

Reasoning with Propositional Logic

Outline

- Knowledge-based agents
- Wumpus world
- General principles of logic
- Propositional logic
 - Basic concepts
 - Inference system
- Combine knowledge-based agent with the technology of propositional logic to build some simple agents for the wumpus world

Review: Searching

- Searching (pencarian) → teknik penyelesaian masalah dengan cara merepresentasikan masalah ke dalam state dan ruang masalah serta menggunakan strategi pencarian untuk menemukan solusi.
- Kesulitan: apakah aturan produksi (operator) sudah lengkap atau belum?
- Membutuhkan representasi state yang sangat banyak saat masalah yang dihadapi cukup kompleks.

Knowledge-based agent

- **Problem solving agent**: memilih solusi di antara kemungkinan yang ada. Apa yang ia “ketahui” tentang dunia tidak berkembang → problem solution (initial state, successor function, goal test)
- **Knowledge-based agent**: lebih “pintar”. Ia “mengetahui” hal-hal tentang dunia dan dapat melakukan **reasoning** (berpikir, bernalar) mengenai:
 - Hal-hal yang tidak diketahui sebelumnya (imperfect/partial information)
 - Tindakan yang paling baik untuk diambil

Knowledge-based agent

- Knowledge-based agent memiliki knowledge base sebagai komponen utama.
- Knowledge base → apa yang diketahui oleh si agent.
- Knowledge base berupa sekumpulan sentences dan setiap sentence diekspresikan ke dalam knowledge representation language.
- Agent harus mempunyai cara untuk menambahkan pengetahuan baru (new sentences) ke dalam knowledge base → operasi TELL
- Agent juga harus dapat melakukan query terhadap knowledge base untuk memutuskan action apa yang harus dilakukan → operasi ASK

function KB-AGENT(*percept*) **returns** an *action*
 persistent: *KB*, a knowledge base
 t, a counter, initially 0, indicating time

 TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept*, *t*))
 action \leftarrow ASK(*KB*, MAKE-ACTION-QUERY(*t*))
 TELL(*KB*, MAKE-ACTION-SENTENCE(*action*, *t*))
 t \leftarrow *t* + 1
 return *action*

Figure 7.1 A generic knowledge-based agent. Given a percept, the agent adds the percept to its knowledge base, asks the knowledge base for the best action, and tells the knowledge base that it has in fact taken that action.

- Knowledge Base:

- Himpunan representasi fakta yang diketahui tentang lingkungannya
- Tiap fakta disebut **sentence**.
- Dinyatakan dalam bahasa **formal** → bisa diolah
- TELL: menambahkan sentence baru ke KB.

- Inference Engine:

- Menentukan fakta baru yang dapat diturunkan dari pengetahuan yang sudah ada dalam KB.
- Menjawab pertanyaan (ASK) berdasarkan KB yang sudah.

- Agent dapat dipandang dari **knowledge level**: informasi apa yang diketahuinya? Mis: sebuah robot “mengetahui” bahwa gedung B ada di antara gedung A dan gedung C.
- Agent dapat dipandang dari **implementation level**: bagaimana **representasi** informasi yang diketahuinya?
 - Logical sentence: `di_antara(gdB, gdA, gdC)`
 - Natural language: “Gedung B ada di antara gedung A dan gedung C”
 - Tabel posisi koordinat gedung-gedung
 - Gambar diagram peta Fasilkom (bitmap? vector?)
- Pilihan representasi berpengaruh thd. apa yang bisa dilakukan oleh inference engine.

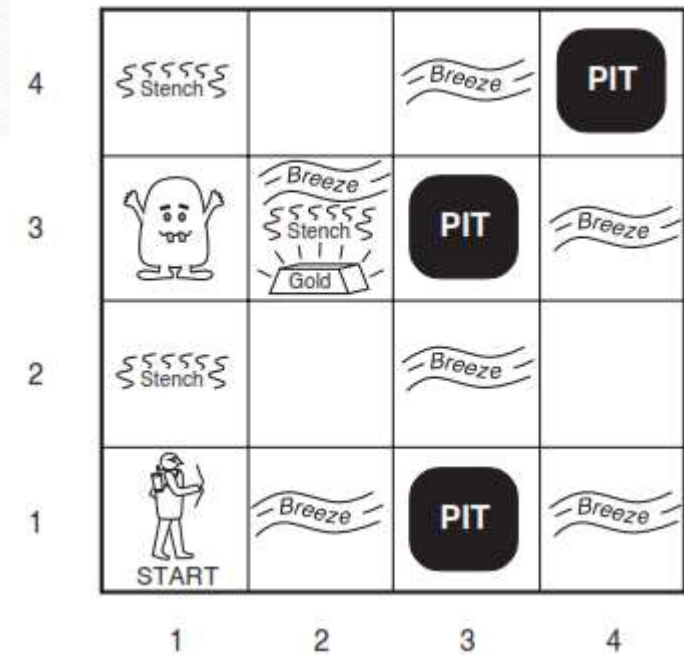
Wumpus World

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.



