Logic

Chen-Yu Wei

Wumpus World

Performance

Gold +1000, death -1000, -1 per step, -10 for using the arrow

Environment

Perceive stench if adjacent to wumpus

Perceive breeze if adjacent to pit

Perceive glitter if in the square of gold

Can grab gold if in the square of gold

Can shoot and kill wumpus if you're facing it (shooting uses up the only arrow)

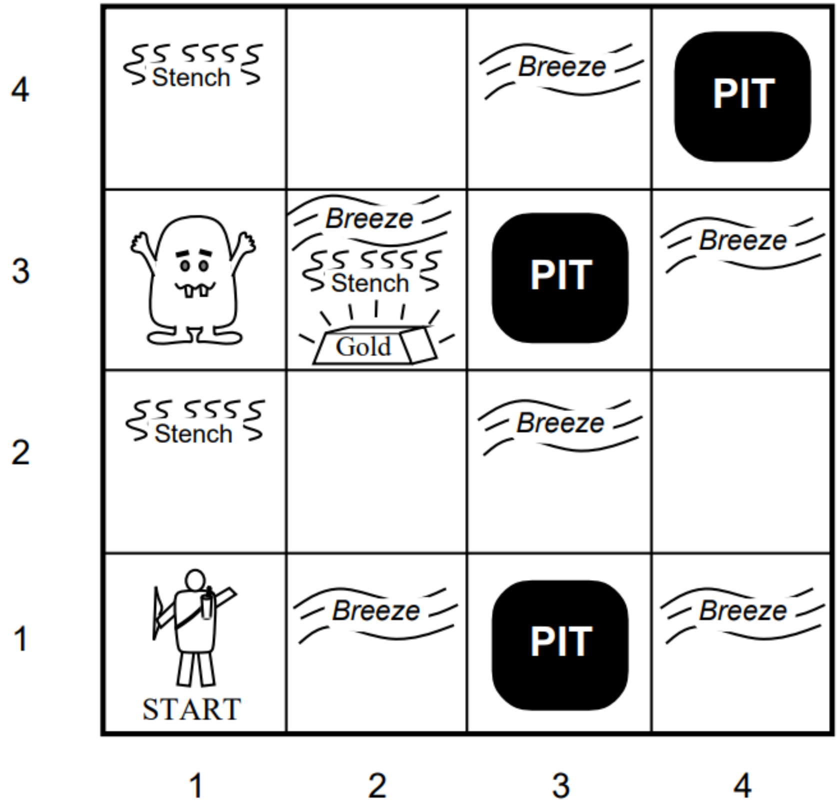
Die if entering a square with pit or living wumpus

Actions

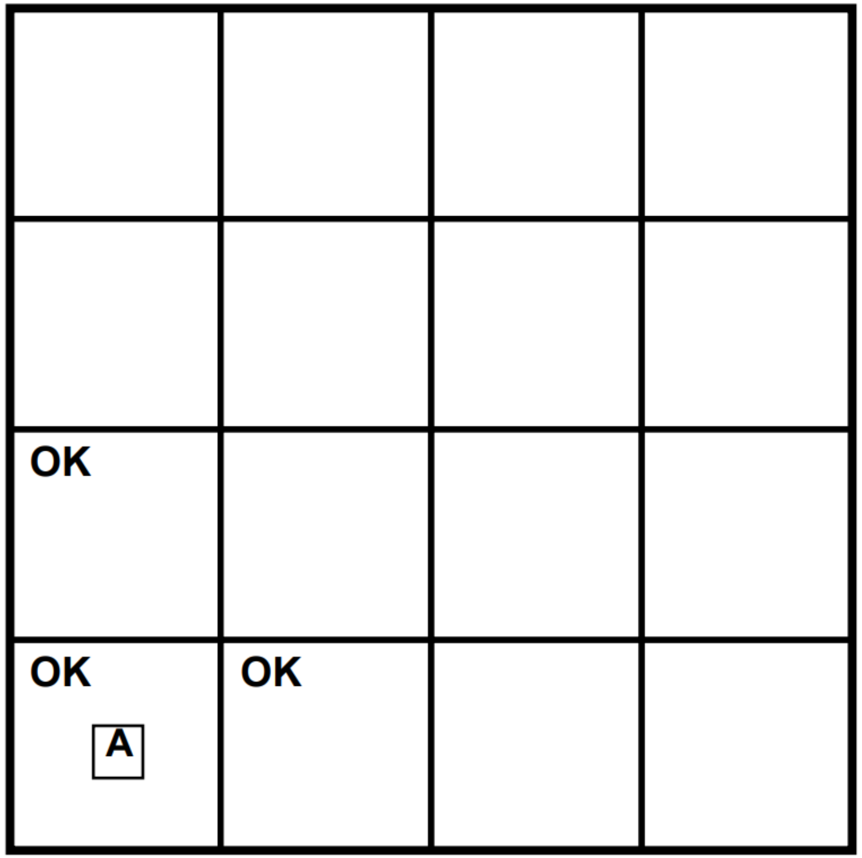
Left turn, right turn, forward, grab, shoot

