Logic
    - Geometry
    - Calculus
    - Proving properties of programs
- **2nd Generation :** Expert Tasks
  - Knowledge Represenation
  - Enginnering
    - Design
    - fault finding
    - Manufacturing planning
  - Medical
    - Diagnosis
    - Medical Image Analysis
  - Finanical
    - Stock market prdeictions
- **3rd Generation of AI :** Perceptual Tasks
  - Perception
    - Vision
    - Speech
  - Natual Language
    - Understanding
    - Generation
    - Translation
  - Robot Control

Exercise : Identify the Product

Logical Agents

Logical Agent

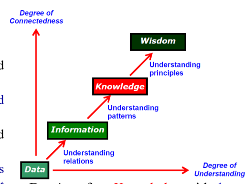
- Knowledge can be defined as the body of facts and principles accumulated by human-kind or the act, fact or state of knowing.
- includes having a familiarity with
  - language, concepts,
  - procedures, rules,
  - ideas, abstractions ,
  - places, customs,
  - facts associations
  - ability to use these notions effectively in modeling different aspects of the world.



✓ Represent the knowledge in state-space
✓ Reason about the solution in logical steps

Knowledge Model

- **Facts**
- **Data** is viewed as collection of disconnected facts.
  - Example : It is raining.
- **Information** emerges when relationships among facts are established and understood; Provides answers to "who", "what", "where", and "when".
  - Example : The temperature dropped 15 degrees and then it started raining.
- **Knowledge** emerges when relationships among patterns are identified and understood; Provides answers as "how" .
  - Example : If the humidity is very high and the temperature drops substantially, then atmospheres is unlikely to hold the moisture, so it rains.
- **Wisdom** is the pinnacle of understanding, uncovers the principles of relationships that describe patterns. Provides answers as "why"
  - Example : Encompasses understanding of all the interactions that happen between raining, evaporation, air currents, temperature gradients, changes, and raining.



- Dont' confuse Knowledge with data.
- Epistemology – study of nature of knowledge.
- Metaknowldge – knowledge about knowledge.

State-space representation

- In Search Strategies, the states are represented as
  - Atomic representation
  - Treated as black box
  - Many factors are ignored, e.g., gas in car, money available for toll, etc.
- To include more factors in our state representation, we have
  - Factored Representation
  - Each state will be represented with
    - Factors
    - Each factor can be a Boolean or real-valued variable
    - E.g., City Arad can now be represented as
    - {"lat": 46.1866, "lon": 21.3123, "has_gas_till_next_station": yes}

Factored Representation

- Why do we need Factored representation
  - To reason about steps
  - To learn new knowledge about the environment
  - To adapt to changes to the existing knowledge
  - Accept new tasks in the form of explicit goals
  - To overcome partial observability of environment

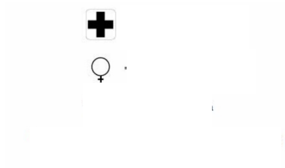
  - E.g., Arad to Bucharest via Sibiu
  - If Sibiu has road blocked, then the agent should take a different path
  - i.e., to build Knowledge-based Agents

Knowledge-based Agents

- **Solving problems by**
  - Representing the knowledge in state-space
  - Reasoning about the solution in logical steps
- **Central Component:**
  - Knowledge Base (KB)
    - set of sentences, not the English sentences.,
  - Represents some assertion about the world
- **Sentence**
  - representation of knowledge in a language called Knowledge representation language
  - Represents an axiom, when the sentence is taken as given without being derived from other sentences
    - TELL operation: Add new sentences to the knowledge base
    - ASK operation: Query what is known

What is representation

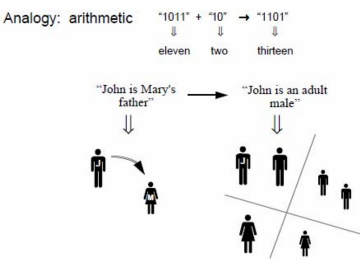
- Symbols standing for things in the world



- Knowledge represenation :
  - symbolic encoding of preposition beloved (by some agent)

What is reasoning

- Manipulation of symbols encoding propositions to produce representations of new propositions
- Benefits of Reasoning
  - Given
    - Patient X allergic to medication M
    - Anyone allergic to medication M is also allergic to medication M'
  - Reasoning helps us derive
    - Patient X is allergic to medication M'

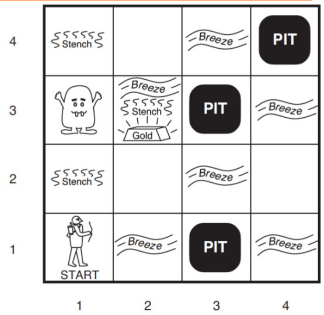


Knowledge-based Agents

- Each time a knowledge-based agent is called, it does three things
  - TELL – the Knowledge Base about the percepts (inputs)
  - ASK – the Knowledge Base what action it should perform.
    - Extensive reasoning about the outcomes of possible action and current state
  - TELL – the Knowledge Base about selected action (to update) and agent executes the action
- The agent must
  - Represent states and actions
  - Incorporate new percepts
  - Update internal representations of the world
  - Deduce hidden properties of the world
  - Deduce appropriate actions

