Modul - Introduction to AI (AI1)

Bachelor Programme AAI

07 - Knowledge Base

Prof. Dr. Marcel Tilly

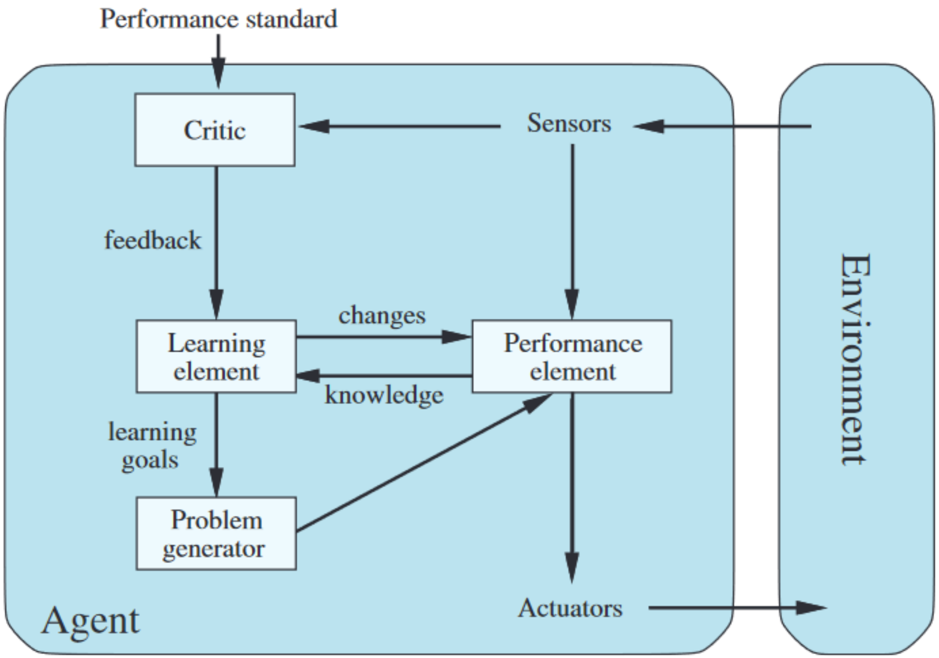
Faculty of Computer Science, Cloud Computing

Goals (formal)

- Students know the concept knowledge and knowledge bases.
- Students understand propositional logics.
- Students know about entailment.
- Students can use logics for reasoning.

RECAP: General learning Agent

A general learning agent. The “performance element” box represents what we have previously considered to be the whole agent program. Now, the “learning element” box gets to modify that program to improve its performance.



RECAP: Agent

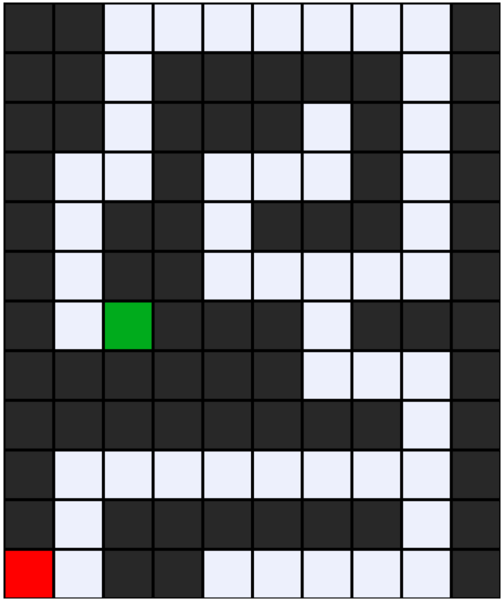
Technische Hochschule Rosenheim

- **Agent**: entity that perceives its environment and acts upon that environment
- **State**: a configuration of the agent and its environment
- **Initial State**: the state in which the agent begins
- **Actions**: choices that can be made in a state
- **Transition Model**: RESULT(s, a) returns the state resulting from performing action a in state s

RECAP: Agent

- **Agent**: entity that perceives its environment and acts upon that environment
- **State**: a configuration of the agent and its environment
- **Initial State**: the state in which the agent begins
- **Actions**: choices that can be made in a state
- **Transition Model**: RESULT(s, a) returns the state resulting from performing action a in state s
- **State Space**: the set of all states reachable from the initial state by any sequence of actions
- **Goal Test**: way to determine whether a given state is a goal state
- **Path Cost**: numerical cost associated with a given path

