Artificial Intelligence

Chapter 4: Knowledge and Reasoning

- 4.1Knowledge based Agents, Brief Overview of propositional logic, First Order Logic: Syntax and Semantic, Inference in FOL, Forward chaining, backward Chaining.
- 4.2Knowledge Engineering in First-Order Logic, Unification, Resolution
- 4.3Uncertain Knowledge and Reasoning: Uncertainty, Representing knowledge in an uncertain domain, The semantics of belief network, Simple Inference in belief network

A knowledge-based agent

- A knowledge-based agent includes a knowledge base and an inference system.
- A knowledge base is a set of representations of facts of the world.
- Each individual representation is called a **sentence**.
- The sentences are expressed in a **knowledge representation language**.

- The agent operates as follows:

1. It TELLs the knowledge base what it perceives.
2. It ASKs the knowledge base what action it should perform.
3. It performs the chosen action.

Examples of sentences

The moon is made of green cheese

If A is true then B is true

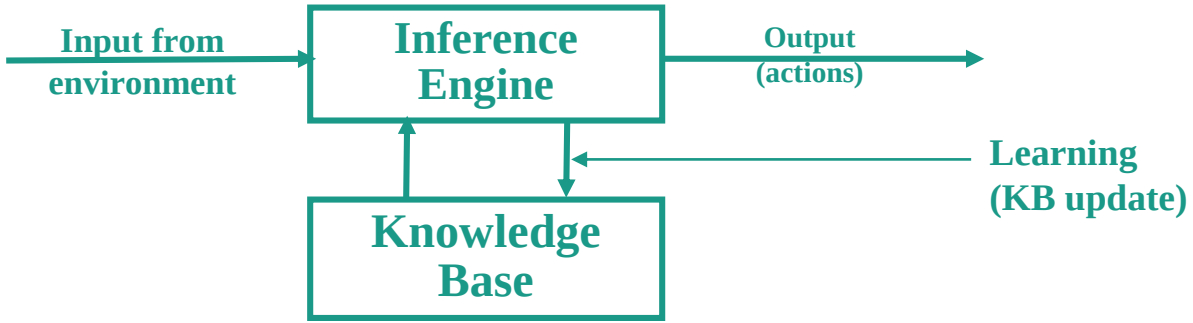
A is false

All humans are mortal

Architecture of a knowledge-based agent

- **Knowledge Level.**
  - The most abstract level: describe agent by saying what it knows.
  - Example: A taxi agent might know that the Golden Gate Bridge connects San Francisco with the Marin County.
- **Logical Level.**
  - The level at which the knowledge is encoded into sentences.
  - Example: Links(GoldenGateBridge, SanFrancisco, MarinCounty).
- **Implementation Level.**
  - The physical representation of the sentences in the logical level.
  - Example: `'(links goldengatebridge sanfrancisco marincounty)`

- The **Inference engine** derives new sentences from the input and KB
- The inference mechanism depends on representation in KB
- The agent operates as follows:
  1. It receives percepts from environment
  2. It computes what action it should perform (by IE and KB)
  3. It performs the chosen action (some actions are simply inserting inferred new facts into KB).



KB can be viewed at different levels

- **Knowledge Level.** ***review***
  - The most abstract level -- describe agent by saying what it knows.
  - Example: A taxi agent might know that the Golden Gate Bridge connects San Francisco with the Marin County.
- **Logical Level.**
  - The level at which the knowledge is encoded into sentences.
  - Example: Links(GoldenGateBridge, SanFrancisco, MarinCounty).
- **Implementation Level.**
  - The physical representation of the sentences in the logical level.
  - Example: "(Links GoldenGateBridge, SanFrancisco, MarinCounty)"

Knowledge representation

- The objective of knowledge representation is to express the knowledge about the world in a computer-tractable form
- Key aspects of knowledge representation languages:

-**Syntax:** describes how sentences are formed in the language

- **Semantics**: describes the meaning of sentences, what is it the sentence refers to in the real world

- **Computational aspect**: describes how sentences and objects are manipulated in concordance with semantical conventions Many KB systems rely on some variant of logic

The agent's goal

The agent's goal is to find the gold and bring it back to the start as quickly as possible, without getting killed.

