CS 4700: Foundations of Artificial Intelligence

Bart Selman
selman@cs.cornell.edu

Module: Knowledge, Reasoning, and Planning
Part 2

Logical Agents
R&N: Chapter 7

Illustrative example: Wumpus World

Performance measure (Somewhat whimsical!)

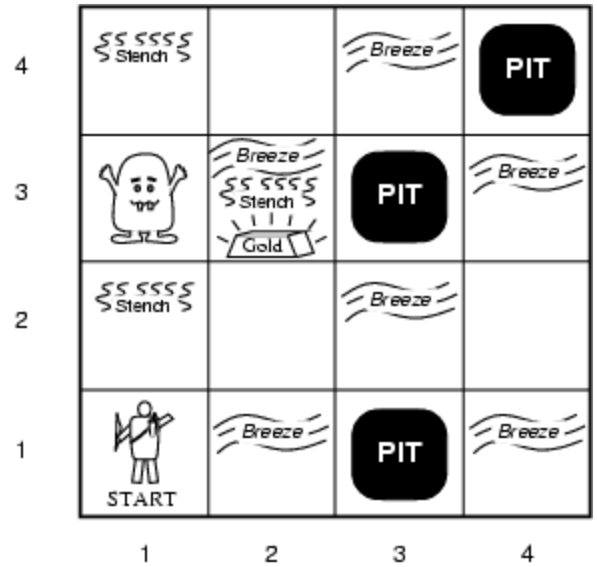
- gold +1000,
- death -1000

(falling into a pit or being eaten by the wumpus)

- -1 per step, -10 for using the arrow

Environment

- Rooms / squares connected by doors.
- 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
- Randomly generated at start of game. Wumpus only senses current room.



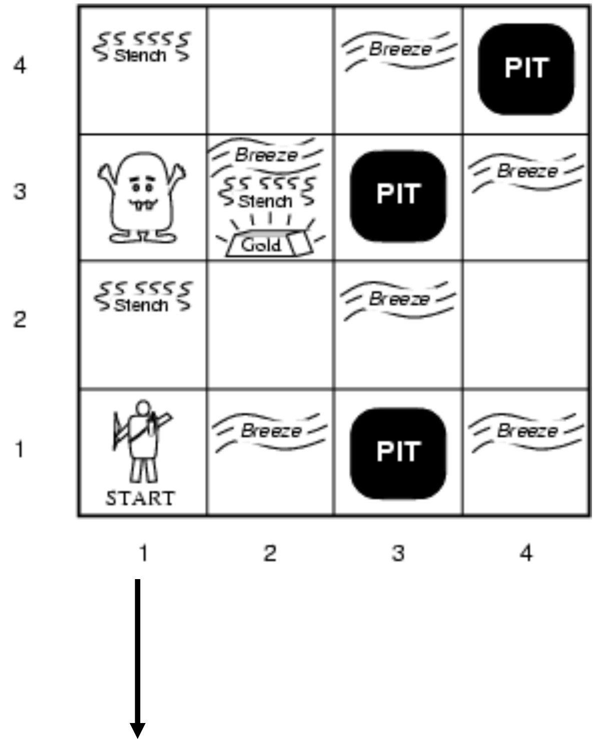
Sensors: Stench, Breeze, Glitter, Bump, Scream [perceptual inputs]

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

Wumpus world characterization

Fully Observable	No – only local perception
Deterministic	Yes – outcomes exactly specified
Static	Yes – Wumpus and Pits do not move
Discrete	Yes
Single-agent?	Yes – Wumpus is essentially a “natural feature.”

Exploring a wumpus world



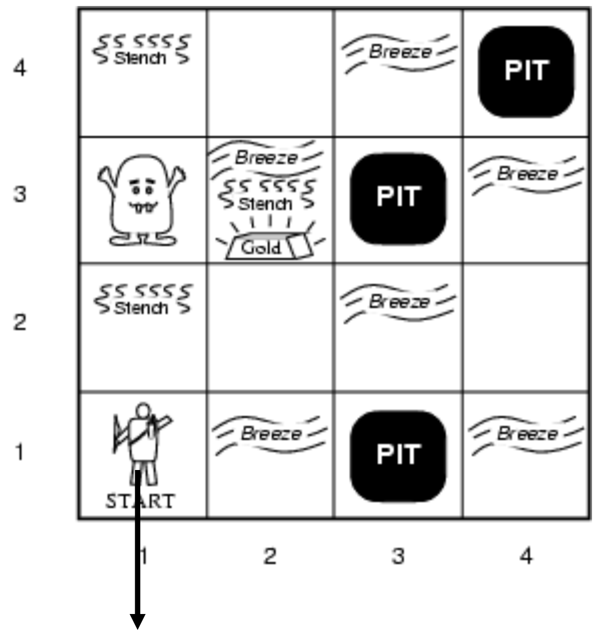
The knowledge base of the agent consists of the rules of the Wumpus world plus the percept "nothing" in [1,1]

None, none, none, none, none

Stench, Breeze, Glitter, Bump, Scream

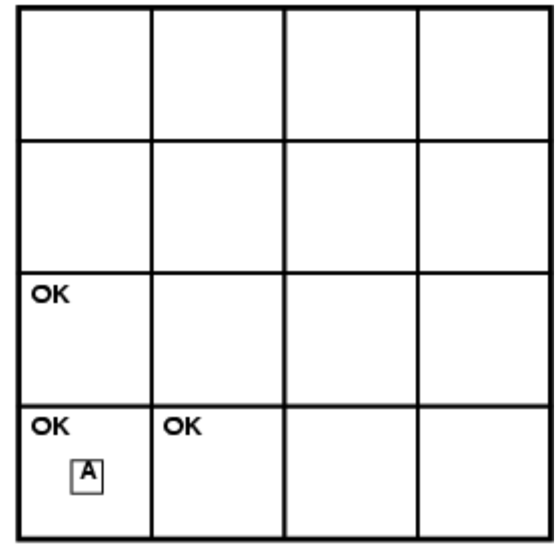
Boolean percept feature values:
<0, 0, 0, 0, 0>

World "known" to agent at time = 0.