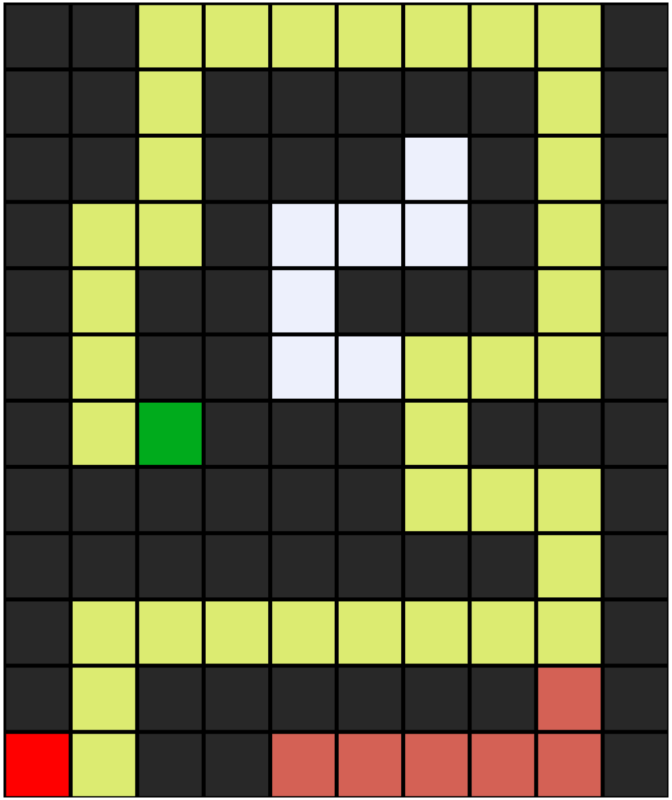
RECAP: Maze



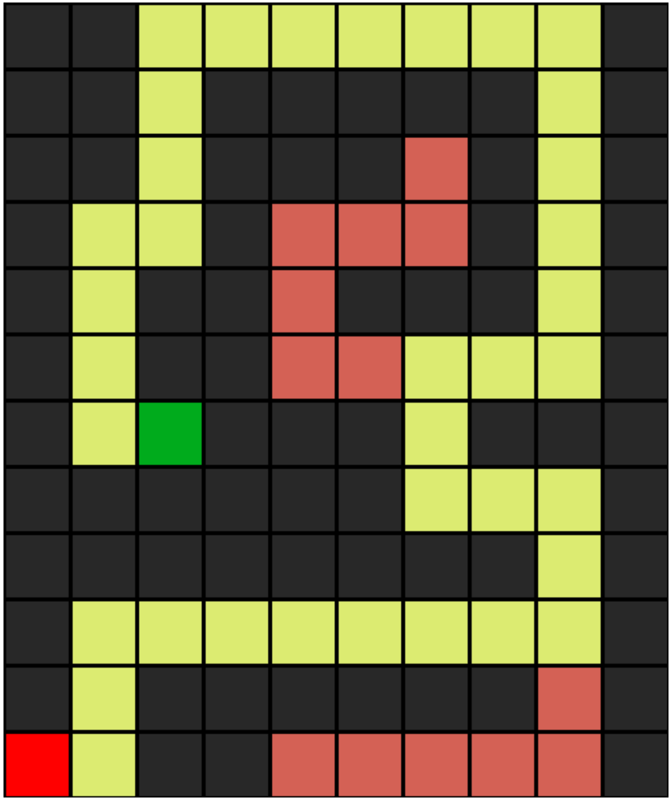
Depth First Search?

Breadth First Search?

Maze - DSF



Maze - BFS



Task 1: Uniformed vs. Informed Search

- **Uninformed Search**: search strategy that uses no problem- specific knowledge
- **Informed Search**: search strategy that uses problem-specific knowledge to find solutions more efficiently

Give me some search strategies for both and explain!

