Intro to AI,
Autumn, 2025

PROPOSITIONAL LOGIC

Faculty of DS & AI
Autumn semester, 2025

Trong-Nghia Nguyen

Business AI Lab

2025-10

Content

- **Representing problems using logic**
- **Propositional logic**

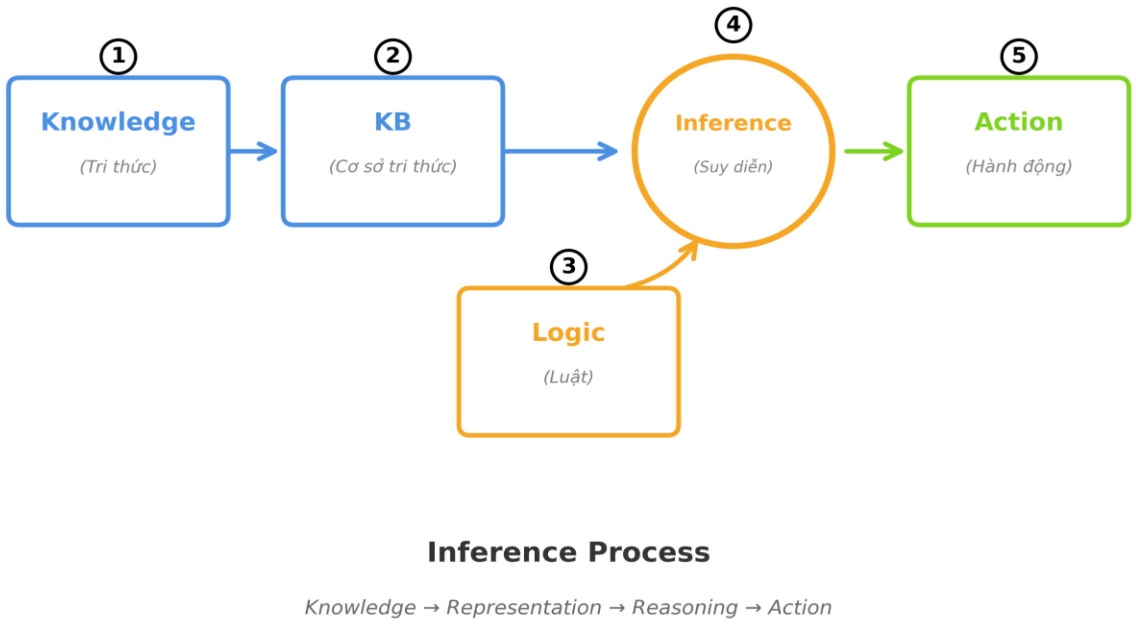
Representing problems using logic

Inference (Reasoning)

- Is the process of **generating a new correct statement** from previously **existing correct statements/ knowledge,** to make appropriate **actions**
  - **Knowledge:** Information and knowledge about a certain field. Represented in the form of sentences or propositions
  - Knowledge base (KB): represented in a specific form, creating a knowledge language

Logic: is a language in which each sentence in the language represents a **true** or **false value**. Logic:

