Artificial Intelligence

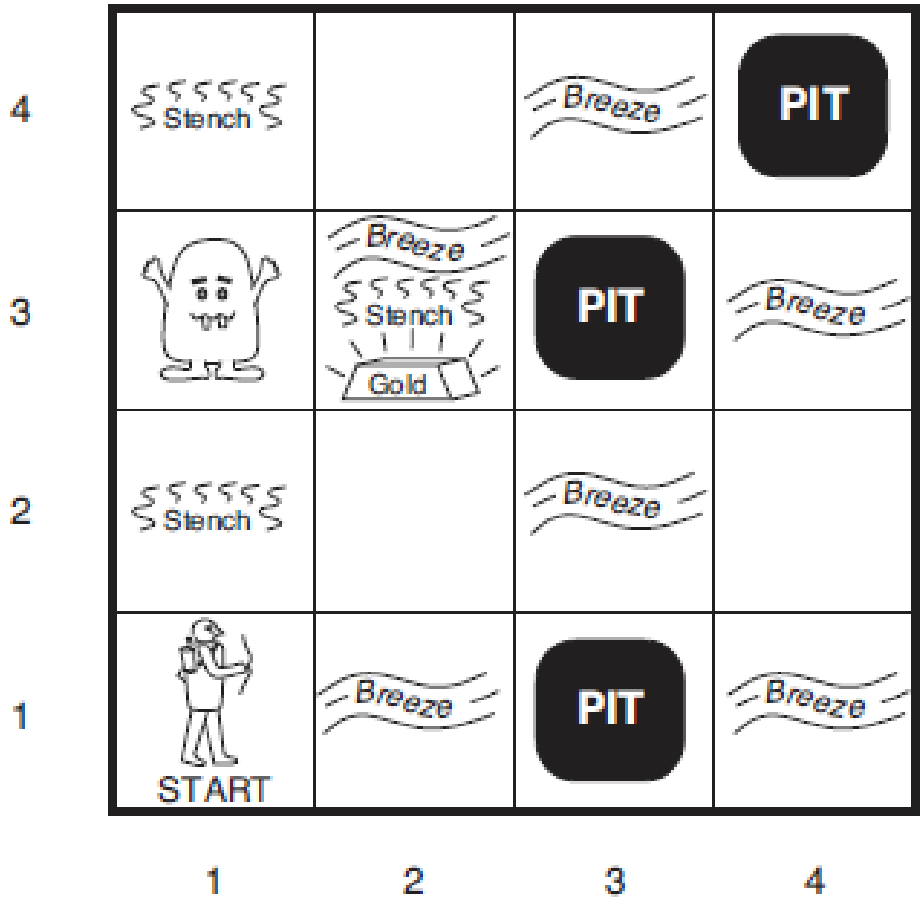
CS482, CS682, MW 1 – 2:15, SEM 201, MS 227

Prerequisites: 302, 365

Instructor: Sushil Louis, sushil@cse.unr.edu, http://www.cse.unr.edu/~sushil

Logic

Logical Agents



Truth tables you know

E1	E2	E1&E2	E1V E2	E1➔E2	!E1 V E2
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	T
F	F	F	F	T	T

Logical agents

- Logical agents **reason** on internal **representations** of knowledge
- **Knowledge-based agent** (KBA)

- Previously states were represented as
  - Black boxes – is state a goal or not?
  - A set of variables and their assignments – Do these variable assignments satisfy problem's constraints?
- Now
  - **Logic** is a general class of representations to support KBAs
  - Combine and recombine information, old and new
- Logic is old and the rules are well developed. If a problem permits a logic representation → we can use well understood tools to solve it
- Many problems do not permit logic representations

Initial Vocabulary

- Logic is old and has thus developed an extensive vocabulary

- Knowledge base (KB)
  - Is a set of **sentences** expressed in a **knowledge representation language**
  - Some sentences are **axioms**
  - **Tell** adds a sentence to the KB
  - **Ask** queries the KB
  - Both operations may require **inference** → deriving new sentences from old
    - Not allowed to make up stuff when deriving new sentences from old