None, none, none, none, none

Stench, Breeze, Glitter, Bump, Scream



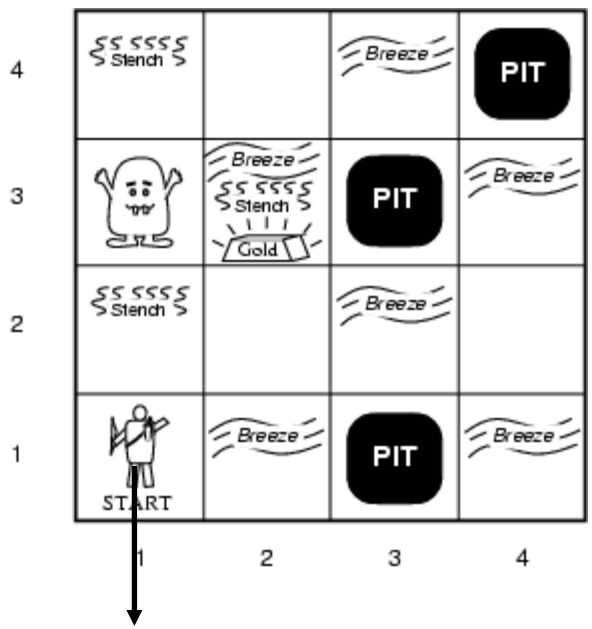
T=0 The KB of the agent consists of the rules of the Wumpus world plus the percept "nothing" in [1,1].

By inference, the agent's knowledge base also has the information that [1,2] and [2,1] are okay.

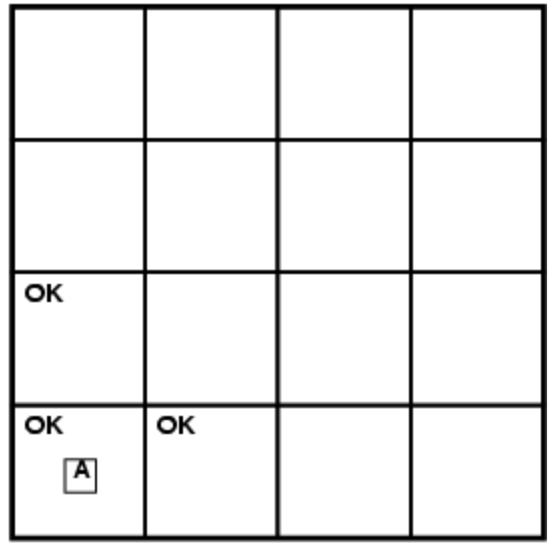
Added as propositions.

Further exploration

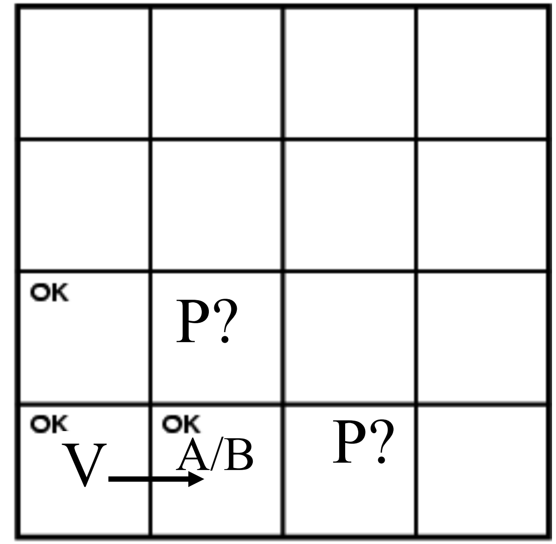
T = 0



None, none, none, none, none



T = 1



None, breeze, none, none, none

A – agent

V – visited

B - breeze

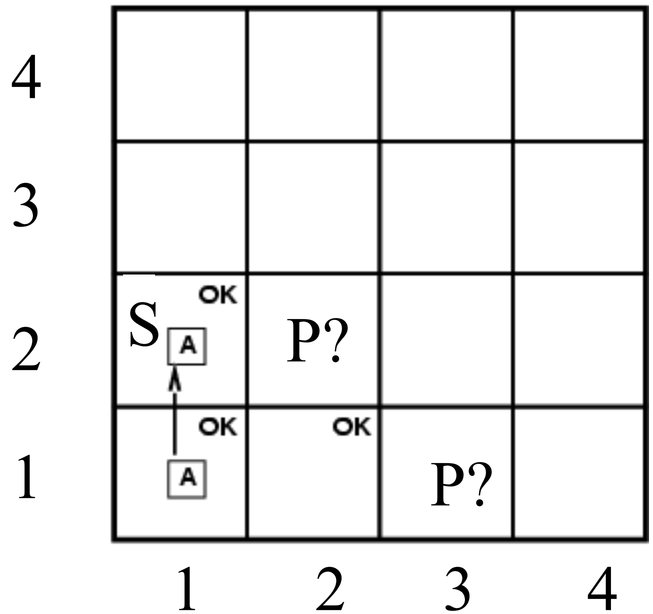
Stench, Breeze, Glitter, Bump, Scream

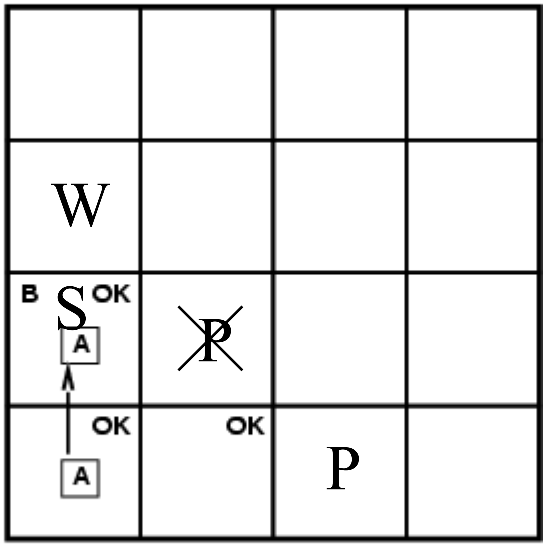
Where next?

@ T = 1 What follows?

