Knowledge representation, principles, propositional logic

Jiří Kléma

Department of Computer Science,
Czech Technical University in Prague

http://cw.felk.cvut.cz/doku.php/courses/a4b33zui/start

Knowledge, inference and decision making – outline

- L1: knowledge and its representation (KR)
  - motivation examples, principles, propositional logic,
  - KR language, close relationship with consequent inference,
- L2: predicate logic, situation calculus, general planning
  - how to model change in classical logic, frame problem,
  - other traditional approaches to KR,
- L3: decision making under uncertainty
  - individual agent's decision in an uncertain world,
  - are people rational agents?
- L4: sequential decision making under uncertainty
  - how to maximize cumulative reward if decisions interact?
  - Markov decision process.

Motivation example 1 – integers

:: Let us compare two forms of integer representation – **Arabic** and **Roman**:

- Length of numbers
  - 1000 vs. M, 1997 vs. MCMXCVII, 100000 vs. MMMM...M,
  - average length (AL) of numbers,
  - redundancy (R), codes with 10 resp. 7 symbols,

System	Arabic		Roman	
Interval	AL[chars]	R[%]	AL[chars]	R[%]
1..1000	2.89	0.3	6	11.5
1..3000	3.63	2.6	7	5.5

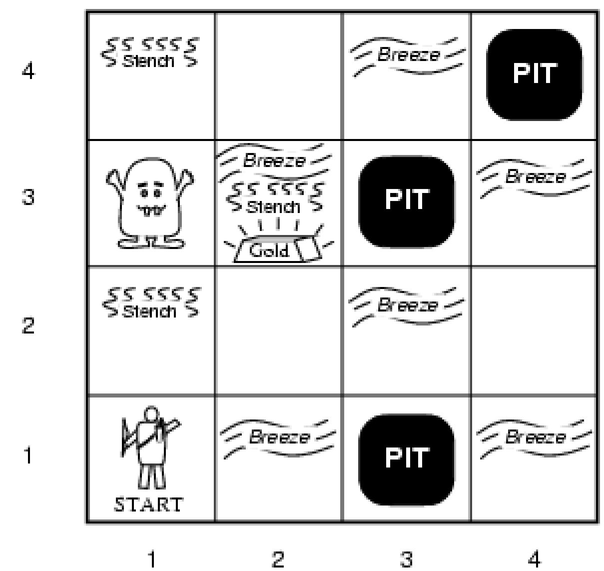
- Complexity of arithmetic operations, e.g. addition and division
  - Arabic numbers are positional (digits match), 0 symbol exists,
  - Roman numbers arithmetically improper (IV $\rightarrow$ IIII, L $\rightarrow$ XXXXX, etc.).

:: Conclusion: numeral systems differ in efficiency and comprehensibility.

Motivation example 2 – wumpus world

:: Hunt the Wumpus

- **Evaluation:** gold +1000, death -1000, -1 for step, -10 for sending arrow,
- **Environment:** cave rooms adjacent to wumpus stench, rooms adjacent to a pit are breezy, the cave with gold glitters,
- **Sensors:** bump (walk into a wall), stench, breeze and glitter (all of them room local), scream (audible in all the cave rooms),
- **Actions:** turn left/right, go forward, grab (pick up the gold when in a cave with gold), lay (gold), shoot (arrow kills wumpus if the agent faces it – the agent has the only arrow).



©Russel, Norvig: AI: A Modern Approach.

:: We take actions maximizing gain. The agent both starts and ends at [1,1].

Wumpus world – game characteristics

- deterministic – outcomes exactly specified,
- static – wumpus and pits do not move,
- discrete – sequence of distinct steps,
- environment is partially observable only – local sensors,

- let us concern 4x4 grid with the only wumpus and gold treasure,
- pit probability in each room set to 0.2 (the starting field always void).

- most tasks have no solution
  (gold in a pit or pit encircled, agent cannot make a safe move),
- still, a clear difference between “brainless” and “clever” agent.

Wumpus world – intentional and reactive agent

- **Reactive agent**
- no internal world model, driven purely by current state of sensors,
- if Glitter=yes then Action=grab,
- if Bump=yes then Action=rand(turn left, turn right),
- if Stench=yes or Breeze=yes then Action=[turn left, turn left, go forward, rand(turn left, turn right, go forward)].