KBA

- Tell an agent what it needs to **know**
- Tell an agent goals to achieve
- This is a declarative approach to building an agent/system

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*

Wumpus world

- Gold and wumpus locations chosen randomly
- 0.2 probability that a square contains a pit

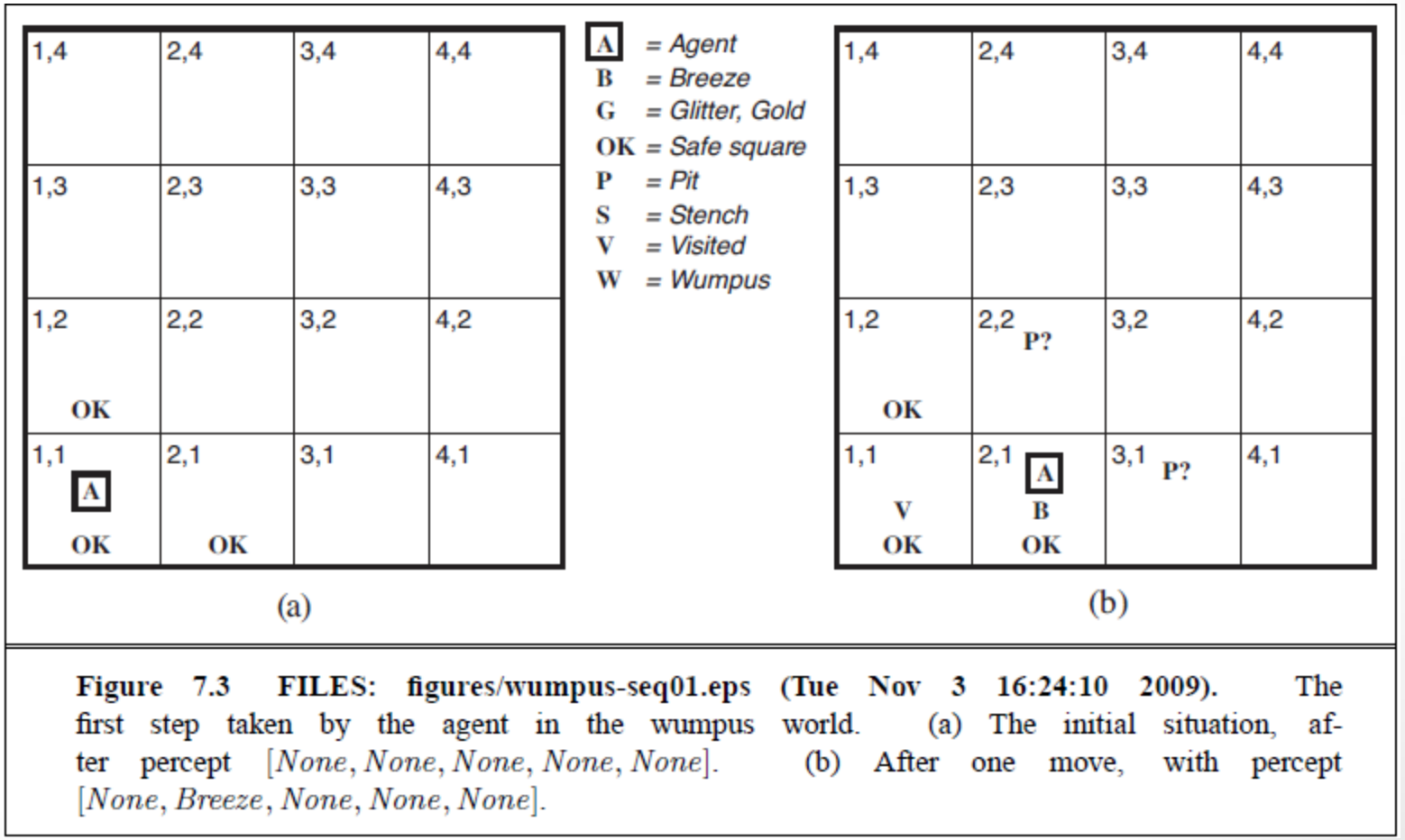


Figure 7.3 FILES: figures/wumpus-seq01.eps (Tue Nov 3 16:24:10 2009). The first step taken by the agent in the wumpus world. (a) The initial situation, after percept [*None*, *None*, *None*, *None*, *None*]. (b) After one move, with percept [*None*, *Breeze*, *None*, *None*, *None*].