- Represent the Knowledge
- Reasoning(Inference) based on the Knowledge

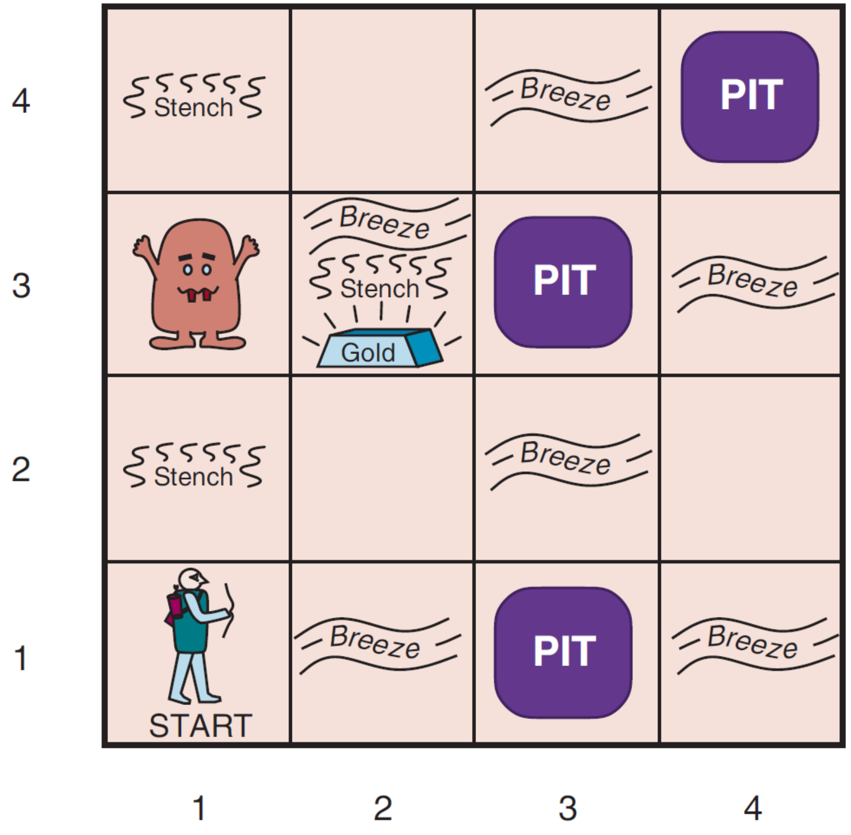


Representing problems using logic

Problem Description

- A Cave of 4 x 4 rooms
- Wumpus, the monster
- PIT, a bottomless room
- Agent: 5 sensors, 1 arrow,
- Goal: find the Gold and escape from the cave

☞ A **Knowledge-Based Agent** is needed

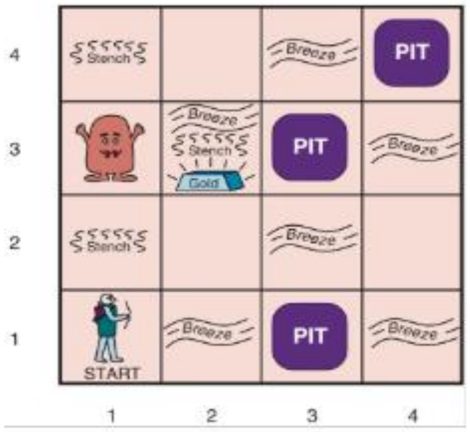


Representing problems using logic

Inference (Reasoning)

PEAS Descriptions

- Performance measure
- Environment
- Actuators
- Sensors



- **Performance measure**: +1000 for climbing out of the cave with the gold, –1000 for falling into a pit or being eaten by the wumpus, –1 for each action taken and –10 for using up the arrow. The game ends either when the agent dies or when the agent climbs out of the cave.
- **Environment**: A 4×4 grid of rooms. The agent always starts in the square labeled [1,1], facing to the right. The locations of the gold and the wumpus are chosen randomly, with a uniform distribution, from the squares other than the start square. In addition, each square other than the start can be a pit, with probability 0.2.
- **Actuators**: The agent can move *Forward*, *TurnLeft* by 90°, or *TurnRight* by 90°. The agent dies a miserable death if it enters a square containing a pit or a live wumpus. (It is safe, albeit smelly, to enter a square with a dead wumpus.) If an agent tries to move forward and bumps into a wall, then the agent does not move. The action *Grab* can be used to pick up the gold if it is in the same square as the agent. The action *Shoot* can be used to fire an arrow in a straight line in the direction the agent is facing. The arrow continues until it either hits (and hence kills) the wumpus or hits a wall. The agent has only one arrow, so only the first *Shoot* action has any effect. Finally, the action *Climb* can be used to climb out of the cave, but only from square [1,1].
- **Sensors**: The agent has five sensors, each of which gives a single bit of information:
  - In the square containing the wumpus and in the directly (not diagonally) adjacent squares, the agent will perceive a *Stench*.
  - In the squares directly adjacent to a pit, the agent will perceive a *Breeze*.
  - In the square where the gold is, the agent will perceive a *Glitter*.
  - When an agent walks into a wall, it will perceive a *Bump*.
  - When the wumpus is killed, it emits a woeful *Scream* that can be perceived anywhere in the cave.