Pit(2,2) or Pit(3,1)

T=3





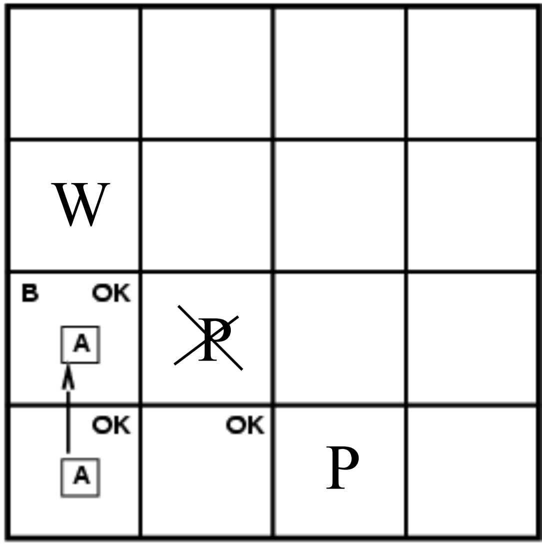
Stench, none, none, none, none

Stench, Breeze, Glitter, Bump, Scream

Where is Wumpus?

Wumpus cannot be in (1,1) or in (2,2) (Why?)➔ Wumpus in (1,3)
Not breeze in (1,2) ➔ no pit in (2,2); but we know there is pit in (2,2) or (3,1) ➔ pit in (3,1)

We reasoned about the possible states the Wumpus world can be in, given our percepts and our knowledge of the rules of the Wumpus world.
I.e., the content of KB at T=3.



What follows is what holds true in all those worlds that satisfy what is known at that time T=3 about the particular Wumpus world we are in.

Example property: P_in_(3,1)

Models(KB) $\subseteq$ Models(P_in_(3,1))

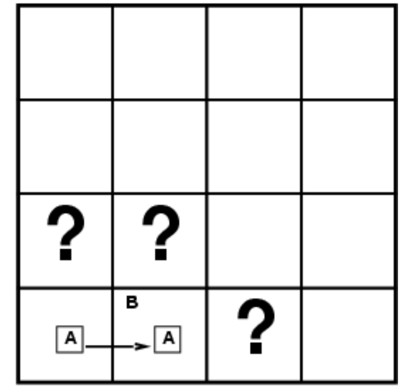
Essence of logical reasoning:

Given *all we know*, Pit_in_(3,1) holds.
("The world cannot be different.")

Formally: Entailment

Knowledge Base (KB) in the Wumpus World → Rules of the wumpus world + new percepts

Situation after detecting nothing in [1,1], moving right, breeze in [2,1]. I.e. T=1.



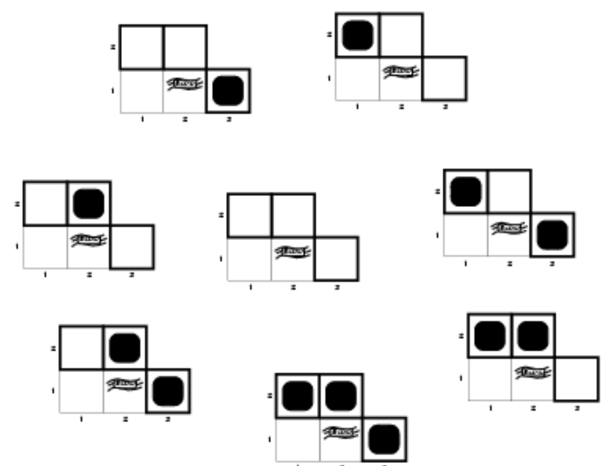
T = 1

Consider possible models for *KB* with respect to the cells (1,2), (2,2) and (3,1), with respect to the existence or non existence of pits

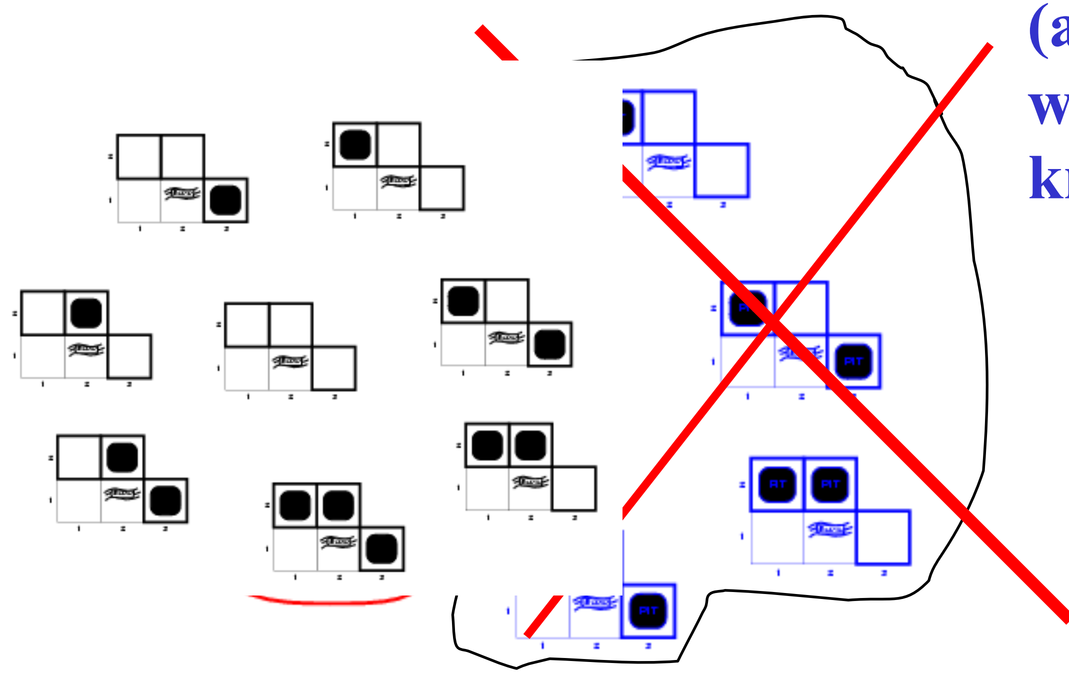
3 Boolean choices ⇒

8 possible interpretations

(enumerate all the models or "possible worlds" wrt Pit location)

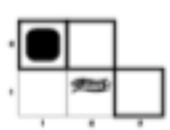


Is KB consistent with all 8 possible worlds?