Move safely, but where?

- Based on guaranteed correct updating of new knowledge
- aka → Sound Rules of Inference

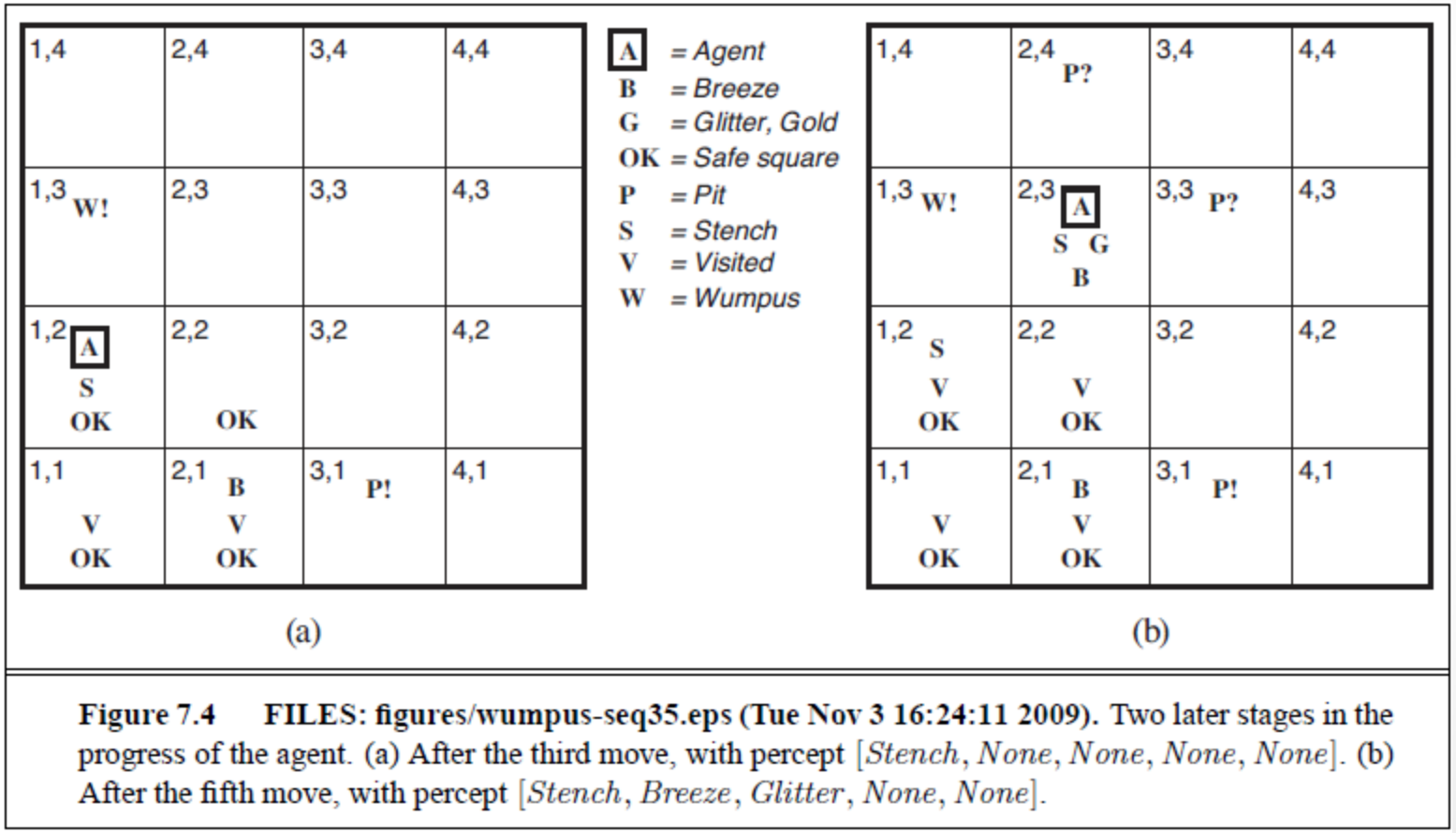


Figure 7.4 **FILES: figures/wumpus-seq35.eps (Tue Nov 3 16:24:11 2009).** Two later stages in the progress of the agent. (a) After the third move, with percept [*Stench*, *None*, *None*, *None*, *None*]. (b) After the fifth move, with percept [*Stench*, *Breeze*, *Glitter*, *None*, *None*].