Representing problems using logic

Inference (Reasoning)

The First Two Steps

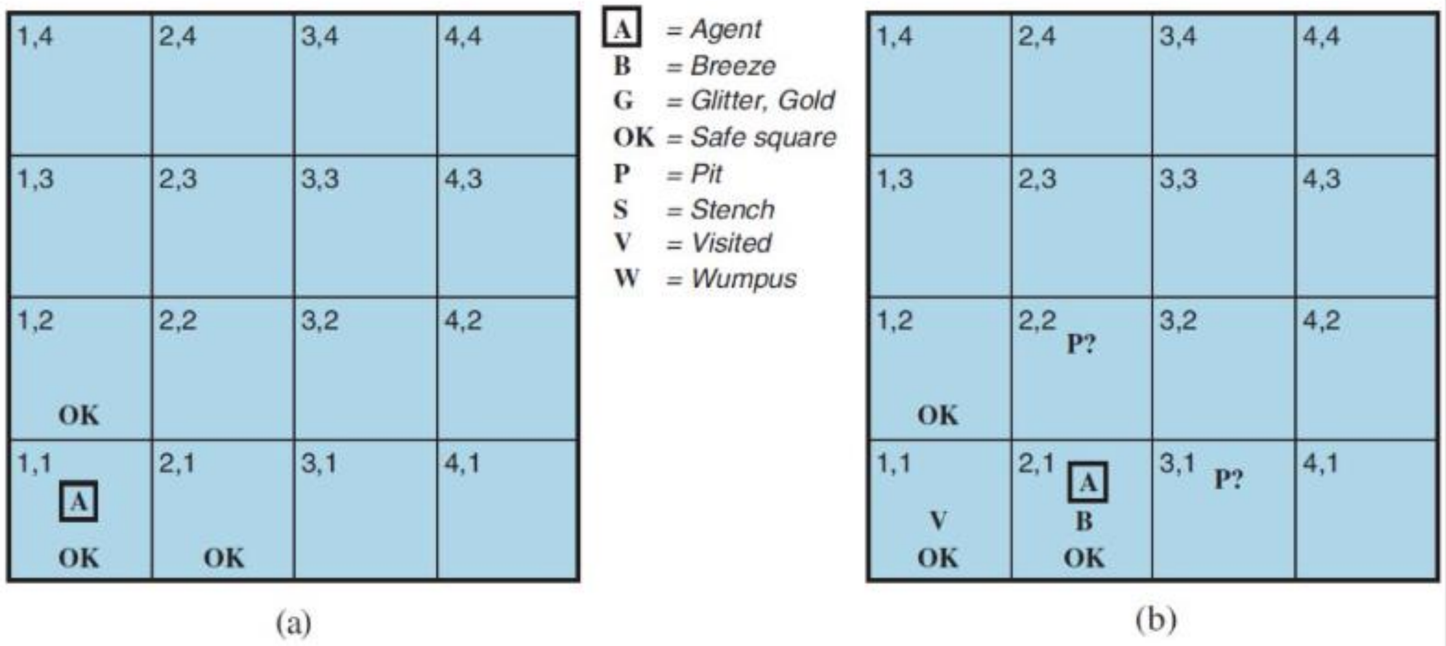


Figure 7.3 The first step taken by the agent in the wumpus world. (a) The initial situation, after percept [*None*, *None*, *None*, *None*, *None*]. (b) After moving to [2,1] and perceiving [*None*, *Breeze*, *None*, *None*, *None*].

Representing problems using logic

Inference (Reasoning)