Worlds that violate KB (are inconsistent with what we know)

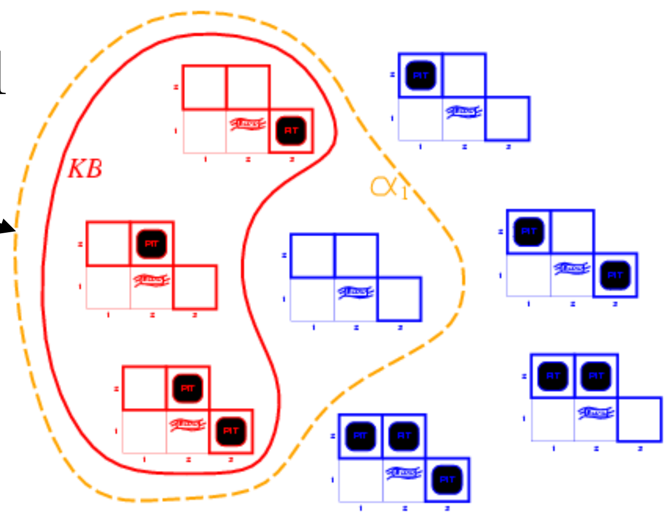
***KB* = Wumpus-world rules + observations (T=1)**

Q: Why does world  violate KB?

Entailment in Wumpus World

So, KB defines all worlds that we hold possible.
Queries: we want to know the properties of those worlds.
That's how the semantics of logical entailment is defined.

Models of the KB and α1



Note: \alpha_1 holds in more models than KB. That's OK, but we don't care about those worlds.

***KB* = Wumpus-world rules + observations**

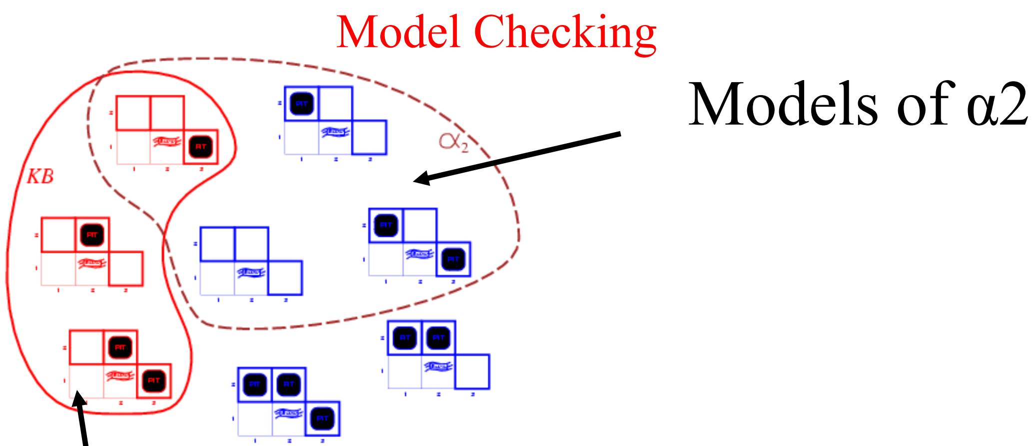
α_1 = "[1,2] has no pit", $KB \models \alpha_1$

- **In every model in which KB is true, α_1 is True (proved by "model checking")**

Wumpus models

KB = wumpus-world rules + observations

α2 = "[2,2] has no pit", this is only True in some of the models for which KB is True, therefore KB $\not\models$ α2



A model of KB where α2 does NOT hold!

Entailment via "Model Checking"

Inference by Model checking –

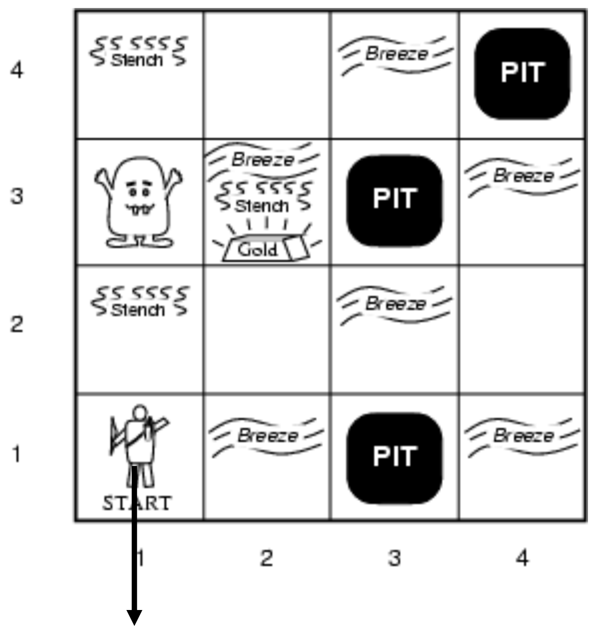
We enumerate all the KB models and check if α_1 and α_2 are True in all the models (which implies that we can only use it when we have a finite number of models).

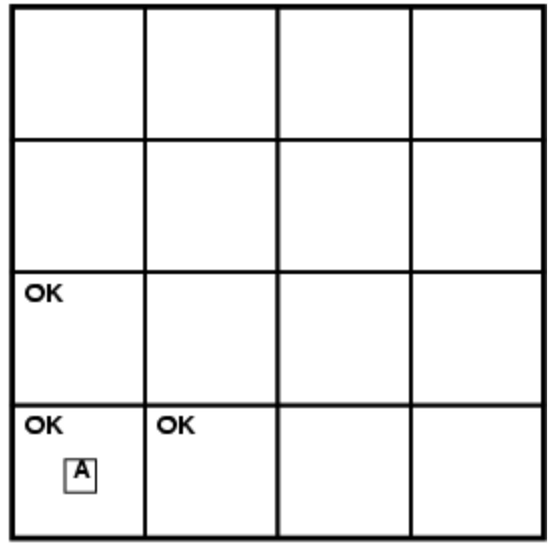
I.e. using semantics directly.