More Vocabulary

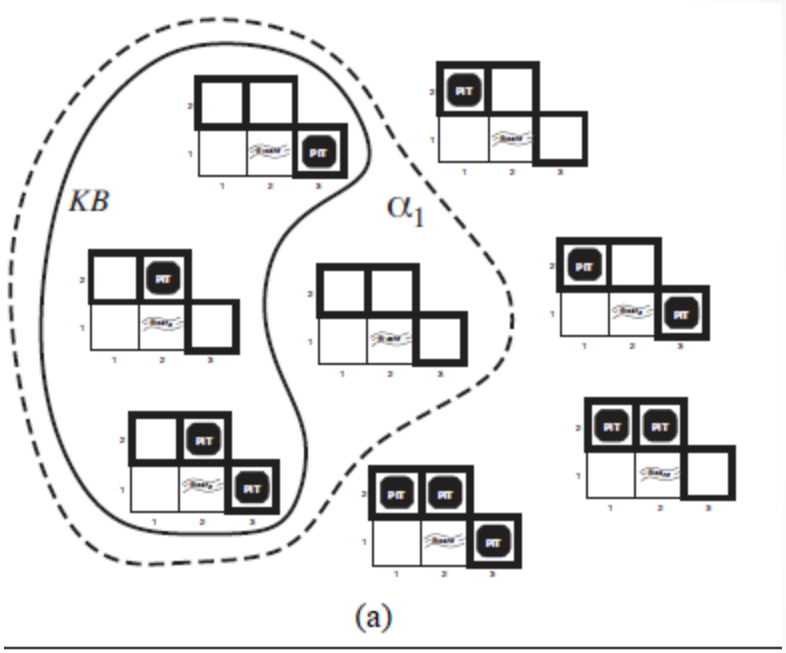
- x + y = 4 is a sentence (call the sentence → alpha)
  - Sentence has syntax. Well formed sentence (well formed formula (wff))
  - Sentence has semantics
    - Semantics defines the **truth** of sentence w.r.t to each **possible world**
- There are **possible worlds** in which alpha is true **or** false
- In classical logic only possibilities are true and false
- In Fuzzy logic, we can have "in-between"
- There are **models** in which alpha is true or false (but a model need not have any connection to the real world)
- A model **m** satisfies alpha if alpha is true in **m**.
  - Ex: **m** = (2, 2)
  - Also stated as: **m** is a **model of** alpha
- **M**(alpha) is the set of all models of alpha
  - **M** = {(x = 0, y = 4), (x = 1, y = 3), (x = 2, y = 2), (x = 3, y = 1), (x = 4, y = 0)}
  - (x = 2, y = 3) is **not** a model of alpha and not a member of **M**

Entailment

- A sentence beta **logically follows from** alpha
- alpha entails beta
  - Iff M(alpha) is a subset of M(beta)
  - Alpha is a **stronger** assertion than beta, it rules out more possible worlds
  - x = 0 entails xy = 0, since in any world where x is 0, xy is zero

Knowledge in the Wumpus world

- Possible **models** for the presence of pits in [1,2], [2,2], [3,1]
- Each square may or may not contain pit
  - 2^3 == 8 possibilities
- KB is what is **known**
  - Cannot have pit in [2,1]
  - Must have pit in [2,2] or [3,1]