Example: Knowledge-based Agents

- Wumpus World



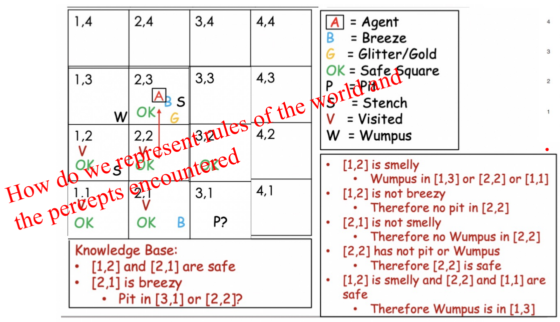
Knowledge-based Agents: Wumpus World PEAS description

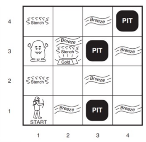
- Performance measure
  - gold +1000, death -1000
  - -1 per step, -10 for using the arrow
- Environment
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square
- **Sensors:** Stench, Breeze, Glitter, Bump, Scream
- **Actuators:** Left turn, Right turn, Forward, Grab, Release, Shoot

Wumpus world characterization

- Fully Observable
  - No - only local perception
- Deterministic
  - Yes – outcomes exactly specified
- Episodic
  - No – sequential at the level of actions
- Static
  - Yes – Wumpus and Pits do not move
- Discrete
  - Yes
- Single-agent?
  - Yes – Wumpus is essentially a natural feature

Exploring a Wumpus World

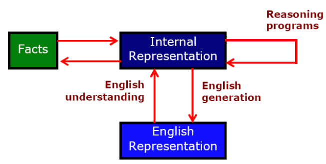




- Logical Agents:
  - combine the current percept with the existing knowledge (percept history) and
  - do logical reasoning in identifying state information

Knowledge Representation (KR)

- Knowledge is a description of the world;
- Knowledge can be of following types
  - Declarative (statements)
  - Procedural (facts)
  - Heuristics (rule of thumb / experience)
- Representation is the way knowledge is encoded
- Represent knowledge about the world in a manner that facilitates inferencing
  - (drawing conclusions) from knowledge.
- Example: Arithmetic logic
  - x >= 5
  - i,e x=7 is a model of world
- In AI: typically based on
  - Logic
  - Probability
  - Logic and Probability

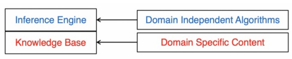


Components of KR

- Syntax:
  - defines the sentences in the language
- Semantics:
  - defines the "meaning" to sentences
- Inference Procedure
  - Algorithm
  - Sound?
  - Complete?
  - Complexity
- Knowledge Base
  - Set of facts associated with a particualr problem or situation

Knowledge bases

- **Two main components**
  - **Knowledge base** = set of sentences in a formal language
    - Represents some assertion about the world
  - **Inference Engine**
    - Mechanism to derive new sentences from old
    - Declarative approach to building an agent (or other system):
      - Tell it what it needs to know
      - Then it can Ask itself what to do - answers should follow from the KB
    - Both "Tell" and "Ask" involve inference – deriving new sentences from old



- Agents can be viewed at the knowledge level
  - i.e., what they know, regardless of how implemented
- Or at the implementation level
  - i.e., data structures in KB and algorithms that manipulate them

Logical Representation

- Logic - formal languages for representing information such that conclusions can be drawn
- Sentence – individual piece of knowledge
- Logic includes : Syntax , Semantics and Inference Procedure.
- **Syntax** – What expressions are legal?
  - In the language of arithmetic $x + 2 \geq y$ is a sentence (well-formed sentence) , but $x2 + y \geq$ is not a sentence
- **Semantics** – What the legal expressions mean?
  - $x + 2 \geq y$ is true iff the number $x + 2$ is no less than the number y
  - $x + 2 \geq y$ is true in a world where $x = 5, y = 4$
  - $x + 2 \geq y$ is false in a world where $x = 3, y = 7$
- **Inference Procedure :**

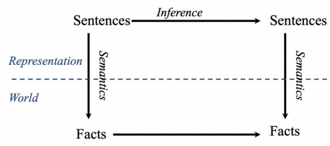
  Specifies methods for computing new sentences from an existing sentences.

Logic

- Reasoning conducted or assessed according to available knowledge about the domain
- **Syntax**: Every sentence in our knowledge base are expressed according to a syntax of the representing language.
  - E.g., in Arithmetic,
  - "x + y = 4" is a well-formed sentence, "xy+4=" is not
- Atoms
  - P, Q, R, …
- Literals
  - P, ¬P
- Sentences
  - Any literal is a sentence
  - If S is a sentence
    - Then (S ∧ S) is a sentence
    - Then (S ∨ S) is a sentence
- Conveniences
  - P → Q same as ¬P ∨ Q

