Artificial Intelligence

Instructor: Kietikul Jearanaitanakij
Department of Computer Engineering
King Mongkut's Institute of Technology Ladkrabang

Lecture 5
Logical Agents

เป็น agent อีกรูปแบบหนึ่ง

ไม่ได้ค้นหาด้วยการ search

แต่ใช้การสรุปความ & เหตุผล

- Knowledge-based agents
- Propositional logic
- Semantic of propositional logic
- Proof by inference
- Proof by resolution

Knowledge-based agents

เขียนแค่ text file ก็ได้ → text จะถูกเขียน rule เอาไว้ (เรา pre program)
↳ จะมี fact เพิ่มระหว่างเรารันโปรแกรม

- Knowledge-based agent is the agent that has the reasoning process that operates on internal knowledge.

  ↳ มีกระบวนการสรุปความ

- The central component of a knowledge-based agent is its knowledge base, or KB, which is a set of sentences.

  ↳ ต้องรู้ knowledge base ของมัน → มี sentence, (fact)

- Each **sentence** is expressed in a special form and represents some fact about the world.

  ↳ sentence กับ fact อยู่คนละ world กัน

  today is hot ⇒ fact
  is(today, hot) ⇒ senten
  รูปแบบอื่น
  ความหมายเหมือนเดิม

- Knowledge-based agent can operate by being told or learning new knowledge about the environment and they can adapt to changes in the environment by updating the relevant knowledge.

  ↳ เมื่อ agent ได้รับ percept จาก env (สิ่งเร้า)
  ↳ การ "tell"

method

- **TELL** is a way to add new sentences to the knowledge base.
- **ASK** is a way to query what is known.

เดี๋ยวอธิบายต่อ

update knowledge
ได้สรุปความออกมา
กระบวนการเหตุผล ← agent เก็บเข้า knowledge base
เราบอก percept ให้ agent

มนุษย์เข้าใจ fact
แต่ไม่เข้าใจ → แปลง fact to sentence
จะให้มันเข้าใจ
ให้เราถามแต่ sentence

เรียนแบบการเรียนรู้ของเด็กผ่านการบอกกล่าว

② เริ่มจากเราโดน tell → บอก fact มา 9 > 6

ต้องการให้เรียนรู้เรื่องนี้

Fact: 9 is greater than 6

TELL

④ โดน tell ต่อ

Fact:6 is greater than 3

TELL

KB

① เค้าเขียนกฎไว้ล่วงหน้า

Rule: If (> x y) and (> y z) then (> x z)

③ แปลง fact → sentence

Sentence: (> 9 6)

Sentence: (> 6 3)

⑤ แปลง

⑥ agent ต้องมาทำการสรุปความออกมาแล้ว

Reasoning (Inference)

↓ การสรุปความ

↓ A => B

true => true

New sentence: (> 9 3)

⑨ ไปค้นใน kb

↓ ได้คำตอบออกมา

↓ มองจากกฎ

if (> 9 6) and (> 6 3)

↳ สรุป fact ออกมา 9>3

↳ แปลงเป็น sentence (>9 3)

⑧ วันหนึ่งอยากถามไปใน kb แทน

(> 9 ?)

ASK

คำถามไร

↓ sentence

↓ fact: อยากรู้ว่าอะไรที่ 9 มากกว่า

ทั้งหมดในขอบนี้เป็นแบบนี้เลย

? = 3, 6

sudocode

สั่งสมตะเข้ามา

return กลับอันเดียว

function KB-AGENT(*percept*) **returns** an *action*

  persistent: *KB*, a knowledge base

    t, a counter, initially 0, indicating time

สร้าง sentence ให้ percept จาก fact

① TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept*, *t*))

  action ← ASK(*KB*, MAKE-ACTION-QUERY(*t*)) ② ถาม kb ว่าเวลา t ควรทำ action อะไร

  TELL(*KB*, MAKE-ACTION-SENTENCE(*action*, *t*))

③ tell กลับ

  t ← *t* + 1

ว่า agent ทำ action อะไรที่เวลา t

action log at t.

  return *action*

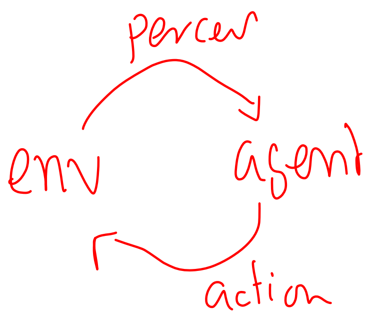
ให้ agent กลับไปทำ

Constructs a sentence asserting that the agent perceived the given percept at the given time.

Constructs a sentence that asks what action should be done at the current time.