RECAP: Environments

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

taken from http://aima.cs.berkeley.edu/figures.pdf

RECAP: PEAS

In order to specify a scenario in which an agent performs a certain task, we need to define:

- a **P**erformance measure;
- an **E**nvironment;
- a set of **A**ctuators;
- a set of **S**ensors;

This is called the PEAS description of the agent/task.

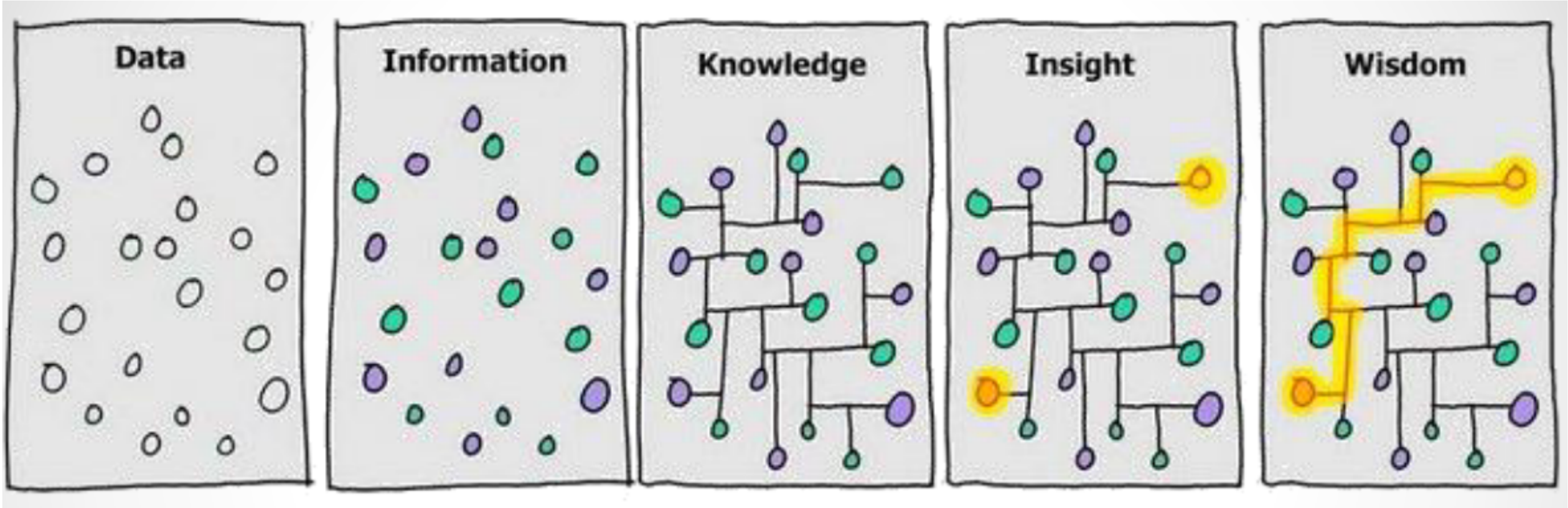
Once the PEAS description has been given, we're ready to define the Agent function f.

Knowledge Based Agents

Technische Hochschule Rosenheim

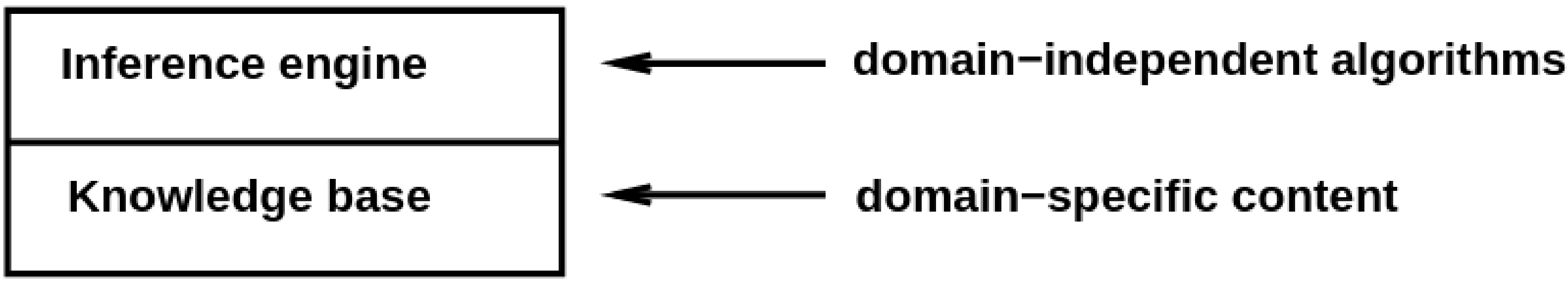
- Representation
  - How is the knowledge stored by the agent?
  - Procedural vs. Declarative
- Reasoning
  - How is the knowledge used to solve a problem?
  - How is the knowledge used to generate more knowledge?
- Generic Functions
  - **TELL** : add a fact to the knowledge base
  - **ASK** : (get next action based on info in KB)