Logic: World to Representation

- **Syntax:** which arrangements of symbols are legal
  - (Def "sentences")
- **Semantics:** what the symbols mean in the world
  - (Mapping between symbols and worlds)
  - Meaning of sqentence.
  - They define the truth of a sentence according to a possible world.
  - E.g., in a world where x=2, y=2, "x + y = 4" is a true sentence, whereas in a world where x=1, y=1, the same sentence is not true
  - In standard Logic, every sentence has to be either True or False but cannot be in between

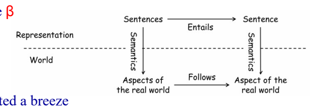


Logic – Model

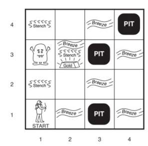
- Formally structured worlds with respect to which truth can be evaluated
- Model is Any possible world. Not necessarily the reality. Any combination of assignment of truth values to sentences in our KB.
- Models are mathematical abstractions
  - We say *m* is a model of a sentence α if α is true in *m*
  - M(α) is the set of all models of α
  - Example: $x + 2 \geq y$ is true in a world
    - where x = 5, y = 4
    - where x = 4, y = 4
- E.g., Two sentences S1 and S2 in our KB.
- Possible models:
  - m1: {S1: True, S2: True}, m2: {S1: False, S2: True}, m3: {S1:True, S2: False}, m4: {S1:False, S2:False}
- S1 is true in m1, m3 →i.e m1 satisfies S1 or m3 satisfies S1
- M(S1) is set of models that satisfy S1, i.e., [m1, m3]

Logic – Entailment

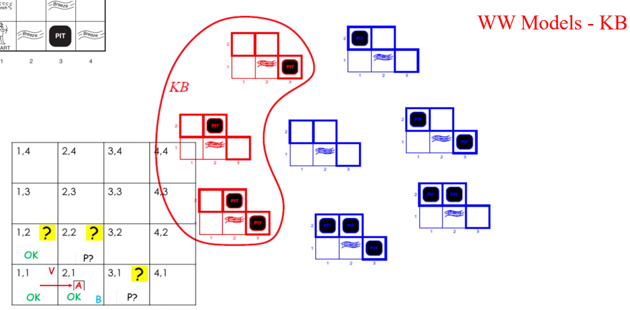
- Means that one thing follows from another: KB ⊨ α
- Knowledge Base KB entails a sentence α iff α is true in all worlds where KB is true
- Relationship between sentences that is based on semantics
- Logical Reasoning: Logical entailment between sentences
  - i.e., a sentence follows from another sentence
  - α\|=β, it means that sentence α entails the sentence β
  - In every model where α is true, β is also true
  - α \|=β if and only if M(α) ⊆ M(β)



- **Example**:
  - In Wumpus World, the agent is in [2,1] and detected a breeze
  - The agent is interested in squares [1, 2], [2, 2], [3, 1] for next move.
  - Now, each square might or might not contain a pit (total 2^3 = 8 possible models)
  - Our KB tells us that in [1, 1] we didn't receive a breeze and hence [2, 1] doesn't have a pit



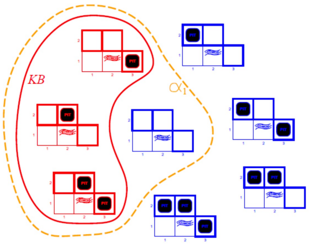
8 possible models



Entailment by model checking

- Define α_1= "There is no pit in [1, 2]"
- In every model, where KB is true, α_1 is also true
- α_1 = "[1, 2] is safe"
- Hence, KB $\models \alpha_1$
- M(KB) $\subseteq$ M(α_1)

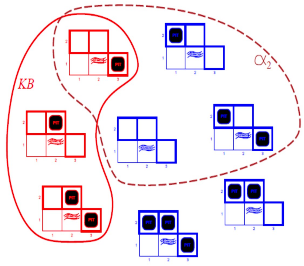
- **Model Checking:** This enumeration of all models to verify logical inference (entailment) is called Model Checking



Logic – WW models - Entailment

- Define α_2= "There is no pit in [2, 2]"
- Here, α_2 is true in models where KB is not true

- Hence, KB $\not\models \alpha_2$



Logic – Entailment

- **Sound**
  - An inference algorithm that derives only entailed sentences
  - E.g., Model Checking is a sound inference procedure
- **Completeness**
  - An inference algorithm is complete if it can derive any sentence that is entailed
- If KB is true, any sentence derived using Sound inference procedure is also true in real world.
- Eg:
  - KB:
    - Learning is maximum in fun classes
    - AI class is a fun class
  - Entails : ??
    - Learning is maximum in AI class

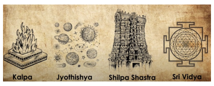
Logic - Grounding

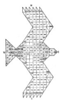
- Grounding
  - connection between logical reasoning and real environment in which agent exists
- *How do we know that KB is true in the real world?*
- Agent's sensors create the connection
- Agent creates sentence from percept (e.g., smell in wumpus world)
- This sentence when stored in Knowledge base will be true in real world
- However, the existing Knowledge Base's sentences like "Wumpus cause smell in adjacent squares" is not a direct representation of single percept but is a general rule.
- Such rules can be learned.

