7

Logical Agents

Claudia Chirita

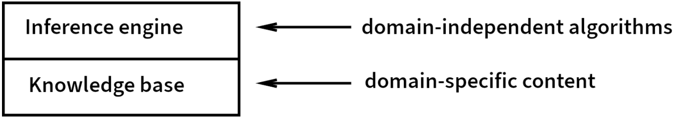
School of Informatics, University of Edinburgh

Based on slides by: Jacques Fleuriot, Michael Rovatsos, Michael Herrmann, Vaishak Belle

7.a

Knowledge-based agents

KNOWLEDGE BASES



Knowledge base (KB) = set of sentences in a *formal* language

Declarative approach to building an agent:
    TELL it what it needs to know

Then it can ASK itself what to do
    answers should follow from the KB

Agents can be viewed at the **knowledge level**
    i.e. *what they know*, regardless of how implemented

or at the **implementation level**
    i.e. data structures in KB and algorithms that manipulate them

A SIMPLE KNOWLEDGE-BASED AGENT

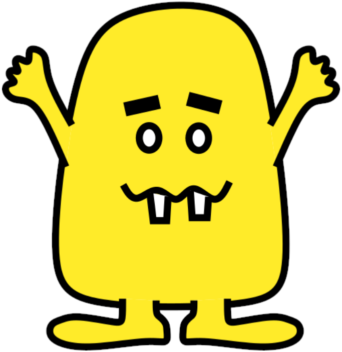
```
function KB-AGENT(percept) returns an action
   persistent: KB, a knowledge base
               t, a counter, initially 0, indicating time

   TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
   action ← ASK(KB, MAKE-ACTION-QUERY(t))
   TELL(KB, MAKE-ACTION-SENTENCE(action, t))
   t ← t + 1
   return action
```

The agent must be able to:

- represent states, actions, etc.
- incorporate new percepts
- update internal representations of the world
- deduce hidden properties of the world
- deduce appropriate actions

WUMPUS WORLD



WUMPUS WORLD

PERFORMANCE MEASURE

gold +1000, death -1000

-1 per step, -10 for using arrow

ENVIRONMENT

Squares adjacent to Wumpus are smelly

Squares adjacent to pits are breezy

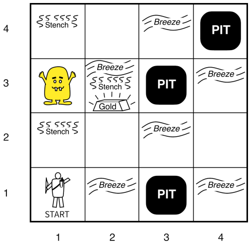
Glitter iff gold is in the same square

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



ACTUATORS

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

SENSORS

Breeze, Glitter, Smell, Scream, Bump

WUMPUS WORLD · CHARACTERIZATION

❓ observable

deterministic

episodic

static

discrete

single-agent

WUMPUS WORLD · CHARACTERIZATION

observable	No	only local perception
deterministic		
episodic		
static		
discrete		
single-agent		

WUMPUS WORLD · CHARACTERIZATION

observable	No	only local perception
deterministic	Yes	outcomes exactly specified
episodic		
static		
discrete		
single-agent		

WUMPUS WORLD · CHARACTERIZATION

observable	No	only local perception
deterministic	Yes	outcomes exactly specified
episodic	No	sequential at the level of actions
static		
discrete		
single-agent		

WUMPUS WORLD · CHARACTERIZATION

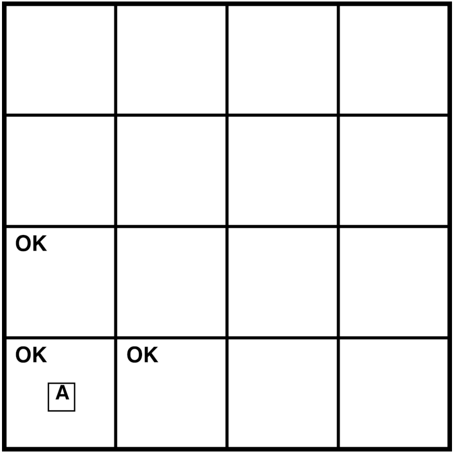
observable	No	only local perception
deterministic	Yes	outcomes exactly specified
episodic	No	sequential at the level of actions
static	Yes	Wumpus and Pits do not move
discrete		
single-agent		