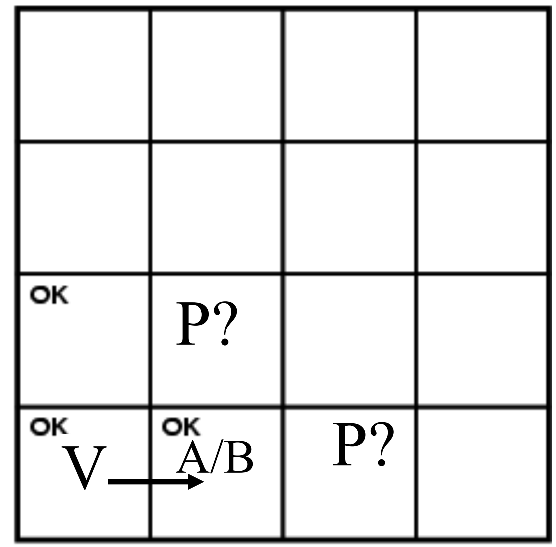
$$\text{Models(KB)} \subseteq \text{Models}(\alpha)$$

$$KB \models \alpha$$

Example redux: More formal







None, none, none, none, none

None, breeze, none, none, none

Stench, Breeze, Glitter, Bump, Scream

A – agent

V – visited

B - breeze

How do we actually encode background knowledge and percepts in formal language?

Wumpus World KB

Define propositions:

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].

Sentence 1 (R1): $\neg P_{1,1}$ [Given.]

Sentence 2 (R2): $\neg B_{1,1}$ [Observation T = 0.]

Sentence 3 (R3): $B_{2,1}$ [Observation T = 1.]

"Pits cause breezes in adjacent squares"

Sentence 4 (R4): $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

Sentence 5 (R5): $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

etc.

Notes: (1) one such statement about Breeze for each square.

(2) similar statements about Wumpus, and stench and Gold and glitter. (Need more propositional letters.)

What about Time? What about Actions?

Is Time represented?

No!

Can include time in propositions:

Explicit time $P_{i,j,t}$ $B_{i,j,t}$ $L_{i,j,t}$ etc.

Many more props: $O(TN^2)$ ($L_{i,j,t}$ for agent at (i,j) at time t)

Now, we can also model actions, use props: Move(i, j, k, l ,t)

E.g. Move(1, 1, 2, 1, 0)

What knowledge axiom(s) capture(s) the effect of an Agent move?

$$Move(i, j, k, l, t) \Rightarrow (\neg L(i, j, t+1) \wedge L(k, l, t+1))$$

Is this it?

What about i, j, k, and l?

What about Agent location at time t?

Improved: *Move implies a change in the world state;*
a change in the world state, implies a move occurred!

$$Move(i, j, k, l, t) \Leftrightarrow (L(i, j, t) \wedge \neg L(i, j, t+1) \wedge L(k, l, t+1))$$

For all tuples (i, j, k, l) that represent legitimate possible moves.

E.g. (1, 1, 2, 1) or (1, 1, 1, 2)

Still, some remaining subtleties when representing time and actions. What happens to propositions at time t+1 compared to at time t, that are *not* involved in any action?

E.g. P(1, 3, 3) is derived at some point.

What about P(1, 3, 4), True or False?

R&N suggests having P as an "atemporal var" since it cannot change over time. Nevertheless, we have many other vars that can change over time, called "fluents".

Values of propositions not involved in any action should not change! "The Frame Problem" / Frame Axioms R&N 7.7.1

Axiom schema:

F is a fluent (prop. that can change over time)

For example:

$$
\begin{aligned}
L_{1,1}^{t+1} = & (L_{1,1}^{t} \wedge (\neg Forward^{t} \vee Bump^{t+1})) \\
& \vee (L_{1,2}^{t} \wedge (South^{t} \wedge Forward^{t})) \\
& \vee (L_{2,1}^{t} \wedge (West^{t} \wedge Forward^{t}))
\end{aligned}
$$

i.e. L_1,1 was “as before” with [no movement action or bump into wall] or resulted from some action (movement into L_1,1).

Actions and inputs up to time 6

Note: includes turns!

Some example inferences
Section 7.7.1 R&N

$$\neg Stench^0 \land \neg Breeze^0 \land \neg Glitter^0 \land \neg Bump^0 \land \neg Scream^0 \quad ; \quad Forward^0$$
$$\neg Stench^1 \land Breeze^1 \land \neg Glitter^1 \land \neg Bump^1 \land \neg Scream^1 \quad ; \quad TurnRight^1$$
$$\neg Stench^2 \land Breeze^2 \land \neg Glitter^2 \land \neg Bump^2 \land \neg Scream^2 \quad ; \quad TurnRight^2$$
$$\neg Stench^3 \land Breeze^3 \land \neg Glitter^3 \land \neg Bump^3 \land \neg Scream^3 \quad ; \quad Forward^3$$
$$\neg Stench^4 \land \neg Breeze^4 \land \neg Glitter^4 \land \neg Bump^4 \land \neg Scream^4 \quad ; \quad TurnRight^4$$
$$\neg Stench^5 \land \neg Breeze^5 \land \neg Glitter^5 \land \neg Bump^5 \land \neg Scream^5 \quad ; \quad Forward^5$$
$$Stench^6 \land \neg Breeze^6 \land \neg Glitter^6 \land \neg Bump^6 \land \neg Scream^6$$

