

Knowledge Representation

Introduction to Artificial Intelligence

G. Lakemeyer

Winter Term 2018/19

Rationally Thinking Agents

- So far the focus was on **rationally acting** agents. (Decision making is done implicitly via the evaluation function, i.e. the designer “thinks.”)
- Often rational action requires **rational thought** by the agent itself.
- Part of the world must be represented explicitly in a **Knowledge Base (KB)**:
 - KB contains sentences in a language with a **truth theory** (Logic) which we can interpret as **propositions** about the world.
 - The sentences, through their **form** alone, have a **causal effect** on the agent's behavior in correlation with the contents of the sentences.
- Ideally, interaction with a KB through **ASK** and **TELL**:
$$\text{ASK}(\text{KB}, \alpha) = \text{YES} \quad \text{iff } \alpha \text{ follows from KB}$$
$$\text{TELL}(\text{KB}, \alpha) = \text{KB}', \quad \text{so that } \alpha \text{ follows from KB}'$$

3 Levels

In **knowledge representation** one distinguishes 3 levels [Newell 1990]:

Knowledge Level: the most abstract level; addresses what is **known** by the KB. E.g.: automatic taxi driver knows that *Vaalser St* connects *Aachen* and *Vaals*.

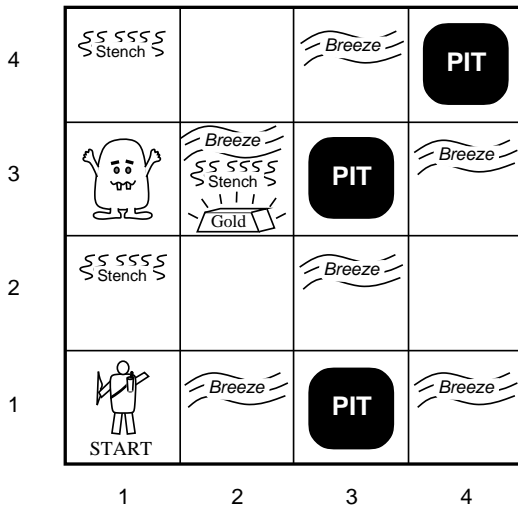
Symbolic Level: Encoding of the KB as sentences in a formal language: *Connects(Vaalser_St, Aachen, Vaals)*

Implementation Level: The internal representation of sentences.
Taxi-example:

- a string "Connects(Vaalser_St, Aachen, Vaals)"
- a bit in in a 3-D-matrix representing connections between places.

When ASK and TELL work correctly, it suffices to stay at the knowledge level.
Advantage: very nice user interface. A user has her own mental world model (propositions about the world) and simply tells that to the agent.

The Wumpus World 1



The Wumpus World 2

- In the square where the Wumpus is and those next to it there is **stench**.
- In the square next to a pit there is a **breeze**.
- In the square with the gold there is **glitter**.
- When hitting a wall, the agent receives a **bump**.
- When the **Wumpus dies**, its howl is heard everywhere.
- **Perceptions** are quintuples. [Stench, Breeze, Glitter, None, None] means that there is stench, breeze, and glitter, but there is no bump nor a howl. There is no location sensor.
- **Actions**: forward, turn right (90°), turn left (90°), grab an object, shoot (only one arrow), leave the cave (only from square (1,1)).
- The **agent dies**, if he falls into a pit or meets the live Wumpus.
- **Goal**: fetch the gold and leave the cave.
- **initial state**: agent in (1,1), 1 Wumpus, 1 pile of gold and 3 pits, randomly distributed.

The Wumpus World 3

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK			
1,1	2,1	3,1	4,1
OK	OK		

(a)

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

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK	P?		
1,1	2,1	3,1	4,1
V	OK	P?	

