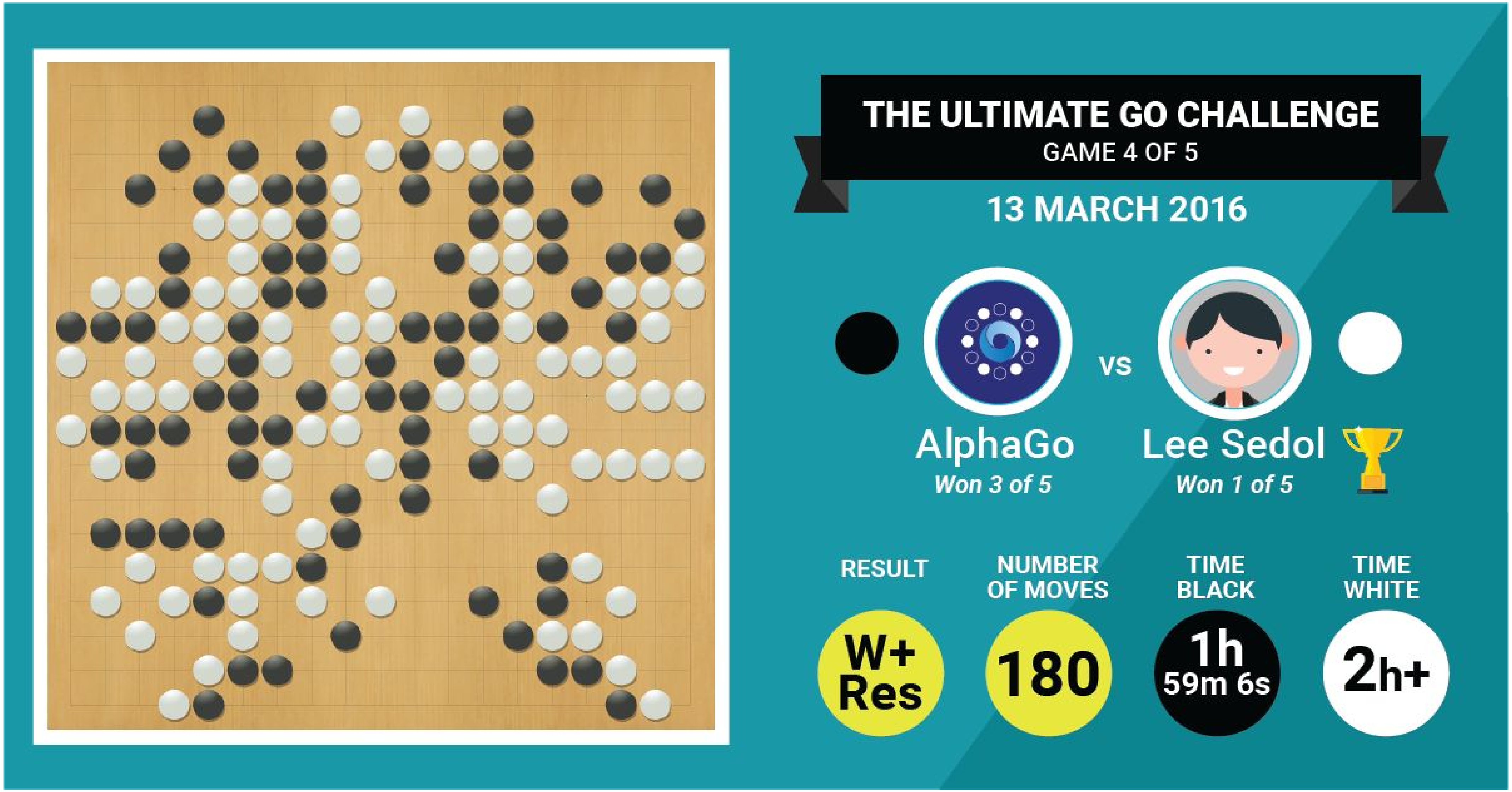
Logical Agents

Chocolate is dessert.

I love all desserts.

∴ I love chocolate.

Update: http://googleasiapacific.blogspot.co.uk/



THE ULTIMATE GO CHALLENGE

GAME 4 OF 5

13 MARCH 2016

AlphaGo

Won 3 of 5

vs

Lee Sedol

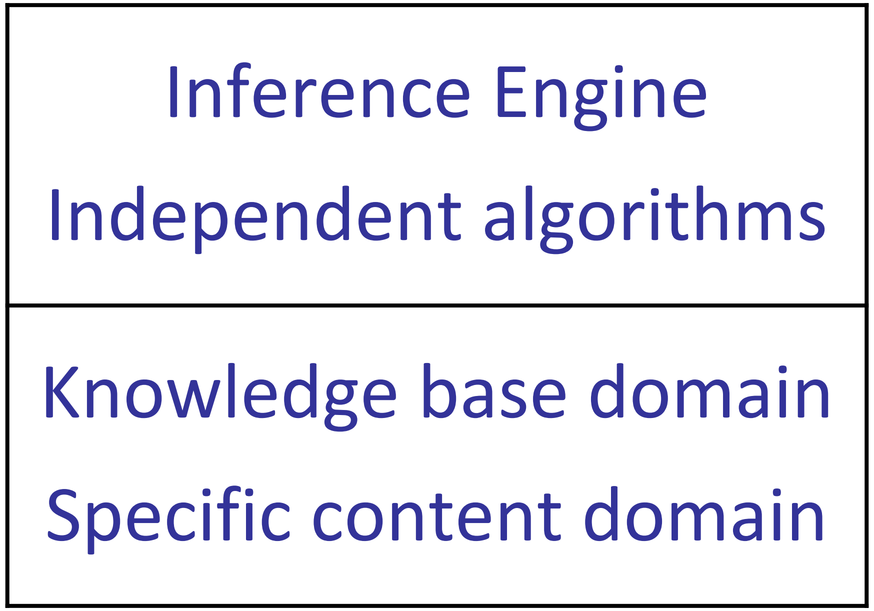
Won 1 of 5

RESULT	NUMBER OF MOVES	TIME BLACK	TIME WHITE
W+ Res	180	1h 59m 6s	2h+

Today

- Knowledge-based agents
- Logic - entailment and inference
- Propositional logic
- Inference rules & resolution