Data(base) vs Knowledge(base)



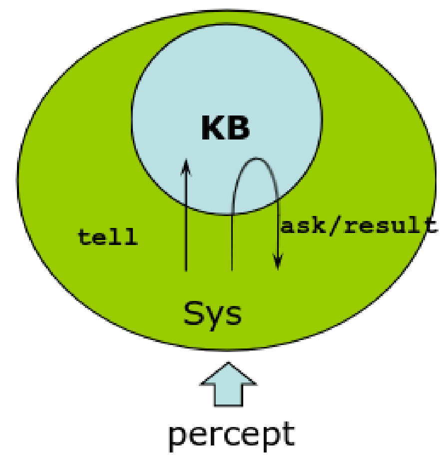
- **Database**: flat data representation (tabular)
- **Knowledge base**: “The ideal representation for a knowledge base is an object model (ontology) with classes, subclasses, and instances.” -- *Wikipedia*

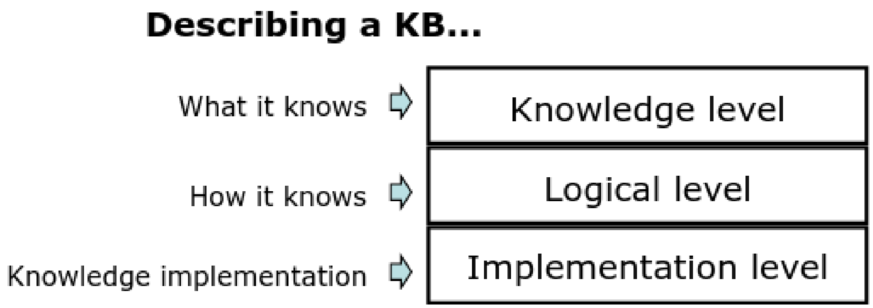
Knowledge Bases



- 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 follow from the *knowledge base*

Knowledge base System





Example: Wumpus World

- Game is played in a MxN grid
- One player, one wumpus, one or more pits
- Goal: find gold while avoiding wumpus and pits
- Percepts:
    - Glitter (gold is in this square)
    - Stench (wumpus is within 1 square N,E,S,W)
    - Breeze (pit is within 1 square N, E, S, W)

	stench		stench
	Glitter [gold]	stench breeze	
start	breeze	pit	breeze

Wumpus Environement

	n/y	Comment
Observable		
Deterministic		
Episodic		
Static		
Discrete		
Single agent		

Wumpus Environement

	n/y	Comment
Observable	No	only local perception
Deterministic	Yes	outcomeofactionexactlyspecified
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

Example: Reasoning

- If the player is in square (1, 0) and the percept is breeze, then there must be a pit in (0,0) or a pit in (2,0) or a pit in (1,1).
- If the player is in (0,0) [and still alive], there is not a pit in (0,0).
- If there is no breeze in (0,0), there is no pit in (0,1)
- If there is also no breeze in (0,1) then there is no pit in (1,1).
- Therefore, there must be a pit in (2,0)

	stench		stench
	Glitter [gold]	stench breeze	
start	breeze	pit	breeze

Formalizing Reasoning

- Information is represented in *sentences*, which must have correct syntax:
    - (1 + 2) *7 = 21 vs. 2) + 7 =* (1 21
- The meaning of a sentence (*semantics*) defines its truth in each *model* (possible world)
- One sentence *entails* another sentence if the second one follows logically from the first
    - i.e. every model that has the first true, also has the second true.
- *Inference* is the process of deriving a specific sentence from a KB (where the sentence must be entailed by the KB)

Desirable properties

Technische Hochschule Rosenheim

Desirable properties of an inference algorithm:

- Soundness
  - Only sentences that are entailed by a KB will be derived by the inference algorithm
- Completeness
  - Every sentence that is entailed by a KB will be derived by the inference algorithm (eventually)
- If these properties are true, then every sentence derived from a true KB will be true in the world.
- All reasoning can be done in a model, not the world!