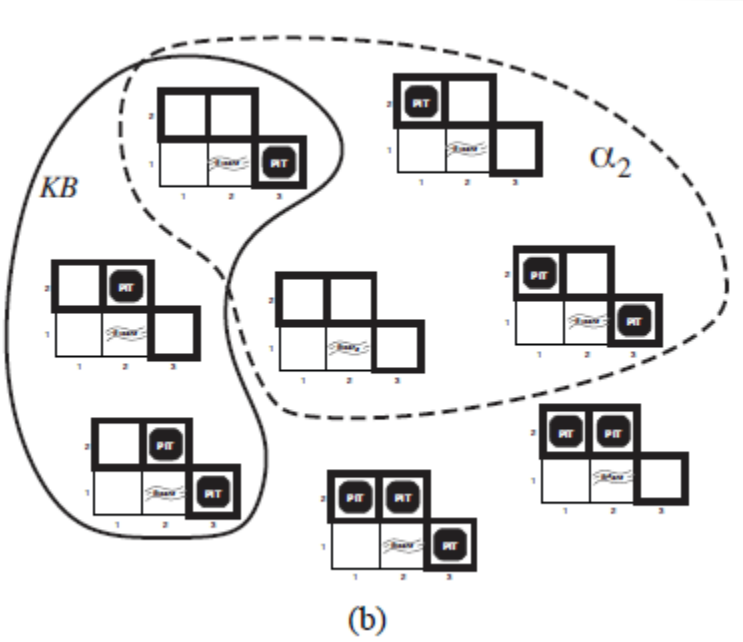


Consider:

alpha1 = There is no pit in [1,2]

Knowledge in the Wumpus world

- Possible **models** for the presence of pits in [1,2], [2,2], [3,1]
- Each square may or may not contain pit
  - 2^3 == 8 possibilities
- KB is what is **known**
  - Cannot have pit in [2,1]
  - Must have pit in [2,2] or [3,1]



Consider:

alpha2 = There is no pit in [3,1]