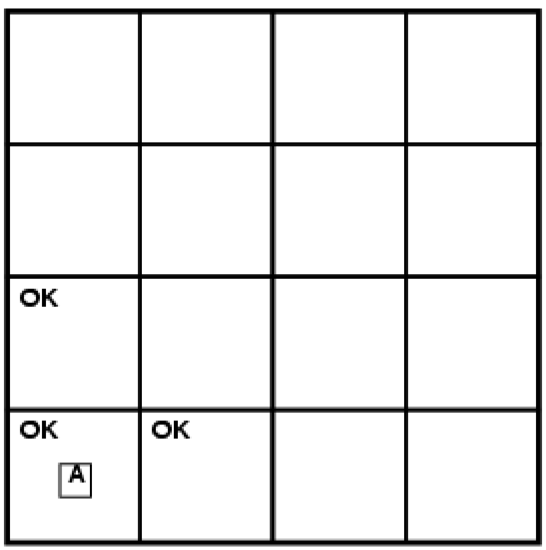
- **Intentional agent**
- works with internal world model (i.e. memory),
- uses knowledge base and inference,
- his implementation more difficult, see later.

Wumpus world (1)

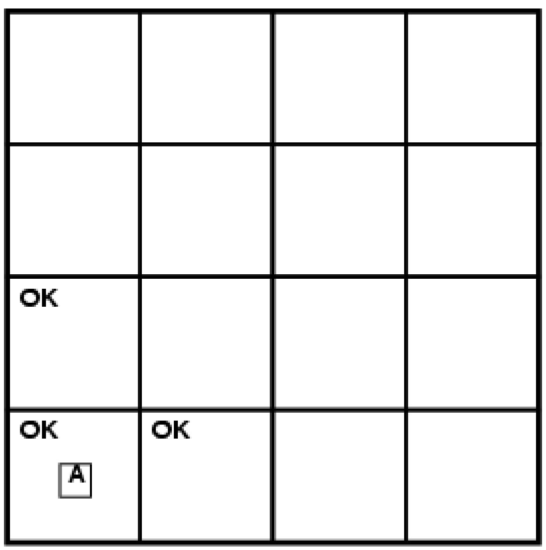
:: Success rate of intentional (IA) and reactive agent (RA):

- IA in 30% tasks finds gold and safely gets gets back to the start,
- in the rest of tasks must take risk or fail to reach gold,
- RA succeeds in 20% tasks only, it performs many more actions.

:: Progress of the agents, examples:



RA – evaluation 1



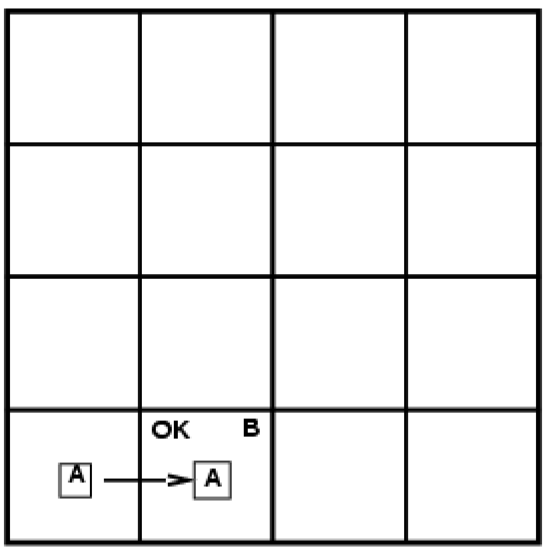
IA – inference 1

Wumpus world (2)

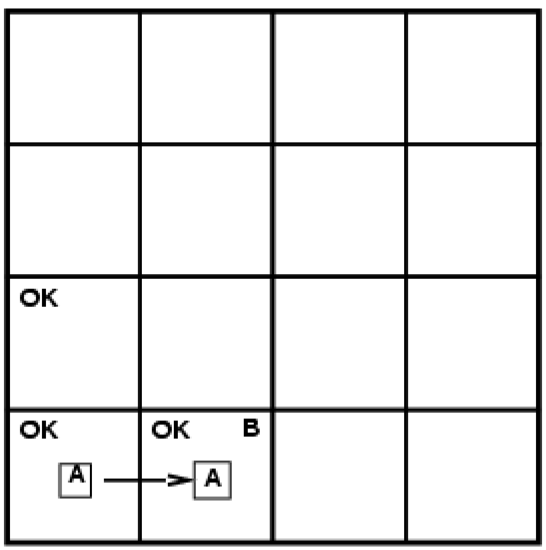
:: Success rate of intentional (IA) and reactive agent (RA):

- IA in 30% tasks finds gold and safely gets gets back to the start,
- in the rest of tasks must take risk or fail to reach gold,
- RA succeeds in 20% tasks only, it performs many more actions.