Knowledge Base



Inference Engine

Independent algorithms

Knowledge base domain

Specific content domain

Knowledge base = 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

Wumpus World

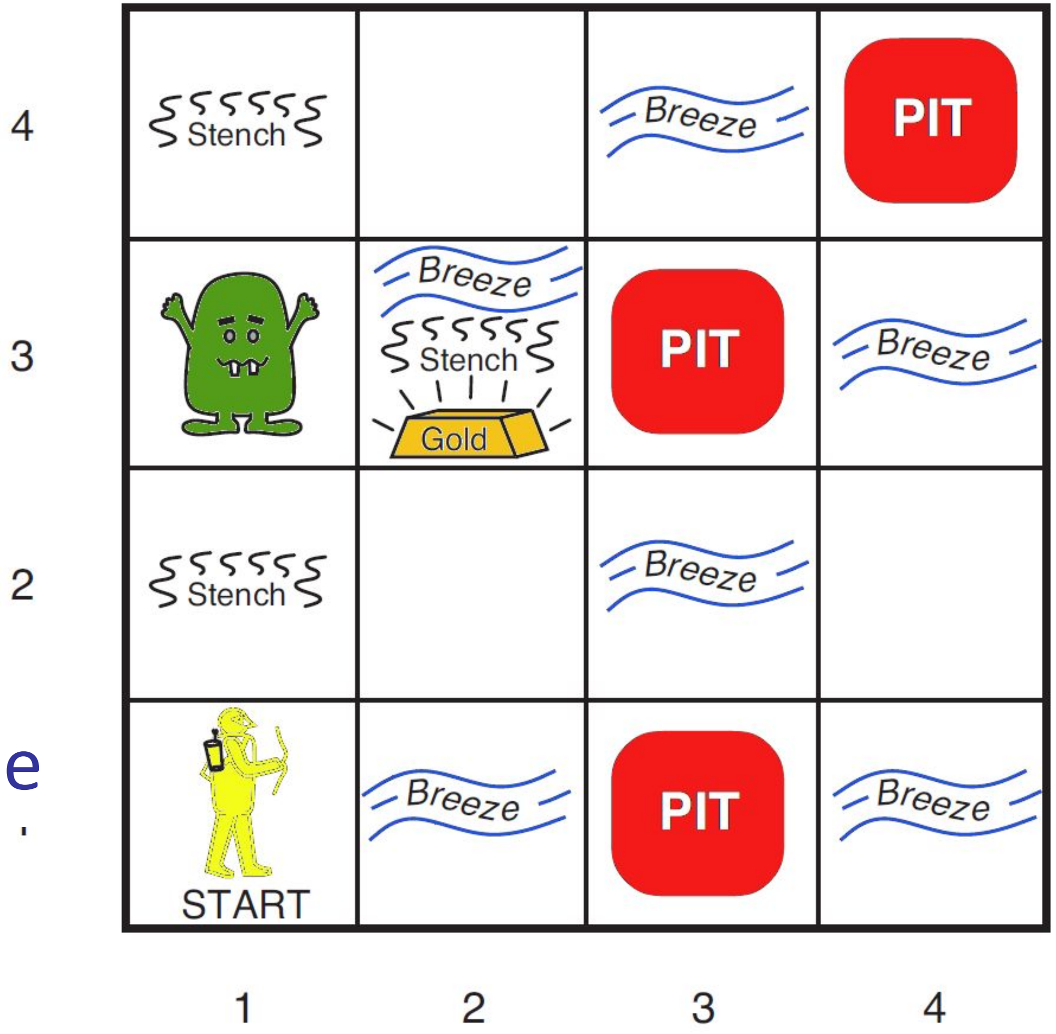
A cave with rooms connected by passageways.

The wumpus is somewhere in the cave and eats anyone who enters it room.

The wumpus doesn't move.

Some rooms contain bottomless pits.

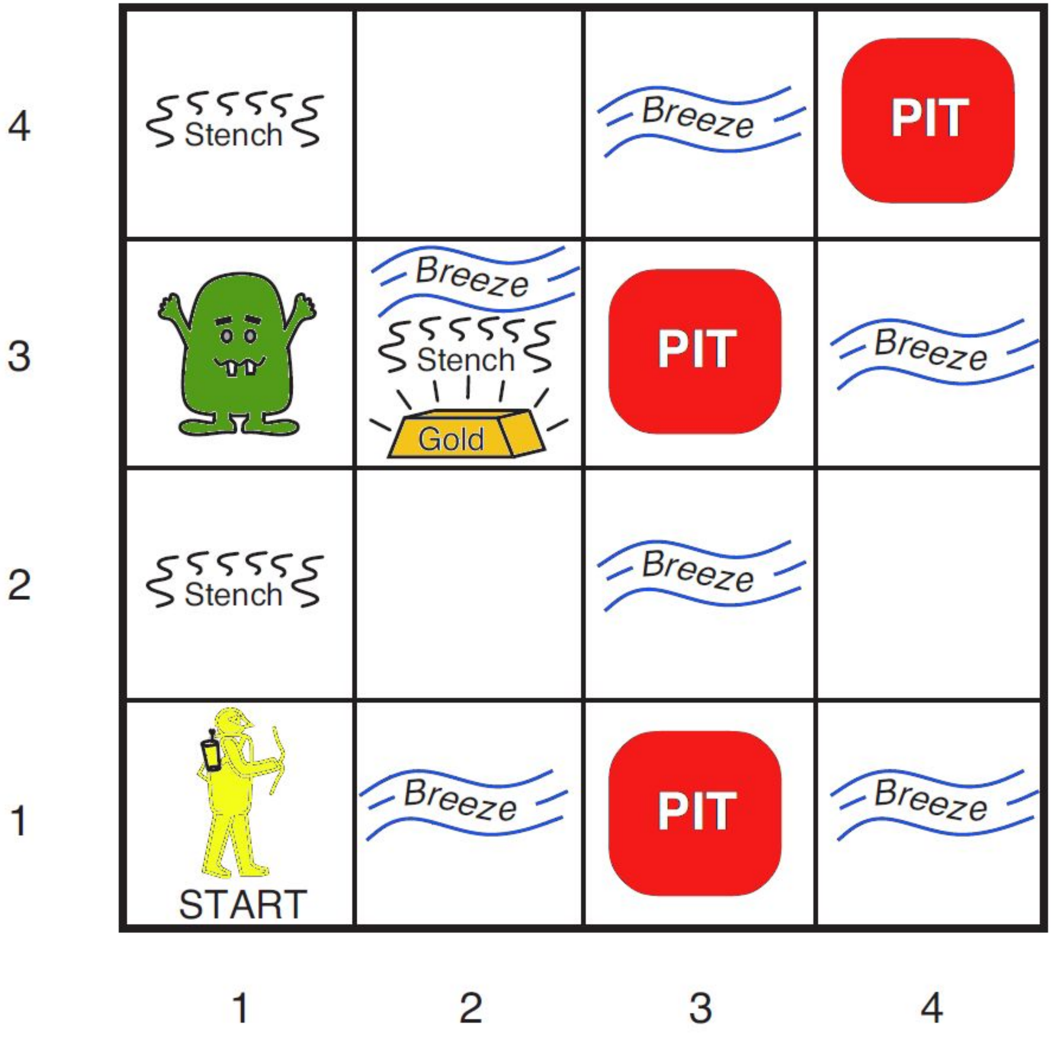
There is a heap of gold somewhere in the cave.