WUMPUS WORLD · CHARACTERIZATION

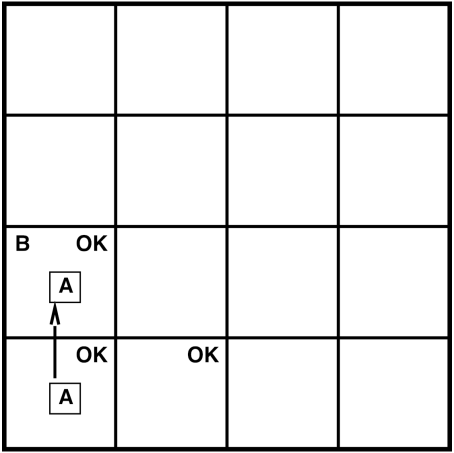
observable	No	only local perception
deterministic	Yes	outcomes exactly specified
episodic	No	sequential at the level of actions
static	Yes	Wumpus and Pits do not move
discrete	Yes	
single-agent		

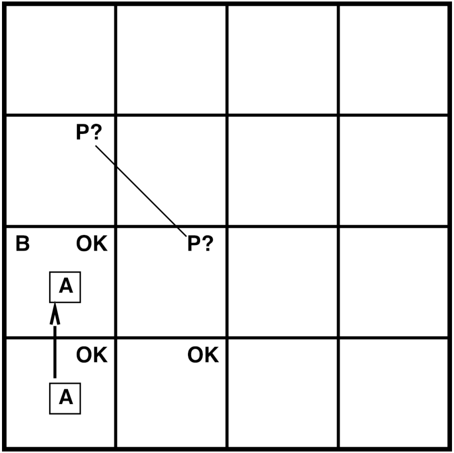
WUMPUS WORLD · CHARACTERIZATION

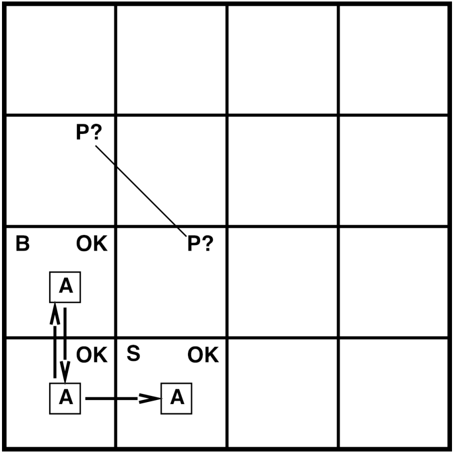
observable	No	only local perception
deterministic	Yes	outcomes exactly specified
episodic	No	sequential at the level of actions
static	Yes	Wumpus and Pits do not move
discrete	Yes	
single-agent	Yes	Wumpus is essentially a natural feature