Two Later Stages

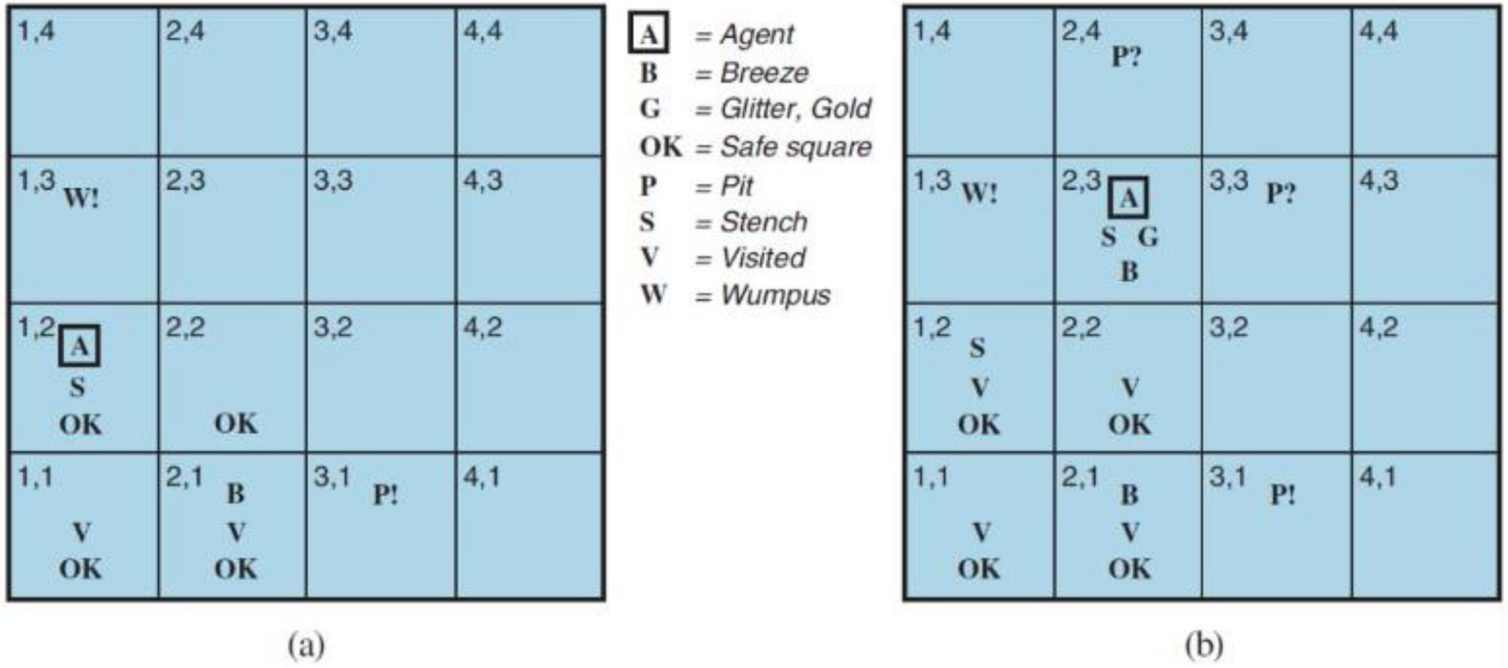


Figure 7.4 Two later stages in the progress of the agent. (a) After moving to [1,1] and then [1,2], and perceiving [*Stench*, *None*, *None*, *None*, *None*]. (b) After moving to [2,2] and then [2,3], and perceiving [*Stench*, *Breeze*, *Glitter*, *None*, *None*].

Propositional logic

- Definition
- Syntax & semantic
- Canonical Form
- Transform to CNF
- Inference rules

Propositional logic

Definition

- A proposition is a statement that can have only one of two truth values: **true (T, 1)** or **false (F, 0)**

Reality

"*If it rains, the sky is cloudy.*"

It's raining

So: "The sky is **cloudy**"

Propositional logic

P="it rains"

Q="the sky is cloudy"

Assuming:

- P->Q : True
- P : True

So -> Q : True (*Modus Ponens rule*)

So: "The sky is **cloudy**"

Propositional logic

Definition

- A proposition is a statement that can have only one of two truth values: **true (T, 1)** or **false (F, 0)**

Reality

"*If An had a lot of money, she would go shopping.*"

*An **wouldn't** go shopping*

So: "An **don't** have a lot of money"

Propositional logic

P="An had a lot of money"

Q="An go shopping"

Assuming:

- P->Q : True
- ¬Q : True

So -> ¬P : True (*Modus Ponens rule*)

So: "An **don't** have a lot of money"

Propositional logic

Syntax & semantic

- 2 logic constant: True, False
- Parentheses: ()
- Logical operations
  - Conjunction (and): ∧
  - Disjunction (or): ∨
  - Negation (not): ¬
  - Implication: =>
  - Equivalence: <=>

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*

If P and Q are formulas:

- P ∧ Q : P **AND** Q
- P ∨ Q : P **OR** Q
- ¬ P: **NOT** P
- P => Q : P **Infer** Q
- P <=> Q : P **equivalent to** Q