:: Progress of the agents, examples:



RA – step 1



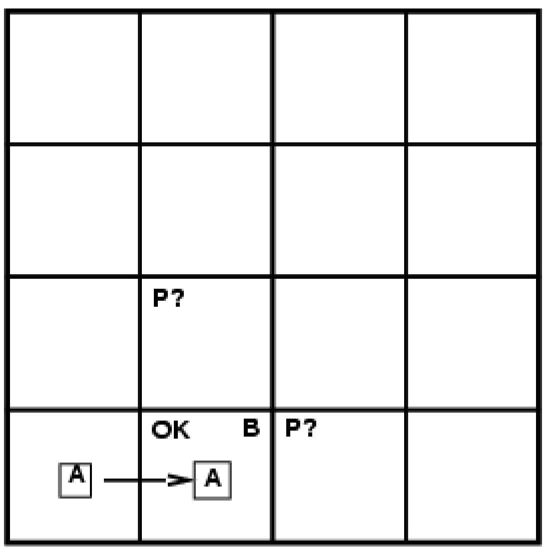
IA – step 1

Wumpus world (3)

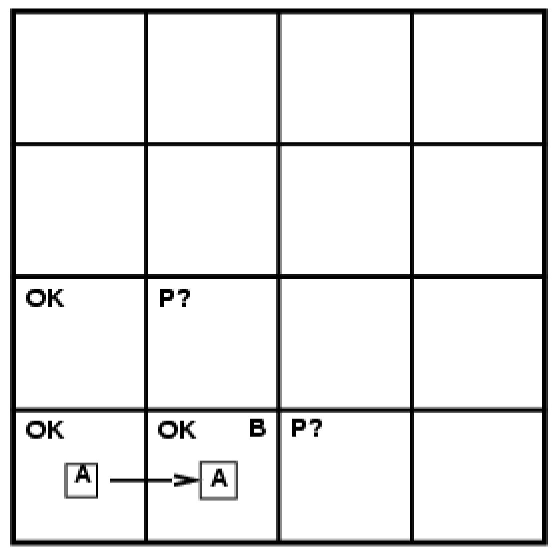
:: Success rate of intentional (IA) and reactive agent (RA):

- IA in 30% tasks finds gold and safely gets gets back to the start,
- in the rest of tasks must take risk or fail to reach gold,
- RA succeeds in 20% tasks only, it performs many more actions.

:: Progress of the agents, examples:



RA – evaluation 2



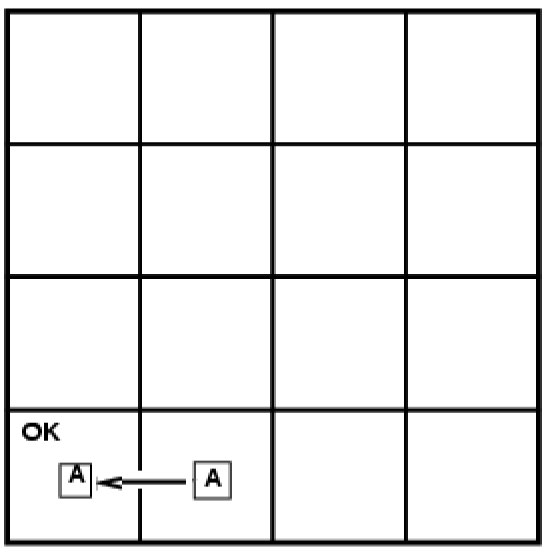
IA – inference 2

Wumpus world (4)

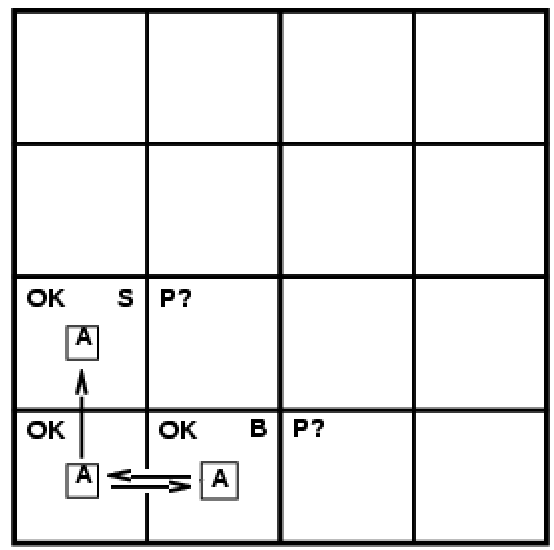
:: Success rate of intentional (IA) and reactive agent (RA):

- IA in 30% tasks finds gold and safely gets gets back to the start,
- in the rest of tasks must take risk or fail to reach gold,
- RA succeeds in 20% tasks only, it performs many more actions.