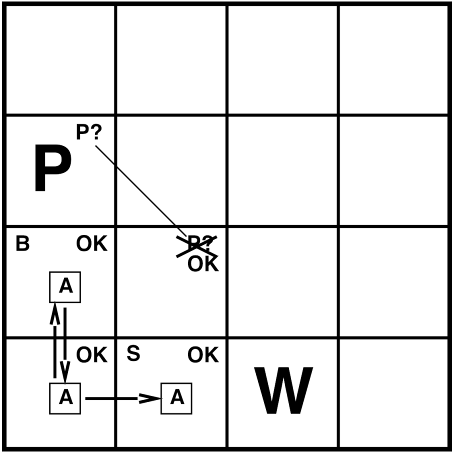

OK
OK
A
OK

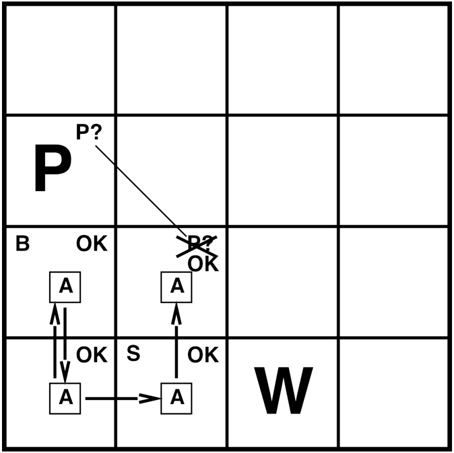
WUMPUS WORLD · EXPLORATION



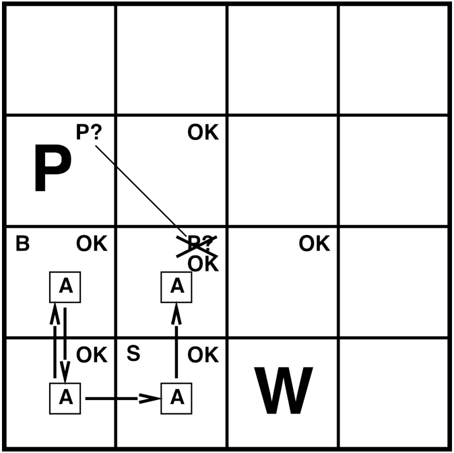

P?
B
OK
P?
A
OK
OK
A


P?
B
OK
P?
A
OK
S
OK
A
A

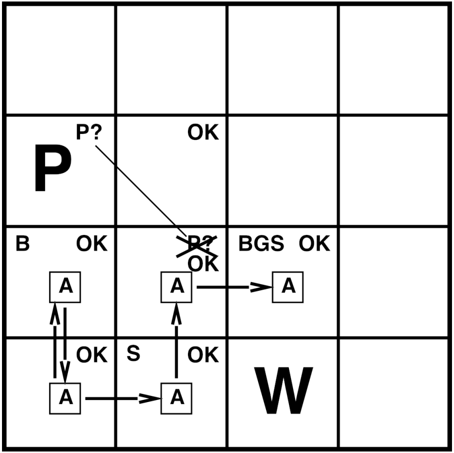

P?
P
B
OK
~~P?~~
OK
A
OK
S
OK
A
A
W


P?
P
B
OK
~~P?~~
OK
A
A
OK
S
OK
A
A
W

WUMPUS WORLD · EXPLORATION



WUMPUS WORLD · EXPLORATION



TAKE-AWAY MESSAGE

Logical agents apply inference to a knowledge base
to derive new information and make decisions.

7.b

What is a logic?

LOGICS

LOGICS formal languages for representing information
such that conclusions can be drawn

SYNTAX the symbols and sentences of the language

SEMANTICS the *meaning* of symbols and sentences
from which we can infer if a sentence holds in a world

EXAMPLE **THE LANGUAGE OF ARITHMETIC**

$x + 2 \geqslant y$ is a sentence

$x2 + y >$ is not a sentence

$x + 2 \geqslant y$ is true iff the number $x + 2$ is no less than the number y

$x + 2 \geqslant y$ is true in a world where $x = 7, y = 1$

$x + 2 \geqslant y$ is false in a world where $x = 0\,, y = 6$

ENTAILMENT

INFORMALLY one thing follows from another

FORMALLY relationship between sets of sentences (*premises*) and sentences (*conclusion*)

KB $\models \alpha$

Knowledge base KB entails sentence α if and only if α is true in all worlds where KB is true.

EXAMPLE

- $x + y = 4$ entails $4 = x + y$.
- The KB containing “Kasparov won” and “Deep Mind won” entails “Kasparov won or Deep Mind won”.

  Considering only worlds where “Kasparov” plays “Deep Mind” (no draws) it entails “either Kasparov won or Deep Mind won”.

MODELS

Logicians typically think in terms of **models**, which are formally structured worlds with respect to which truth can be evaluated.

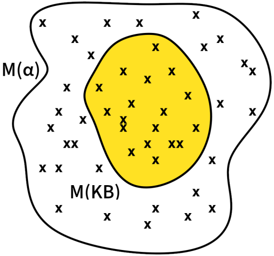
$M \models \alpha$

A model M satisfies a sentence α if α is true in M.
We say that M **is a model** of α.

$\mathrm{M}(\alpha)$ is the set of all models of α.

$\mathrm{KB} \models \alpha$ iff $\mathrm{M}(\mathrm{KB}) \subseteq \mathrm{M}(\alpha)$

The stronger an assertion, the fewer models it has.