Sensors

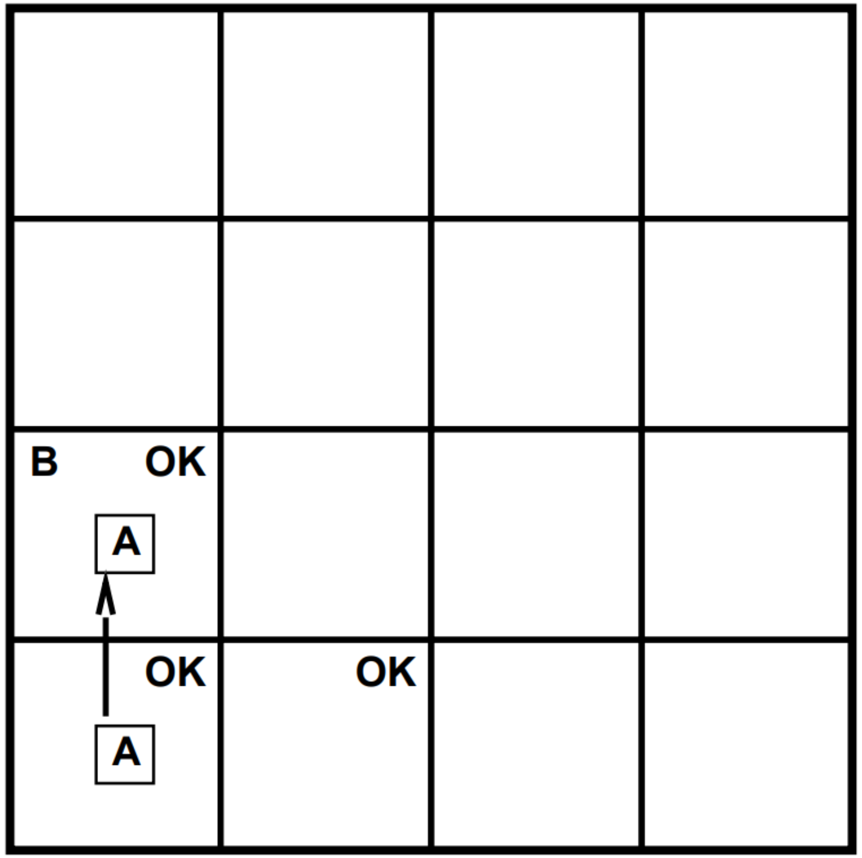
Breeze, glitter, smell



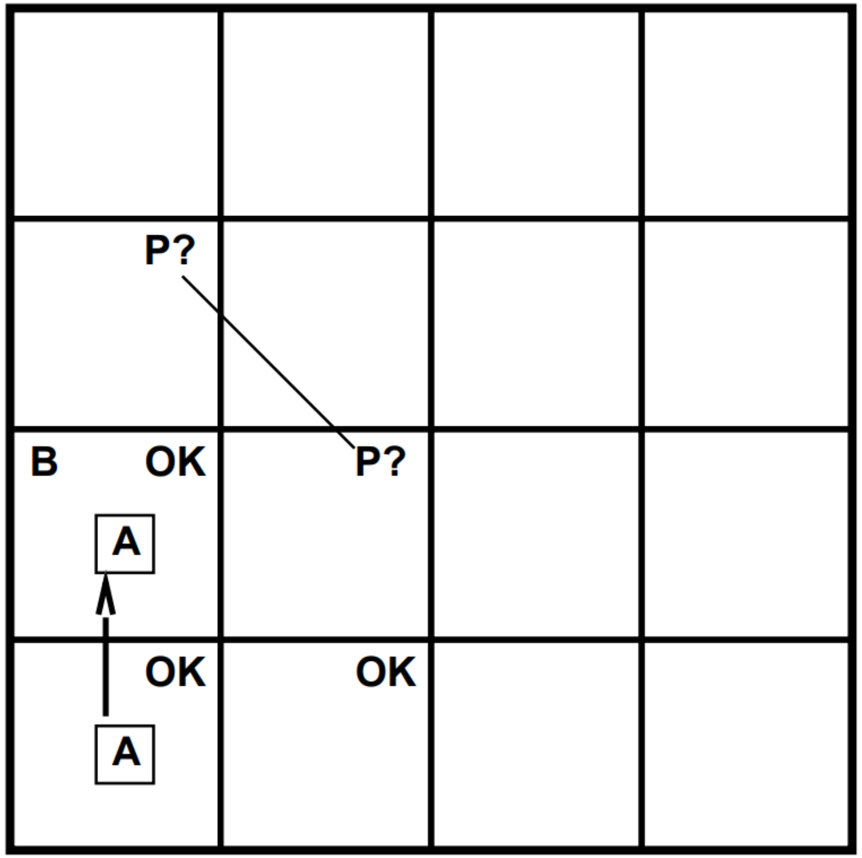
Exploring a wumpus world



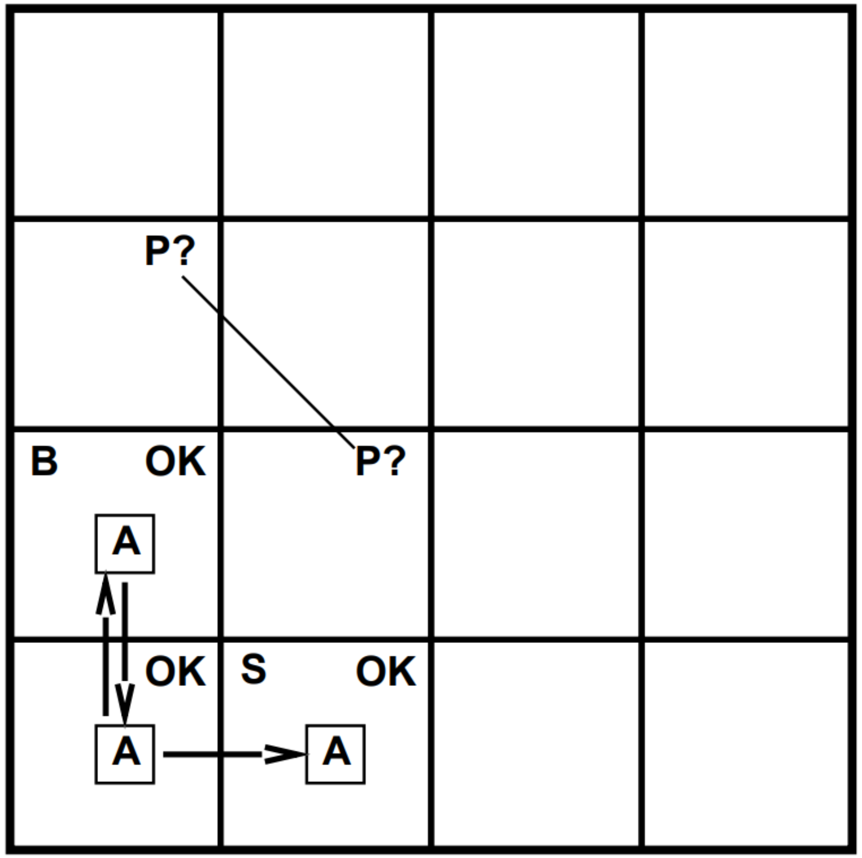
Exploring a wumpus world



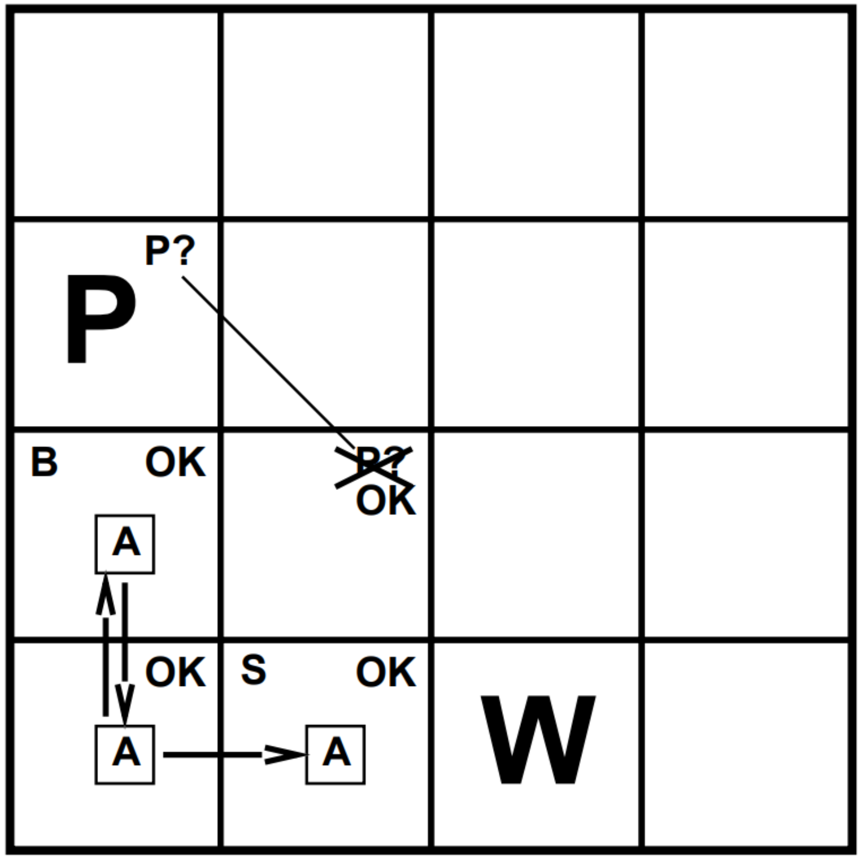
Exploring a wumpus world



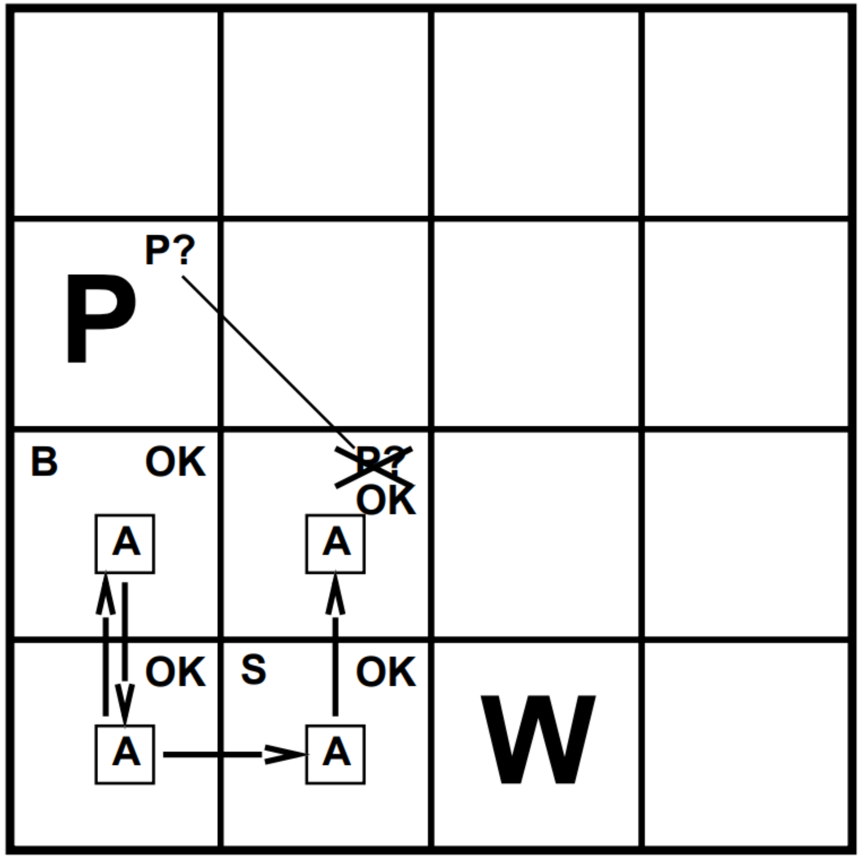
