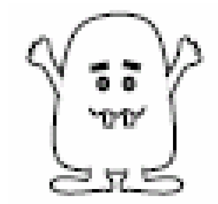
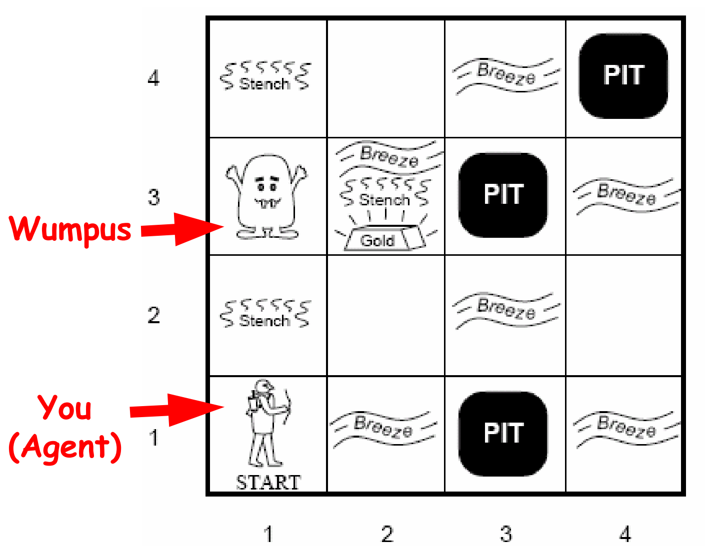
CSE 473

Chapter 7

Logic for the Wumpus: Propositional Logic



A Typical Wumpus World



2

Wumpus World PEAS Description

Performance measure
- gold +1000, death -1000
- -1 per step, -10 for using the arrow

Environment
- Squares adjacent to wumpus are smelly
- Squares adjacent to pit are breezy
- Glitter iff gold is in the same square
- Shooting kills wumpus if you are facing it
- Shooting uses up the only arrow
- Grabbing picks up gold if in same square
- Releasing drops the gold in same square

Sensors Breeze, Glitter, Smell

Actuators Left turn, Right turn,
Forward, Grab, Release, Shoot

3

How do we represent rules of the world and percepts encountered so far?

Why not use logic?

What is a logic?

- *A formal language*
  - Syntax – what expressions are legal (well-formed)
  - Semantics – what legal expressions mean
    - In logic the truth of each sentence evaluated with respect to each possible world
- E.g the language of arithmetic
  - X+2 >= y is a sentence, x2+y is not a sentence
  - X+2 >= y is true in a world where x=7 and y =1
  - X+2 >= y is false in a world where x=0 and y =6

5

How do we draw conclusions and deduce new facts about the world?

Entailment

Knowledge Base = KB
Sentence α

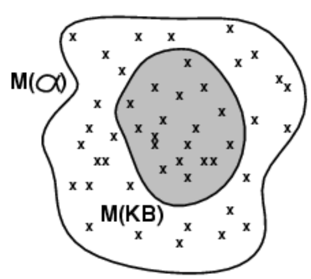
KB $\models \alpha$ (KB "entails" sentence α)
if and only if α is true in all worlds (models) where KB is true.

E.g. x+y=4 entails 4=x+y
(because 4=x+y is true for all values of x, y for which x+y=4 is true)

7

Models and Entailment

- *m* is a model of a sentence α if α is true in *m*
  e.g. α is "4=x+y" and *m* = {*x=2, y=2*}
- *M(α)* is the set of all models of α
- Then KB $\models \alpha$ iff $M(KB) \subseteq M(\alpha)$



8

Soundness and Completeness

If an inference algorithm only derives entailed sentences, it is called *sound* (or *truth preserving*).

Otherwise it just makes things up.

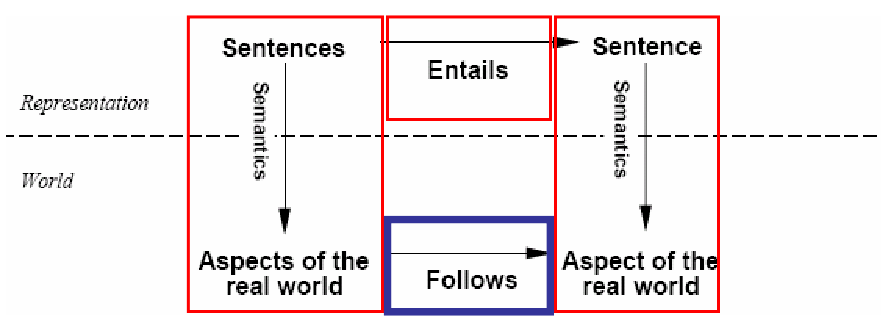
Algorithm i is sound if whenever $KB \mid\!-_i \alpha$ (i.e. α is derived by i from KB) it is also true that $KB \models \alpha$

- Completeness: An algorithm is complete if it can derive any sentence that is entailed.

i is complete if whenever $KB \models \alpha$ it is also true that $KB \mid\!-_i \alpha$

9

Relating to the Real World



If KB is true in the real world, then any sentence α derived from KB by a sound inference procedure is also true in the real world

10

Propositional Logic: Syntax

- Propositional logic is the simplest logic – illustrates basic ideas
- Atomic sentences = proposition symbols = A, B, $P_{1,2}$, $P_{2,2}$ etc. used to denote properties of the world