Wumpus World

Knowledge:

- Rooms adjacent to Wumpus are smelly (a stench is detected)
- Rooms adjacent to pits are breezy
- Glitter in the room where the gold is



Wumpus World

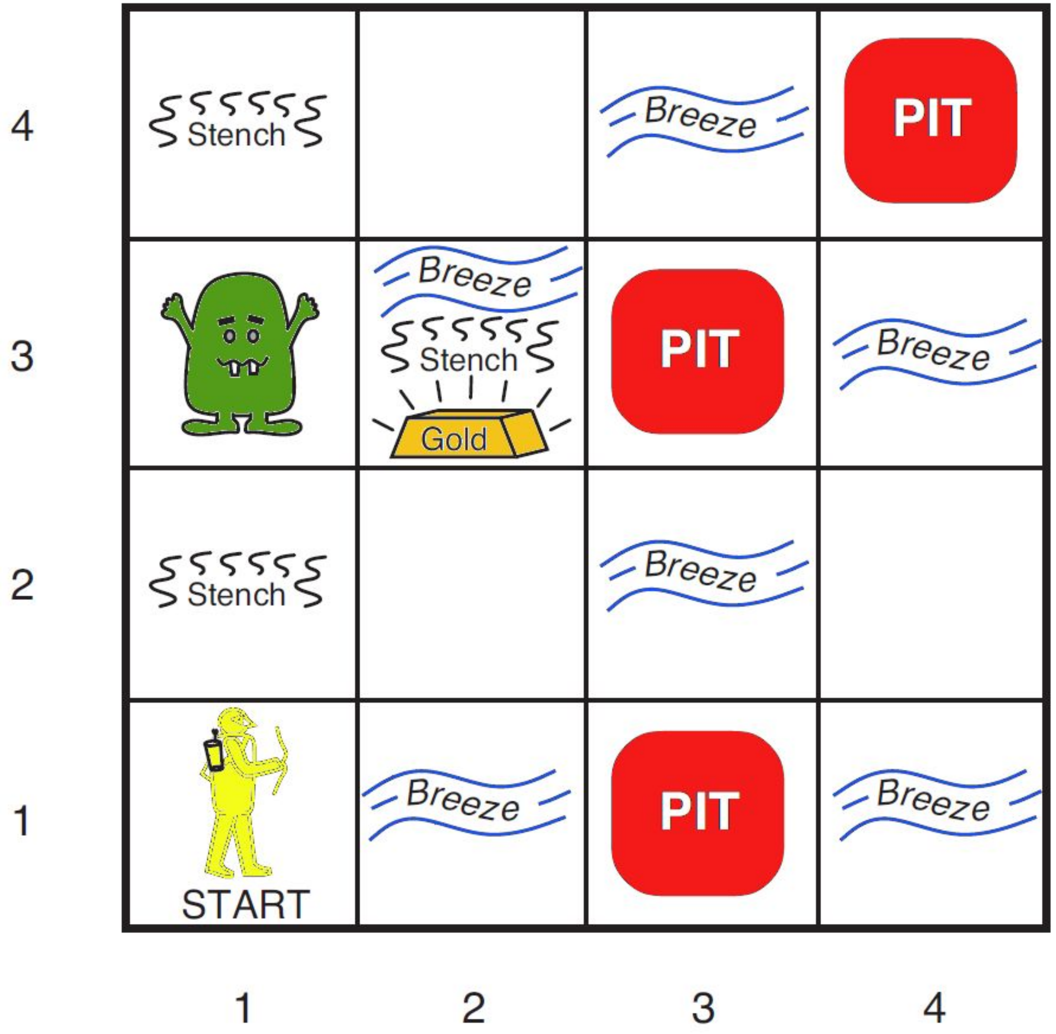
Performance measure

- gold: +1000
- death: -1000
- -1 per step

Environment: 16 rooms

Actuators: Move, Grab

Sensors: Smell, Breeze, Glitter



Wumpus World

Fully Observable?

No - it's a cave

Deterministic?

Yes – outcomes exactly specified

Episodic?

No – sequential

Static?