ข้อ ④ เผื่อกี้

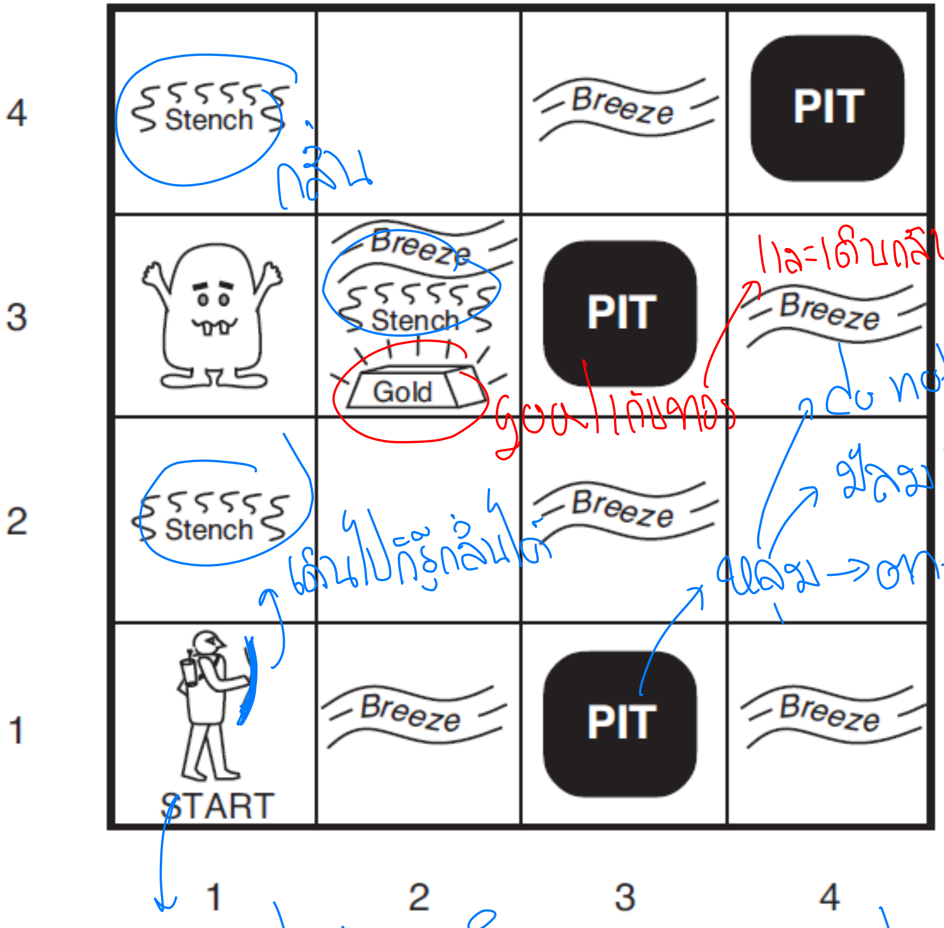
Constructs a sentence asserting that the chosen action was executed.



action might change env or percept

→ monster ที่กิน agent ได้ → คุณสมบัติ → มีกลิ่นที่แรงมาก

Case study : The Wumpus World → เหมือนกิจกรรมวันที่ 2



- The wumpus world is a cave consisting of rooms connected by passageways.
- Lurking somewhere in the cave is the terrible **wumpus**, a beast that cannot move but eats anyone who enters its room.
- The wumpus can be shot by an agent, but the agent has only one **arrow**.
- Some rooms contain bottomless **pits** that will trap anyone who wanders into these rooms (except for the wumpus, which is too big to fall in).
- The agent has to find path that leads to a heap of **gold** and safely return to the origin (1,1). หลังจากกลับมา
- Actions : { Forward, TurnLeft, TurnRight, Grab, Shoot, Climb }
- Percepts: { Stench, Breeze, Glitter, Bump, Scream }
  เหม็น, ลม, ทอง, ชน, ยิงโดน
- Environment : { Sequential, Partial observable, discrete, static, single-agent }