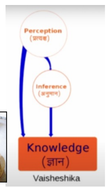
Logic in Ancient Times

Indic
Geometry, Calculations
Nyaya, Vaisisekha
Theory of Argumentation (शास्त्रार्थ)
Sanskrit language with Binary-Level arguments
Logical Argumentation: Chatustoki
Buddhist and Jain Philosophies
Formal Systems
Vedanta

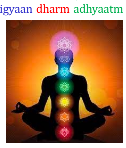
China
Confucious, Mozi,
Master Mo (Mohist School)
Basic Formal Systems
Buddhist Systems from India

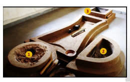


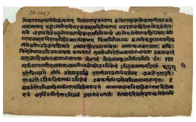




विज्ञान धर्म अध्यात्म
vigyaan dharm adhyaatm







Logic in Ancient Times

Indic
Geometry, Calculations
Nyaya, Vaisisekha
Theory of Argumentation
Sanskrit language with Binary-Level arguments
Logical Argumentation: Chatustoki
Buddhist and Jain Philosophies
Formal Systems
Vedanta

China
Confucious, Mozi,
Master Mo (Mohist School)
Basic Formal Systems
Buddhist Systems from India

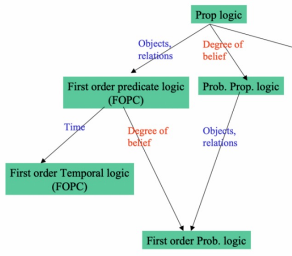
Greek
Thales, Pythagoras (Propositions and Geometry)
Heraclitus, Parmenides (Logos)
Plato (Logic beyond Geometry)
Aristotle (Syllogism, Syntax)
Stoics

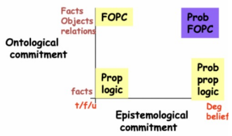
Middle East
Ancient Egypt, Babylon
Arab (Avisennian Logic)
Inductive Logic

Medieval Europe
Post Aristotle
Precursor to First Order Logic

Today
Propositional
Predicate
Higher Order
Logic, Numbers & Computation
Psychology
Philosophy
Brain/ Neural Network

Common KR Languages in AI





- Propositional Logic
- Predicate Calculus
- Frame Systems
- Rules with Certainty Factors
- Bayesian Belief Networks
- Influence Diagrams
- Ontologies
- Semantic Networks
- Concept Description Languages
- Non-monotonic Logic
- Fuzzy Logic

Logic

- **Assumptions about Knowledge Representation (KR) :**
  - Intelligent Behaviour can be achieved by manipulation languages of symbol structures.
  - KR are designed to facilitate operations over symbol structures, have precise syntax and semantics;
  - Make Inferences, draw new conclusions from existing facts.
- To satisfy these assumptions about KR, we need formal notation *that allow* automated inference and problem solving. ***One popular choice is use of logic.***
- Logic is a language for reasoning, a collection of rules used while doing logical reasoning.
- Language of Logic : way of representing fact
- Logic is studied as Knowledge Representation languages in AI.
- Propositional logic and Predicate logic are fundamental to all logic.
  1) **Propositional logic** -is the study of statements and their connectivity.
  - Propositional: p⟹p
  2) **Predicate Logic**- is the study of individuals and their properties.
  - Predicate: ∀x:p(x)⟹p(x)
- They represent things and allow more or less efficient inference.

Representing simple facts in logic

To present real world facts as logical prepositions written as well-formed formulas (wff)

If it is hot it will rain.

=> hot(X)→ rain(X)

Propositional Logic

- A proposition is a statement, which in English would be a declarative sentence.
- Every proposition is either TRUE or FALSE.

Examples:

(a) The sky is blue.
(b) Snow is cold.
(c) 12 * 12=144

- Propositions are “sentences” , either true or false but not both.
- A sentence is smallest unit in propositional logic.
- If proposition is true, then truth value is "true"
  - if proposition is false, then truth value is "false"

PROPOSITIONAL LOGIC

- A simple representation language for building knowledge-based agents
- Proposition Symbol – A symbol that stands for a proposition.
  - E.g., $W_{1,3}$ – “Wumpus in [1,3]” is a proposition and $W_{1,3}$ is the symbol
- **Syntax:**
  - Atomic sentences: P, Q, …
    - Usually represented with uppercase letter and may contain other letters or subscripts. E.g., P, Q, R, W1,3, North
  - Connectives: ~ / ¬, ∧, ∨, →
    - Large, compound or complex statement are constructed from basic propositions by combining them with connectives.
- **Semantics**
  - Truth Tables : True and False
- **Inference**
  - Modus Ponens
  - Resolution

BNF (Backus–Naur Form) grammar of sentences in propositional logic

Sentence → *AtomicSentence* | *ComplexSentence*

AtomicSentence → *True* | *False* | *P* | *Q* | *R* | ...