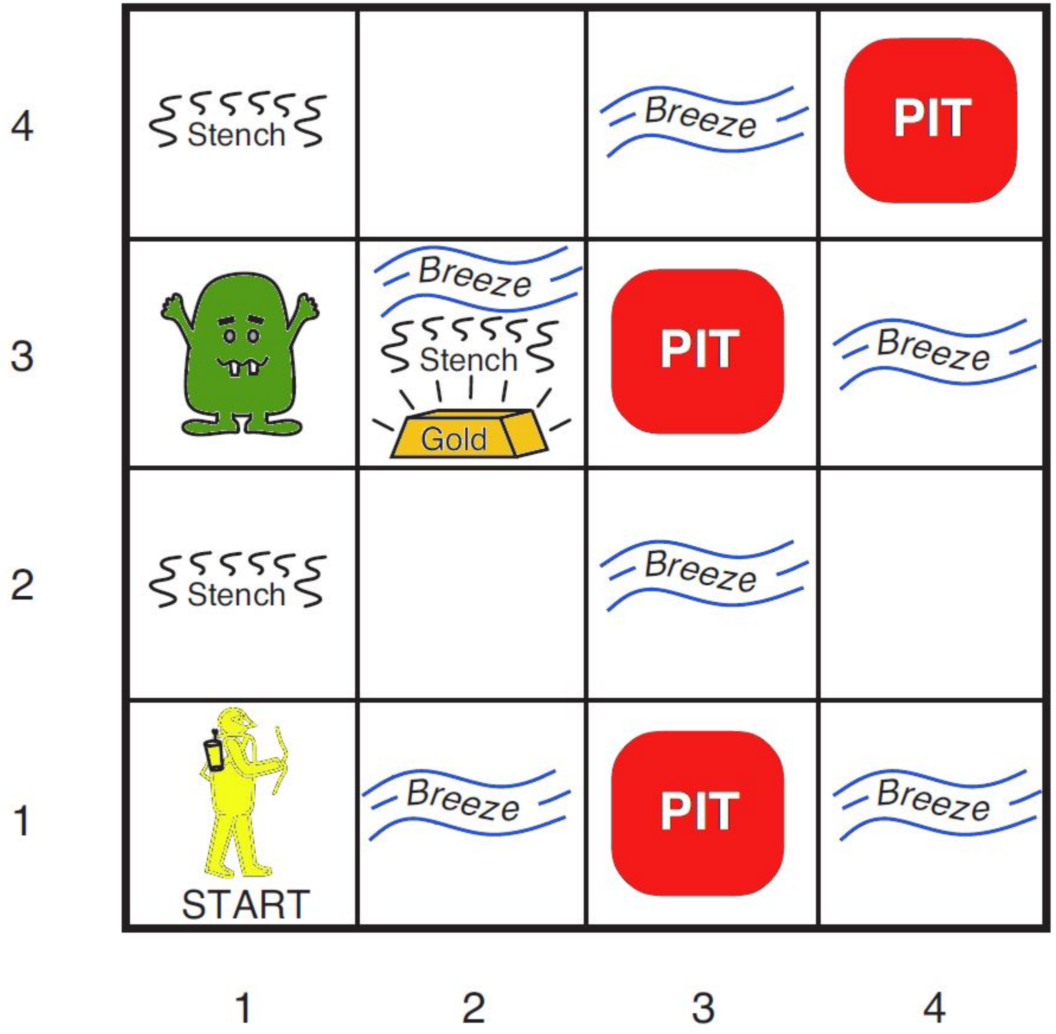
Yes – Wumpus and Pits do not move

Discrete?

Yes

Single-agent?

Yes – Wumpus is essentially a natural feature

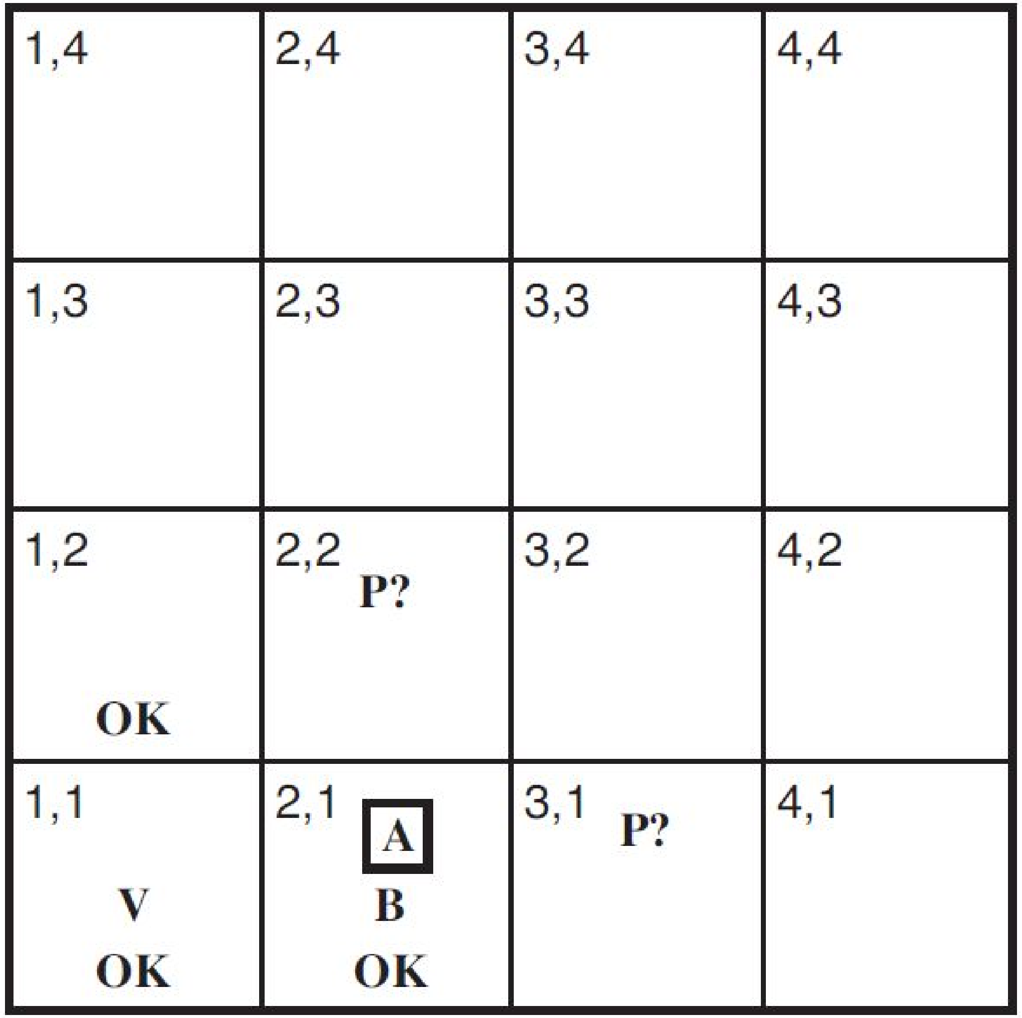


Wumpus World

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 $\boxed{\text{A}}$ OK	2,1 OK	3,1	4,1