agent มีธนู ยิงแนว x or y only ทะลุ

1 ดอกonly → รู้ได้ว่ายิงโดน → มี percept scream ของmonster

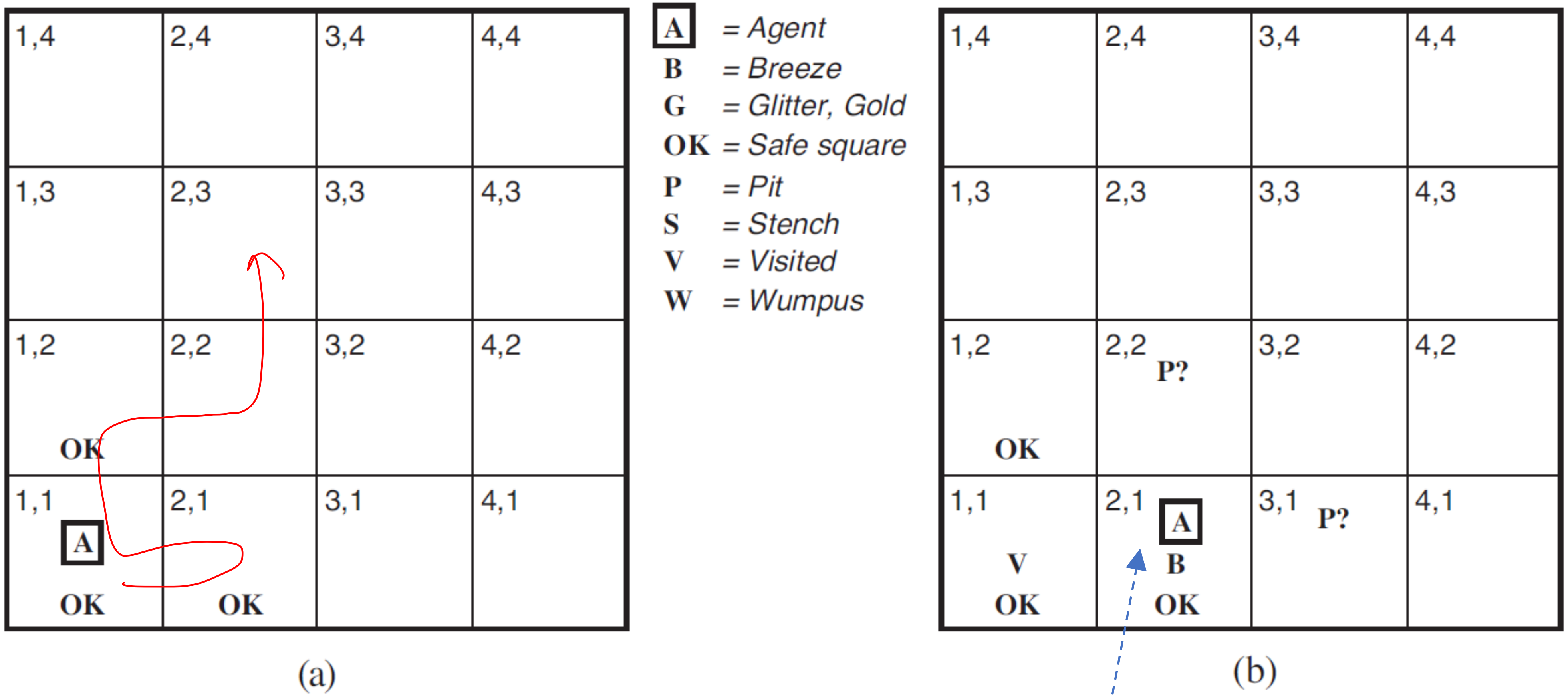


Figure 7.3 The first step taken by the agent in the wumpus world. (a) The initial situation, after percept $[None, None, None, None, None]$. (b) After one move, with percept $[None, Breeze, None, None, None]$.

The prudent agent will turn around, go back to [1,1], and then proceed to [1,2]

Assume that the agent turns and moves to [2,3].