Wumpus World (cont.)

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.
- 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 agent has only one arrow).
- The action *Climb* can be used to climb out of the cave, but only from square [1,1].

Wumpus World (cont.)

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 any-where* in the cave.
- Example of percept : [Stench, Breeze, None, None, None]

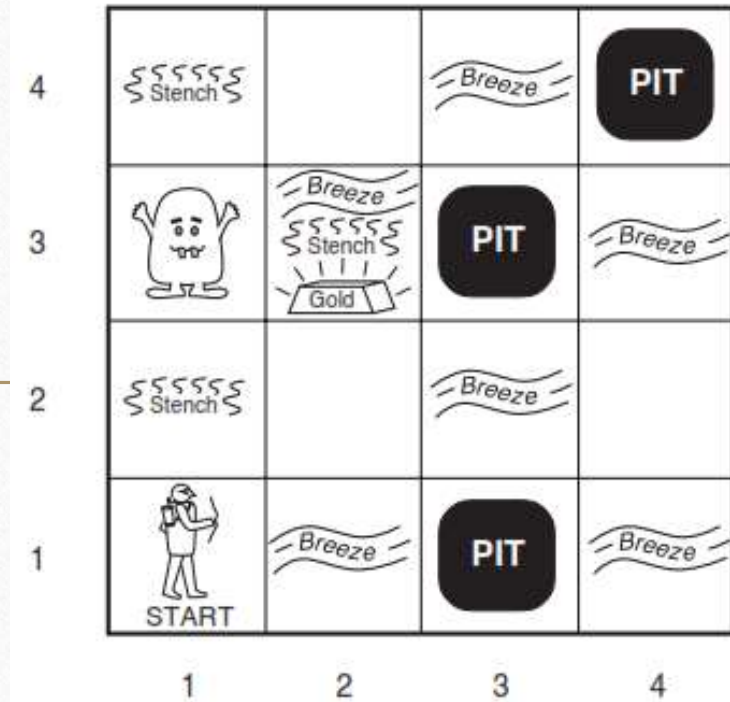
Sifat Wumpus World

- Discrete
- Static (lokasi wumpus dan pit tidak berubah)
- Single-agent
- Sequential → reward (Gold) baru didapat setelah sederetan aksi dilakukan .
- Partially observable → lokasi wumpus dan pit tidak diketahui secara pasti, namun dapat diindikasikan dari adanya percept Stench atau Breeze.

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

(a)

A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus



- Agent berada di posisi awal, yaitu [1,1] dan menghadap ke kanan
- Agent menerima percept: [None, None, None, None, None] → posisi aman (OK)
- Kemungkinan posisi berikutnya: [2,1] atau [1,2]

A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2 P?	3,2	4,2
1,1 V OK	2,1 A B OK	3,1 P?	4,1

(b)

- Agent melangkah ke posisi [2,1] dan menerima percept: [None, Breeze, None, None, None]
- Kemungkinan ada Pit di neighboring square [1,1], [2,2] atau [3,1].
- Pit tidak ada di [1,1] by the rules of the game → Pit di [2,2] atau [3,1]
- Kemungkinan posisi berikutnya adalah **[2,2]** atau **[3,1]** atau [1,1] (kembali ke posisi awal)

- Agent kembali ke posisi [1,1] dan melangkah ke [1,2]
- Agent menerima percept: [Stench, None, None, None, None]
- Kemungkinan ada Wumpus di [1,1] atau [2,2] atau [1,3]
 - [1,1] tidak mungkin by the rules of the game
 - [2,2] tidak mungkin, karena jika Wumpus di [2,2] harusnya ada Stench di [2,1]
 - Wumpus pasti [1,3]
- Pengetahuan sebelumnya: Pit diantara [2,2] atau [3,1]
 - [2,2] tidak mungkin karena jika Pit di [2,2] harusnya ada Breeze di [1,2]
 - Pit pasti di [3,1]
- Kemungkinan agent melangkah dari posisi [1,2] adalah : **[1,3]** atau [2,2] atau **[1,1]** (kembali)