ComplexSentence → (*Sentence*) | [*Sentence*]
| ¬ *Sentence*
| *Sentence* ∧ *Sentence*
| *Sentence* ∨ *Sentence*
| *Sentence* ⇒ *Sentence*
| *Sentence* ⇔ *Sentence*

OPERATOR PRECEDENCE : ¬, ∧, ∨, ⇒, ⇔

Connectives and Symbols in decreasing order of operation priority

Connective	Symbols					Read as
assertion	P					"p is true"
negation	¬p	~	!		NOT	"p is false"
conjunction	p ∧ q	·	&&	&	AND	"both p and q are true"
disjunction	P ∨ q	||	|		OR	"either p is true, or q is true, or both "
implication	p → q	⊃	⇒		if ..then	"if p is true, then q is true" " p implies q "
equivalence	↔	≡	⇔		if and only if	"p and q are either both true or both false"

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

Propositional Theorem Proving

- Theorem Proving
  - applying rules of inference directly to sentences in our KB to prove query sentence without consulting models
- Logical Equivalence
  - two sentences α and β are logically equivalent if they are true in the same set of models denoted as $\alpha \equiv \beta$

Propositional Theorem Proving - Logical Equivalence Laws

- Two sentences are logically equivalent iff true in same models:
- $\alpha \equiv \beta$ if and only if $\alpha \models \beta$ and $\beta \models \alpha$

$$
\begin{array}{rcll}
(\alpha \wedge \beta) & \equiv & (\beta \wedge \alpha) & \text{commutativity of } \wedge \\
(\alpha \vee \beta) & \equiv & (\beta \vee \alpha) & \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) & \equiv & (\alpha \wedge (\beta \wedge \gamma)) & \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) & \equiv & (\alpha \vee (\beta \vee \gamma)) & \text{associativity of } \vee \\
\neg(\neg \alpha) & \equiv & \alpha & \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg \beta \Rightarrow \neg \alpha) & \text{contraposition} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg \alpha \vee \beta) & \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) & \equiv & ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) & \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) & \equiv & (\neg \alpha \vee \neg \beta) & \text{De Morgan} \\
\neg(\alpha \vee \beta) & \equiv & (\neg \alpha \wedge \neg \beta) & \text{De Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) & \equiv & ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) & \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) & \equiv & ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) & \text{distributivity of } \vee \text{ over } \wedge
\end{array}
$$

Inference Rules and Proffs

- Inference rules that can be applied to derive a proof
  - a chain of conclusions that leads to the desired goal.
- **Inference Rules:**
  - Modus Ponens (Latin for mode that affirms) and is written $\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta}$
    - if (WumpusAhead ∧WumpusAlive) ⇒ Shoot and (WumpusAhead ∧ WumpusAlive) are given, then Shoot can be inferred.
  - And-Elimination, says that, from a conjunction, any of the conjuncts can be inferred : $\frac{\alpha \wedge \beta}{\alpha}$
    - from (WumpusAhead ∧ WumpusAlive), WumpusAlive can be inferred.
  - Resolution Technique :
    - assumes sentences are in conjunctive normal form (CNF) – conjunction of clauses,
    - e.g. $(\neg B1,1 \vee P1,2 \vee P2,1) \wedge (\neg P1,2 \vee B1,1) \wedge (\neg P2,1 \vee B1,1)$

Preposition Examples

- If I am the Student President then I am well-known. I am the Student President. So I am wellknown.
- If I am the Student President then I am well-known. I am not the Student President. So I am not well-known.
- If Rajat is the Student President then Rajat is well-known. Rajat is the Student President. So Rajat is well known.
- If Asha is elected VP then Rajat is chosen as G-Sec and Bharati is chosen as Treasurer. Rajat is not chosen as G-Sec. Therefore Asha is not elected VP.
- If Asha is elected VP then Rajat is chosen as G-Sec and Bharati is chosen as Treasurer. Rajat is chosen as G-Sec. Therefore Asha is elected VP.

Deduction Using Propositional Logic: Steps

- **Atoms**
  - lower-case letters, *p, q, r, are symbols for simple statements.*
  - Choice of Boolean Variables `a, b, c, d, …`
  - which can take values true or false.
- Boolean Formulae developed using well defined connectors `~/¬, ∧, ∨, →,⇔` etc, whose meaning (semantics) is given by their truth tables.
  - P → Q same as ¬P ∨ Q
- Codification of Sentences of the argument into Boolean Formulae.
- Developing the Deduction Process as obtaining truth of a Combined Formula expressing the complete argument.
- Determining the Truth or Validity of the formula and thereby proving or disproving the argument and Analyzing its truth under various Interpretations.

Deduction Using Propositional Logic: Example 1

Atoms Choice of Boolean Variables `a, b, c, d, …` which can take values true or false.

Boolean Formulae developed using well defined connectors `~/¬, ∧, ∨, →,⇔` etc, whose meaning (semantics) is given by their truth tables.

P → Q same as ¬P ∨ Q

Codification of Sentences of the argument into Boolean Formulae.

Developing the Deduction Process as obtaining truth of a Combined Formula expressing the complete argument.

Determining the Truth or Validity of the formula and thereby proving or disproving the argument and Analyzing its truth under various Interpretations.