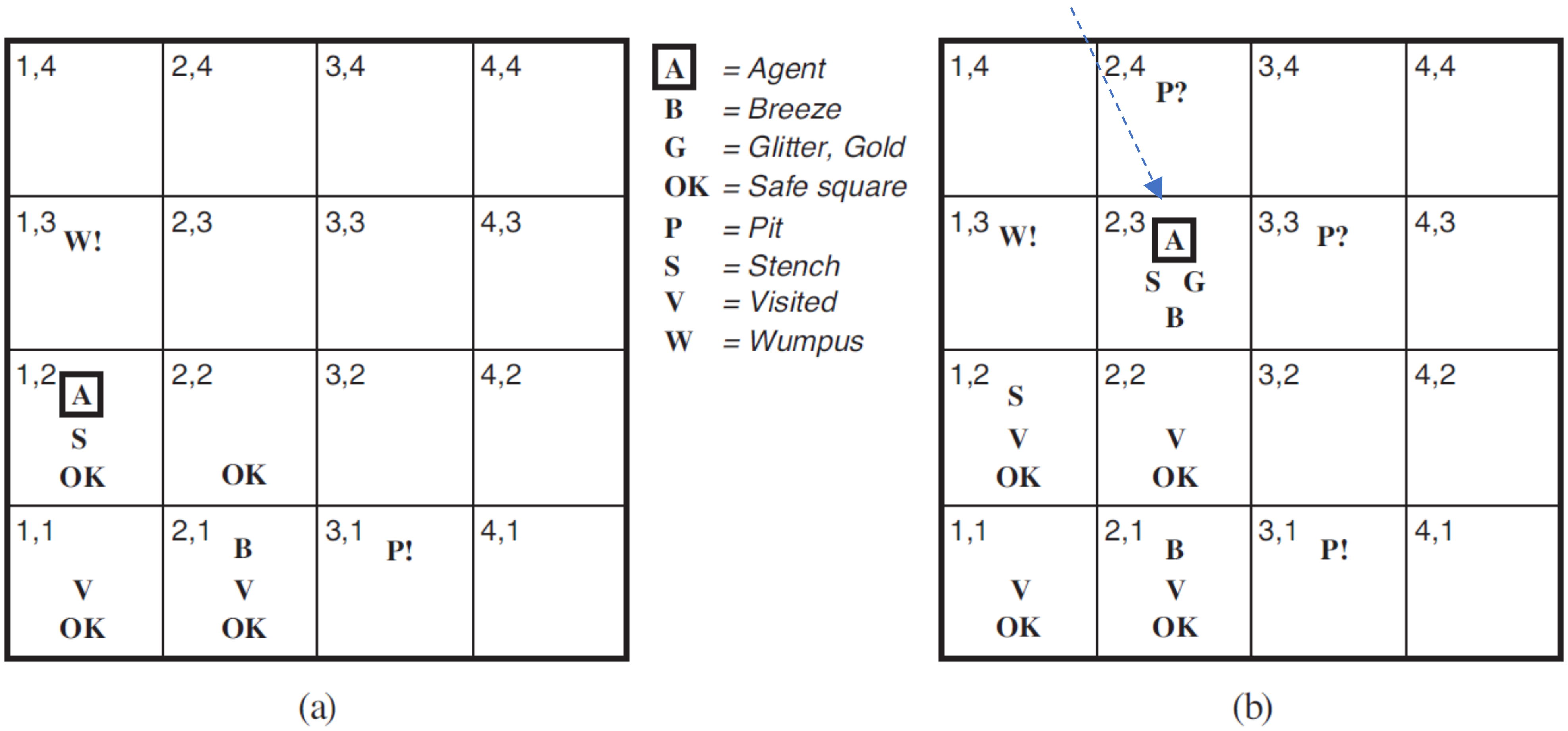


Figure 7.4 Two later stages in the progress of the agent. (a) After the third move, with percept [*Stench*, *None*, *None*, *None*, *None*]. (b) After the fifth move, with percept [*Stench*, *Breeze*, *Glitter*, *None*, *None*].

การที่สมองจะคิดได้แบบเรา → เราต้องอาศัยหลัก logic ต่อไป

Reasoning

“การอนุมาน” หรือ “inference”

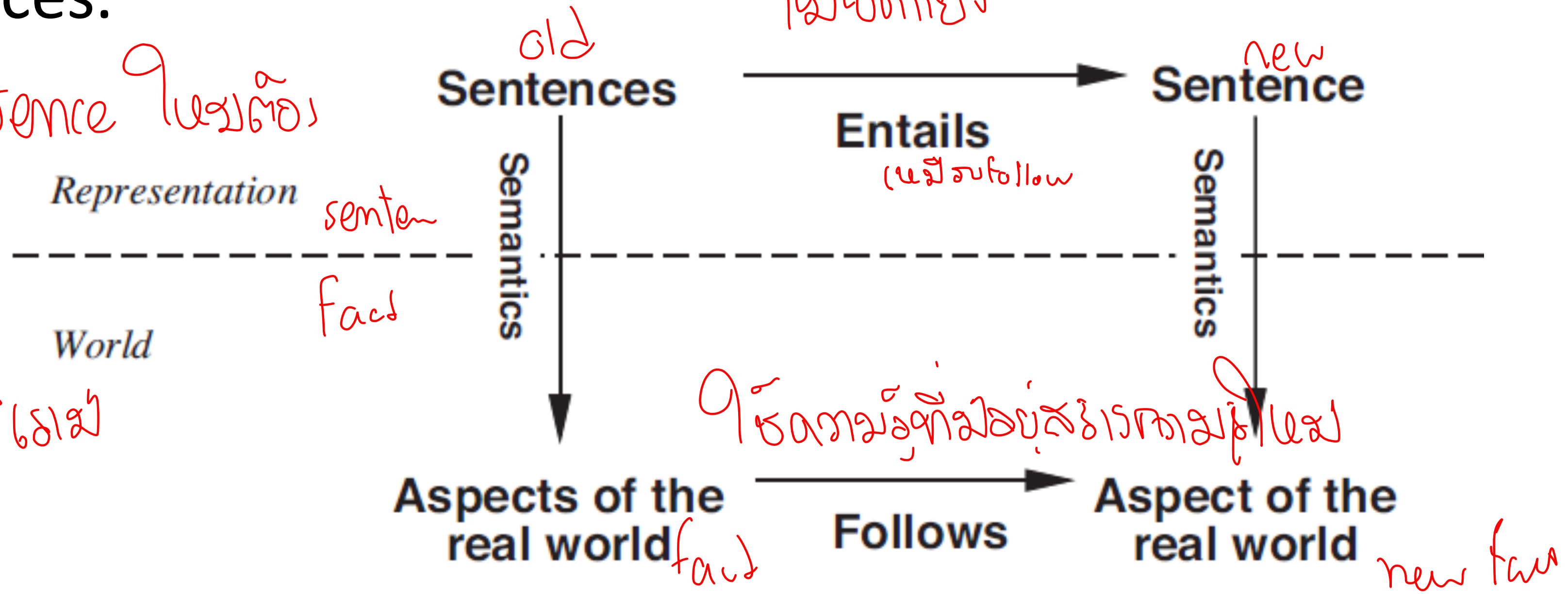
- Reasoning is the process of constructing new physical configurations (sentences) from old ones.

การสร้าง sentence ใหม่จาก sentence เดิมที่มีอยู่

- Proper reasoning should ensure that the new configuration represents facts that actually **follow** from the fact that the old configurations represent. In other words, new sentence **entails** the old sentences.