Exploring a wumpus world



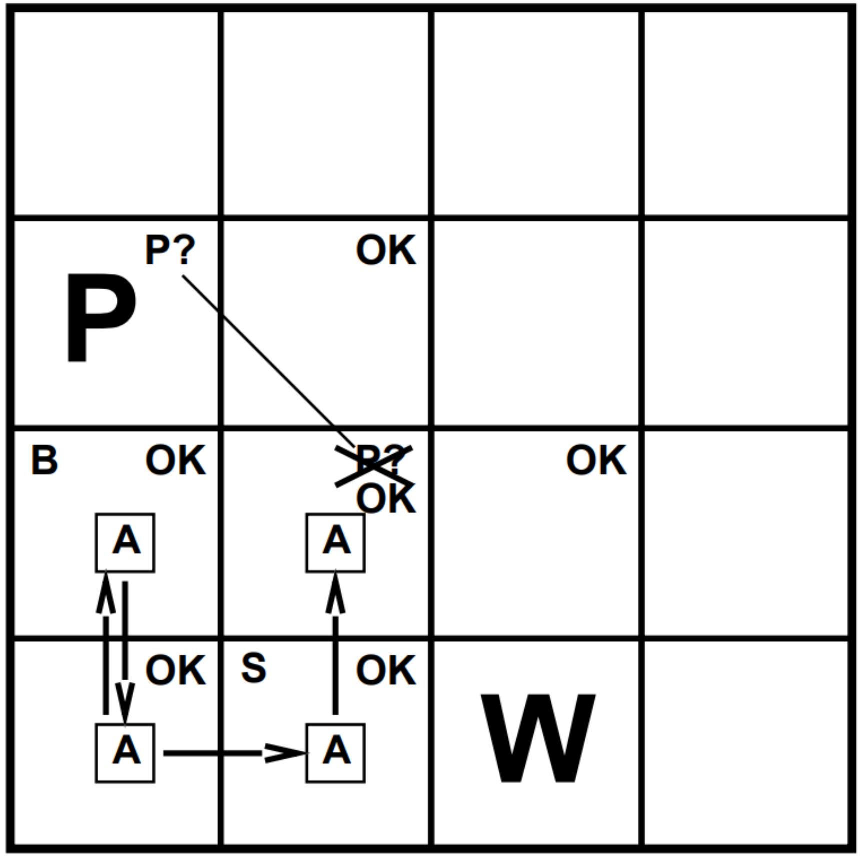
Exploring a wumpus world



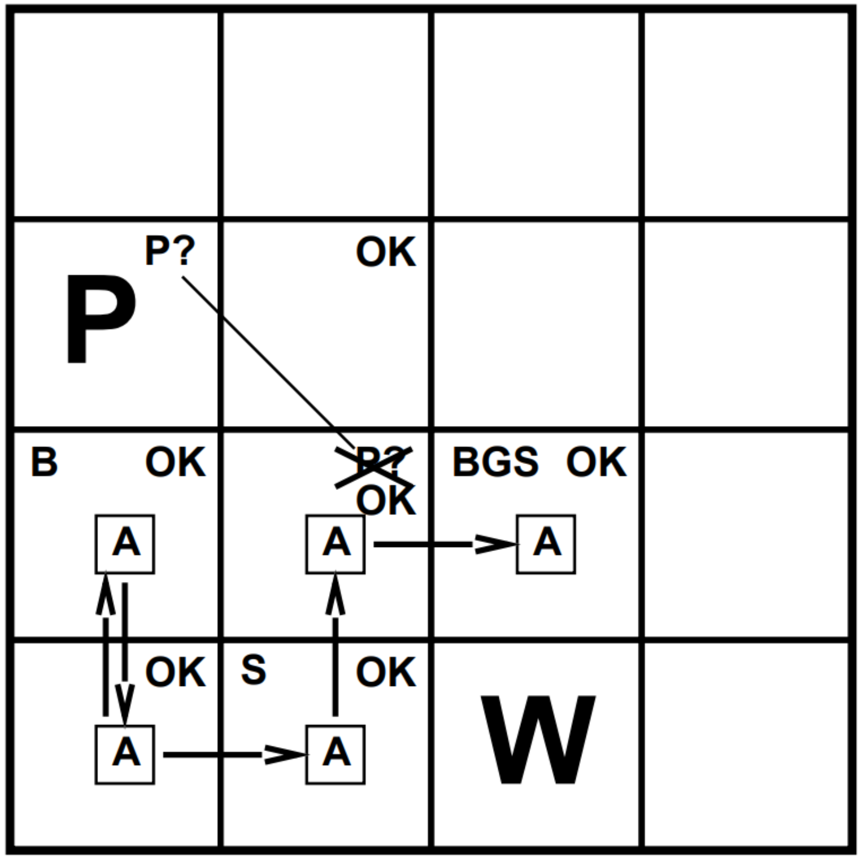
Exploring a wumpus world



Exploring a wumpus world



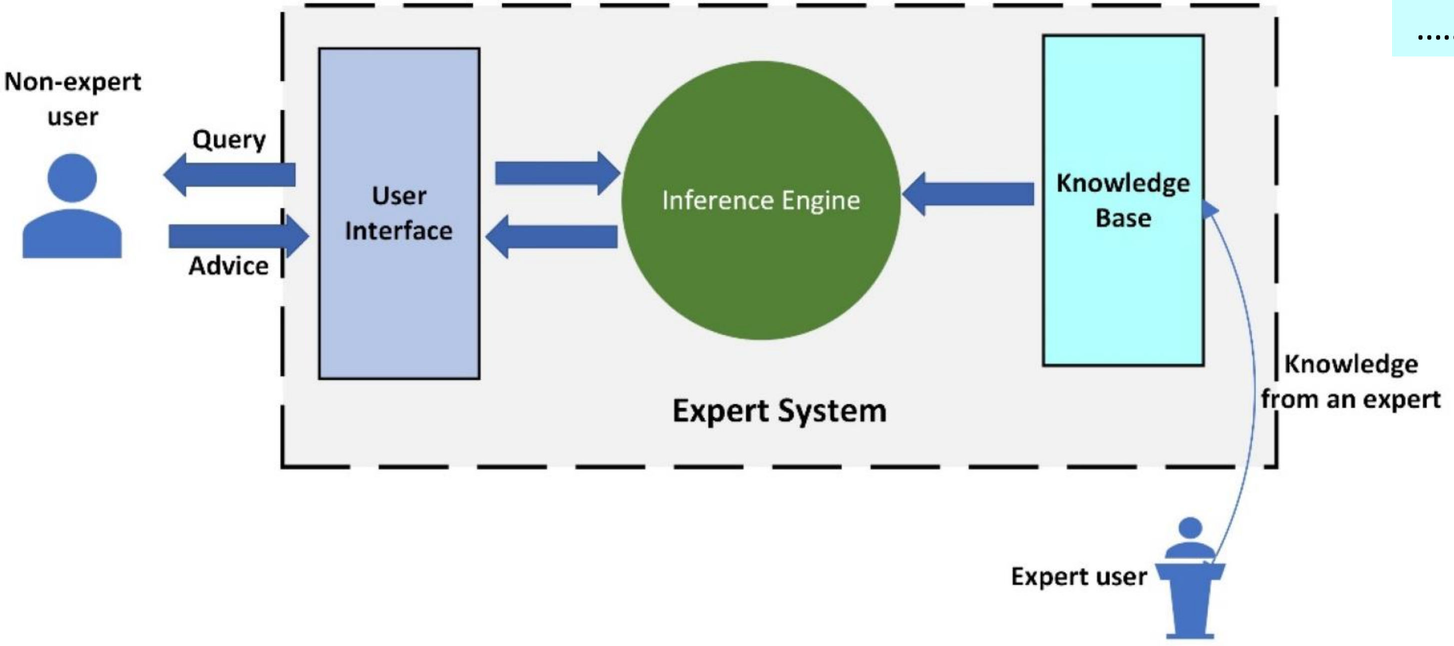
Exploring a wumpus world



Systems with Logical Reasoning

- Knowledge base
  - Consists of some prior knowledge