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus

(a)

- Misal dari posisi [2,2] agent menerima percept [None, None, None, None, None]
- Selanjutnya agent memutuskan untuk melangkah ke posisi [2,3] dan menerima percept [Stench, Breeze, Glitter, None, None]
 - Glitter → Agent dapat langsung melakukan aksi Grab untuk mengambil Emas
 - Stench → karena ada Wumpus di [1,3] (sudah diketahui sebelumnya)
 - Breeze → kemungkinan ada Pit di [1,3] atau [2,4] atau [3,3]
 - [1,3] tidak mungkin karena sudah ada Wumpus di lokasi tersebut

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 A S G B	3,3 P?	4,3
1,2 S V OK	2,2 V OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

(b)

-
- Note that in each case for which the agent draws a conclusion from the available information.
 - That conclusion is *guaranteed to be correct if the available information is correct*.
 - This is a fundamental property of logical reasoning.

Logic

- Knowledge base berupa sekumpulan sentences dan setiap sentence diekspresikan ke dalam knowledge representation language (KRL).
- Syntax: aturan yang mendefinisikan sentence yang sah dalam Bahasa.
- Semantics: aturan yang mendefinisikan arti dari sebuah sentence.
 - Semantik mendefinisikan kebenaran sentence di dalam dunia (*possible world*).

KRL bahasa aritmatika

- **Syntax:**

- $x + 2 \geq y$ adalah kalimat sah.
- $x2 + y \geq$ bukan kalimat sah.

- **Semantics:** $x + 2 \geq y$ benar jhj bilangan $x + 2$ tidak lebih kecil dari bilangan y :

- $x + 2 \geq y$ benar dalam "dunia" di mana $x = 7, y = 1$
- $x + 2 \geq y$ salah dalam "dunia" di mana $x = 0, y = 6$

KRL bahasa Indonesia

- **Syntax:**

- "Jakarta adalah ibukota Indonesia" adalah kalimat sah.
- "Ibu Indonesia kota Jakarta adalah" bukan kalimat sah.

- **Semantics:** "X adalah ibukota Y" benar jhj X adalah pusat pemerintahan negara Y.

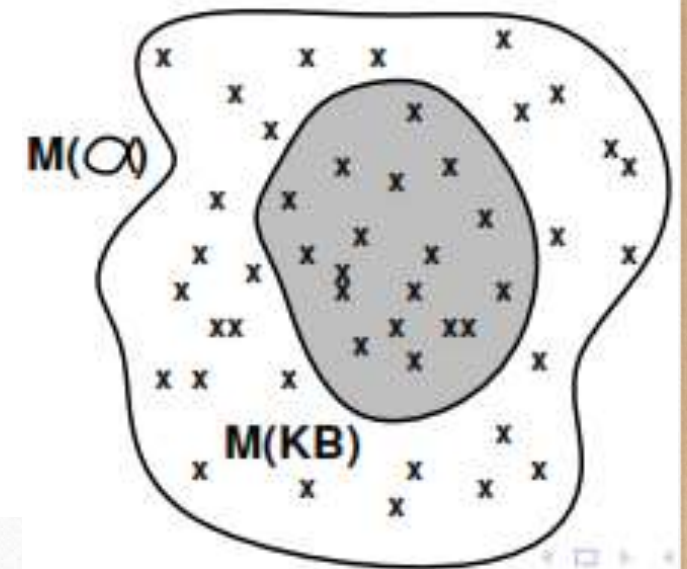
- "Jakarta adalah ibukota Indonesia" benar dalam "dunia" kita sekarang.
- "Jakarta adalah ibukota Indonesia" salah dalam "dunia" th. 1948 (Yogya? Bukittinggi?).