ไม่ขัดแย้ง

ควรจะมีได้ sentence ใหม่ต้อง ไม่ขัดแย้ง กับ fact เดิมที่เรามี



Propositional Logic: A Very Simple Logic

Syntax: The **syntax** of propositional logic defines the allowable sentences.

กฎต่างๆ ที่เราอนุญาตให้ใช้

แยก sentence ออกไปไม่ได้แล้ว → เช่น $W_{1,3}$ → W, W_1, W_3 } ไร้ความหมาย

- The **atomic sentences** consist of a single proposition symbol.

  symbol

  - **Symbol** represents proposition (fact) that can be true or false; e.g., $W_{1,3}$ to stand for the proposition that the wumpus is in [1,3]. Symbol is atomic, i.e., W, 1, and 3 are not meaningful parts of the symbol.
  - There are **two proposition symbols** with fixed meanings: **True** is the always-true proposition and **False** is the always-false proposition

ค่าคงที่ในเชิง logic เลย

นำ atomic sentence มาเชื่อมกัน

- **Complex sentences** are constructed from simpler sentences, using parentheses and logical connectives.

operator

There are five logical connectives in common use:

- **¬ (not)** A sentence such as $\neg W_{1,3}$ is called the negation of $W_{1,3}$.
- **∧ (and)** A sentence whose main connective is ∧, such as $W_{1,3} \land P_{3,1}$, is called a conjunction.
- **∨ (or)** A sentence using ∨, such as $(W_{1,3} \land P_{3,1}) \lor W_{2,2}$, is a disjunction of the disjuncts $(W_{1,3} \land P_{3,1})$ and $W_{2,2}$.
- **⇒ (implies)** A sentence such as $(W_{1,3} \land P_{3,1}) \Rightarrow \neg W_{2,2}$ is called an implication (or conditional). Its premise or antecedent is $(W_{1,3} \land P_{3,1})$, and its conclusion or consequent is $\neg W_{2,2}$.
- **⇔ (if and only if)** The sentence $W_{1,3} \Leftrightarrow \neg W_{2,2}$ is a biconditional. Some other books write this as ≡.

$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathit{True} \mid \mathit{False} \mid P \mid Q \mid R \mid \ldots \\
\mathit{ComplexSentence} & \rightarrow & (\ \mathit{Sentence}\ ) \mid [\ \mathit{Sentence}\ ] \\
 & \mid & \neg\ \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \land \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \lor \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \ \Rightarrow\ \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \ \Leftrightarrow\ \mathit{Sentence}
\end{array}
$$

การประกอบ

ศักย

$$
\text{OPERATOR PRECEDENCE} \quad : \quad \neg, \land, \lor, \Rightarrow, \Leftrightarrow
$$

Figure 7.7 A BNF (Backus–Naur Form) grammar of sentences in propositional logic, along with operator precedences, from highest to lowest.

Note that we can use parenthesis in propositional logic along with other operators.

Semantics

Propositional logic → syntax $(A \wedge B \Rightarrow A)$
→ semantic → "measure"

- The semantics defines the rules for determining the truth of a sentence with respect to a particular model.
- The truth of a sentence can be evaluated recursively like a **truth table**.

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*

- For example, the sentence $\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1})$

  when $P_{1,2}$ = *false*, $P_{2,2}$ = *false*, $P_{3,1}$ = *true*, gives

  true ∧ (*false* ∨ *true*)= *true* ∧ *true* =*true*.

กฎที่ map ความถูกต้องของข้อสรุป หลักการอ้าง $\frac{\text{Premise}}{\text{Conclusion}}$ $\frac{\text{ที่มา}}{\text{สรุป}}$

Seven inference rules for propositional logic

- Modus Ponens

$$\frac{\propto \rightarrow \beta, \alpha}{\beta}$$

เป็นจริง
จริง

ตาราง truth table ก็แสดงได้

$T \rightarrow X \approx T$

- And-Elimination

$$\frac{\propto_1 \wedge \propto_2 \wedge \ldots \wedge \propto_n}{\propto_i} \; ; 1 \leq i \leq n$$

→ ตัวเป็นจริงใดๆ ก็เป็นจริง

- And-Introduction

$$\frac{\propto_1, \propto_2, \ldots, \propto_n}{\propto_1 \wedge \propto_2 \wedge \ldots \wedge \propto_n}$$

ต้องเป็นจริง
เอามาเชื่อมกันได้

- Or-Introduction

$$\frac{\propto_i}{\propto_1 \vee \propto_2 \vee \ldots \vee \propto_n} \; ; 1 \leq i \leq n$$