If I am the Student President then I am well-known. I am the Student President. So I am well-known

Coding: Variables

a: I am the Student President

b: I am well-known

Coding the sentences:

F1: $a \to b$

F2: a

G: b

The final formula for deduction: $(F1 \wedge F2) \to G$,

that is: $((a \to b) \wedge a) \to b$

Deduction Using Propositional Logic: Example 1

Atoms Choice of Boolean Variables a, b, c, d, … which can take values true or false.

Boolean Formulae developed using well defined connectors ~/¬, ∧, ∨, →, ⇔ etc, whose meaning (semantics) is given by their truth tables.

$P \rightarrow Q$ same as $\neg P \vee Q$

Codification of Sentences of the argument into Boolean Formulae.

Developing the Deduction Process as obtaining truth of a Combined Formula expressing the complete argument.

Determining the Truth or Validity of the formula and thereby proving or disproving the argument and Analyzing its truth under various Interpretations.

If I am the Student President then I am well-known. I am the Student President. So I am well-known

Coding: Variables

a: I am the Student President

b: I am well-known

Coding the sentences:

F1: $a \rightarrow b$

F2: a

G: b

The final formula for deduction: $(F1 \wedge F2) \rightarrow G$, that is: $((a \rightarrow b) \wedge a) \rightarrow b$

a	b	a → b	(a → b) ∧ a	((a → b) ∧ a) → b
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

Deduction Using Propositional Logic: Example 2

Atoms Choice of Boolean Variables a, b, c, d, … which can take values true or false.

Boolean Formulae developed using well defined connectors ~/¬, ∧, ∨, →, ⇔ etc, whose meaning (semantics) is given by their truth tables.

$P \rightarrow Q$ same as $\neg P \vee Q$

Codification of Sentences of the argument into Boolean Formulae.

Developing the Deduction Process as obtaining truth of a Combined Formula expressing the complete argument.

Determining the Truth or Validity of the formula and thereby proving or disproving the argument and Analyzing its truth under various Interpretations.

If I am the Student President then I am well-known. I am not the Student President. So I am not well-known

Coding: Variables

a: I am the Student President

b: I am well-known

Coding the sentences:

F1: $a \rightarrow b$

F2: ~a

G: ~b

The final formula for deduction: $(F1 \wedge F2) \rightarrow G$,

that is: $((a \rightarrow b) \wedge \sim a) \rightarrow \sim b$

a	b	a → b	(a → b) ∧ ~a	((a → b) ∧ ~a) → ~b
T	T	T	F	T
T	F	F	F	T
F	T	T	T	F
F	F	T	T	T

Deduction Using Propositional Logic: Example 3

If I am the Student President then I am well-known. I am the Student President. So I am well-known

Coding: Variables

a: I am the Student President

b: I am well-known

Coding the sentences:

F1: $a \rightarrow b$

F2: a

G: b

The final formula for deduction: $(F1 \wedge F2) \rightarrow G$, that is: $((a \rightarrow b) \wedge a) \rightarrow b$

a	b	a → b	(a → b) ∧ a	((a → b) ∧ a) → b
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

If Rajat is the Student President then Rajat is wellknown. Rajat is the Student President. So Rajat is well known

Coding: Variables

a: Rajat is the Student President

b: Rajat is well-known

Coding the sentences:

F1: $a \rightarrow b$

F2: a

G: b

The final formula for deduction:

$(F1 \wedge F2) \rightarrow G$,

that is: $((a \rightarrow b) \wedge a) \rightarrow b$

Deduction Using Propositional Logic: Example 4 & 5

If Asha is elected VP then Rajat is chosen as G-Sec and Bharati is chosen as Treasurer. Rajat is not chosen as G-Sec. Therefore Asha is not elected VP.

If Asha is elected VP then Rajat is chosen as GSec and Bharati is chosen as Treasurer. Rajat is chosen as G-Sec. Therefore Asha is elected VP.

More Examples

If Asha is elected VP then Rajat is chosen as G-Sec or Bharati is chosen as Treasurer. Rajat is not chosen as G-Sec. Therefore if Asha is elected as VP then Bharati is chosen as Treasurer.

If Asha is elected VP then either Rajat is chosen as GSec or Bharati is chosen as Treasurer but not both. Rajat is not chosen as G-Sec. Therefore if Asha is elected as VP then Bharati is chosen as Treasurer

Methods for Deduction in Propositional Logic

Interpretation of a Formula

Valid, non-valid, Satisfiable, Unsatisfiable

Decidable but NP-Hard

Truth Table Method

Faster Methods for validity checking:- Tree Method

Data Structures: Binary Decision Diagrams

Symbolic Method: Natural Deduction

Soundness and Completeness of a Method

Methods for Deduction in Propositional Logic

Interpretation of a Formula

Valid, non-valid, Satisfiable, Unsatisfiable

Decidable but NP-Hard

Truth Table Method

Faster Methods for validity checking:- Tree Method

Data Structures: Binary Decision Diagrams

Symbolic Method: Natural Deduction

Soundness and Completeness of a Method

NATURAL DEDUCTION:

Modus Ponens:
(a → b), a :- therefore b

Modus Tollens:
(a → b), ~b :- therefore ~a

Hypothetical Syllogism:
(a → b), (b → c):- therefore (a → c)

Disjunctive Syllogism:
(a V b), ~a:- therefore b

Constructive Dilemma:
(a → b) Λ (c → d), (a Vc) :- therefore (b V d)

Destructive Dilemma:
(a → b) Λ (c → d), (~b V~d) :- therefore (~a V ~c)

Simplification: a Λ b:- therefore a

Conjunction: a, b:- therefore a Λ b

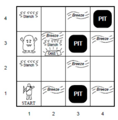
Addition: a :- therefore a V b

Natural Deduction is Sound and Complete

Insufficiency of Propositional Logic

- Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.
- No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.
- All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.
- Every passenger is either in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.