Entailment

- **Entailment** berarti sesuatu fakta bisa disimpulkan dari (kumpulan) fakta lain.
- $KB \models \alpha$: KB entails sentence α jh α true dalam semua “dunia” di mana KB true.
- Contoh:
 - KB mengandung sentence “*Anto ganteng*” dan “*Ani cantik*”.
 - $KB \models \alpha_1$: “*Anto ganteng dan Ani cantik*”
 - $KB \not\models \alpha_2$: “*Anto pintar*”
 - $x + y = 4 \models 4 = x + y$

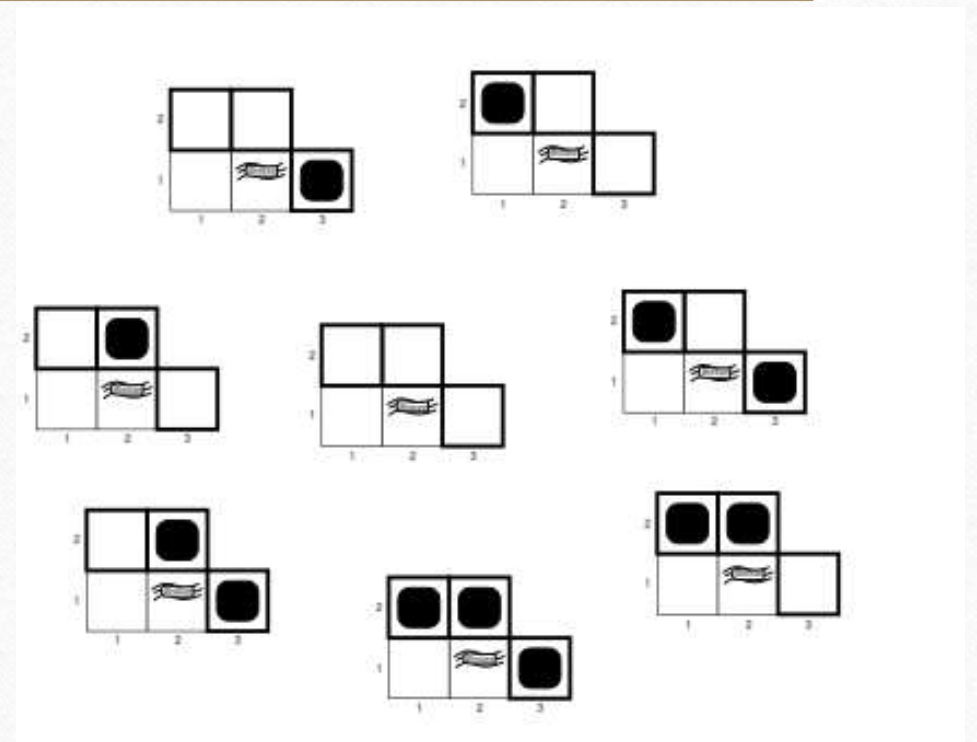
Model dan Entailment

- **Model**: sebuah “dunia” di mana kebenaran suatu sentence bisa diuji.
- m adalah model α jika α **true** di “dalam” m .
- $M(\alpha)$ adalah himpunan semua model dari α
- $KB \models \alpha$ jh $M(KB) \subseteq M(\alpha)$
- Mis:
 $KB = \text{Anto ganteng dan Ani cantik.}$
 $\alpha = \text{Anto ganteng.}$

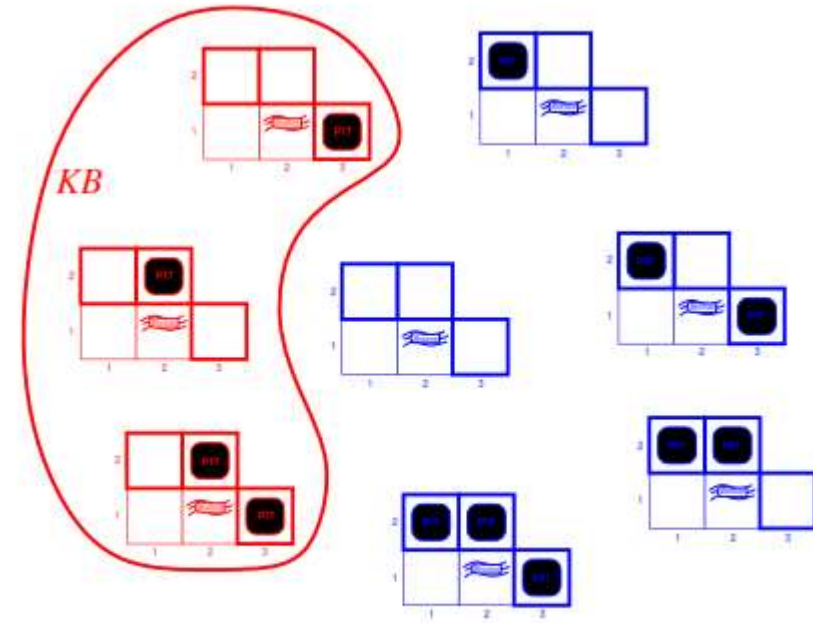


Model dan Entailment (cont.)