- 1000 points reward for climbing out of the cave with the gold
- 1 point deducted for every action taken
- 10000 points penalty for getting killed

LOGIC

- A formal language for expressing knowledge and ways of reasoning. Logic is defined by:
- **A set of sentences** – A sentence is constructed from a set of primitives according to syntax rules.
- **A set of interpretations** – An interpretation gives a semantic to primitives. It associates primitives with values.
- **The valuation (meaning) function V** – Assigns a value (typically the truth value) to a given sentence under some interpretation V : sentence × interpretation → {True , False }

Example Of Logic

- Example of logic Language of numerical constraints:

- **A sentence:**

$x + 3 \leq z$ x , z - variable symbols (primitives in the language)

- **An interpretation**: x = 5, z = 2 Variables mapped to specific real numbers

- **Valuation (meaning):**

V ($x + 3 \leq z$, I) is False for I: x = 5, z = 2 is True

for I: x = 5, z = 10

Big Ideas

- Logic is a great knowledge representation language for many AI problems
- **Propositional logic** is the simple foundation and fine for some AI problems
- **First order logic** (FOL) is much more expressive as a KR language and more commonly used in AI
- There are many variations: horn logic, higher order logic, three-valued logic, probabilistic logics, etc.

Propositional logic Syntax s

- **Propositional logic P**: – defines a language for symbolic reasoning

- **Proposition**: a statement that is either true or false

- **Examples of propositions:**

– France is in Europe.

– It rains outside.

– 2 is a prime number and 6 is a prime

– How are you? Not a proposition.

Propositional logic

- **Logical constants**: true, false
- **Propositional symbols**: P, Q,... (**atomic sentences**)
- Wrapping **parentheses**: (…)
- Sentences are combined by **connectives**:

  ∧ and [conjunction]

  ∨ or [disjunction]

  ⇒ implies [implication / conditional]

  ⇔ is equivalent [biconditional]

  ¬ not [negation]
- **Literal**: atomic sentence or negated atomic sentence

  P, ¬ P

Examples of PL sentences

- $(P \wedge Q) \rightarrow R$

  "If it is hot and humid, then it is raining"
- $Q \rightarrow P$

  "If it is humid, then it is hot"
- Q

  "It is humid."
- We're free to choose better symbols, btw:

  Ho = "It is hot"

  Hu = "It is humid"

  R = "It is raining"

Propositional logic (PL)

- Simple language for showing key ideas and definitions
- User defines set of propositional symbols, like P and Q
- User defines **semantics** of each propositional symbol:
  - P means “It is hot”, Q means “It is humid”, etc.
- A sentence (well formed formula) is defined as follows:
  - A symbol is a sentence
  - If S is a sentence, then ¬S is a sentence
  - If S is a sentence, then (S) is a sentence
  - If S and T are sentences, then (S ∨ T), (S ∧ T), (S → T), and (S ↔ T) are sentences
  - A sentence results from a finite number of applications of the rules

Some terms

- The meaning or **semantics** of a sentence determines its **interpretation**
- 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* – an assignment of truth values to propositional symbols that makes each sentence in the KB True

Model for a KB

- Let the KB be [P∧Q→R, Q → P]
- What are the possible models? Consider all possible assignments of T|F to P, Q and R and check truth tables
  - **FFF: OK**
  - **FFT: OK**
  - FTF: NO
  - FTT: NO
  - **TFF: OK**
  - **TFT: OK**
  - TTF: NO
  - **TTT: OK**
- If KB is [P∧Q→R, Q → P, Q], then the only model is TTT

P: it's hot
Q: it's humid
R: it's raining

More terms

- A **valid sentence** or **tautology** is a sentence that is True under all interpretations, no matter what the world is actually like or what the semantics is. Example: “It’s raining or it’s not raining”
- An **inconsistent sentence** or **contradictio**n 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 |= Q, means that whenever P is True, so is Q. In other words, all models of P are also models of Q.