(b)

1,4	2,4	3,4	4,4
1,3	W!	3,3	4,3
1,2		3,2	4,2
S	OK	OK	
1,1	2,1	3,1	4,1
V	B	P!	
OK	V	OK	

(a)

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

1,4	2,4	3,4	4,4
	P?		
1,3	W!	3,3	4,3
	A	P?	
	S		
	G		
	B		
1,2		3,2	4,2
S			
V	V		
OK	OK		
1,1	2,1	3,1	4,1
V	B	P!	
OK	V	OK	

(b)

A Declarative Language

We need a precise declarative language for representation and reasoning.

- **declarative**: System believes P iff P is thought to be **true**.
(One cannot believe P without having an idea what it means for P to be true (satisfied) in the world.)
- **precise**: need to know
 - which strings are **sentences**;
 - what it means for a sentence to be **true** (without having to explicitly specify for every sentence whether or not it is true).

Here: the language of **first-order logic** (FOL)

Note: this is only one of many languages which satisfy the above criteria.

Alphabet

Logical symbols:

- Delimiter : $() ,$
- Operators: $\neg, \wedge, \vee, \forall, \exists, =$
- Variables: $x, x_1, x_2, \dots, x', x'', \dots, y, \dots, z, \dots$
fixed meaning and use (like keywords in a programming language).

Nonlogical symbols

- Predicate symbols (like Friend)
- Function symbols (like bestFriendOf)
 - 0-ary predicates: often called **propositional variables**
 - 0-ary function symbols: **constants**

the meaning is application dependent (like identifiers in a PL)

Note: $=$ is not considered a predicate!

Grammar

Expressions: terms and formulas (wffs).

Terms:

- 1 Every variable is a term.
- 2 If t_1, t_2, \dots, t_n are terms and f is an n -ary function symbol, then $f(t_1, t_2, \dots, t_n)$ is a term.

Atomic wffs:

- 1 If t_1, t_2, \dots, t_n are terms and P is an n -ary predicate symbol, then $P(t_1, t_2, \dots, t_n)$ is an atomic formula.
- 2 If t_1 and t_2 are terms, then $t_1 = t_2$ is an atomic formula.

Formulas:

- 1 Every atomic wff is a wff.
- 2 For wffs α, β and variable x , $\neg\alpha, (\alpha \wedge \beta), (\alpha \vee \beta), \exists x\alpha, \forall x\alpha$ are wffs.

Special Case: Propositional Logic

Propositional logic as a sublanguage of FOL:

- No terms;
- atomic wffs: only 0-ary predicate symbols (propositional variables);
- neither variables nor quantifiers;
- **Example:** $(p \wedge \neg(q \vee r))$.

Notation

We often omit parentheses or use other kinds ($\{, \}, [,]$) for better readability.

Abbreviations: $(\alpha \supset \beta)$ instead of $(\neg\alpha \vee \beta)$
 $(\alpha \equiv \beta)$ instead of $(\alpha \supset \beta) \wedge (\beta \supset \alpha)$

Nonlogical symbols:

Predicates: Person, Nice, OlderThan

Functions: fatherOf, successor, janeDoe

Lexical binding of variables:

$P(x)$	\wedge	$\exists x[P(x) \vee Q(x)]$	
free		bound	occurrence of x

Sentences = wffs without free variables.

Substitution: $\alpha[x/t]$ means α with all free occ. of x replaced by t .

Semantics

How are sentences interpreted?

- What do sentences tell us about what is **true** in the world?
- What does it mean to **know/believe** a sentence?

Without answers to these questions knowledge representation is impossible!

Problem: The semantics of sentences takes us outside the language because of the nonlogical symbols.

Therefore it is important to make precise the dependence of interpretations from the nonlogical symbols.

Logical interpretations: Specification of how to understand predicates and functions. Can be very complex!

Examples: *FavoriteMovie*, *SoccerCoach*, *DemocraticCountry*.

Interpretations (informal)

There are **objects**, some of which satisfy a predicate P , others don't. Each interpretation determines the **extension** of P .

Each interpretation determines a mapping from objects to objects for each function symbol.

Functions are single-valued and defined everywhere.