So: KB entails alpha1

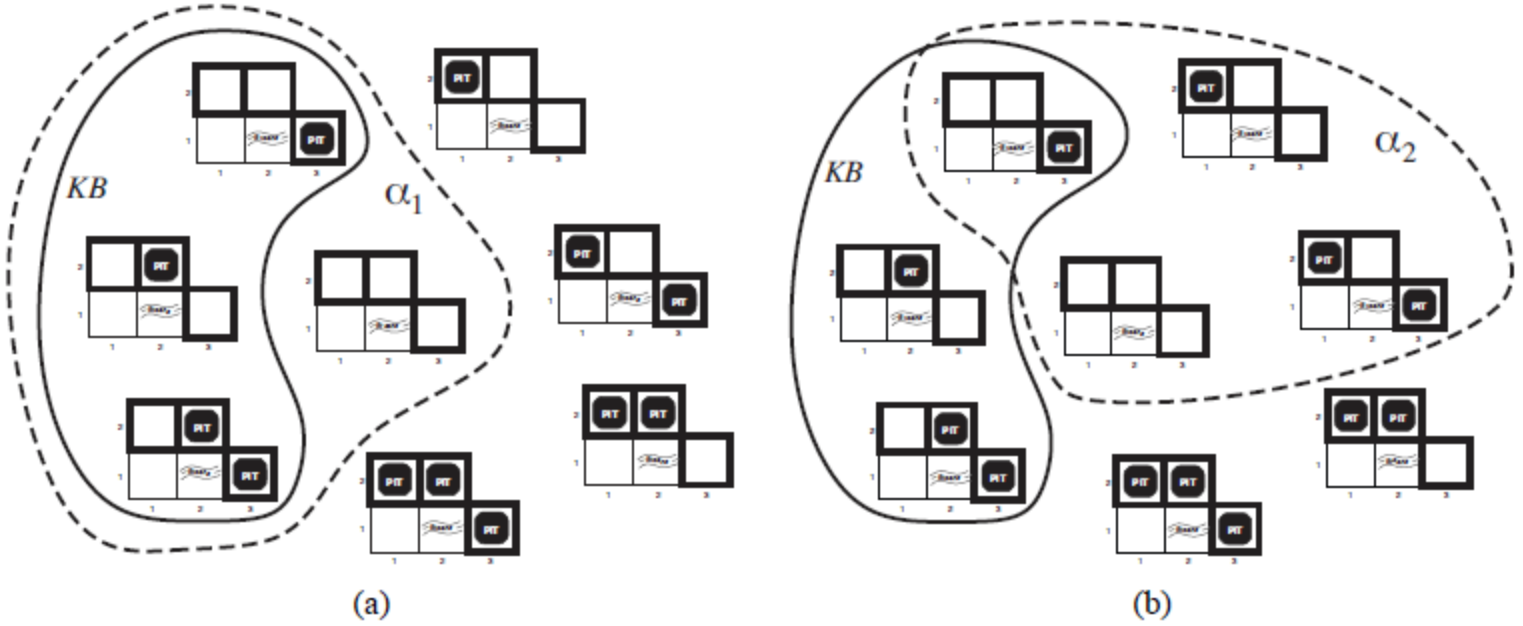


Figure 7.5 **FILES: figures/wumpus-entailment.eps (Tue Nov 3 16:24:09 2009) figures/wumpus-nonentailment.eps (Tue Nov 3 16:24:10 2009).** Possible models for the presence of pits in squares [1,2], [2,2], and [3,1]. The KB corresponding to the observations of nothing in [1,1] and a breeze in [2,1] is shown by the solid line. (a) Dotted line shows models of α_1 (no pit in [1,2]). (b) Dotted line shows models of α_2 (no pit in [2,2]).

KB does not entail alpha2

Logical inference

- Entailment can be used to derive conclusions
  - This is Logical Inference
- **Model Checking** checks that M(alpha) entails M(beta)
  - Previous figure model checked that M(KB) entails M(alpha1)

- If an inference algorithm *i* can derive alpha from KB then
  - *i derives alpha from KB*
- An inference algorithm that derives only entailed sentences is
  - **SOUND**
  - **Model Checking** is a sound procedure
  - **Model Checking** is a sound inference algorithm
- An inference algorithm is **complete** if it can derive any sentence that is entailed

We are back to searching to check that KB entails alpha

Sound Inference

- Guarantees that a conclusion arrived through sound inference is true in any world in which the premise (KB) is true
- Sound inference operates on a representation of the real world
- If good representation then this means
  - Conclusions correspond to aspects of the real world

Some philosophy

- Grounding
  - Connects the logical reasoning on a representation of the real world with the real world

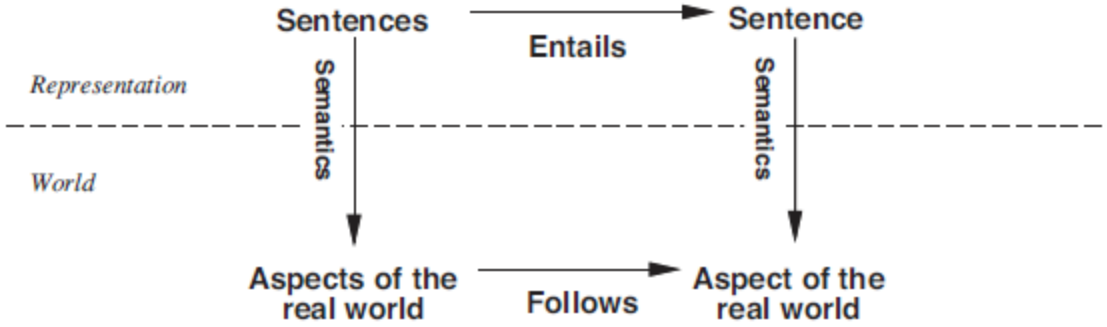


Figure 7.6 **FILES: figures/follows+entails.eps (Tue Nov 3 16:22:52 2009).** Sentences are physical configurations of the agent, and reasoning is a process of constructing new physical configurations from old ones. Logical reasoning should ensure that the new configurations represent aspects of the world that actually follow from the aspects that the old configurations represent.

Sad about Marcus

- Marcus is a man
- Marcus is a pompein
- Marcus was born in 40AD
- All men are mortal
- All pompeins died when the volcano erupted in 79AD
- No mortal lives longer than 150 years
- It is now 2013AD

- How do we represent this as sentences that a sound inference procedure can work with?