:: Progress of the agents, examples:



RA – step 2



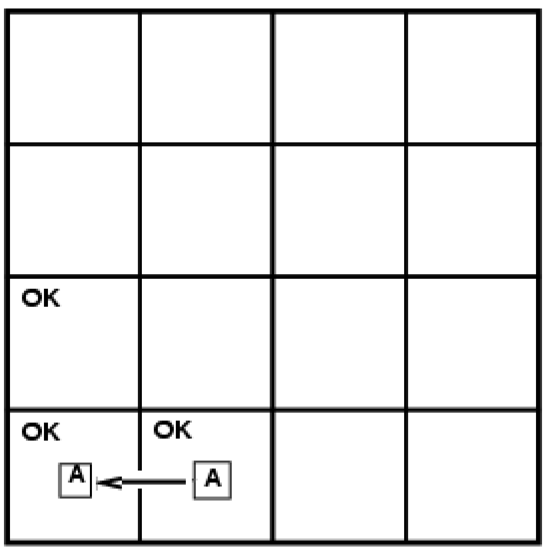
IA – steps 2 and 3

Wumpus world (5)

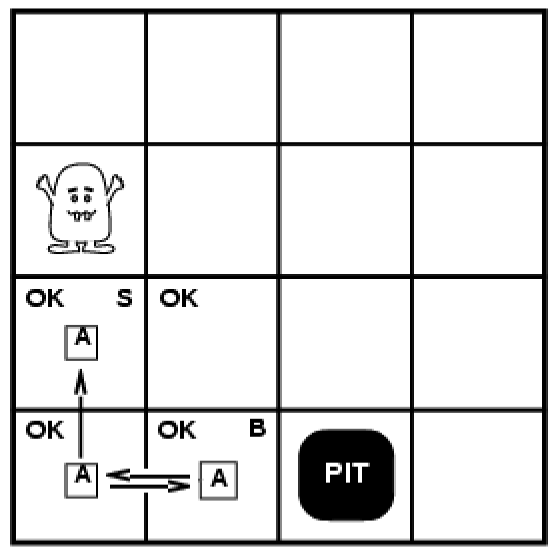
:: Success rate of intentional (IA) and reactive agent (RA):

- IA in 30% tasks finds gold and safely gets gets back to the start,
- in the rest of tasks must take risk or fail to reach gold,
- RA succeeds in 20% tasks only, it performs many more actions.

:: Progress of the agents, examples:



RA – evaluation 3



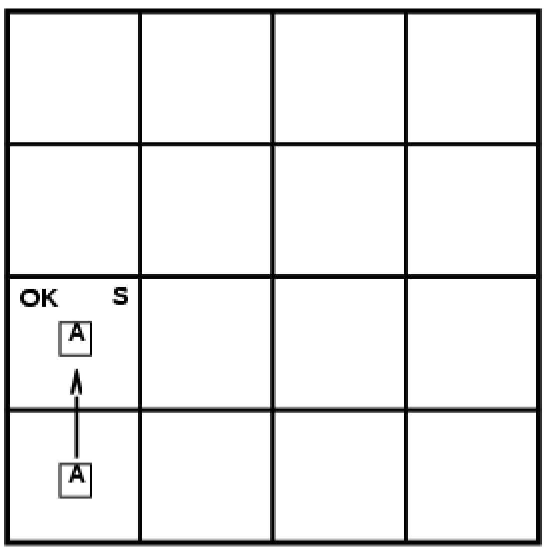
IA – inference 3

Wumpus world (6)