- Inference engine
  - Derive new knowledge or make some claims
- User Interaction
  - **Tell** information
  - **Ask** question

Example: Expert System



Knowledge base

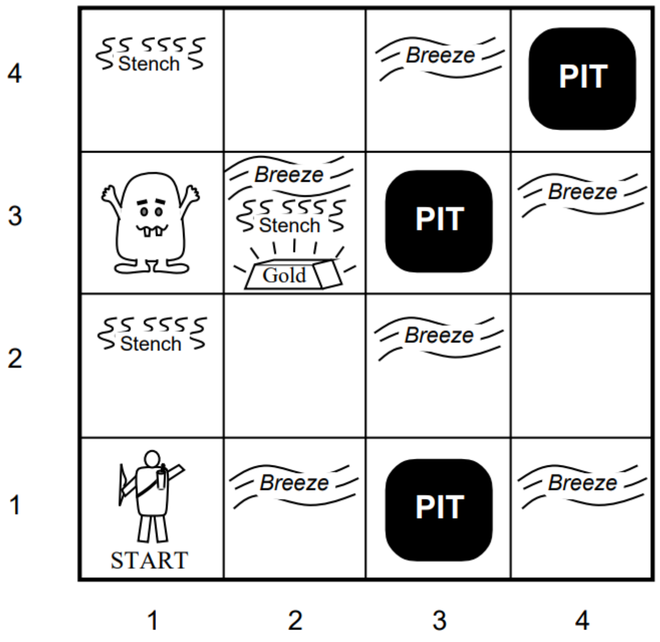
If **has_hair**, then **mammal**.
If **mammal** and **has_hooves**, then **ungulate**.
If **has_feathers**, then **bird**.
If **mammal** and **carnivore** and **has_dark_spots**, then **cheetah**.
If **mammal** and **carnivore** and **has_black_stripes**, then **tiger**.
If **bird** and **does_not_fly** and **has_long_neck**, then **ostrich**.
……

User interaction

File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.6)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- go.
Does the animal have hair? yes.

Does the animal eat meat? |: no.

Does the animal have pointed teeth? |: no.

Does the animal have hooves? |: yes.

Does the animal have a long neck? |: yes.

Does the animal have long legs? |: yes.

I guess that the animal is: giraffe
true.