Truth tables

- Truth tables are used to define logical connectives
- and to determine when a complex sentence is true given the values of the symbols in it

Truth tables for the five logical connectives

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*

Example of a truth table used for a complex sentence

P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
False	*False*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
True	*True*	*True*	*False*	*True*

On the implies connective: P → Q

- Note that → is a logical connective
- So $P \rightarrow Q$ is a logical sentence and has a truth value, i.e., is either true or false
- If we add this sentence to the KB, it can be used by an inference rule, *Modes Ponens*, to derive/infer/prove Q if P is also in the KB
- Given a KB where P=True and Q=True, we can also derive/infer/prove that P→Q is True

P → Q

- When is $P \rightarrow Q$ true? Check all that apply
  - ❑ P=Q=true
  - ❑ P=Q=false
  - ❑ P=true, Q=false
  - ❑ P=false, Q=true

P → Q

- When is $P \rightarrow Q$ true? Check all that apply
  - ☑ P=Q=true
  - ☐ P=Q=false
  - ☑ P=true, Q=false
  - ☐ P=false, Q=true
- We can get this from the truth table for →

Inference rules

- **Logical inference** creates new sentences that logically follow from a set of sentences (KB)
- An inference rule is **sound** if every sentence X it produces when operating on a KB logically follows from the KB
  - i.e., inference rule creates no contradictions
- An inference rule is **complete** if it can produce every expression that logically follows from (is entailed by) the KB.
  - Note analogy to complete search algorithms

Sound rules of inference

- Here are some examples of sound rules of inference
- Each can be shown to be sound using a truth table

RULE	PREMISE	CONCLUSION
Modus Ponens	A, A → B	B
And Introduction	A, B	A ∧ B
And Elimination	A ∧ B	A
Double Negation	¬ ¬A	A
Unit Resolution	A ∨ B, ¬B	A
Resolution	**A ∨ B, ¬B ∨ C**	**A ∨ C**

Inference Rules

Modus Ponens or Implication Elimination

$$\frac{\alpha \Rightarrow \beta, \alpha}{\beta}$$

From two sentences **α ⇒ β** and **α** that are **true** (so called **axioms**) the new **true** sentence **β** can be concluded (a **theorem** is proved with respect to the axioms, i.e. the theorem logically follows from the axioms).

Inference Rules

Example:

Sentence: If sun shines it is warm

A – “sun shines”; **B** – “it is warm”.

Axioms: $\mathbf{A \Rightarrow B}$

$\mathbf{A}$

Theorem: **B**, i.e., “it is warm”.

Inference Rules

AND-Elimination

$$\frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}{\alpha_i}$$

From a conjunction of sentences **any** of conjuncts can be inferred.

Inference Rules

AND-Introduction

$$\frac{\alpha_1, \alpha_2, \ldots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}$$

From a **list** of sentences their conjunction can be inferred.

Inference Rules

OR-Introduction

$$\frac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee \ldots \vee \alpha_n}$$

From a sentence its disjunction with **anything else** at all can be inferred.

Inference Rules

Double-Negation Elimination

$$\frac{\neg\neg\alpha}{\alpha}$$

From a double negated sentence a **positive** sentence can be inferred.

Inference Rules

Unit Resolution

$$\frac{\alpha \vee \beta, \neg\beta}{\alpha}$$

From a disjunction, if one of the disjuncts is **false** it can be inferred that the other one is **true**.

Soundness of modus ponens

A	B	A → B	OK?
True	True	True	√
True	False	False	√
False	True	True	√
False	False	True	√

Resolution

- **Resolution** is a valid inference rule producing a new clause implied by two clauses containing *complementary literals*
  - A literal is an atomic symbol or its negation, i.e., P, ~P
- This is the only inference rule you need to build a sound and complete theorem prover
  - Based on proof by contradiction and usually called resolution refutation
- The resolution rule was discovered by Alan Robinson (CS, U. of Syracuse) in the mid 60s

Resolution

- A KB is actually a set of sentences all of which are true, i.e., a conjunction of sentences.
- To use resolution, put KB into *conjunctive normal form* (CNF), where each sentence written as a disjunction of (one or more) literals