$\boxed{\text{A}}$	= *Agent*
B	= *Breeze*
G	= *Glitter, Gold*
OK	= *Safe square*
P	= *Pit*
S	= *Stench*
V	= *Visited*
W	= *Wumpus*

Wumpus World



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

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

Logic

A logic is a formal language for representing information such that conclusions can be drawn

Syntax defines the sentences in the language

Semantics define the meaning of sentences (their truth)

Example: the language of arithmetic

$x + 2 \geq y$ is a sentence

$x2 + y >$ is not a sentence

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

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

Entailment

Entailment means that one thing follows from another:

$$KB \models \alpha$$

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

Examples:

The KB containing "the Giants won" and "the Reds won" entails "Either the Giants won or the Reds won"

$x + y = 4$ entails $4 = x + y$

$x = 0$ entails $x * y = 0$

Inference

$$KB \vdash_i \alpha$$

sentence α can be derived from *KB* by procedure *i*

Consequences of KB are a haystack; α is a needle.

Entailment = needle in haystack

Inference = finding it

Inference

$$KB \vdash_i \alpha$$

sentence α can be derived from *KB* by procedure *i*

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 $KB \vdash_i \alpha$

Propositional Logic: Syntax

Propositional logic is the simplest logic

The proposition symbols P1, P2 etc are sentences

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

If P1 and P2 are sentences, P1 $\wedge$ P2 is a sentence (conjunction)

If P1 and P2 are sentences, P1 $\vee$ P2 is a sentence (disjunction)

If P1 and P2 are sentences, P1 $\Rightarrow$ P2 is a sentence (implication)

If P1 and P2 are sentences, P1 $\Leftrightarrow$ P2 is a sentence (biconditional)

Propositional Logic: Semantics & Models

Each model specifies true/false for each proposition symbol

With these symbols, 4 possible models, can be enumerated automatically.

P1	P2
true	true
true	false
false	true
false	false

Propositional Logic

Rules for evaluating truth with respect to a model:

¬S is true iff S is false

S1 ∧ S2 is true iff S1 is true and S2 is true

S1 ∨ S2 is true iff S1 is true or S2 is true

S1 ⇒ S2 is true iff S1 is false or S2 is true

S1 ⇔ S2 is true iff S1 ⇒ S2 is true and S2 ⇒ S1 is true

Propositional Logic

S1 ⇒ S2 is true iff S1 is false or S2 is true

S1: 3 is odd

S2: Tokyo is the capital of Japan

S1 ⇒ S2 ?

A. true (because S2 is true)
B. false

Propositional Logic

S1 ⇒ S2 is true iff S1 is false or S2 is true

S1: 3 is even

S2: Paris is the capital of Japan

S1 ⇒ S2 ?

A. true (because S1 is false)
B. false

Propositional Logic

Simple recursive process evaluates an arbitrary sentence.

P1 is false

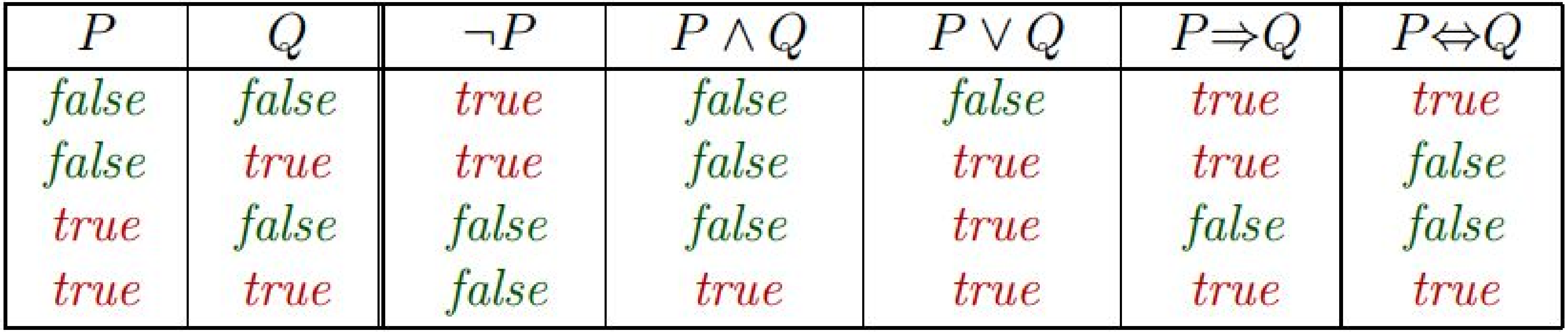
P2 is false

P3 is true

$\neg P1 \land (P2 \lor P3) = \neg \text{false} \land (\text{false} \lor \text{true}) = \text{true} \land \text{true} = \text{true}$

Truth Table

A truth table lists all the possibilities for the propositional symbols and the corresponding truth values of the compound sentences



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*

Wumpus World

$P_{i,j}$: true if there is a pit in [i, j].

$B_{i,j:}$ true if there is a breeze in [i, j].