Note:

That's all one needs to know about the nonlogical symbols to determine which sentences are true and which are false.

In other words, after specifying

- which objects there are,
- which objects satisfy P and
- which mapping corresponds to f ,

it is possible to determine the truth value of all sentences.

Interpretations (formal)

$I = \langle D, \Phi \rangle$ is an interpretation where

- D is the **universe of discourse** or **domain**
can be **any** non-empty set (mathematical objects, but also students, tables, sentences, cars, etc.)
- Φ is an **interpretation function**

where P is an n -ary predicate symbol,

$\Phi(P) \subseteq D \times D \times \dots \times D$ an n -ary relation over D

where f is an n -ary function symbol,

$\Phi(f) \in [D \times D \times \dots \times D \rightarrow D]$ an n -ary function over D

In **propositional logic**:

$\Phi(P) = \{\}$ (FALSE) or $\Phi(P) = \{\langle \rangle\}$ (TRUE)

Simplification: $I = \Phi \in [\text{prop. variables} \rightarrow \{\text{TRUE}, \text{FALSE}\}]$

Denotation

The denotation of a term is the element of D assigned by $I = \langle D, \Phi \rangle$.

Notation: $I \llbracket t \rrbracket$

For terms with free variables the denotation depends on the assignment to the variables as well:

Notation: $I, \nu \llbracket t \rrbracket$, where $\nu \in [\text{Variables} \rightarrow D]$ defines a variable map.

Rules:

- ① $I, \nu \llbracket x \rrbracket = \nu(x)$ (for every x)
- ② $I, \nu \llbracket f(t_1, \dots, t_n) \rrbracket = H(d_1, \dots, d_n)$,
where $H = \Phi(f)$ and $d_i = I, \nu \llbracket t_i \rrbracket$ (recursively)

Satisfaction

For a given I the truth value of a wff depends also on the assignment to the variables.

$I, \nu \models \alpha$ stands for “ α is satisfied by I and ν .”

We write $I \models \alpha$ if α is a sentence and $I \models S$ if S is a set of sentences.

Rules:

- ① $I, \nu \models P(t_1, \dots, t_n)$ iff $\langle d_1, \dots, d_n \rangle \in R$,
where $R = \Phi(P)$ and $d_i = I, \nu \models t_i$;
- ② $I, \nu \models (t_1 = t_2)$ iff $I, \nu \models t_1$ equals $I, \nu \models t_2$;
- ③ $I, \nu \models \neg \alpha$ iff $I, \nu \not\models \alpha$;
- ④ $I, \nu \models (\alpha \wedge \beta)$ iff $I, \nu \models \alpha$ and $I, \nu \models \beta$;
- ⑤ $I, \nu \models (\alpha \vee \beta)$ iff $I, \nu \models \alpha$ or $I, \nu \models \beta$;
- ⑥ $I, \nu \models \exists x \alpha$ iff for some $d \in D$, $I, \nu_d^x \models \alpha$,
where ν_d^x is like ν except that $\nu_d^x(x) = d$;
- ⑦ $I, \nu \models \forall x \alpha$ iff for all $d \in D$, $I, \nu_d^x \models \alpha$;

In **propositional logic**: $I \models p$ iff $\Phi(p) \neq \{\}$ and else as above.

Logical Consequence

Rules tell us how the truth value of a sentence depends on the meaning of the nonlogical symbols.

But not all connections between sentences depend on them.

Example: if α is satisfied by I , then $\neg(\beta \wedge \neg\alpha)$ is true independent of why α is true and what β is!

Logical Consequence:

S implies α or α is a **logical consequence** of S :

$$S \models \alpha \text{ iff for all } I, \text{ if } I \models S \text{ then } I \models \alpha$$

In other words: for all I , $I \not\models S \cup \{\neg\alpha\}$, or, $S \cup \{\neg\alpha\}$ is **unsatisfiable**.

Special case: S is empty: $\models \alpha$ iff for all I , $I \models \alpha$. (α is **valid**.)