— ใดๆ เป็นจริง

or กับอะไร $\approx T$

- Double-negation Elimination

$$\frac{\neg\neg\propto}{\propto}$$

- Unit Resolution

$$\frac{\propto \vee \beta \;, \neg\beta}{\propto}$$

$T \vee F = T$

- Resolution

$$\frac{\propto \vee \beta \;, \neg\beta \vee \gamma}{\propto \vee \gamma}$$

$\sim\alpha \rightarrow \beta \;, \; \beta \rightarrow \gamma$

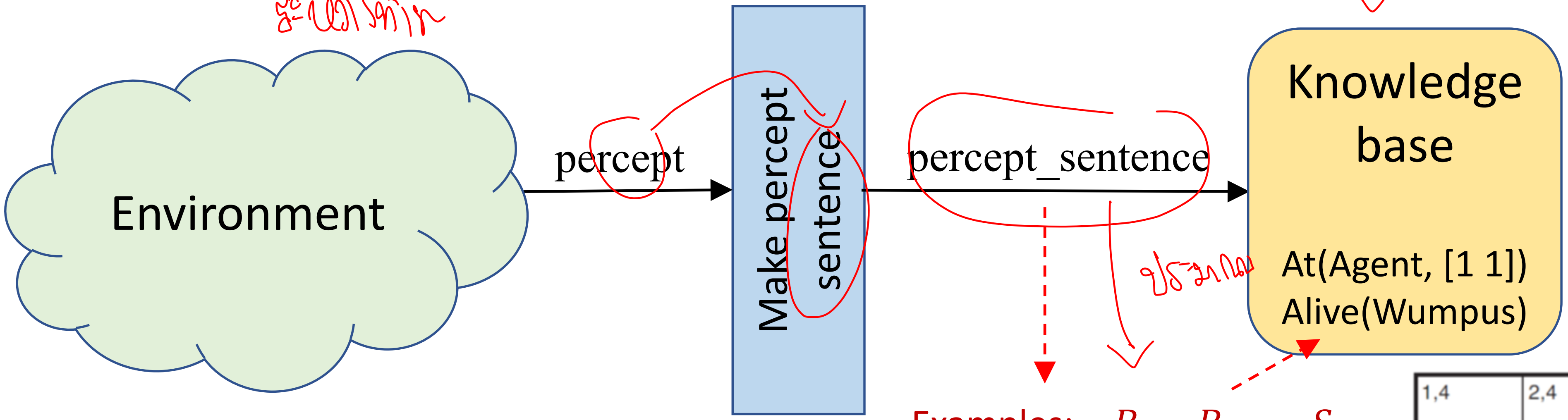
$\sim\alpha \rightarrow \gamma \approx \alpha \vee \gamma$

- The agent must start out with some knowledge of environment. For example, agent’s position is [1,1], wumpus is alive.

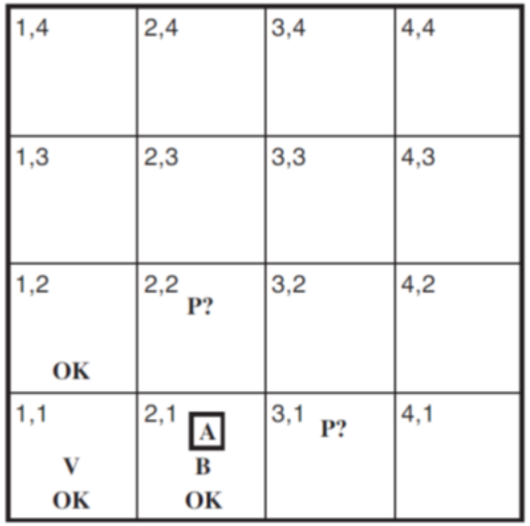
→ มีการโปรแกรมเชื่อมต่อไว้

- Knowledge base is told new percept sentences or inferred sentences at each time step.

ระบบจริง



Examples: $\neg B_{1,1}$, $B_{2,1}$, $\neg S_{1,1}$, $\neg S_{2,1}$



Example

- Let us see how these inference rules and equivalences can be used in the wumpus world.
- We start with the knowledge base containing R1 through R5 and show how to prove $\neg P_{1,2}$, that is, there is **no pit in [1,2]**.

ต้องการ

Predefine

กฎเบื้องต้น

$$R_1 : \quad \neg P_{1,1} .$$

ไม่มีหลุม

$$R_2 : \quad B_{1,1} \quad \Leftrightarrow \quad (P_{1,2} \vee P_{2,1}) .$$

ถ้ามี b แปลว่าจะ → จริง

$$R_3 : \quad B_{2,1} \quad \Leftrightarrow \quad (P_{1,1} \vee P_{2,2} \vee P_{3,1}) .$$

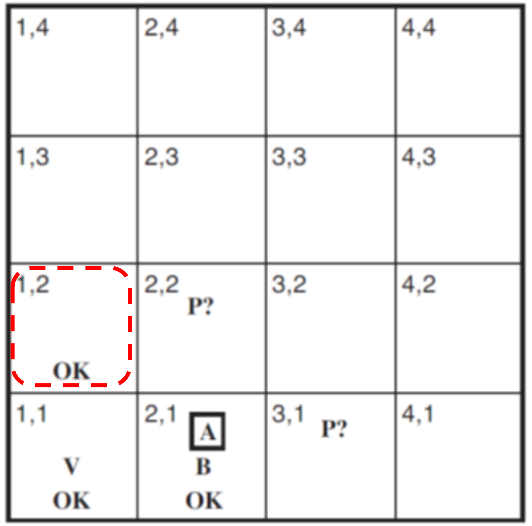
$$R_4 : \quad \neg B_{1,1} .$$

$$R_5 : \quad B_{2,1} .$$

ไม่ต้องกำหนดกฎไว้ตอนเริ่ม

ความรู้ใหม่ต้องไปต่อกับของเดิม

ข้อมูล agent จะ tell



2 ระ

Proof by inference $\neg P_{1,2}$: There is no pit in [1,2].