Limitations of Propositional Logic



- Does not scale to large size environments
- For the current WW
  - 64 proposition symbols
  - 155 sentences
- Lacks concise representations
- Propositional logic identify individual states but fail to associate with objects.
  - We would like to talk about objects and group of objects and relationships between them.
- Doesn't capture relations within objects
  - E.g., every pit would have a breeze in adjacent squares need to be written separately for each square, instead a simple relation should be able to encode the knowledge to every pit
  - E.g., Any sentence in natural language would contain objects (nouns, verbs) and relations (subject, object) between them

First order Logic

or

PREDICATE LOGIC

Using Propositional Logic

Representing simple facts

It is raining
RAINING

It is sunny
SUNNY

It is windy
WINDY

If it is raining, then it is not sunny
RAINING → ¬SUNNY

- If we want to represent
- Mohit is a man
  - Mohitman
- Rohit is a man
  - Rohitman
- Not able to draw any conclusion about similarities between Mohit and Rohit.

Better Representation

MAN(MOHIT)
MAN(ROHIT)

- Structure of representation reflects the structure of knowledge itself.
- Need to use predicate applied to arguments

All man are Mortal
- MORTALMAN

– Need variable and quantification

Methods for Deduction in First Order Logic

1. Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.

2. No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.

3. All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.

4. Every passenger is either in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.

New Additions in Proposition (First Order Logic)

Variables,
Constants,
Predicate Symbols and
New Connectors: ∃ (there exists), V(for all)

Try Now

Formulating Predicate Logic Statements

New Additions in Proposition (First Order Logic)
Variables, Constants, Predicate Symbols and
New Connectors: ∃ (there exists), V(for all)

1. Wherever Mary goes, so does the Lamb.
Mary goes to School. So the Lamb goes to School.

Predicate: goes(x,y) to represent x goes to y
New Connectors: ∃ (there exists), V(for all)
F1: Vx(goes(Mary, x) → goes(Lamb, x))
F2: goes(Mary, School)
G: goes(Lamb, School)
To prove: (F1 Λ F2) → G) is always true

2. No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.

Predicates: contractor(x), dependable(x), engineer(x)

F1: Vx(contractor(x) → ~dependable(x))
[Alternative: ~∃x (contractor(x) Λ dependable(x))]

F2: ∃x(engineer(x) Λ contractor(x))

G: ∃x(engineer(x) Λ ~dependable(x))

To prove: (F1 Λ F2) → G) is always true

More Examples

3. All dancers are graceful.
Ayesha is a student. Ayesha is a dancer.
Therefore some student is graceful.

4. Every passenger is either in first class or second class. Each passenger is in second class if and only if the passenger is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.

Use of Quantifiers

EXAMPLES:

Someone likes everyone
Everyone likes someone
There is someone whom everyone likes
Everyone likes everyone
If everyone likes everyone then someone likes everyone
If there is a person whom everyone likes then that person likes himself

LAWS of NEGATION:

Use of Function Symbols

If x is greater than y and y is greater than z then x is greater than z.

The age of a person is greater than the age of his child.

Therefore the age of a person is greater than the age of his grandchild.

The sum of ages of two children are never more than the sum of ages of their parents.

Variables and Predicate / Function Symbols

Variables, Free variables, Bound variables

Symbols – proposition symbols,constant symbols, function symbols,predicate symbols

Variables can be quantified in first order predicate logic

Symbols cannot be quantified in first order predicate logic

Interpretations are mappings of symbols to relevant aspects of a domain

Terminology for Predicate Logic

Domain: D
Constant Symbols: M, N, O, P, ….
Variable Symbols: x,y,z,….
Function Symbols: F(x), G(x,y), H(x,y,z)
Predicate Symbols: p(x), q(x,y), r(x,y,z),
Connectors: ~, Λ, V, →, ∃, V

Terms:
Well-formed Formula:
Free and Bound Variables:
Interpretation, Valid, Non-Valid,
Satisfiable, Unsatisfiable

Validity, Satisfiability, Structur

F1:∀x(goes(Mary, x) → goes(Lamb, x))
F2: goes(Mary, School)
G: goes(Lamb, School)
To prove: (F1 Λ F2) → G) is always true

Is the same as:
F1: ∀x(www(M, x) → www(L, x))
F2: www(M, S)
G: www(L, S)
To prove: (F1 Λ F2) → G) is always true

More Examples : Predicate Logic

1. Marcus was a man.
2. Marcus was a Pompeian.
3. All Pompeians were Romans.
4. Caesar was a ruler.
5. All Pompeians were either loyal to Caesar or hated him.
6. Every one is loyal to someone.
7. People only try to assassinate rulers they are not loyal to.
8. Marcus tried to assassinate Caesar.