- Contoh kasus entailment dan model dalam dunia wumpus
- Himpunan model: Pit dapat berada di posisi [1,2] atau [2,2] atau [3,1] → ada $2^3 = 8$ kemungkinan.



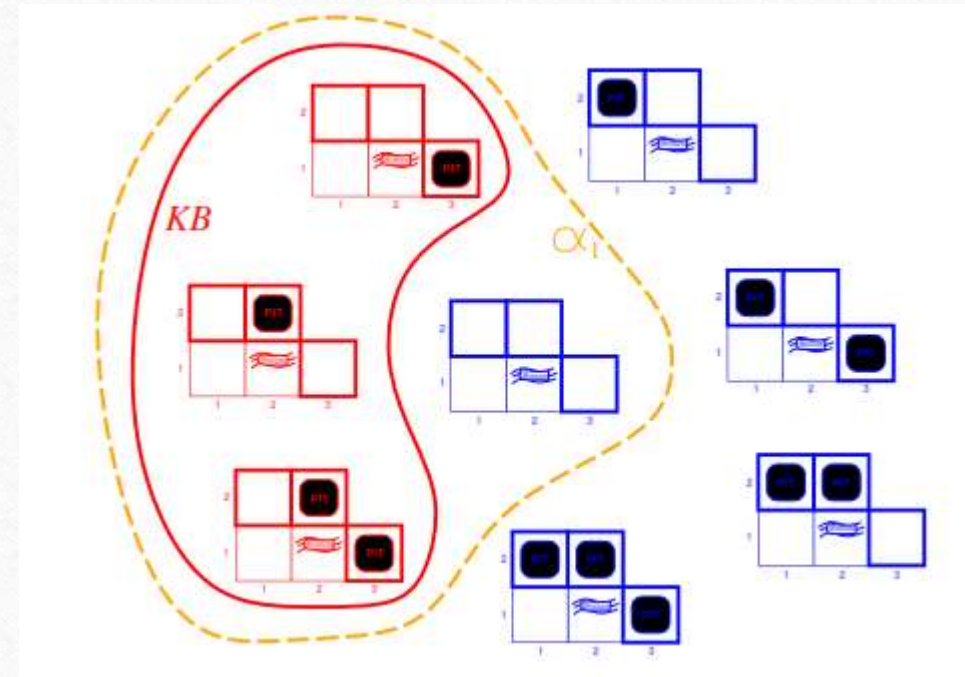
- Knowledge: Agent tidak menerima percept apapun di posisi [1,1] dan menerima percept Breeze di posisi [2,1]
- KB adalah False dalam model yang kontradiksi dengan pengetahuan agent.
- KB adalah False dalam setiap model yang mana posisi [1,2] terdapat Pit karena tidak ada Breeze di posisi [1,1].
- Hanya ada 3 model dimana KB bernilai True.



KB = pengamatan (percept) + aturan main Wumpus World

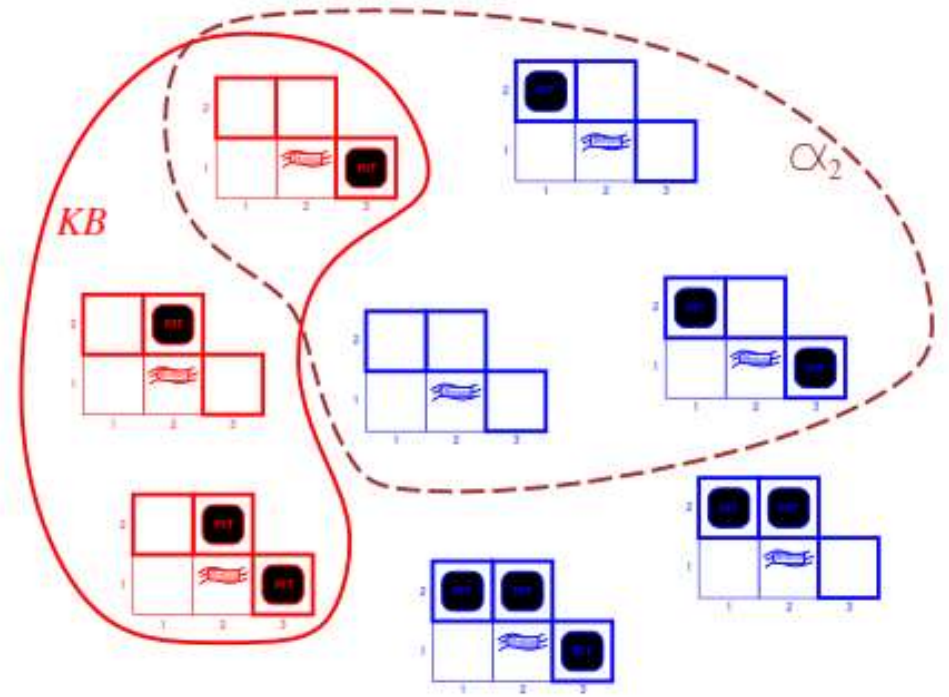
Model dan Entailment (cont.)