$\textsc{Ask}(KB, P_{3,1}) = true$

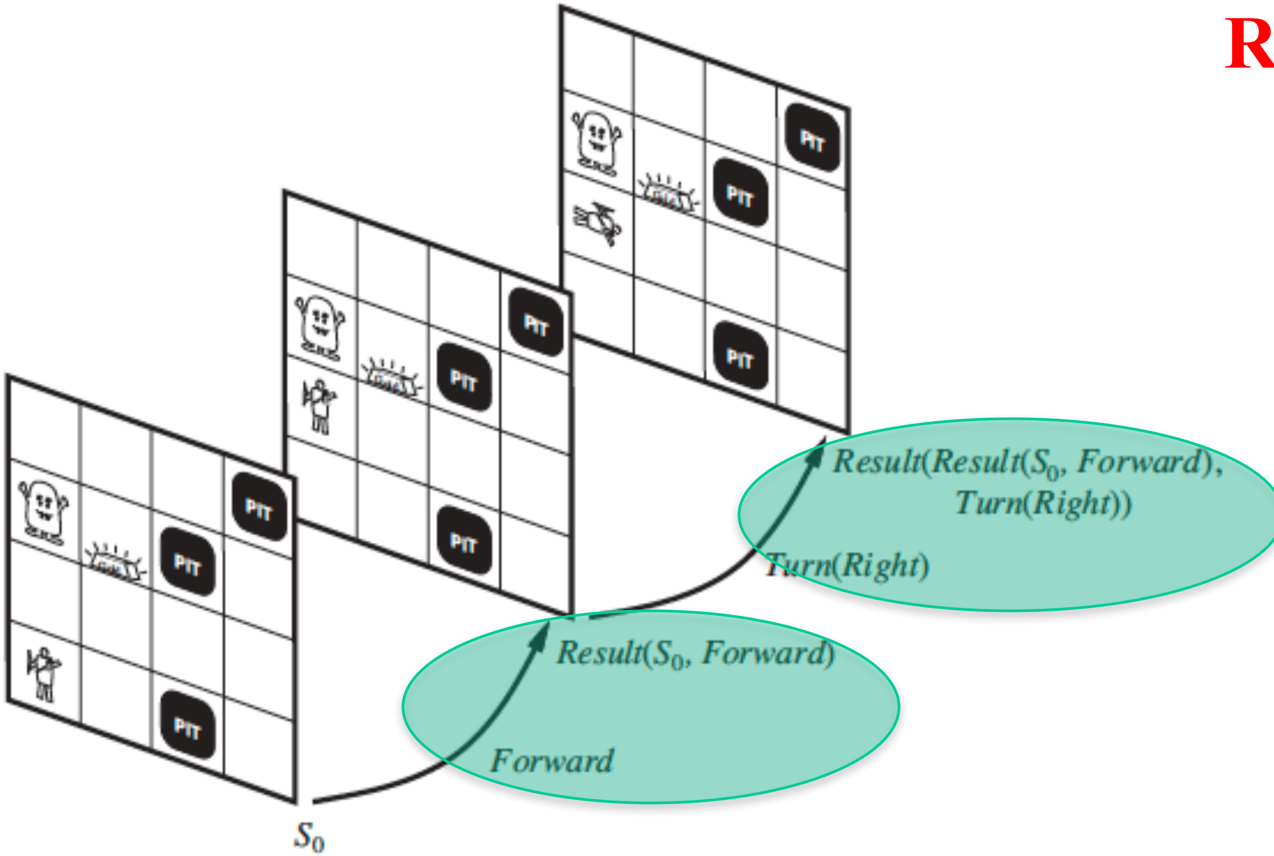
$\textsc{Ask}(KB, W_{1,3}) = true$

Define "OK": $OK^t_{x,y} \Leftrightarrow \neg P_{x,y} \land \neg(W_{x,y} \land WumpusAlive^t)$

$\textsc{Ask}(KB, OK^6_{2,2}) = true$, so the square $[2, 2]$ is OK

In milliseconds, with modern SAT solver.

Alternative formulation: Situation Calculus
R&N 10.4.2



No explicit time. Actions are what changes the world from "situation" to "situation". More elegant, but still need frame axioms to capture what stays the same. Inherent with many representation formalisms: "physical" persistance does not come for free! (and probably shouldn't)

Inference by enumeration / "model checking" Style I

The goal of logical inference is to decide whether $KB \models \alpha$, for some α.

For example, given the rules of the Wumpus World, is P_{22} entailed? Relevant propositional symbols:

R1: $\neg P_{1,1}$

R2: $\neg B_{1,1}$

R3: $B_{2,1}$

$$\text{Models(KB)} \overset{?}{\subseteq} \text{Models}(P_{22})$$

"Pits cause breezes in adjacent squares"

R4: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

R5: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

Inference by enumeration. We have 7 relevant symbols Therefore $2^7 = 128$ interpretations. Need to check if P_{22} is true in all of the KB models (interpretations that satisfy KB sentences).

Q.: KB has many more symbols. Why can we restrict ourselves to these symbols here? But, be careful, typically we can't!!

All equivalent
Prop. / FO Logic

1) $\text{KB} \models \alpha$ **entailment**

Proof techniques

$M(KB) \subseteq M(\alpha)$

by defn. / semantic proofs / truth tables
"model checking"
(style I, R&N 7.4.4) Done.

$KB \vdash \alpha$

soundness and completeness
logical deduction / symbol pushing
proof by inference rules (style II)
e.g. modus ponens (R&N 7.5.1)

$(KB \wedge \neg \alpha)$ is inconsistent

Proof by contradiction
use CNF / clausal form
Resolution (style III, R&N 7.5)
SAT solvers (style IV, R&N 7.6)
most effective

Aside

Standard syntax and semantics for propositional logic. (CS-2800; see 7.4.1 and 7.4.2.)

Syntax:

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

Semantics

Note: Truth value of a sentence is built from its parts "compositional semantics"

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

Logical equivalences

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

(*) key to go to clausal (Conjunctive Normal Form)
Implication for "humans"; clauses for machines.
de Morgan laws also very useful in going to clausal form.

KB at T = 1:

R1: $\neg P_{1,1}$

R2: $\neg B_{1,1}$

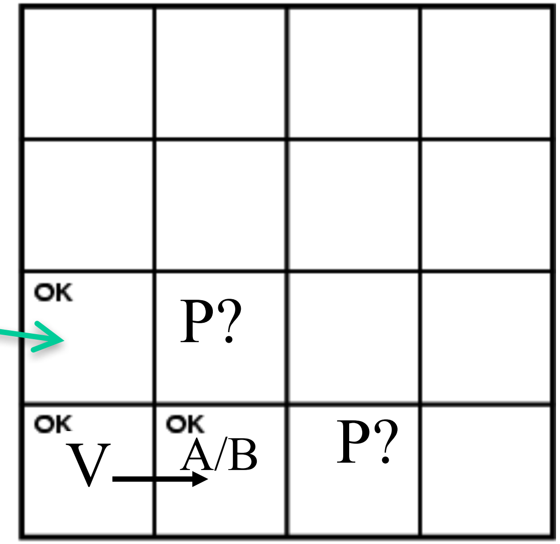
R3: $B_{2,1}$

R4: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

R5: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

Style II: Proof by inference rules
Modus Ponens (MP)

How can we show that KR $\models \neg P_{1,2}$?



Wumpus world
at T = 1

Note: In formal proof, every step needs to be justified.