Example

- KB: [P→Q , Q→R∧S]
- KB in CNF: [~P∨Q , ~Q∨R , ~Q∨S]
- Resolve KB(1) and KB(2) producing: ~P∨R *(i.e., P→R)*
- Resolve KB(1) and KB(3) producing: ~P∨S *(i.e., P→S)*
- New KB: [~P∨Q , ~Q∨~R∨~S , ~P∨R , ~P∨S]

Tautologies

(A→B)↔(~A∨B)

(A∨(B∧C)) ↔(A∨B)∧(A∨C)

Soundness of the resolution inference rule

α	β	γ	$\alpha \vee \beta$	$\neg\beta \vee \gamma$	$\alpha \vee \gamma$
False	False	False	False	True	False
False	False	True	False	True	True
False	True	False	True	False	False
False	True	True	True	True	True
True	False	False	True	True	True
True	False	True	True	True	True
True	True	False	True	False	True
True	True	True	True	True	True

From the rightmost three columns of this truth table, we can see that

$(\alpha \vee \beta) \wedge (\beta \vee \gamma) \leftrightarrow (\alpha \vee \gamma)$

is valid (i.e., always true regardless of the truth values assigned to α, β and γ

Simple Resolution Algorithm for PL

function PL-RESOLUTION(KB, α) **returns** *true* or *false*
  inputs: KB, the knowledge base, a sentence in propositional logic
    α, the query, a sentence in propositional logic

  clauses $\leftarrow$ the set of clauses in the CNF representation of $KB \wedge \neg\alpha$
  new $\leftarrow$ { }
  loop do
    for each C_i, C_j **in** *clauses* **do**
      resolvents $\leftarrow$ PL-RESOLVE(C_i, C_j)
      if *resolvents* contains the empty clause **then return** *true*
      new $\leftarrow$ *new* $\cup$ *resolvents*
    if *new* $\subseteq$ *clauses* **then return** *false*
    clauses $\leftarrow$ *clauses* $\cup$ *new*

Figure 7.12 A simple resolution algorithm for propositional logic. The function PL-RESOLVE returns the set of all possible clauses obtained by resolving its two inputs.

Application of PL Resolution

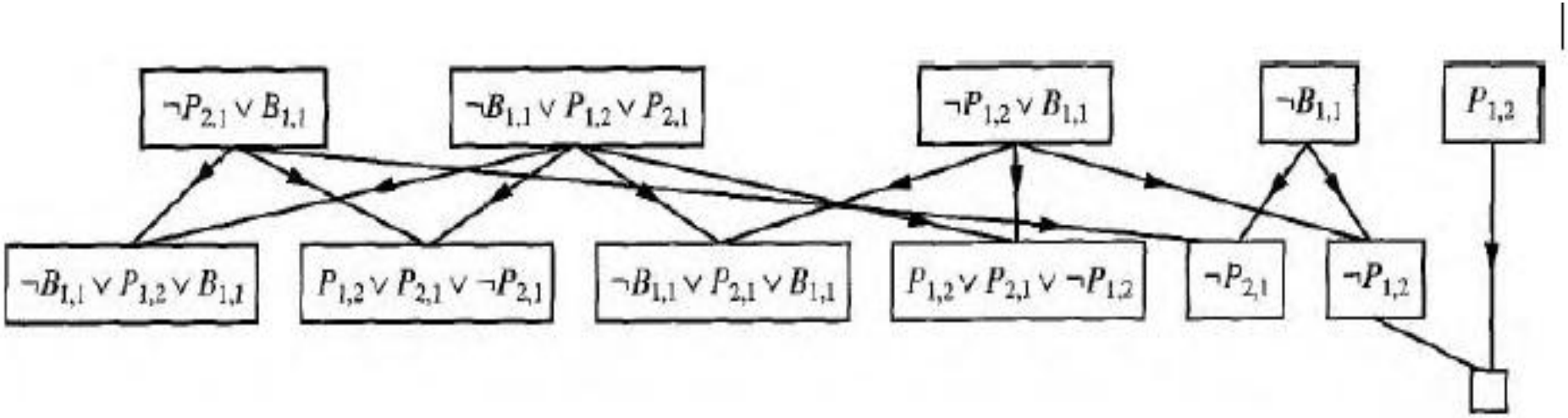


Figure 7.13 Partial application of PL-RESOLUTION to a simple inference in the wumpus world. $\neg P_{1,2}$ is shown to follow from the first four clauses in the top row.