- Misal kita tarik sebuah sentence α_1 = "tidak ada Pit di [1,2]"
- Dalam setiap model dimana KB bernilai True, α_1 juga bernilai True
- Dapat dikatakan bahwa KB **entails** α_1



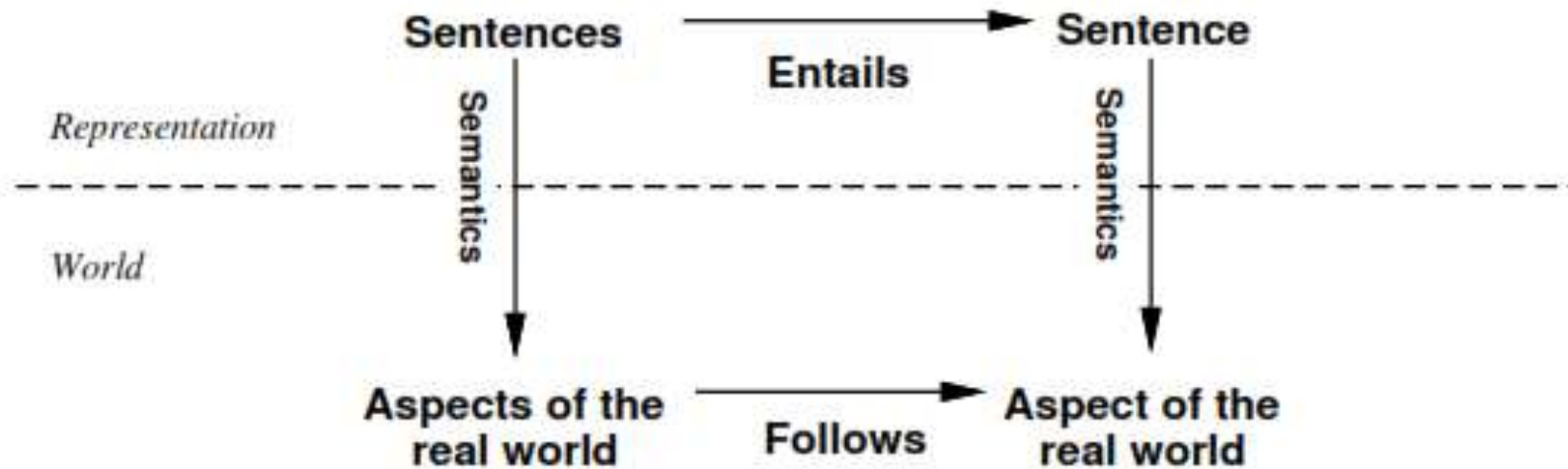
Model dan Entailment (cont.)

- Misal kita tarik sentence berikutnya:
 α_2 = "tidak ada Pit di [2,2]"
- Dalam setiap model dimana KB bernilai True, α_2 tidak selalu bernilai true
- Dengan demikian, KB not entails α_2



The preceding example not only illustrates entailment but also shows how the definition of entailment can be applied to derive conclusions—that is, to carry out **logical inference**. The inference algorithm illustrated in Figure 7.5 is called **model checking**, because it enumerates all possible models to check that α is true in all models in which KB is true, that is, that $M(KB) \subseteq M(\alpha)$.

- **Inference**, atau **reasoning**: pembentukan fakta (sentence) baru yang meng-**entail** fakta-fakta lama.
- **Reasoning** bukan dilakukan pada fakta di dunia (semantics), melainkan **representasi** fakta dalam KRL si agent (syntax).
- Otak manusia melakukan proses reasoning dalam suatu bentuk syntax!