?-
```

Example: wumpus world



Knowledge base

Perceive stench if adjacent to wumpus

Perceive breeze if adjacent to pit

Perceive glitter if in the square of gold

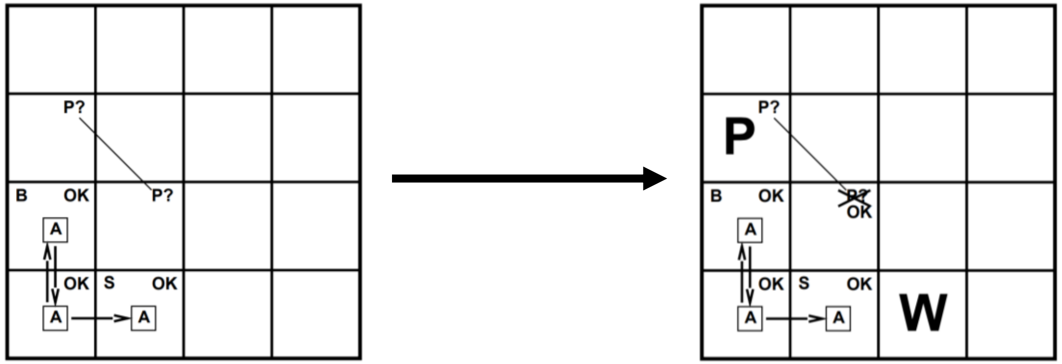
…

User interaction

Tell the logic system whether stench, breeze, glitter is perceived

Ask for the next action

Inference Engine



Ingredients of Propositional Logic

Sentence

Knowledge base consists of “sentences”

Inference algorithm derives new “sentences” and add them to the knowledge base

Example:

KB = { “Rain→Wet”, “Rain” }

Inference algorithm derives a new sentence “Wet” based on KB

Now KB becomes

KB = {“Rain→Wet”, “Rain”, “Wet” }

Ingredients of Logic – Syntax

Define what are valid sentences.

E.g., syntax in **python**:

“ for x in range(10): ”	Valid	
“ for x range(10): ”	Invalid	(the python interpreter cannot understand)

E.g. syntax in **math**:

“ $x + y = 5$”	Valid
“ $x\ 5 = y +$ ”	Invalid

Ingredients of Logic – Syntax

Syntax in **propositional logic**:

- A proposition symbols X is a sentence
  (a propositional symbol is a Boolean variable)
- If α is a sentence then $\neg\alpha$ is a sentence
- If α and β are sentences then $\alpha \wedge \beta$ is a sentence
- If α and β are sentences then $\alpha \vee \beta$ is a sentence
- If α and β are sentences then $\alpha \Rightarrow \beta$ is a sentence
- If α and β are sentences then $\alpha \Leftrightarrow \beta$ is a sentence

The $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ symbols have no meaning here. Their meanings are specified by the "semantics" of logic (discussed next).

Elements of Logic – Semantics

Let's first define "models". A model is a configuration of the world.

In propositional logic, a model is an **assignment of truth values** to propositional symbols.

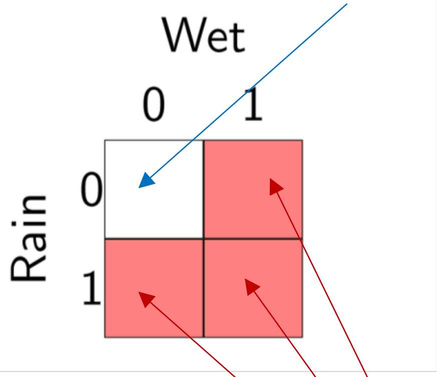
E.g., There are four possible models in the raining example:

		Wet 0	Wet 1
Rain	0		
Rain	1		

Elements of Logic – Semantics

$$f = \text{Rain} \lor \text{Wet}$$

models where the sentence f is false



P	Q	(P v Q)
T	T	T
T	F	T
F	T	T
F	F	F

models where the sentence f is true

Elements of Logic – Semantics

P	~P
T	F
F	T

P	Q	(P ^ Q)
T	T	T
T	F	F
F	T	F
F	F	F

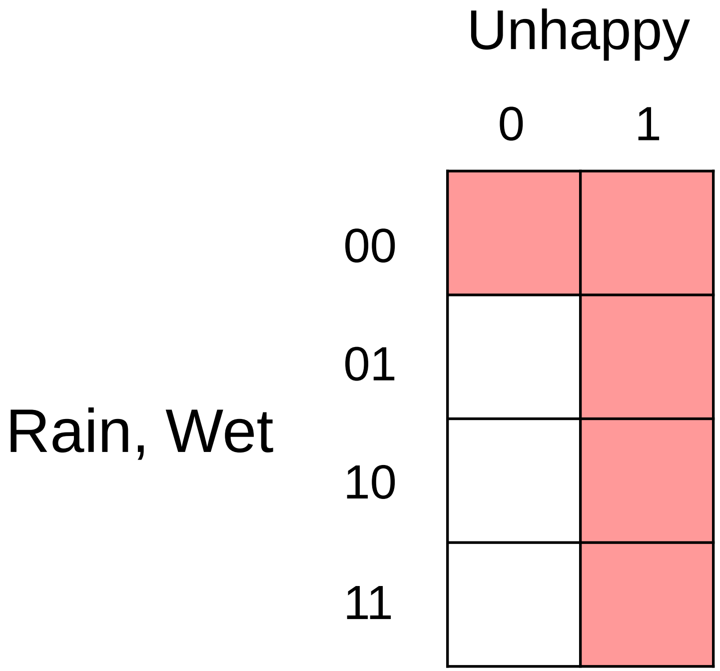
P	Q	(P v Q)
T	T	T
T	F	T
F	T	T
F	F	F

P	Q	(P =>Q)
T	T	T