KB:

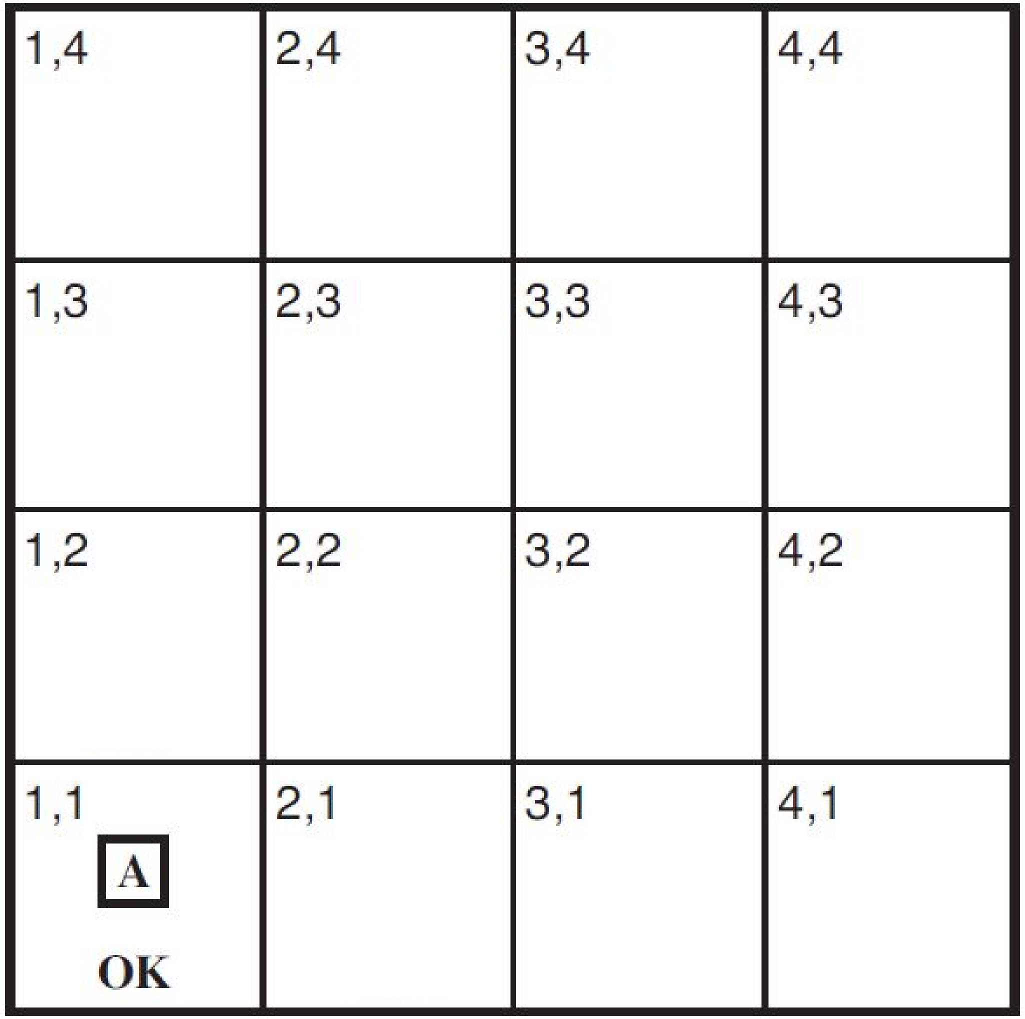
There is no pit in [1, 1]: $\neg P_{1,1}$

A room is breezy if and only if there is an adjacent pit:

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

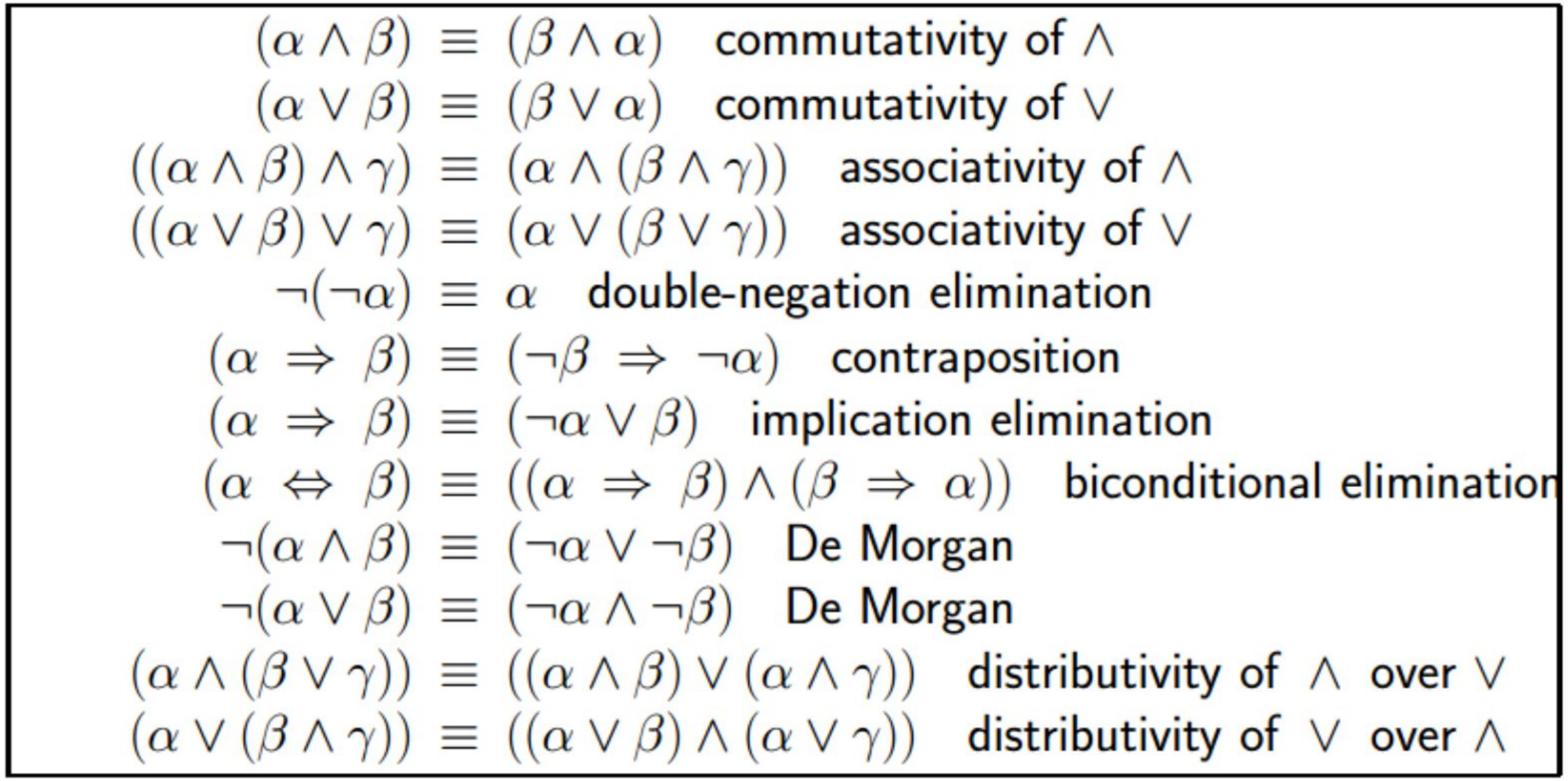
Percept: $\neg B_{1,1}$

Is $\neg P_{1,2}$ entailed?