Inference

- **Inference** adalah proses/algorithm yang “menurunkan” fakta baru dari fakta-fakta lama.
- $KB \vdash_i \alpha$: sentence α bisa diturunkan dari KB oleh prosedur i
- **Soundness**: i dikatakan *sound* jika untuk semua $KB \vdash_i \alpha$, $KB \models \alpha$ benar
- **Completeness**: i dikatakan *complete* untuk semua $KB \models \alpha$, $KB \vdash_i \alpha$ benar

Lima Jenis Logic [RUS95]

Jenis <i>logic</i>	Apa yang ada di dunia nyata	Apa yang dipercaya <i>Agent</i> tentang fakta
<i>Propositional logic</i>	fakta	benar/salah /tidak diketahui
<i>First-order logic</i>	fakta, objek, relasi	benar/salah /tidak diketahui
<i>Temporal logic</i>	fakta, objek, relasi, waktu	benar/salah /tidak diketahui
<i>Probability theory</i>	fakta	derajat kepercayaan [0,1]
<i>Fuzzy logic</i>	derajat kebenaran	derajat kepercayaan [0,1]

Propositional Logic (PL)

- *Logic* yang paling sederhana
- Sangat mudah dipahami
- Membuat kita lebih mudah membedakan teknik *reasoning* dengan teknik *searching*.

$$\begin{aligned}
 \textit{Sentence} &\rightarrow \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
 \textit{AtomicSentence} &\rightarrow \textit{True} \mid \textit{False} \mid P \mid Q \mid R \mid \dots \\
 \textit{ComplexSentence} &\rightarrow (\textit{Sentence}) \mid [\textit{Sentence}] \\
 &\mid \neg \textit{Sentence} \\
 &\mid \textit{Sentence} \wedge \textit{Sentence} \\
 &\mid \textit{Sentence} \vee \textit{Sentence} \\
 &\mid \textit{Sentence} \Rightarrow \textit{Sentence} \\
 &\mid \textit{Sentence} \Leftrightarrow \textit{Sentence}
 \end{aligned}$$

OPERATOR PRECEDENCE : $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

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

Propositional Logic (cont.)

- Atomic sentence hanya berisi satu proposition symbol.
- Sebuah proposition symbol dapat bernilai True atau False dan biasanya ditulis dengan huruf kapita, misal P, Q, atau R.
- Sebuah proposition symbol dapat menyatakan fakta apapun yang kita inginkan, misal:
 - P = Bandung adalah kota yang nyaman
 - Q = Bandung adalah kota yang macet

Semantics of Propositional Logic

- Semantik mendefinisikan aturan untuk menentukan kebenaran dari sebuah sentence di dalam model tertentu.
- Aturan untuk menentukan kebenaran dari atomic sentence
 - *True* is true in every model and *False* is false in every model.
 - The truth value of every other proposition symbol must be specified directly in the model. For example, in the model m_1 given earlier, $P_{1,2}$ is false.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>
<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>

Figure 7.8 Truth tables for the five logical connectives. To use the table to compute, for example, the value of $P \vee Q$ when P is true and Q is false, first look on the left for the row where P is *true* and Q is *false* (the third row). Then look in that row under the $P \vee Q$ column to see the result: *true*.

A simple knowledge base

- We need the following symbols for each $[x,y]$ location

$P_{x,y}$ is true if there is a pit in $[x, y]$.

$W_{x,y}$ is true if there is a wumpus in $[x, y]$, dead or alive.

$B_{x,y}$ is true if the agent perceives a breeze in $[x, y]$.

$S_{x,y}$ is true if the agent perceives a stench in $[x, y]$.

- Seperti contoh sebelumnya, kita ingin membuktikan bahwa KB entails α_1 (tidak ada pit di [1,2]).
- Misal kita memberi label setiap sentence dengan R_i

- There is no pit in [1,1]:

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

- A square is breezy if and only if there is a pit in a neighboring square. This has to be stated for each square; for now, we include just the relevant squares:

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

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

- The preceding sentences are true in all wumpus worlds. Now we include the breeze percepts for the first two squares visited in the specific world the agent is in, leading up to the situation in Figure 7.3(b).

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

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

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	false	false	false	false	false	false	true	true	true	true	false	false
false	false	false	false	false	false	true	true	true	false	true	false	false
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
false	true	false	false	false	false	false	true	true	false	true	true	false
false	true	false	false	false	false	true	true	true	true	true	true	<u>true</u>
false	true	false	false	false	true	false	true	true	true	true	true	<u>true</u>
false	true	false	false	false	true	true	true	true	true	true	true	<u>true</u>
false	true	false	false	true	false	false	true	false	false	true	true	false
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
true	true	true	true	true	true	true	false	true	true	false	true	false

Figure 7.9 A truth table constructed for the knowledge base given in the text. KB is true if R_1 through R_5 are true, which occurs in just 3 of the 128 rows (the ones underlined in the right-hand column). In all 3 rows, $P_{1,2}$ is false, so there is no pit in $[1,2]$. On the other hand, there might (or might not) be a pit in $[2,2]$.