T	F	F
F	T	T
F	F	T

P	Q	(P ⇔Q)
T	T	T
T	F	F
F	T	F
F	F	T

Elements of Logic – Semantics

f: (Rain ∨ Wet) ⇒ Unhappy

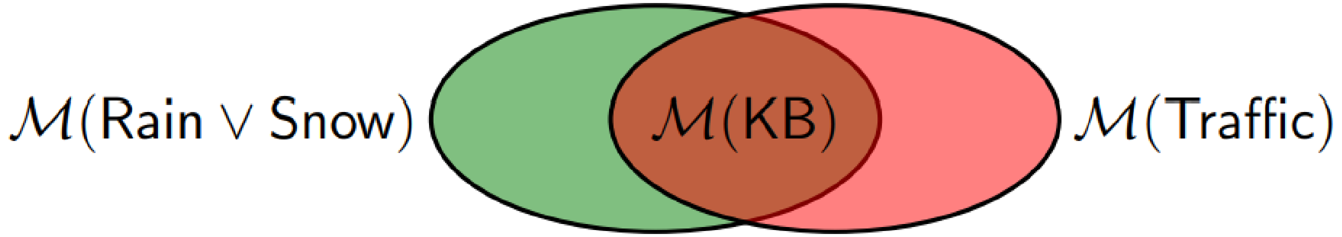


$\mathcal{M}(f)$: the set of models where sentence f is true.

Elements of Logic – Knowledge Base

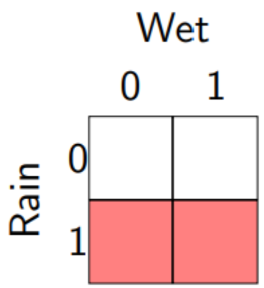
Knowledge base = a collection of sentences

Let $\mathsf{KB} = \{\mathsf{Rain} \vee \mathsf{Snow}, \mathsf{Traffic}\}$.

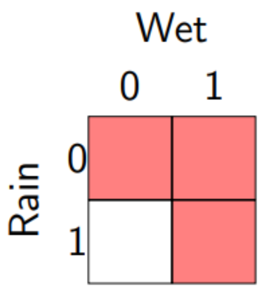


Elements of Logic – Knowledge Base

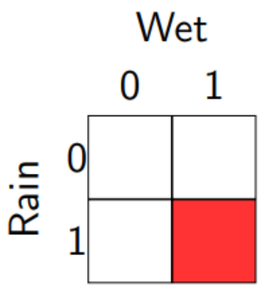
$\mathcal{M}(\text{Rain})$



$\mathcal{M}(\text{Rain} \rightarrow \text{Wet})$



$\mathcal{M}(\{\text{Rain}, \text{Rain} \rightarrow \text{Wet}\})$



Adding more formulas to the knowledge base:

$$\text{KB} \rightarrow \text{KB} \cup \{f\}$$

Shrinks the set of models:

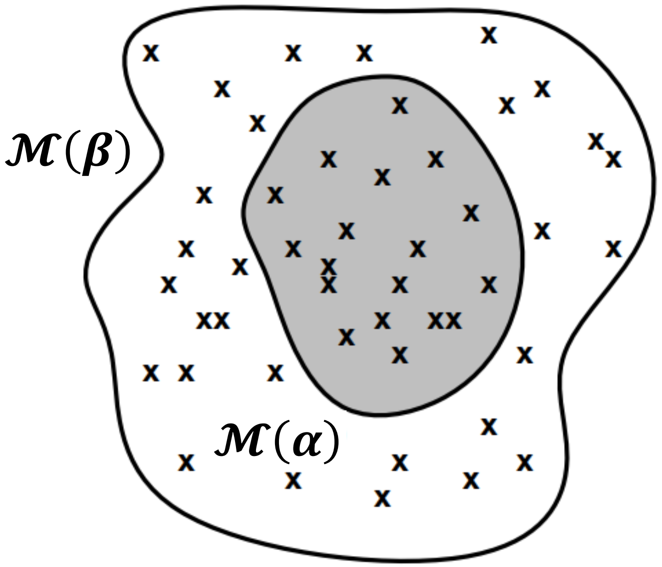
$$\mathcal{M}(\text{KB}) \rightarrow \mathcal{M}(\text{KB}) \cap \mathcal{M}(f)$$

Recap: Propositional Logic

- **Sentence:** propositional symbols, or their negations ($\neg$), or their combinations through $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$.
- **Models:** An assignment of truth values to propositional symbols.
- **Knowledge base:** a set of sentences
- $\mathcal{M}(f)$: the set of models where sentence f is true.

Entailment

- Sentence α **entails** sentence β means that (in high level) sentence β follows logically from sentence α
- Denoted as $\alpha \vDash \beta$
- $\alpha \vDash \beta$ if and only if $\mathcal{M}(\alpha) \subset \mathcal{M}(\beta)$
- **Example:** Rain ∧ Snow ⊨ Snow



Inference Algorithms

- Given KB, the algorithm decides whether sentence α can be entailed.
  - KB $\vDash \alpha$?
- Soundness (correctness)
  - The algorithm only say yes when α is entailed by KB.
- Completeness
  - For any α that KB entails, the algorithm says yes.

A (Simple) Inference Algorithm: Model Checking

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
    $P \leftarrow$ FIRST($symbols$)
    $rest \leftarrow$ REST($symbols$)
    return (TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P = true\}$)
      and
      TT-CHECK-ALL($KB, \alpha, rest, model \cup \{P = false\}$))

Theorem Proving

Idea: Instead of checking all models, will just perform manipulations on the sentence level.