Logical equivalence

Two sentences are logically equivalent only if they are true in the same models:
α ≡ β if and only if α ╞ β and β ╞ α



$(\alpha \wedge \beta)$	$\equiv$	$(\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv$	$(\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv$	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv$	$(\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

Validity & Satisfiability

A valid sentence is a sentence that is true in every possible model

A satisfiable sentence is a sentence that is true in some model

An unsatisfiable sentence is a sentence that is false in all models

Validity & Satisfiability

A valid sentence is a sentence that is true in every possible model

A satisfiable sentence is a sentence that is true in some model

An unsatisfiable sentence is a sentence that is false in all models

$A \lor \neg A$

A. valid

B. satisfiable

C. unsatisfiable

Validity & Satisfiability

A valid sentence is a sentence that is true in every possible model

A satisfiable sentence is a sentence that is true in some model

An unsatisfiable sentence is a sentence that is false in all models

$A \land \neg A$

A. valid
B. satisfiable
C. unsatisfiable

Validity & Satisfiability

A valid sentence is a sentence that is true in every possible model

A satisfiable sentence is a sentence that is true in some model

An unsatisfiable sentence is a sentence that is false in all models

A V B

A. valid
B. satisfiable
C. unsatisfiable

Validity & Satisfiability

Validity is connected to inference via the Deduction Theorem:

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

Satisfiability is connected to inference via the following:

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

Applying Inference

Legitimate (sound) generation of new sentences from old

Proof = a sequence of inference rule applications

Typically require transformation of sentences into a normal form

Inference Rules

Whenever any sentences of the form $\alpha \Rightarrow \beta$ and α are given, then β can be inferred. (Modus Ponens).

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

KB:

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

$B_{1,1}$

$(P_{1,2} \vee P_{2,1})$ can be inferred

Inference Rules

Any of the sentences can be inferred from a conjunction of sentences (and elimination):

$$\frac{\alpha \land \beta}{\alpha}$$

It is sunny and I have an umbrella

It is sunny

Inference Rules - Unit Resolution

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

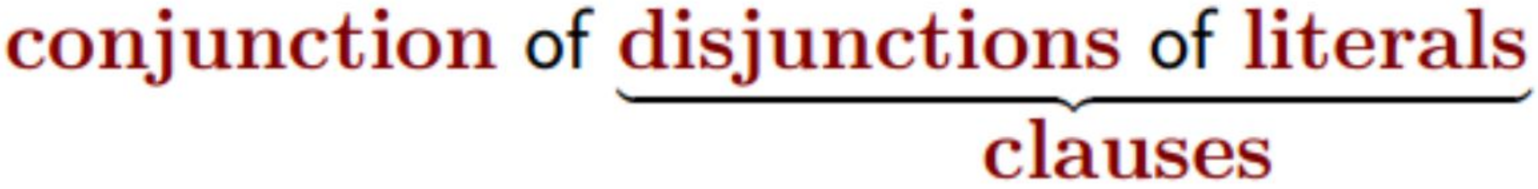
There is a pit in [1, 2] or [2, 1]: $P_{1,2} \lor P_{2,1}$

There is no pit in [1, 2]: $\neg P_{1,2}$

There is a pit in [2, 1]: $P_{2,1}$

Conjunctive Normal Form (CNF)

a conjunction (and) of clauses, where each clause is a disjunction (or) of literals.



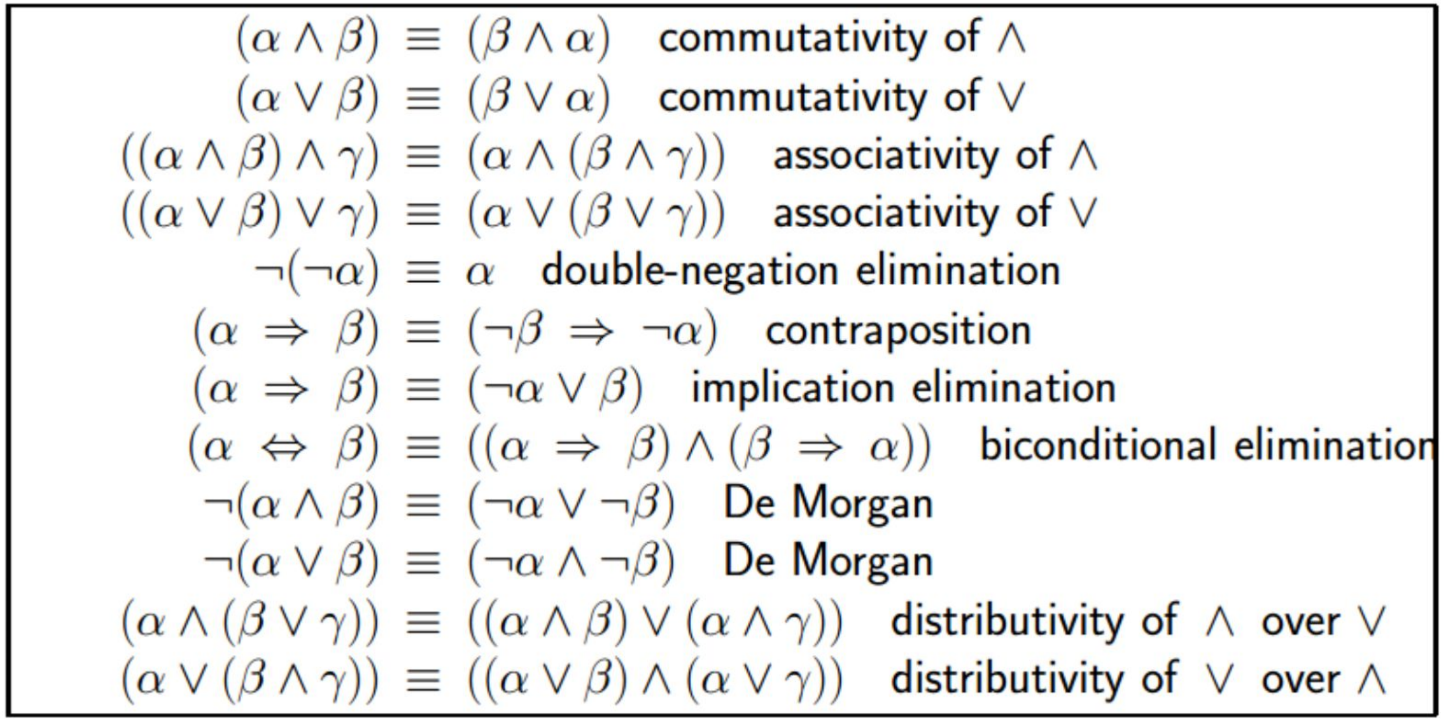
$$\text{conjunction of } \underbrace{\text{disjunctions of literals}}_{\text{clauses}}$$