จาก R2

$a \Leftrightarrow b$

$\Leftrightarrow b \wedge b \rightarrow a$

- First, we apply biconditional elimination to R2 to obtain:

$$R_6 : \quad (B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}) .$$

จะมีวิธีตามลำดับ/ตัวเอง

- Then we apply And-Elimination to R6 to obtain:

$$R_7 : \quad ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}) .$$

เลือกก้อนหลังสุด

- Logical equivalence for contrapositives gives:

$a \rightarrow b \quad -b \rightarrow -a$

$T \rightarrow N$

$$R_8 : \quad (\neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1})) .$$

$\approx N$

This is called "**inference**".

- Now we can apply Modus Ponens with R8 and the percept R4 (i.e., $\neg B_{1,1}$), to obtain

assume ว่าเป็น ถูกต้องเสมอ

$$R_9 : \quad \neg(P_{1,2} \vee P_{2,1}) .$$

อ่านติดแล้วไหม

- Finally, we apply De Morgan's rule, giving the conclusion

$$R_{10} : \quad \neg P_{1,2} \wedge \neg P_{2,1} .$$

ได้ของแต่ละตัว → ก่อนจะสรุป ย้อนไปดูอดีต (เป็น)

Proof by resolution : $P_{3,1}$: There is a pit at [3,1]

2

Let us consider the steps which the agent returns from [2,1] to [1,1] and then goes to [1,2], where it perceives a stench, but no breeze. We add the following facts to the KB:

$$R_{11}: \quad \neg B_{1,2}\,.$$

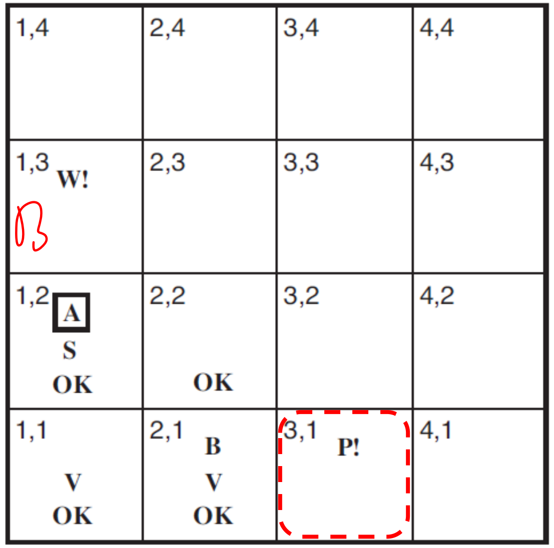
→ เกิดตามนั้น

$$R_{12}: \quad B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3})\,.$$

→ เดินจนกลับมาว่า 1,2 ไม่มี B

define กฎเพิ่ม

$B_{1,2} \Rightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3}) \wedge (P_{1,1} \vee P_{2,2} \vee P_{1,3}) \Rightarrow B_{1,2}$



By the same process that led to R10 earlier, we can derive

split

$\neg B_{1,2} \Rightarrow \neg(P_{1,1} \vee P_{2,2} \vee P_{1,3})$

$\neg P_{1,1} \wedge \neg P_{2,2} \wedge \neg P_{1,3}$

มีอยู่แล้ว

$$R_{13}: \quad \neg P_{2,2}\,.$$

$$R_{14}: \quad \neg P_{1,3}\,.$$

We can also apply biconditional elimination to R3, followed by Modus Ponens with R5, to obtain R15

$R_3: B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

$B_{2,1} \Rightarrow (\;) \wedge (\;) \Rightarrow B_{2,1}$

$$R_{15}: \quad P_{1,1} \vee P_{2,2} \vee P_{3,1}\,.$$

F

$P_{2,2} \vee P_{3,1}$

( ) ใช้ resolution ต่อจัด or

Now using the resolution rule: the literal $\neg P_{2,2}$ in R13 resolves with the literal $P_{2,2}$ in R15 to give the resolvent.

$$R_{16}: \quad P_{1,1} \vee P_{3,1} .$$

F

Similarly, the literal $\neg P_{1,1}$ in R1 resolves with the literal $P_{1,1}$ in R16 to give

$$R_{17}: \quad P_{3,1} .$$

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 $\boxed{\text{A}}$ S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

Proof by resolution : $W_{1,3}$: There is the Wumpus at [1,3]