Proving things

- A **proof** is a sequence of sentences, where each is a premise or is derived from earlier sentences in the proof by an inference rule
- The last sentence is the **theorem** (also called goal or query) that we want to prove
- Example for the "weather problem"

1 Hu	premise		"It's humid"
2 Hu→Ho	premise		"If it's humid, it's hot"
3 Ho	modus ponens(1,2)		"It's hot"
4 (Ho∧Hu)→R	premise		"If it's hot & humid, it's raining"
5 Ho∧Hu	and introduction(1,3)		"It's hot and humid"
6 R	modus ponens(4,5)		"It's raining"

Entailment and derivation

- **Entailment: KB |= Q**
  - Q is entailed by KB (set sentences) iff there is no logically possible world where Q is false while all the sentences in KB are true
  - Or, stated positively, Q is entailed by KB iff the conclusion is true in every logically possible world in which all the premises in KB are true
- **Derivation: KB |- Q**
  - We can derive Q from KB if there's a proof consisting of a sequence of valid inference steps starting from the premises in KB and resulting in Q

Two important properties for inference

Soundness: If KB |- Q then KB |= Q

- If Q is derived from KB using a given set of rules of inference, then Q is entailed by KB
- Hence, inference produces only real entailments, or any sentence that follows deductively from the premises is valid

Completeness: If KB |= Q then KB |- Q

- If Q is entailed by KB, then Q can be derived from KB using the rules of inference
- Hence, inference produces all entailments, or all valid sentences can be proved from the premises

PL Example

- 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?

PL Example

- In PL we have to create propositional symbols to stand for all or part of each sentence, e.g.:

  P = “person”; Q = “mortal”; R = “Confucius”

- The above 3 sentences are represented as:

  $P \rightarrow Q$; $R \rightarrow P$; $R \rightarrow Q$

- The 3rd sentence is entailed by the first two, but we need an explicit symbol, R, to represent an individual, Confucius, who is a member of the classes *person* and *mortal*
- Representing other individuals requires introducing separate symbols for each, with some way to represent the fact that all individuals who are “people” are also “mortal”

Propositional logic: pro and con

- Advantages
  - Simple KR language sufficient for some problems
  - Lays the foundation for higher logics (e.g., FOL)
  - Reasoning is decidable, though NP complete, and efficient techniques exist for many problems
- Disadvantages
  - Not expressive enough for most problems
  - Even when it is, it can very “un-concise”

PL is a weak KR 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 (FOL) is expressive enough to represent this kind of information using relations, variables and quantifiers, e.g.,
  - *Every elephant is gray:* $\forall$ x (elephant(x) $\rightarrow$ gray(x))
  - *There is a white alligator:* $\exists$ x (alligator(X) $\wedge$ white(X))

PL Example

- In PL we have to create propositional symbols to stand for all or part of each sentence, e.g.:

  P = “person”; Q = “mortal”; R = “Confucius”

- The above 3 sentences are represented as:

  $P \rightarrow Q$; $R \rightarrow P$; $R \rightarrow Q$

- The 3rd sentence is entailed by the first two, but we need an explicit symbol, R, to represent an individual, Confucius, who is a member of the classes *person* and *mortal*
- Representing other individuals requires introducing separate symbols for each, with some way to represent the fact that all individuals who are “people” are also “mortal”

Propositional logic summary

- Inference is the process of deriving new sentences from old
  - **Sound** inference derives true conclusions given true premises
  - **Complete** inference derives 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
- **Propositional logic** commits only to the existence of facts that may or may not be the case in the world being represented
  - Simple syntax and semantics suffices to illustrate the process of inference
  - Propositional logic can become impractical, even for very small worlds