Conversion to CNF

A ⇒ B

Implication elimination:

¬A V B



$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha) \equiv \alpha$	double-negation elimination
$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

Wumpus World

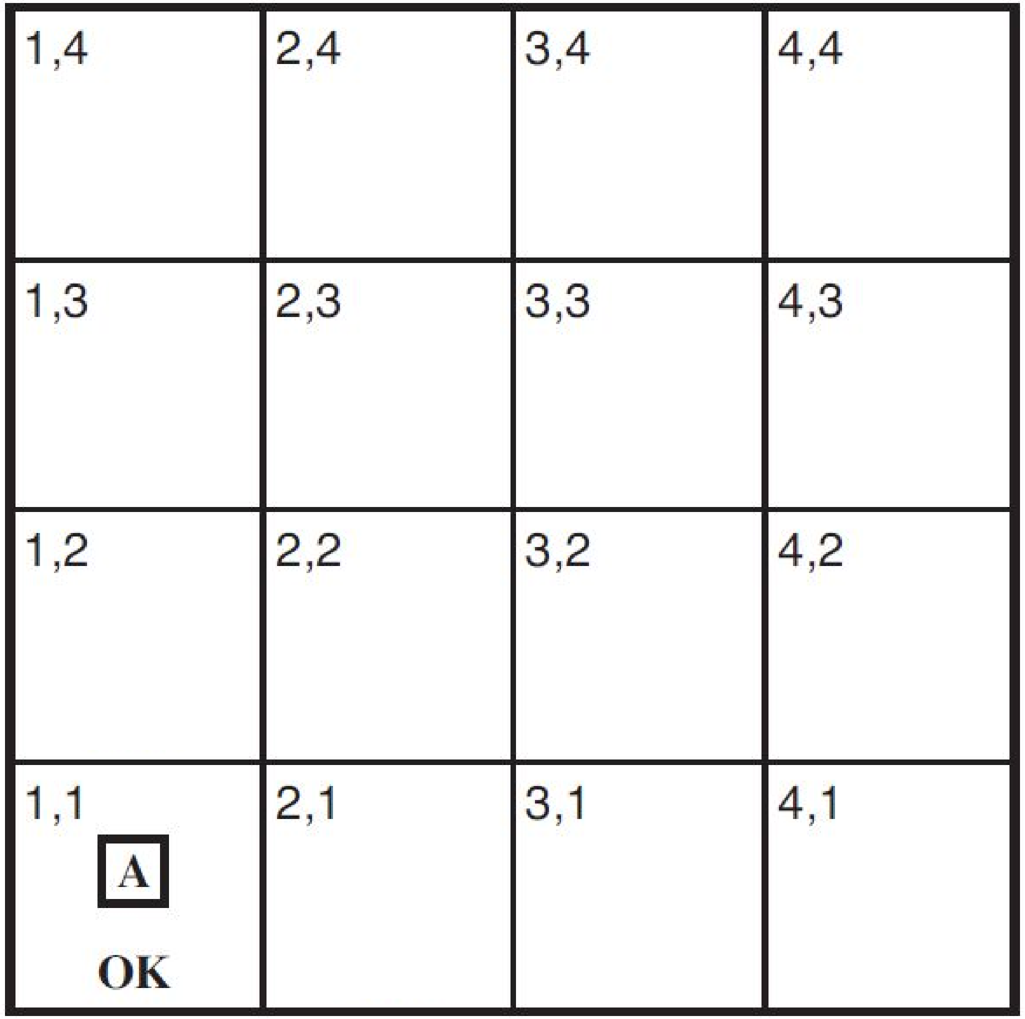
$P_{i,j}$: true if there is a pit in [i, j].
$B_{i,j:}$ true if there is a breeze in [i, j].
KB:

There is no pit in [1, 1]: $\neg P_{1,1}$
A room is breezy if and only if there is an adjacent pit:

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

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

Is $\neg P_{1,2}$ entailed?



Conversion to CNF

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

1. Eliminate ⇔, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)$.

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

2. Eliminate ⇒, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \lor \beta$.

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

3. Move ¬ inwards using de Morgan's rule:

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

4. Apply distributivity law (∨ over ∧) and flatten:

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

Resolution Algorithm

To show that $KB \models \alpha$, we show KB ∧ ¬α is unsatisfiable.

1. Convert KB ∧ ¬α to CNF
2. Apply resolution rule repeatedly
3. At the end: empty clause - unsatisfiable (A ∧ ¬A)

Resolution is sound and complete for propositional logic.