So, we used R2 and R4.

Length of Proofs

Why bother with inference rules? We could always use a truth table to check the validity of a conclusion from a set of premises.

But, ***resulting proof can be much shorter than truth table method.***

Consider KB:
p_1, p_1 → p_2, p_2 → p_3, …, p_(n-1) → p_n

To prove conclusion: p_n

Inference rules: n-1 MP steps Truth table: 2^n

Key open question: Is there always a short proof for any valid conclusion? Probably not. The NP vs. co-NP question.
(The closely related: P vs. NP question carries a $1M prize.)

Style III: Resolution

First, we need a conversion to Conjunctive Normal Form (CNF) or Clausal Form.

Let's consider converting R4 in clausal form:

R4: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

We have:

$B_{1,1}$) ($P_{1,2}$ Ç $P_{2,1}$)

which gives (implication elimination):

(: $B_{1,1}$ Ç $P_{1,2}$ Ç $P_{2,1}$)

Also

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

which gives:

(: ($P_{1,2}$ Ç $P_{2,1}$) Ç $B_{1,1}$)

Thus,

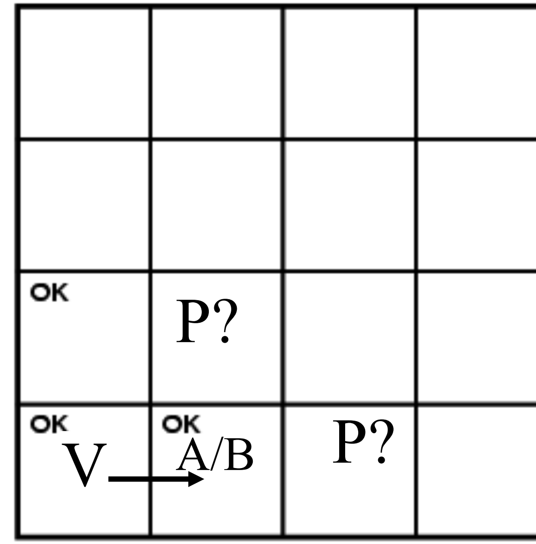
(: $P_{1,2}$ Æ : $P_{2,1}$) Ç $B_{1,1}$

leaving,

(: $P_{1,2}$ Ç $B_{1,1}$)

(: $P_{2,1}$ Ç $B_{1,1}$)

(note: clauses in red)



Wumpus world at T = 1

Style III: Resolution

First, we need a conversion to Conjunctive Normal Form (CNF) or Clausal Form.

Let's consider converting R4 in clausal form:

$$\mathbf{R4:}\ \ B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$

We have:

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

which gives (implication elimination):

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

Also

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

which gives:

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

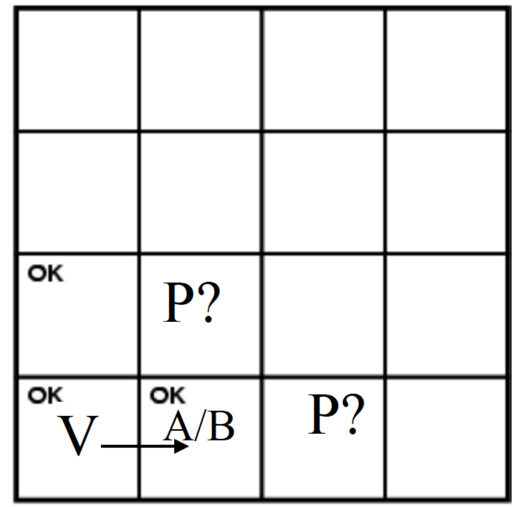
Thus,

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

leaving,

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

(note: clauses in red)



Wumpus world at T = 1

KB at T = 1:

R1: $\neg P_{1,1}$

R2: $\neg B_{1,1}$

R3: $B_{2,1}$

R4: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

R5: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

KB at T=1 in clausal form:

R1: $\neg P_{1,1}$

R2: $\neg B_{1,1}$

R3: $B_{2,1}$

R4a: $\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$

R4b: $\neg P_{1,2} \vee B_{1,1}$

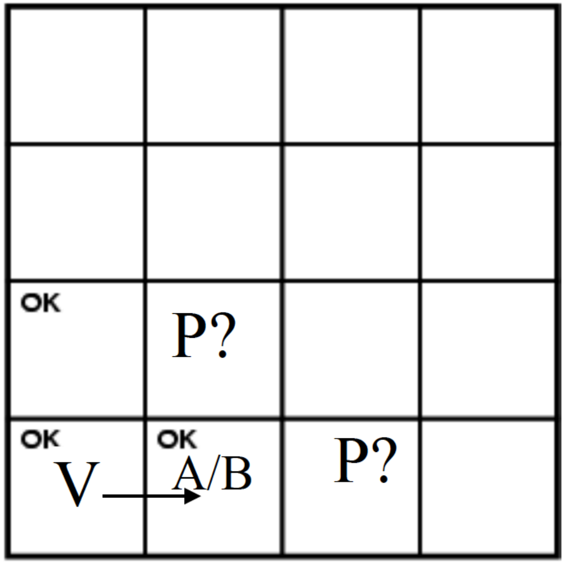
R4c: $\neg P_{2,1} \vee B_{1,1}$

R5a: $\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$

R5b: $\neg P_{1,1} \vee B_{2,1}$

R5c: $\neg P_{2,2} \vee B_{2,1}$

R5d: $\neg P_{3,1} \vee B_{2,1}$



Wumpus world at T = 1

How can we show that KR $\models \neg P_{1,2}$?

Proof by contradiction:
Need to show that (KB $\wedge$ $P_{1,2}$) is
inconsistent (unsatisfiable).

Resolution rule:

($\alpha \vee$ p) and ($\beta \vee \neg$ p)

gives resolvent (logically valid conclusion):

($\alpha \vee \beta$)

If we can reach the empty clause, then KB is inconsistent. (And, vice versa.)