Propositional logic

Syntax & semantic

- These logical operations have the properties of **commutativity**, **associativity**, and **distributivity**.
- Order of **precedence**:
  Negation → Implication → Equivalence → Conjunction → Disjunction
- Use () to group clauses to specify order of priority.
  - For example: (A ∧ B ∨ C) → D ∧ E

¬
∧
∨
⇒
⇔

A ∨ B ∧ C	A ∨ (B ∧ C)
A ∧ B ⇒ C ∨ D	(A ∧ B) ⇒ (C ∨ D)
A ⇒ B ∨ C ⇔ D	(A ⇒ (B ∨ C)) ⇔ D

Propositional logic

Syntax & semantic

- The semantics of propositional logic allows us to determine the meaning of formulas in the real world.
- A fact can only be true or false
- For example: P =. "Hanoi is the capital of Vietnam"

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*

Truth table of logical operations

Propositional logic

Syntax & semantic

Symbols

- Constant, Variable: objects

  $A,\ X_1,\ John,\ a,\ x, \ldots$

- Predicate: relations

  $True,\ False,\ Person,\ King, \ldots$

- Function: functions

  $MotherOf,\ LeftLeg, \ldots$

Terms

- expression that refers to an object

  Constant, Variable, Function(Term, …)

Propositional logic

Syntax & semantic

Atomic Sentences

- An atomic sentence states a fact

$Brother(Richard, John)$ read as "Richard is a Brother of John"

$Married(Father(Richard), Mother(John))$

- can be $True$ or $False$ -- semantics/interpretation

Complex Sentences

- logical connectives($\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, ...) along with atomic sentences

$\neg Brother(Richard, John)$

$\neg King(Richard) \Rightarrow King(John)$

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 logic

Syntax & semantic

Quantifiers

- Universal quantification (∀)

  $\forall x\, King(x) \Rightarrow Person(x)$ "For all, if x is a King, then x is a Person"

  $\forall x\, King(x) \wedge Person(x)$ "For all, x is a King and x is a Person"

- Existential quantification (∃)

  $\exists x\, Crown(x) \wedge OnHead(x, John)$ "There exists an x such that …"

  $\exists x\, Crown(x) \Rightarrow OnHead(x, John)$ "There exists an x, if x is Crwon, then x is …"

- Nested quantifiers

  $\forall x\, \forall y\, Brother(x, y) \Rightarrow Sibling(x, y)$

  $\forall x\, \exists y\, Loves(x, y)$

Propositional logic

Syntax & semantic

Exercises 1

(a) Whoever can read is literate. (읽을 수 있으면 문맹이 아니다)

(b) Monkeys are not literate. (원숭이는 문맹이다)

(c) Some monkeys are intelligent. (어떤 원숭이는 지능적이다)

(d) Some who are intelligent cannot read. (지능적이어도 문맹일 수 있다)

Propositional logic

Syntax & semantic

Exercises 1

(a) Whoever can read is literate. (읽을 수 있으면 문맹이 아니다)

(b) Monkeys are not literate. (원숭이는 문맹이다)

(c) Some monkeys are intelligent. (어떤 원숭이는 지능적이다)

(d) Some who are intelligent cannot read. (지능적이어도 문맹일 수 있다)

(a) $\forall x[CanRead(x) \rightarrow Literate(x)]$

(b) $\forall x[Monkey(x) \rightarrow \neg Literate(x)]$

(c) $\exists x[Monkey(x) \wedge Intelligent(x)]$

(d) $\exists x[Intelligent(x) \wedge \neg CanRead(x)]$

Propositional logic

Syntax & semantic

Exercises 2

Consider a vocabulary with the following symbols:

$Occupation(p, o)$: Predicate. Person p has occupation o.
$Customer(p1, p2)$: Predicate. Person $p1$ is a customer of person $p2$.
$Boss(p1, p2)$: Predicate. Person $p1$ is a boss of person $p2$.
$Doctor$, $Surgeon$, $Lawyer$, $Actor$: Constants denoting occupations.
$Emily$, Joe: Constants denoting people.