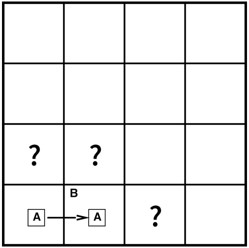


WUMPUS WORLD · ENTAILMENT

Situation after detecting nothing in [1,1], moving right, breeze in [2,1]

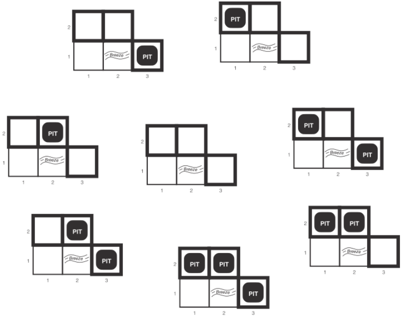
Consider possible models for ?s assuming only pits

3 Boolean choices $\longrightarrow$ 8 possible models

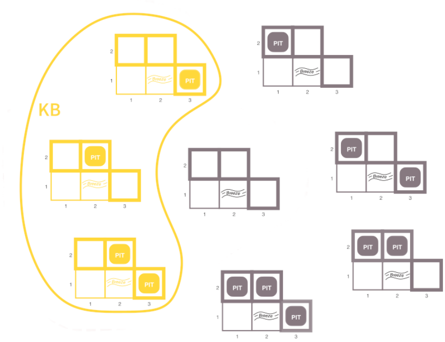


What are these 8 models?

WUMPUS MODELS

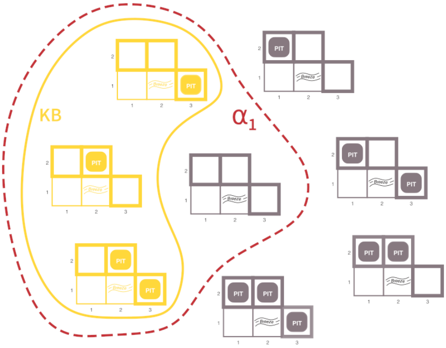


WUMPUS MODELS



KB $=$ wumpus-world rules $+$ observations

WUMPUS MODELS

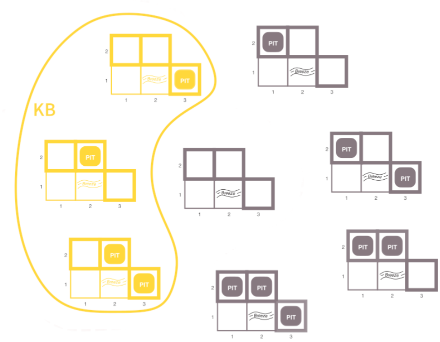


KB $=$ wumpus-world rules $+$ observations

$\alpha_1 =$ "[1,2] has no pit", KB $\models \alpha_1$ proved by **model checking**

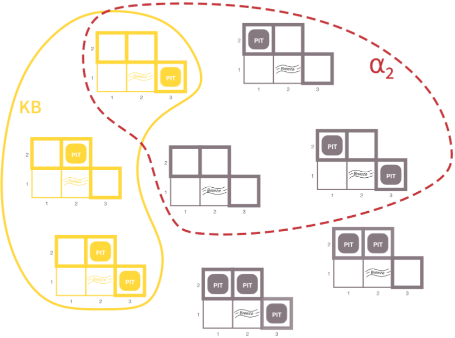
in every model in which KB is true, α_1 is also true

WUMPUS MODELS



KB $=$ wumpus-world rules $+$ observations

WUMPUS MODELS



KB $=$ wumpus-world rules $+$ observations

$\alpha_2 =$ "[2,2] has no pit", KB $\not\models \alpha_2$

in some models in which KB is true, α_2 is also true

INFERENCE

KB $\vdash_i \alpha$ sentence α can be derived from KB by inference procedure i

Consequences of KB are a haystack; α is a needle.
Entailment = needle in haystack; inference = finding it.

SOUNDNESS

i is sound if whenever KB $\vdash_i \alpha$, it is also true that KB $\models \alpha$

COMPLETENESS

i is complete if whenever KB $\models \alpha$, it is also true that KB $\vdash_i \alpha$

❓ What happens when an inference procedure is not sound?

v=zrzMhU_4m-g

INFERENCE

KB $\vdash_i \alpha$ sentence α can be derived from KB by inference procedure i

Consequences of KB are a haystack; α is a needle.
Entailment = needle in haystack; inference = finding it.

SOUNDNESS

i is sound if whenever KB $\vdash_i \alpha$, it is also true that KB $\models \alpha$

COMPLETENESS

i is complete if whenever KB $\models \alpha$, it is also true that KB $\vdash_i \alpha$

SNEAK PEAK

We will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure.

That is, the procedure will answer any question whose answer follows from what is known by the KB.