KB at T=1 in clausal form:

R1: $\neg P_{1,1}$

R2: $\neg B_{1,1}$

R3: $B_{2,1}$

R4a: $\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$

R4b: $\neg P_{1,2} \vee B_{1,1}$

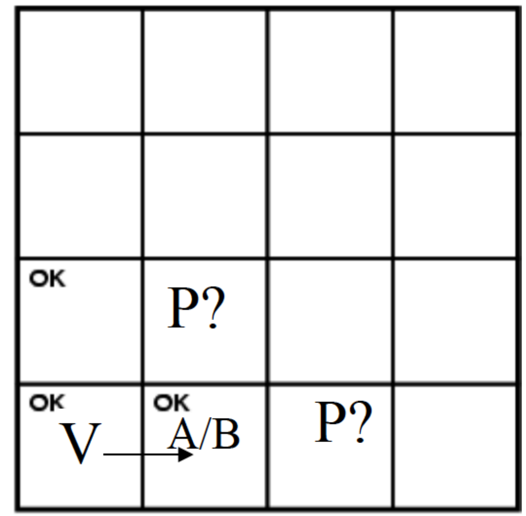
R4c: $\neg P_{2,1} \vee B_{1,1}$

R5a: $\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$

R5b: $\neg P_{1,1} \vee B_{2,1}$

R5c: $\neg P_{2,2} \vee B_{2,1}$

R5d: $\neg P_{3,1} \vee B_{2,1}$



Wumpus world at T = 1

Show that (KB $\wedge$ $P_{1,2}$) is **inconsistent. (unsatisfiable)**

R4b with $P_{1,2}$ resolves to $B_{1,1}$,
which with R2, resolves to the empty clause, $\square$.
So, we can conclude KB $\models \neg P_{1,2}$.
(make sure you use "what you want to prove.")

KB at T=1 in clausal form:

R1: $\neg P_{1,1}$

R2: $\neg B_{1,1}$

R3: $B_{2,1}$

R4a: $\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$

R4b: $\neg P_{1,2} \vee B_{1,1}$

R4c: $\neg P_{2,1} \vee B_{1,1}$

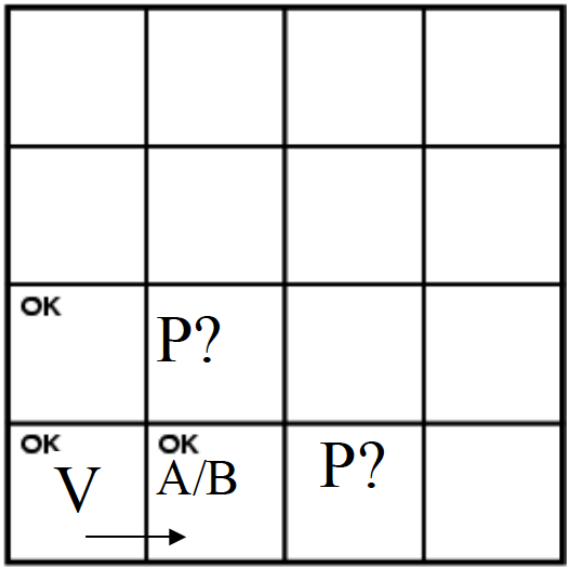
R5a: $\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$

R5b: $\neg P_{1,1} \vee B_{2,1}$

R5c: $\neg P_{2,2} \vee B_{2,1}$

R5d: $\neg P_{3,1} \vee B_{2,1}$

Another example resolution proof



Wumpus world at T = 1

Note that R5a resolved with R1, and then resolved with R3, gives $(P_{2,2} \vee P_{3,1})$.

Almost there… to show $KB \models (P_{2,2} \vee P_{3,1})$, we need to show $KB \wedge (\neg (P_{2,2} \vee P_{3,1}))$ is inconsistent. (Why? Semantically?)
So, show $KB \wedge \neg P_{2,2} \wedge \neg P_{3,1}$ is inconsistent.
This follows from $(P_{2,2} \vee P_{3,1})$; because in two more resolution steps, we get the empty clause (a contradiction).

Length of Proofs

Consider KB:

p_1, p_1 → p_2, p_2 → p_3, …, p_(n-1) → p_n

To prove conclusion: p_n

Resolution. Assert (¬ p_n)
with (¬ p_(n-1) ∨ p_n) gives (¬ p_(n-1))
with (¬ p_(n-2) ∨ p_(n-1) gives (¬ p_(n-2))
…
with (¬ p_1) ∨ p_2) gives (¬ p_1)
with (p_1) gives empty clause (contradiction).
QED
Note how resolution mimics Modus Ponens steps.

Inference rules: n resolution steps   Truth table: 2^n

So, efficient on these proofs!

Length of Proofs

What is hard for resolution?

Consider:

Given a fixed pos. int. N

$(P(i,1) \vee P(i$ … .. N+1

$(\neg P(i,j) \vee \neg$ … .. N;

.. N+1;

..N+1; i =/= i'

What does this encode?

Think of: P(i,j) for "object i in location j"

Pigeon hole problem…

Provable requires exponential number of resolution steps to reach empty clause (Haken 1985). Method "can't count."

Style IV: SAT Solvers

Instead of using resolution to show that

$KB \wedge \neg \alpha$ is inconsistent,

modern Satisfiability (SAT) solvers operating on the clausal form are *much* more effi

The SAT solvers trea constraints (disjunctions) on Boo blem! Current solvers are ve illion+ variables and several millions of clauses.

SAT SOLVERS CAN BE VIEWED AS DOING A SPECIAL FORM OF RESOLUTION

Systematic: Davis Putnam (DPLL) + *series of improvements*
Stochastic local search: WalkSAT (issue?)

See R&N 7.6. "Ironically," we are back to semantic model checking, but way more clever than basic truth assignment enumeration (exponentially faster)!

DPLL improvements

Backtracking + …

1) **Component analysis (disjoint sets of constraints? Problem decomposition?)**
2) **Clever variable and value ordering (e.g. degree heuristics)**
3) **Intelligent backtracking and clause learning (conflict learning)**
4) **Random restarts (heavy tails in search spaces…)**
5) **Clever data structures**

1+ Million Boolean vars & 10+ Million clause/constraints are feasible nowadays. (e.g. Minisat solver)

Has changed the world of verification (hardware/software) over the last decade (incl. Turing award for Clarke). Widely used in industry, Intel, Microsoft, IBM etc.

All equivalent
Prop. / FO Logic

ENDS LOGIC PART

1) $\text{KB} \models \alpha$ **entailment**