Propositional Logic

- The syntax of *propositional logic* is made up of *propositions* and *connectives*.
- A statement in some language that can be evaluated to either *true* or *false* (but it cannot be both).

Example propositions:

- It is raining.
- China is in Asia.
- Berlin is the capital of Germany.

Not propositions:

- Where are you?

> An easy way to determine whether or not a statement is a proposition is to see if you can prefix it with “it is true that” or “it is false that”; and if it subsequently still makes sense.

Propositions

- Propositions are represented using the propositional variables **p, q, r etc.**
- The previous examples of propositions are all atomic.
- We can combine these atomic propositions to form compound propositions by using connectives.
    - Conjunction (and) Λ
    - Disjunction (or) v
    - Negation (not) $\neg$
    - Implication (if..then): $\Rightarrow$
    - Equivalence (if and only if): $\Leftrightarrow$

Propositional Logic

Sentence	Name	Notation
not P	negation (not)	$\neg P$
P and Q	conjunction (and)	$P \land Q$
P or Q	disjunction (or)	$P \lor Q$
if P then Q	implication (implies)	$P \Rightarrow Q$
P exactly if Q	equivalence (if & inly if)	a$P \Leftrightarrow Q$

- If there is a pit at [1,1], there is a breeze at [1,0]: $P11 \Rightarrow B10$
- There is a breeze at [2,2], if and only if there is a pit in the neighborhood:
  $B22 \Leftrightarrow (P21 \lor P23 \lor P12 \lor P32)$
- There is no breeze at [2,2]: $\neg B2$

Entailment

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

We write: $KB \models \alpha$

> Note: Entailment is a relationship between sentences (i.e.,syntax) that is based on semantics.

- Entailment means that one thing follows from another.

Examples:

- Wumpus world: Situation after detecting nothing in [1,1], moving right, breeze in [2,1]
- The KB containing "the shirt is green" and "the shirt is striped" entails "the shirt is green or the shirt is striped"
- x+y=4 entails 4=x+y

Inference

Inference is the process of deriving new sentences from old ones.

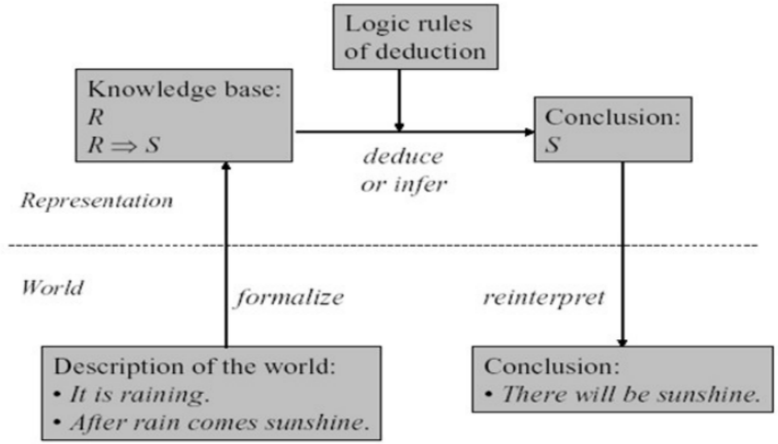
Example

- P: It is a Tuesday.
- Q: It is raining.
- R: Harry will go for a run.
- KB: $(P \land \neg Q) \rightarrow R$
- Inference: R

Inference Algorithm via entailment: **Does KB $\models \alpha$?**

Model Checking

- To determine if KB $\models \alpha$:
  - Enumerate all possible models.
  - If in every model where KB is true, α is true, then KB entails α.
  - Otherwise, KB does not entail α.