  Can be either True or False
- E.g. $P_{1,2}$ = "There's a pit in location [1,2]" is either true or false in the wumpus world

11

Propositional Logic: Syntax

- Complex sentences constructed from simpler ones recursively

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)

12

Propositional Logic: Semantics

A model specifies true/false for each proposition symbol

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

false true false

Rules for evaluating truth w.r.t. 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
$S_1 \Leftrightarrow S_2$ is true iff both $S_1 \Rightarrow S_2$ and $S_2 \Rightarrow S_1$ are true

13

Truth Tables for 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*

14

Propositional Logic: Semantics

Simple recursive process can be used to evaluate an arbitrary sentence

E.g., Model: $P_{1,2}$ $P_{2,2}$ $P_{3,1}$

false true false

$\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1}) = true \wedge (true \vee false)$

$= true \wedge true$

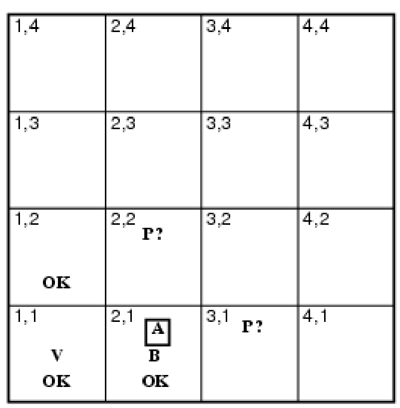
$= true$

Example: Wumpus World

Proposition Symbols and Semantics:

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].



Wumpus KB

Knowledge Base (KB) includes the following sentences:

- Statements currently known to be true:

  $\neg P_{1,1}$

  $\neg B_{1,1}$

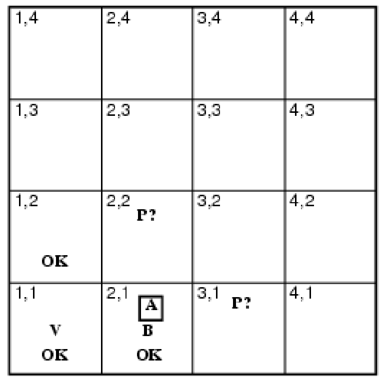
  $B_{2,1}$

- Properties of the world: E.g., "Pits cause breezes in adjacent squares"

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

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

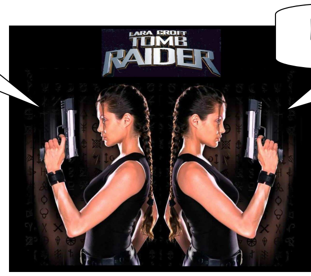
  (and so on for all squares)



17

Can a Wumpus Agent use this logical representation and KB to avoid pits and the wumpus, and find the gold?

Is there no pit in [1,2]?



Does KB $\models \neg P_{1,2}$?

Inference by Truth Table Enumeration

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	$\neg P_{1,2}$
false	false	false	false	false	false	false	false	true
false	false	false	false	false	false	true	false	true
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	true	false	false	false	false	false	false	true
false	true	false	false	false	false	true	<u>true</u>	<u>true</u>
false	true	false	false	false	true	false	<u>true</u>	<u>true</u>
false	true	false	false	false	true	true	<u>true</u>	<u>true</u>
false	true	false	false	true	false	false	false	true
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	true	true	true	true	true	true	false	false

$\neg P_{1,2}$ true in all models in which KB is true
Therefore, KB $\models \neg P_{1,2}$

19

Inference by TT Enumeration

Depth-first enumeration of all models is sound & complete

```
function TT-ENTAILS?(KB, α) returns true or false
   symbols ← a list of the proposition symbols in KB and α
   return TT-CHECK-ALL(KB, α, symbols, [])

function TT-CHECK-ALL(KB, α, symbols, model) returns true or false
   if EMPTY?(symbols) then
      if PL-TRUE?(KB, model) then return PL-TRUE?(α, model)
      else return true
   else do
      P ← FIRST(symbols); rest ← REST(symbols)
      return TT-CHECK-ALL(KB, α, rest, EXTEND(P, true, model)) and
             TT-CHECK-ALL(KB, α, rest, EXTEND(P, false, model))
```

For n symbols, time complexity = $O(2^n)$, space = $O(n)$

20

Concepts for Other Techniques: Logical Equivalence

- Two sentences are logically equivalent iff they are true in the same models: $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

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

21

Concepts for Other Techniques: Validity and Satisfiability

A sentence is *valid* if it is true in *all* models (a tautology)

e.g., *True*, $A \lor \neg A$, $A \Rightarrow A$, $(A \land (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the Deduction Theorem:

$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is *satisfiable* if it is true in *some* model

e.g., $A \lor B$, C

A sentence is *unsatisfiable* if it is true in no models

e.g., $A \land \neg A$

Satisfiability is connected to inference via the following:

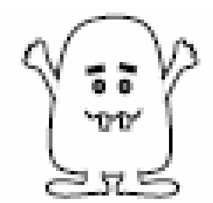
$KB \models \alpha$ if and only if $(KB \land \neg\alpha)$ is unsatisfiable

(proof by contradiction)

22

Next Time

- Inference Techniques
  - Resolution
  - Forward and backward chaining
  - DPLL



23