Use these symbols to write the following assertions in first-order logic:

a. Emily is either a surgeon or a lawyer.
b. Joe is an actor, but he also holds another job.
c. All surgeons are doctors.
d. Joe does not have a lawyer (i.e., is not a customer of any lawyer).
e. Emily has a boss who is a lawyer.
f. There exists a lawyer all of whose customers are doctors.
g. Every surgeon has a lawyer.

Propositional logic

Syntax & semantic

The Kinship Domain

For example, one's mother is one's female parent:

$$\forall m, c \;\; Mother(c) = m \;\Leftrightarrow\; Female(m) \land Parent(m, c)\,.$$

One's husband is one's male spouse:

$$\forall w, h \;\; Husband(h, w) \;\Leftrightarrow\; Male(h) \land Spouse(h, w)\,.$$

Male and female are disjoint categories:

$$\forall x \;\; Male(x) \;\Leftrightarrow\; \neg Female(x)\,.$$

Parent and child are inverse relations:

$$\forall p, c \;\; Parent(p, c) \;\Leftrightarrow\; Child(c, p)\,.$$

A grandparent is a parent of one's parent:

$$\forall g, c \;\; Grandparent(g, c) \;\Leftrightarrow\; \exists p \;\; Parent(g, p) \land Parent(p, c)\,.$$

A sibling is another child of one's parents:

$$\forall x, y \;\; Sibling(x, y) \;\Leftrightarrow\; x \neq y \land \exists p \;\; Parent(p, x) \land Parent(p, y)\,.$$

$$\forall x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x)$$

Propositional logic

Syntax & semantic

The Wumpus World

$$\forall t, s, g, m, c \;\; Percept([s, Breeze, g, m, c], t) \Rightarrow Breeze(t) ,$$
$$\forall t, s, b, m, c \;\; Percept([s, b, Glitter, m, c], t) \Rightarrow Glitter(t) ,$$

$$\forall t \;\; Glitter(t) \Rightarrow BestAction(Grab, t) .$$

$$\forall x, y, a, b \;\; Adjacent([x, y], [a, b]) \Leftrightarrow$$
$$(x = a \land (y = b - 1 \lor y = b + 1)) \lor (y = b \land (x = a - 1 \lor x = a + 1)) .$$

$$\forall x, s_1, s_2, t \;\; At(x, s_1, t) \land At(x, s_2, t) \Rightarrow s_1 = s_2 .$$

$$\forall s, t \;\; At(Agent, s, t) \land Breeze(t) \Rightarrow Breezy(s)$$

$$\forall s \;\; Breezy(s) \Leftrightarrow \exists r \;\; Adjacent(r, s) \land Pit(r) .$$

$$\forall t \;\; HaveArrow(t + 1) \Leftrightarrow (HaveArrow(t) \land \neg Action(Shoot, t)) .$$

…

☞ $\text{ASKVARS}(\exists a \;\; BestAction(a, 5))$

Propositional logic

Canonical Form

- Two propositions **A** and **B** are said to be **equivalent** *(A ≡ B)* if they have the same truth value (both true or both false) in all possible cases.

$(\alpha \wedge \beta)$	$\equiv$	$(\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv$	$(\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv$	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv$	$(\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

Figure 7.11 Standard logical equivalences. The symbols α, β, and γ stand for arbitrary sentences of propositional logic.

Propositional logic

Canonical Form

- Using only (**AND** and **NOT)**, or (**OR** and **NOT)**, we can express **any logical expression**.

- Conjunctive Normal Form (CNF)
  - Is a conjunction (AND conjunction) of clauses
  - Each clause is an OR conjunction of simple propositional symbols

$$(p_1 \vee \ldots \vee p_n) \wedge \ldots \wedge (q_1 \vee \ldots \vee q_m)$$

- Disjunctive Normal Form (DNF)
  - Is is an OR of clauses
  - Each clause is an AND of simple propositional symbols

$$(p_1 \wedge \ldots \wedge p_n) \vee \ldots \vee (q_1 \wedge \ldots \wedge q_m)$$

Propositional logic

Transform to CNF

$B_{1,1} \leftrightarrow (P_{1,2} \vee P_{2,1})$

Step 1: Eliminate Biconditional (↔)

- Rule: Replace $\alpha \leftrightarrow \beta$ with $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$
- Result: $(B_{1,1} \rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \rightarrow B_{1,1})$

Step 2: Eliminate Implication (→)