Model Checking

P: It is a Tuesday. Q: It is raining. R: Harry will go for a run. KB: $(P \land \neg Q) \rightarrow R$ P $\neg Q$ Query: R

P	Q	R	KB
false	false	false	false
false	false	true	false
false	true	false	false
false	true	true	false
true	false	false	false
true	false	true	true
true	true	false	false
true	true	true	false

Propositional Inference: Enumeration Method

Example

$$\alpha = A \vee B \qquad KB = (A \vee C) \wedge (B \vee \neg C)$$

Checking that $KB \models \alpha$ ✓

A	B	C	$A \vee C$	$B \vee \neg C$	KB	α	
false	*false*	*false*	f	t	f	f	
false	*false*	*true*	t	f	f	f	
false	*true*	*false*	f	t	f	t	
false	*true*	*true*	t	t	t	t	✓
true	*false*	*false*	t	t	t	t	✓
true	*false*	*true*	t	f	f	t	
true	*true*	*false*	t	t	t	t	✓
true	*true*	*true*	t	t	t	t	✓

Propositional Inference: Enumeration Method

Technische Hochschule Rosenheim

Example

$$\alpha = A \vee B \qquad KB = (A \vee C) \wedge (B \vee \neg C)$$

Checking that $KB \models \alpha$

A	B	C	$A \vee C$	$B \vee \neg C$	KB	α
false	*false*	*false*	*false*			
false	*false*	*true*	*true*			
false	*true*	*false*	*false*			
false	*true*	*true*	*true*			
true	*false*	*false*	*true*			
true	*false*	*true*	*true*			
true	*true*	*false*	*true*			
true	*true*	*true*	*true*			

Propositional Inference: Enumeration Method

Example

$$\alpha = A \vee B \qquad KB = (A \vee C) \wedge (B \vee \neg C)$$

Checking that $KB \models \alpha$

A	B	C	$A \vee C$	$B \vee \neg C$	KB	α
false	*false*	*false*	*false*	*true*		
false	*false*	*true*	*true*	*false*		
false	*true*	*false*	*false*	*true*		
false	*true*	*true*	*true*	*true*		
true	*false*	*false*	*true*	*true*		
true	*false*	*true*	*true*	*false*		
true	*true*	*false*	*true*	*true*		
true	*true*	*true*	*true*	*true*		

Propositional Inference: Enumeration Method

Example

$$\alpha = A \vee B \qquad KB = (A \vee C) \wedge (B \vee \neg C)$$

Checking that $KB \models \alpha$

A	B	C	$A \vee C$	$B \vee \neg C$	KB	α
false	*false*	*false*	*false*	*true*	*false*	
false	*false*	*true*	*true*	*false*	*false*	
false	*true*	*false*	*false*	*true*	*false*	
false	*true*	*true*	*true*	*true*	*true*	
true	*false*	*false*	*true*	*true*	*true*	
true	*false*	*true*	*true*	*false*	*false*	
true	*true*	*false*	*true*	*true*	*true*	
true	*true*	*true*	*true*	*true*	*true*	

Propositional Inference: Enumeration Method

Example

$$\alpha = A \lor B \qquad KB = (A \lor C) \land (B \lor \neg C)$$

Checking that $KB \models \alpha$

A	B	C	$A \lor C$	$B \lor \neg C$	KB	α
false	*false*	*false*	*false*	*true*	*false*	*false*
false	*false*	*true*	*true*	*false*	*false*	*false*
false	*true*	*false*	*false*	*true*	*false*	*true*
false	*true*	*true*	*true*	*true*	*true*	*true*
true	*false*	*false*	*true*	*true*	*true*	*true*
true	*false*	*true*	*true*	*false*	*false*	*true*
true	*true*	*false*	*true*	*true*	*true*	*true*
true	*true*	*true*	*true*	*true*	*true*	*true*