Predicate Logic Example

1. **Marcus was a man.**
   man(Marcus)
2. **Marcus was a Pompeian.**
   Pompeian(Marcus)
3. **All Pompeians were Romans.**
   ∀x: Pompeian(x) → Roman(x)
4. **Caesar was a ruler.**
   ruler(Caesar)

Predicate Logic Example

5. **All Pompeians were either loyal to Caesar or hated him.**
   inclusive-or
   ∀x: Pompeians (x) → loyalto(x, Caesar) ∨ hate(x, Caesar)
   exclusive-or $p \oplus q = (p \wedge \neg q) \vee (\neg p \wedge q)$
   ∀x: Pompeians (x) → (loyalto(x, Caesar) ∧ ¬hate(x, Caesar)) ∨(¬loyalto(x, Caesar) ∧ hate(x, Caesar))

6. Every one is loyal to someone.
∀x: ∃y: loyalto(x, y)
∃y: ∀x: loyalto(x, y)

7. People only try to assassinate rulers they are not loyal to.
∀x: ∀y: person(x) ∧ ruler(y) ∧ tryassassinate(x, y) → ¬loyalto(x, y)

8. Marcus tried to assassinate Caesar.
tryassassinate(Marcus, Caesar)

First Order Logic – Inference

- First order logic has sentences with quantifiers which makes it hard for inference
- First order inference can be done by
  – reducing the knowledge base to propositional logic and
  – use propositional inference techniques

Was Marcus loyal to Caesar?

- Using 7 & 8 fact, we can predict
- Backward chaining
  man(Marcus)
  ruler(Caesar)
  tryassassinate(Marcus, Caesar)
  ⇓ ∀x: man(x) → person(x)
  ¬loyalto(Marcus, Caesar)

Inferencing in Predicate Logic

Domain: D
Constant Symbols: M, N, O, P, ….
Variable Symbols: x,y,z,….
Function Symbols: F(x), G(x,y), H(x,y,z)
Predicate Symbols: p(x), q(x,y), r(x,y,z),
Connectors: ~, Λ, V, →, ∃, V

Terms:
Well-formed Formula:
Free and Bound Variables:
Interpretation, Valid, Non-Valid,
Satisfiable, Unsatisfiable

What is an Interpretation? Assign a domain set D, map constants, functions, predicates suitably.
The formula will now have a truth value
Example:
F1: Vx(g(M, x) → g(L, x))
F2: g(M, S)
G: g(L, S)
Interpretation 1: D = {Akash, Baby, Home, Play, Ratan, Swim}, etc.,
Interpretation 2: D = Set of Integers, etc.,

How many interpretations can there be?
To prove Validity, means (F1 Λ F2) → G) is true under all interpretations
To prove Satisfiability means (F1 Λ F2) → G) is true under at least one interpretation

Predicate and its Limitations

Russell's Paradox
(The barber shaves all those who do not shave themselves. Does the barber shave himself?)
- There is a single barber in town.
- Those and only those who do not shave themselves are shaved by the barber.
- Who shaves the barber?

Checking Validity of First order logic is undecidable but partially decidable (semi- decidable) {Robinson's Method of Resolution Refutation}

Higher order predicate logic - can quantify symbols in addition to quantifying variables.
∀p((p(0) Λ (∀x(p(x) → p(S(x))) → ∀y(p(y))

Home work

- Translate these sentences into following Logic.
  a) Preposition logic
  b) Predicate logic

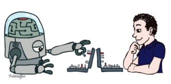
"Mary likes all kinds of food. Pizza is a kind of food. Apple is a food. Anything anyone eats is a food. John eats chicken. Ana eats everything Mary eats."

Next :

- Module 4: Logic and Deduction
  – PART 4.1 : Knowledge Representation
  – PART 4.2 : Propositional logic
  – PART 4.3 : Predicate Logic
    - Propositional Logic to Predicate Logic
    - Predicate Logic Fundamentals
  – PART 4.5 : Deduction / Reasoning Tasks
  – PART 4.6 : Inference By Forward Chaining
  – PART 4.7 : Inferencing By Resolution Refutation
    - Reduction to satisfiability problem : SAT Solver
  –

References

- ***Artificial Intelligence*** *by Elaine Rich & Kevin Knight*, Third Ed, Tata McGraw Hill
- ***Artificial Intelligence and Expert System*** *by Patterson*
- http://www.cs.rmit.edu.au/AI-Search/Product/
- http://aima.cs.berkeley.edu/demos.html (for more demos)
- ***Artificial Intelligence and Expert System*** *by Patterson*
- Slides adapted from CS188 Instructor: Anca Dragan, University of California, Berkeley
- Slides adapted from CS60045 ARTIFICIAL INTELLIGENCE



(some slides adapted from http://aima.cs.berkeley.edu/)