Menentukan entailment dengan model cheking → mengenumerasi semua model dan menunjukkan sentence yang bersesuaian dengan semua model tersebut.

Propositional Theorem Proving

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

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

- **Logical equivalence**, two sentences α dan β logically equivalent jika keduanya bernilai True dalam himpunan model yang sama.

1. Modus Ponens atau **Implication-Elimination:**

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

2. And-Elimination:

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

3. And-Introduction:

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

4. Or-Introduction:

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

5. Double-Negation-Elimination:

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

6. Unit Resolution:

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

7. Resolution:

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

ekivalen dengan

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

Example of Inference and Proofs

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

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

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

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

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

We start with the knowledge base containing R_1 through R_5 and show how to prove $\neg P_{1,2}$, that is, there is no pit in [1,2]. First, we apply biconditional elimination to R_2 to obtain

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

Then we apply And-Elimination to R_6 to obtain

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

Logical equivalence for contrapositives gives

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

Now we can apply Modus Ponens with R_8 and the percept R_4 (i.e., $\neg B_{1,1}$), to obtain

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

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

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

That is, neither [1,2] nor [2,1] contains a pit.

We begin by using a simple version of the resolution rule in the wumpus world. Let us consider the steps leading up to Figure 7.4(a): 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 knowledge base:

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

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

By the same process that led to R_{10} earlier, we can now derive the absence of pits in [2,2] and [1,3] (remember that [1,1] is already known to be pitless):

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

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

We can also apply biconditional elimination to R_3 , followed by Modus Ponens with R_5 , to obtain the fact that there is a pit in [1,1], [2,2], or [3,1]:

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

Now comes the first application of the resolution rule: the literal $\neg P_{2,2}$ in R_{13} *resolves with* the literal $P_{2,2}$ in R_{15} to give the **resolvent**

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

In English; if there's a pit in one of $[1,1]$, $[2,2]$, and $[3,1]$ and it's not in $[2,2]$, then it's in $[1,1]$ or $[3,1]$. Similarly, the literal $\neg P_{1,1}$ in R_1 resolves with the literal $P_{1,1}$ in R_{16} to give

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

In English: if there's a pit in $[1,1]$ or $[3,1]$ and it's not in $[1,1]$, then it's in $[3,1]$. These last two inference steps are examples of the **unit resolution** inference rule,

Langkah Selanjutnya

- Ketika lokasi Wumpus sudah diketahui, *agent* bisa menggunakan panah untuk membunuhnya atau fokus pada pencarian emas dengan menghindari lokasi Wumpus.
- Kita bisa membuat berbagai strategi sehingga bisa membawa emas kembali ke posisi (1,1) dengan jumlah aksi yang semimum mungkin.

Kelebihan PL

- Pada kasus Wumpus tersebut, *agent* harus dibekali aturan yang merupakan pengetahuan tentang *environment* dan aturan untuk menerjemahkan pengetahuan menjadi aksi
- Untuk 16 ruangan yang ada di dalam gua, diperlukan sangat banyak aturan tentang *environment* dan aturan penerjemah.
- Pada kasus Wumpus ini, teknik manakah yang lebih efisien dalam representasi masalah, *searching* atau *reasoning*?
- Tentu saja *reasoning* lebih efisien.

Kelemahan PL

- Bagaimanapun, untuk kasus yang lebih kompleks seperti permainan catur, *propositional logic* akan sulit digunakan.
- Terdapat sangat banyak aturan pada permainan catur sedangkan *propositional logic* merepresentasikan fakta hanya dalam simbol-simbol sederhana.
- Oleh karena itu, kita akan membahas satu *logic* yang lebih tinggi tingkatannya, yaitu *First-Order Logic* yang juga disebut sebagai *Predicate Logic* atau *Predicate Calculus*.

Daftar Pustaka

- **[SUY07]** Suyanto. 2007. Artificial Intelligence: Searching, Reasoning, Planning and Learning. Informatika, Bandung Indonesia. ISBN: 979-1153-05-1.
- **[RUS95]** Russel, Stuart and Norvig, Peter. 1995. Artificial Intelligence: A Modern Approach. Prentice Hall International, Inc.