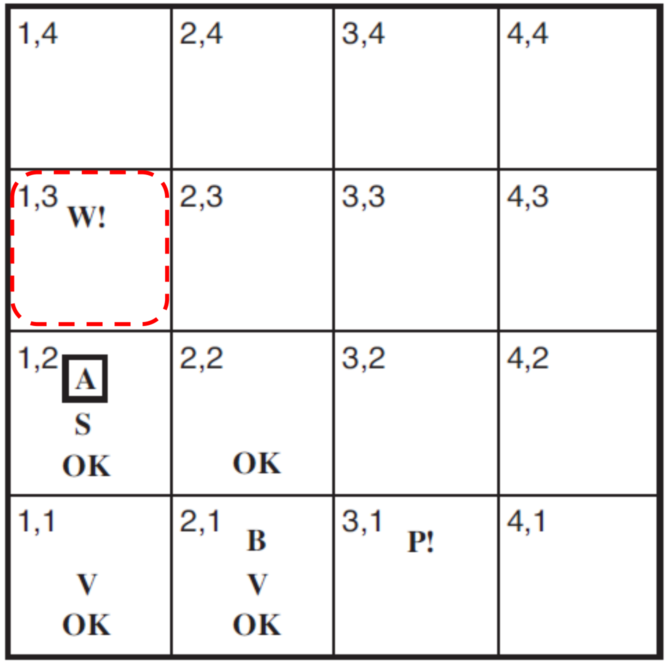
$R_1 : \neg S_{1,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$

$R_2 : \neg S_{2,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1}$

$R_3 : \neg S_{1,2} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,2} \wedge \neg W_{1,3}$

rule

$R_4 : S_{1,2} \Rightarrow W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$



Finding the Wumpus:

1. Agent starts out at [1,1] and gets the percept $\neg S_{1,1}$. By using R1 and Modus Ponens, we obtain

$$\neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$$

✓ and elimination

ยืนยันได้แค่ที่ W, S จริงๆ สนใจ B ด้วย

2. Applying And-Elimination:

$$\neg W_{1,1} \quad \neg W_{1,2} \quad \neg W_{2,1}$$

ได้ความจริงมา 3 ก้อนใส่ KB

3. Agent moves to [2,1] and gets percept $\neg S_{2,1}$.

   By using R2, Modus Ponens, And-Elimination, we obtain

$$\neg W_{2,2} \quad \neg W_{2,1} \quad \neg W_{3,1} \quad \neg W_{1,1}$$

4. Agent moves to [1,1] then [1,2] and get $S_{1,2}$.

   By using R4 and Modus Ponens, we obtain

$$W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee \cancel{W_{1,1}} \rightarrow F$$

5. Applying the unit resolution, where $\propto$ is $W_{1,3} \vee W_{1,2} \vee W_{2,2}$ and β is $W_{1,1}$.

$$W_{1,3} \vee W_{1,2} \vee W_{2,2} \rightarrow F$$

6. Applying the unit resolution again , where $\propto$ is $W_{1,3} \vee W_{1,2}$ and β is $W_{2,2}$.

$$W_{1,3} \vee W_{1,2} \rightarrow F$$

7. One more resolution where $\propto$ is $W_{1,3}$ and β is $W_{1,3}$.

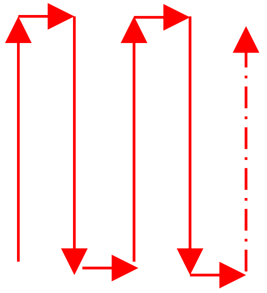
$$W_{1,3}$$

→ ได้ความรู้ใหม่ → ไปดู KB ว่าขัดแย้งไหม → ไม่ → ✓

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 $\boxed{A}$ S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

Exercise : Find a set of positions where agent visits.

Walk pattern



Translating knowledge into action

We need additional rules that relate the current state of the world to the actions the agent should take.

นอกจากจะมีกฎบอก W, B, S อยู่ตรงไหน

เราต้องมีกฎสำหรับการเดินด้วย , การยิง

For example,

กฎสำหรับ action

$$A_{1,1} \wedge East_A \wedge \neg W_{2,1} \Rightarrow Forward$$

$$A_{1,2} \wedge North_A \wedge W_{1,3} \Rightarrow TurnRight$$

$$A_{1,2} \wedge North_A \wedge W_{1,3} \wedge Alive(W) \Rightarrow Shoot$$

Problems of the propositional agent

สร้างกฎนี้ แยกมากมายเพื่อครอบทุกสถานการณ์

1. There are too many propositions to handle.

16 ช่อง 4 ทิศ

For example, The simple rule "Don't go forward if the Wumpus is in front of the agent" can be stated in propositional logic by 64 rules.

เมื่อเวลาเปลี่ยนต่างกัน → ใช้ไม่ได้ → ต้องมีการเปลี่ยนแปลง

2. Since the world can change configuration at each time step, we must specify time in the inference rules.

(Suppose that, when agent wants to turn, it always turns left.)

มีการระบุเวลาเข้าไป

$$A_{1,1}^{0} \wedge East_A^0 \wedge \neg W_{2,1} \Rightarrow Forward^0$$

ตะวันออก

$$A_{1,1}^{6} \wedge East_A^6 \wedge \neg W_{2,1} \Rightarrow TurnLeft^6$$