Propositional Inference: Enumeration Method

Example

$$\alpha = A \vee B \qquad KB = (A \vee C) \wedge (B \vee \neg C)$$

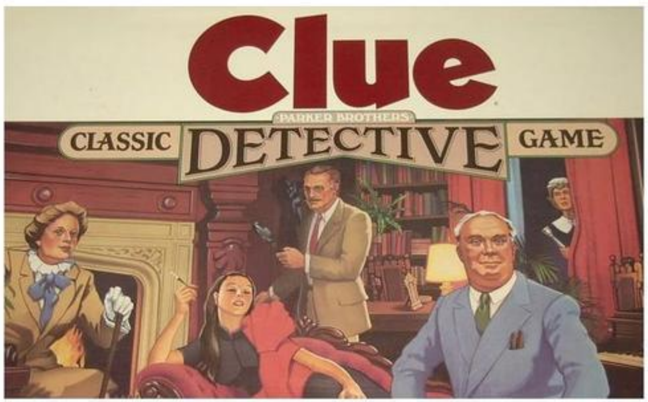
Checking that $KB \models \alpha$

A	B	C	$A \vee C$	$B \vee \neg C$	KB	α
false	*false*	*false*	*false*	*true*	*false*	*false*
false	*false*	*true*	*true*	*false*	*false*	*false*
false	*true*	*false*	*false*	*true*	*false*	*true*
false	*true*	*true*	*true*	*true*	*true*	*true*
true	*false*	*false*	*true*	*true*	*true*	*true*
true	*false*	*true*	*true*	*false*	*false*	*true*
true	*true*	*false*	*true*	*true*	*true*	*true*
true	*true*	*true*	*true*	*true*	*true*	*true*

Note

Table has 2^n rows for n symbols

Clue



Propositional Symbols

Person	Rooms	Weapons
Col. Mustard	Ballroom	Knife
Prof. Plum	Kitchen	Revolver
Ms. Scarlet	Library	Wrench

Clue

Knowledge Base:

```
(mustard ∨ plum ∨ scarlet)

(ballroom ∨ kitchen ∨ library)

(knife ∨ revolver ∨ wrench)

¬plum

¬mustard ∨ ¬library ∨ ¬revolver
```

Logic Puzzles

Technische Hochschule Rosenheim

- Gilderoy, Minerva, Pomona and Horace each belong to a different one of the four houses: Gryffindor, Hufflepuff, Ravenclaw, and Slytherin House.
- Gilderoy belongs to Gryffindor or Ravenclaw.
- Pomona does not belong in Slytherin.
- Minerva belongs to Gryffindor.