Note: $\{\alpha_1, \dots, \alpha_n\} \models \alpha$ iff $\models (\alpha_1 \wedge \dots \wedge \alpha_n) \supset \alpha$.

(reduces finite implication to validity)

Why Implication?

System does not have access to the interpretation of the nonlogical symbols as intended by the user!

With **implication** we know that α is true as long S is true in the intended interpretation.

If the world which the user envisions satisfies S , then it must also satisfy α . Other sentences may be true in the user's model of the world, but α need not be satisfied necessarily.

How about $Dog(fido) \models Mammal(fido)$??

No! Not a logical consequence. $\Phi(Dog) \not\subseteq \Phi(Mammal)$ for some Φ .

Central idea of KR:

such connections are explicitly represented in S :

$$\forall x [Dog(x) \supset Mammal(x)]$$

Then: $S \cup \{Dog(fido)\} \models Mammal(fido)$

Knowledge Bases

A knowledge base (KB) is a set of sentences

explicit representation of the believed sentences

(includes the assumed connections between the nonlogical symbols.)

$KB \models \alpha$: α is a consequence of one's beliefs.

- explicit knowledge: KB
- implicit knowledge: $\{\alpha \mid KB \models \alpha\}$

An Example

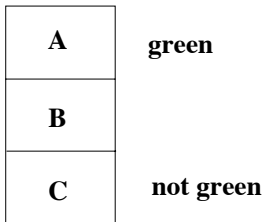
Often not trivial to extract implicit from explicit knowledge:

Three blocks are stacked on top of each other.

The top block is green.

The lowest block is not green.

There is no information about the color of the middle block.



Question: Is there a green block on top of a non-green block?

A Formalization

$$\begin{aligned} S &= \{On(a, b), On(b, c), Green(a), \neg Green(c)\} \\ \alpha &= \exists x \exists y [Green(x) \wedge \neg Green(y) \wedge On(x, y)] \end{aligned}$$

Claim: $S \models \alpha$

Proof: Let I be an arbitrary interpretation such that $I \models S$.

Case 1: $I \models Green(b)$.

Then $I \models Green(b) \wedge \neg Green(c) \wedge On(b, c)$ and thus $I \models \alpha$.

Case 2: $I \not\models Green(b)$.

Then $I \models \neg Green(b)$.

Hence $I \models Green(a) \wedge \neg Green(b) \wedge On(a, b)$ and thus $I \models \alpha$.

In any case: if $I \models S$, then $I \models \alpha$.

Therefore $S \models \alpha$ **QED**

Where is the Wumpus?

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

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

Knowledge about the current situation:

$[B = \text{breeze}, S = \text{stench}, B_{i,j} = \text{there is a breeze in } (i, j)]$

$$\begin{aligned}
 &\neg S_{1,1} \quad \neg B_{1,1} \\
 &\neg S_{2,1} \quad B_{2,1} \\
 &S_{1,2} \quad \neg B_{1,2}
 \end{aligned}$$

Invariant knowledge about the Wumpus and stench:

$$\begin{aligned}
 R_1 : & \neg S_{1,1} \supset \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1} \\
 R_2 : & \neg S_{2,1} \supset \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1} \\
 R_3 : & \neg S_{1,2} \supset \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,2} \wedge \neg W_{1,3} \\
 R_4 : & S_{1,2} \supset W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}
 \end{aligned}$$

...

Knowledge-Based Systems

Start with a large KB representing explicit knowledge (e.g. what it has been told explicitly.)

The behavior of the system should then depend on the **implicit** knowledge. This requires **inference methods**:

Deductive inference:

Process to compute the logical consequences of a KB.
given a KB and a query α , computes whether $KB \models \alpha$.

Process is **correct** if for every derivable α we have that $KB \models \alpha$.
does not allow plausible inferences, which hold only in intended interpretations.

Process is **complete** if every α for which $KB \models \alpha$ holds is derivable.
does not allow the process to abort computing difficult inferences!