TAKE-AWAY MESSAGE

LOGIC = SYNTAX + SEMANTICS

entailment	necessary truth of one sentence given another
inference	deriving sentences from other sentences
soundness	derivations produce only entailed sentences
completeness	derivations can produce all entailed sentences

7.c

Propositional logic

Propositional logic is the simplest logic – illustrates basic ideas.

The proposition symbols P_1, P_2, etc. are sentences.

If S is a sentence, then $\neg S$ 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**]

Each model specifies true/false for each proposition symbol.

EXAMPLE

$P_{1,2}$	$P_{2,2}$	$P_{3,1}$
false	*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:

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

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*

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"
"A square is breezy if and only if there is an adjacent pit"

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

Recall: $\alpha_1 =$ "[1,2] has no pit"

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	α_1
false	*false*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
false	*false*	*false*	*false*	*false*	*false*	*true*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*false*

Enumerate rows (different assignments to symbols):
if KB is true in row, check that α is too

INFERENCE BY ENUMERATION

Depth-first enumeration of all models is sound and complete.

```
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 ← 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 // when KB is false, always return true
   else do
      P ← FIRST(symbols)
      rest ← REST(symbols)
      return (TT-CHECK-ALL(KB, α, rest, model ∪ {P = true})
              and
              TT-CHECK-ALL(KB, α, rest, model ∪ {P = false }))
```

PL-TRUE?	returns true if a sentence holds within a model
EXTEND(P, *val*, *model*)	returns a new partial model in which P has value *val*

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

LOGICAL EQUIVALENCE

Two sentences are **logically equivalent** iff true in the same models

$\alpha \equiv \beta$ if and only if $\alpha \models \beta$ and $\beta \models \alpha$

$(\alpha \land \beta) \equiv (\beta \land \alpha)$	commutativity of $\land$
$(\alpha \lor \beta) \equiv (\beta \lor \alpha)$	commutativity of $\lor$
$((\alpha \land \beta) \land \gamma) \equiv (\alpha \land (\beta \land \gamma))$	associativity of $\land$
$((\alpha \lor \beta) \lor \gamma) \equiv (\alpha \lor (\beta \lor \gamma))$	associativity of $\lor$
$\neg(\neg\alpha) \equiv \alpha$	double-negation elimination
$(\alpha \longrightarrow \beta) \equiv (\neg\beta \longrightarrow \neg\alpha)$	contraposition
$(\alpha \longrightarrow \beta) \equiv (\neg\alpha \lor \beta)$	implication elimination
$(\alpha \leftrightarrow \beta) \equiv ((\alpha \longrightarrow \beta) \land (\beta \longrightarrow \alpha))$	biconditional elimination
$\neg(\alpha \land \beta) \equiv (\neg\alpha \lor \neg\beta)$	de Morgan
$\neg(\alpha \lor \beta) \equiv (\neg\alpha \land \neg\beta)$	de Morgan
$(\alpha \land (\beta \lor \gamma)) \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma))$	distributivity of $\land$ over $\lor$
$(\alpha \lor (\beta \land \gamma)) \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma))$	distributivity of $\lor$ over $\land$

VALIDITY AND SATISFIABILITY

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

true, $A \lor \neg A$, $A \longrightarrow A$, $(A \land (A \longrightarrow B)) \longrightarrow B$

Validity is connected to inference via the **Deduction Theorem**.

KB $\models \alpha$ if and only if $(\text{KB} \longrightarrow \alpha)$ is valid

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

$A \lor B$, C

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

$A \land \neg A$

Satisfiability is connected to inference via the following:

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

prove α by *reductio ad absurdum*

PROOF METHODS

Proof methods divide into (roughly) two kinds:

APPLICATION OF INFERENCE RULES

Legitimate (sound) generation of new sentences from old

Proof = a sequence of inference rule applications

can use inference rules as operators in a standard search algorithm

Typically require translation of sentences into a normal form.

example: resolution

MODEL CHECKING

Truth table enumeration	(always exponential in n)
Improved backtracking	(Davis–Putnam–Logemann–Loveland)
Heuristic search in model space	(sound but incomplete)

ALL WE NEED?

Wumpus world requires the ability to represent partial and negated information, reason by cases, etc.

Does propositional logic provide enough expressive power for statements about the real world?