Sadder about Marcus

- Man(marcus)
- Pompein(marcus)
- Born(marcus, 40)
- A(x) [man(x) → mortal(x)]
- Erupted(Volcano, 79)
- A(x) [Pompein(x) → died(x, 79)]
- A(x) A(t1) A(t2) [mortal(x) & born(x, t1) & gt(t2 – t1, 150) → dead(x, t2)]
- Now = 2013

Propositional Calculus

- Cannot represent all facts about the Marcus problem
- What can it represent? To find out let us define it
- Syntax

Propositional logic is the simplest logic—illustrates basic ideas

The proposition symbols P_1, P_2 etc are sentences

If S is a sentence, $\neg S$ is a sentence (negation)

If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction)

If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (disjunction)

If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)

If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Semantics

Each model specifies true/false for each proposition symbol

E.g. $P_{1,2}$ $P_{2,2}$ $P_{3,1}$
$true$ $true$ $false$

(With these symbols, 8 possible models, can be enumerated automatically.)

Rules for evaluating truth with respect to a model m:

$\neg S$	is true iff	S	is false			
$S_1 \wedge S_2$	is true iff	S_1	is true **and**	S_2	is true	
$S_1 \vee S_2$	is true iff	S_1	is true **or**	S_2	is true	
$S_1 \Rightarrow S_2$	is true iff	S_1	is false **or**	S_2	is true	
i.e.,	is false iff	S_1	is true **and**	S_2	is false	
$S_1 \Leftrightarrow S_2$	is true iff	$S_1 \Rightarrow S_2$	is true **and**	$S_2 \Rightarrow S_1$	is true	

Simple recursive process evaluates an arbitrary sentence, e.g.,
$\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1}) = true \wedge (false \vee true) = true \wedge true = true$

Back to Wumpuses

Propositional logic is the simplest logic—illustrates basic ideas

The proposition symbols P_1, P_2 etc are sentences

If S is a sentence, $\neg S$ is a sentence (negation)

If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction)

If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (disjunction)

If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)

If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

W[1,3] ←→ ! W[2,2]. Wumpus is in [1,3] is true if and only if Wumpus is in [2,2] is false

Truth tables

E1	E2	E1&E2	E1V E2	E1➔E2	!E1 V E2	E1 ←→E2
T	T	T	T	T	T	T
T	F	F	T	F	F	F
F	T	F	T	T	T	F
F	F	F	F	T	T	T

Wumpus representation

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)

Let $P_{i,j}$ be true if there is a pit in $[i, j]$.
Let $B_{i,j}$ be true if there is a breeze in $[i, j]$.

$$\neg P_{1,1}$$
$$\neg B_{1,1}$$
$$B_{2,1}$$

Wumpus Representation (2)

"Pits cause breezes in adjacent squares"

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

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)

!P[1,2] ?

- 5 sentences in KB

Let $P_{i,j}$ be true if there is a pit in $[i, j]$.
Let $B_{i,j}$ be true if there is a breeze in $[i, j]$.

$$\neg P_{1,1}$$
$$\neg B_{1,1}$$
$$B_{2,1}$$

"Pits cause breezes in adjacent squares"

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

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)

Need to model check

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	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*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
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*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

M(KB) entails M(!P[1,2])

Does M(KB) entail M(P[2,2]) ?

Entailment algorithm O(2^n)

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

  symbols $\leftarrow$ a list of the proposition symbols in KB and α
  return TT-CHECK-ALL($KB, \alpha, symbols, \{\}$)

---

function TT-CHECK-ALL($KB, \alpha, symbols, model$) **returns** *true* or *false*
  if EMPTY?($symbols$) **then**
    if PL-TRUE?($KB, model$) **then return** PL-TRUE?($\alpha, model$)
    else return *true* // *when KB is false, always return true*
  else do
    $P \leftarrow$ FIRST($symbols$)
    $rest \leftarrow$ REST($symbols$)
    return (TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P = true\}$)
      and
      TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P = false\}$))

Figure 7.8 A truth-table enumeration algorithm for deciding propositional entailment. (TT stands for truth table.) PL-TRUE? returns *true* if a sentence holds within a model. The variable *model* represents a partial model—an assignment to some of the symbols. The keyword "**and**" is used here as a logical operation on its two arguments, returning *true* or *false*.