- Rule: Replace $\alpha \rightarrow \beta$ with $\neg\alpha \vee \beta$
- Result: $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

Step 3: Move Negation (¬) Inward

Rule:Use De Morgan's laws and double negation

- $\neg(\alpha \vee \beta) = \neg\alpha \wedge \neg\beta$
- $\neg(\alpha \wedge \beta) = \neg\alpha \vee \neg\beta$
- $\neg\neg\alpha = \alpha$

- Result: $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

Step 4: Apply Distributive Law

Rule:Distribute $\vee$ over $\wedge$

- $\alpha \vee (\beta \wedge \gamma) = (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

Result: $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Final CNF Form: $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Propositional logic

Transform to CNF

$(A \lor B) \to (C \to D)$

1. Loại bỏ phép suy ra

   $\neg(A \lor B) \lor (\neg C \lor D)$

2. Chuyển phủ định vào trong ngoặc

   $(\neg A \land \neg B) \lor (\neg C \lor D)$

3. Phân phối

   $(\neg A \lor \neg C \lor D) \land (\neg B \lor \neg C \lor D)$

Propositional logic

Transform to CNF

Excisers: Convert these functions to CNF

1. $P \vee (\neg P \wedge Q \wedge R)$
2. $(\neg P \wedge Q) \vee (P \wedge \neg Q)$
3. $\neg(P \Rightarrow Q) \vee (P \vee Q)$
4. $(P \Rightarrow Q) \Rightarrow R$
5. $(P \Rightarrow (Q \Rightarrow R)) \Rightarrow ((P \wedge S) \Rightarrow R)$
6. $(P \wedge (Q \Rightarrow R)) \Rightarrow S$
7. $P \wedge Q \Rightarrow R \wedge S$

Propositional logic

Inference rules

Modus ponens	Modus tolens	And-Introduction	And-Elimination
$\dfrac{\alpha \Rightarrow \beta \quad \alpha}{\beta}$	$\dfrac{\alpha \Rightarrow \beta \quad \neg\beta}{\neg\alpha}$	$\dfrac{\alpha \quad \beta}{\alpha \wedge \beta}$	$\dfrac{\alpha \wedge \beta}{\alpha}$

Propositional logic

Prove using inference rules

Example: Prove S

Step	Formula	Source
1	$P \wedge Q$	Given
2	$P \Rightarrow Q$	Given
3	$Q \wedge R \Rightarrow S$	Given

Propositional logic

Prove using inference rules

Example: Prove S

Step	Formula	Source
1	P ∧ Q	Given
2	P => Q	Given
3	Q ∧ R => S	Given
4	P	(1): And-Eli
5	R	(4) + (2): Modus Po.
6	Q	(1): And-Eli

Propositional logic

Prove using inference rules

Example: Prove S

Step	Formula	Source
1	$P \wedge Q$	Given
2	$P \Rightarrow Q$	Given
3	$Q \wedge R \Rightarrow S$	Given
4	P	(1): And-Eli
5	R	(4) + (2): Modus Po.
6	Q	(1): And-Eli
7	$Q \wedge R$	(5) + (6): And-Intro
8	S	(3) + (7): Modus Po.

Deduction in propositional logic

Problem Statement

Given a set of premises {p1, p2, …, pn}, where each premise is a true propositional expression. We need to derive the conclusion c (i.e., prove that c is true). Solving the problem is equivalent to proving:

$$p1 \wedge p2 \wedge \ldots \wedge pn \rightarrow c$$

Methods:

- Method 1: Using the truth table
- Method 2: Using Robinson's refutation (proof by contradiction)
- Method 3: Using inference rules
- Method 4: Using forward or backward reasoning

Deduction in propositional logic

Prove using truth table

- Given the hypothesis H = {(p ∨ r) ∧ (q ∨ ¬r)}
- Prove: p ∨ q
- Does (p ∨ r) ∧ (q ∨ ¬r) =>(p ∨ q)?

According to the **truth table**:
whenever (p ∨ r) ∧ (q ∨ ¬r) is true,
we also have (p ∨ q) is true

p	q	r	p ∨ r	q ∨ ¬r	(p∨r)
0	0	0	0	1	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	1	1

Thank you!

You're now ready to explore the exciting world of AI!