Logic Puzzles

Propositional Symbols

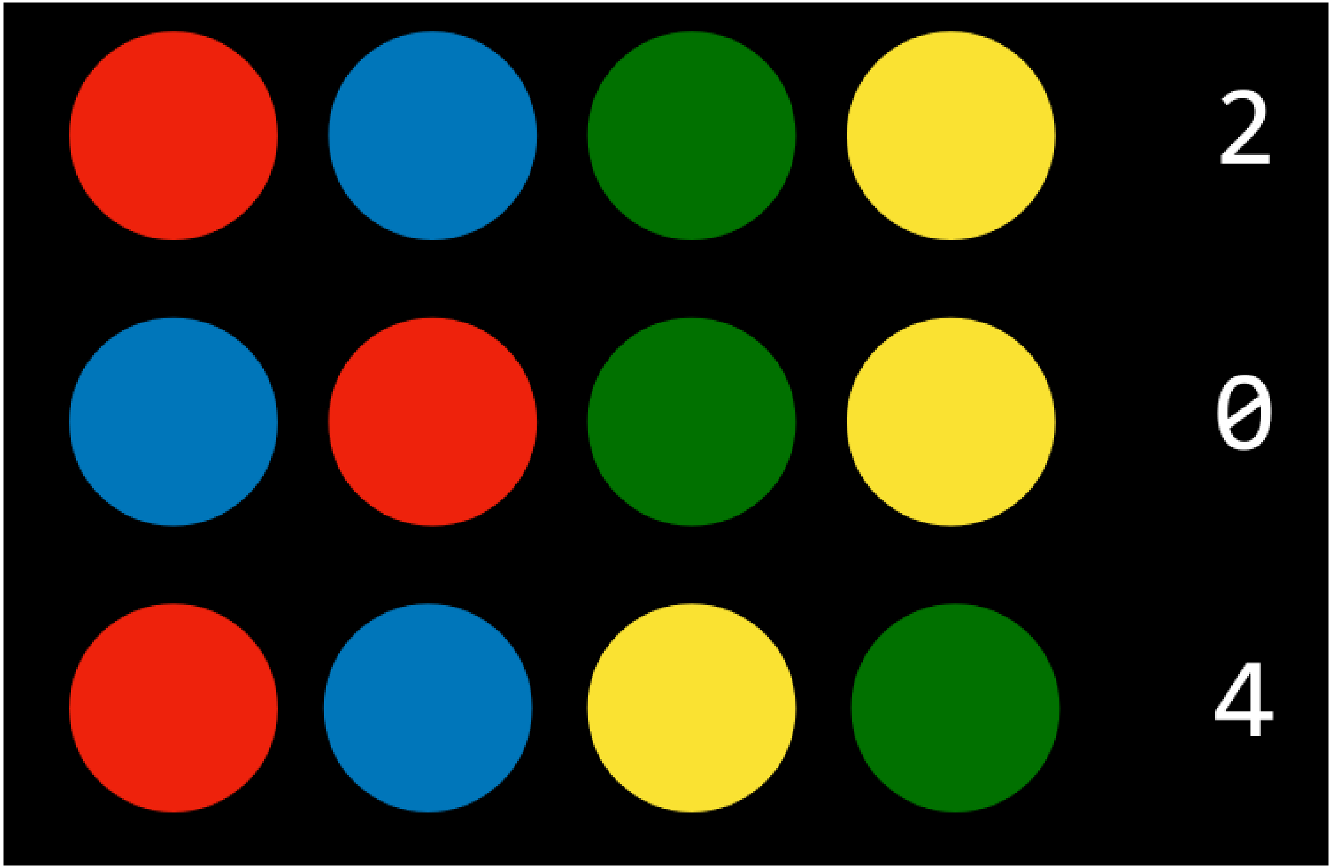
```
GilderoyGryffindor    PomonaGryffindor
GilderoyHufflepuff    PomonaHufflepuff
GilderoyRavenclaw     PomonaRavenclaw
GilderoySlytherin     PomonaSlytherin

MinervaGryffindor     HoraceGryffindor
MinervaHufflepuff     HoraceHufflepuff
MinervaRavenclaw      HoraceRavenclaw
MinervaSlytherin      HoraceSlytherin
```

As logic:

```
(GilderoyGryffindor ∨ GilderoyRavenclaw)
(PomonaSlytherin → ¬PomonaHufflepuff)
(MinervaRavenclaw → ¬GilderoyRavenclaw)
```

Mastermind



Limitations

Entailment KB $\models \alpha$

- What is the computational complexity of the truth table approach?

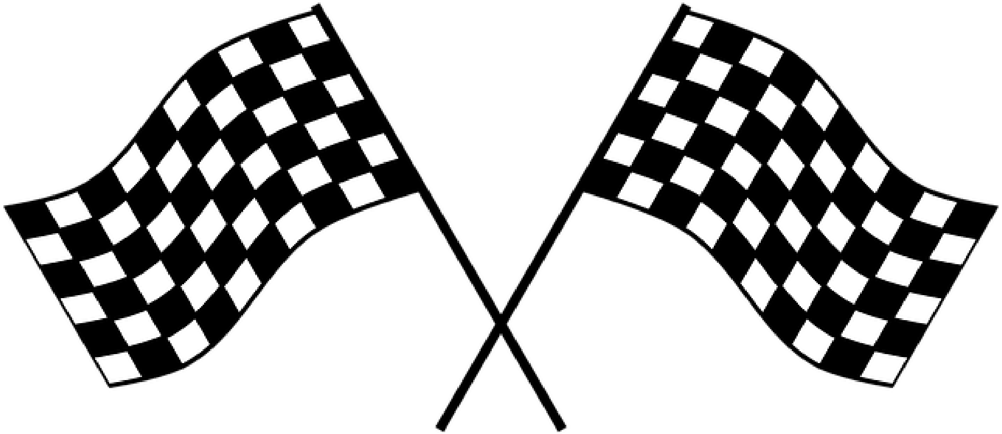
Exponential in the number of the propositional symbols n

$$2^n$$

We check all combinations!

Summary

- Knowledge and Knowledge Base
- Propositional Logic
- Reasoning



More Links

- CS50's Introduction to Artificial Intelligence with Python:
  https://cs50.harvard.edu/ai/2020/weeks/1/