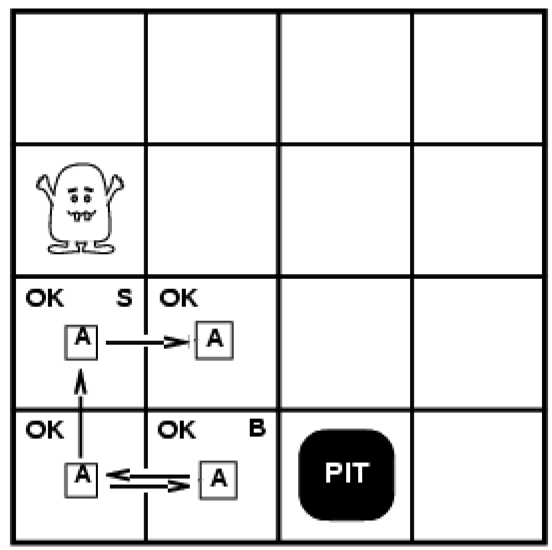
:: Success rate of intentional (IA) and reactive agent (RA):

- IA in 30% tasks finds gold and safely gets gets back to the start,
- in the rest of tasks must take risk or fail to reach gold,
- RA succeeds in 20% tasks only, it performs many more actions.

:: Progress of the agents, examples:



RA – step 3



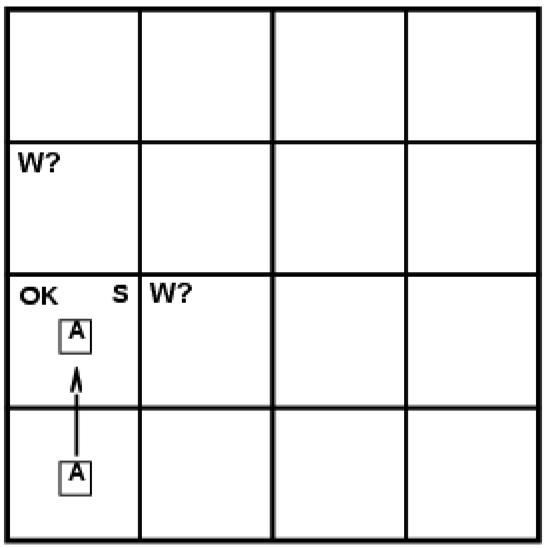
IA – step 4

Wumpus world (7)

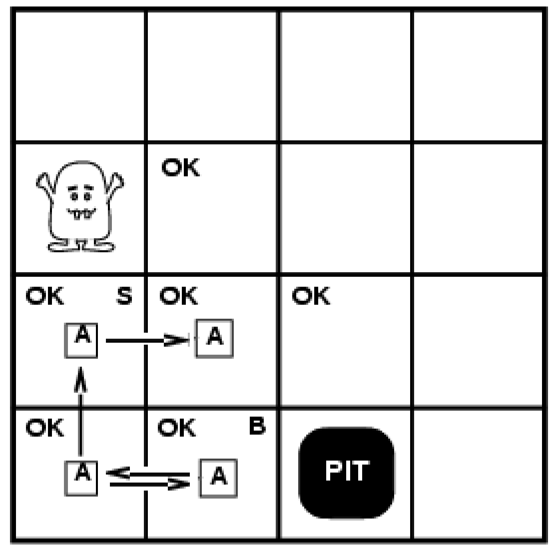
:: Success rate of intentional (IA) and reactive agent (RA):

- IA in 30% tasks finds gold and safely gets gets back to the start,
- in the rest of tasks must take risk or fail to reach gold,
- RA succeeds in 20% tasks only, it performs many more actions.

:: Progress of the agents, examples:



RA – evaluation 4



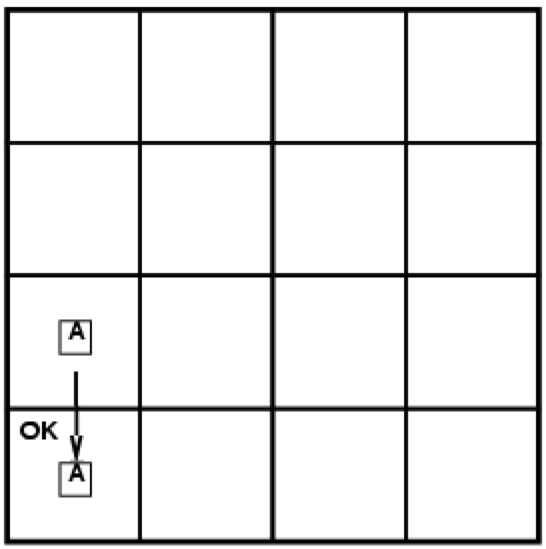
IA – inference 4

Wumpus world (8)

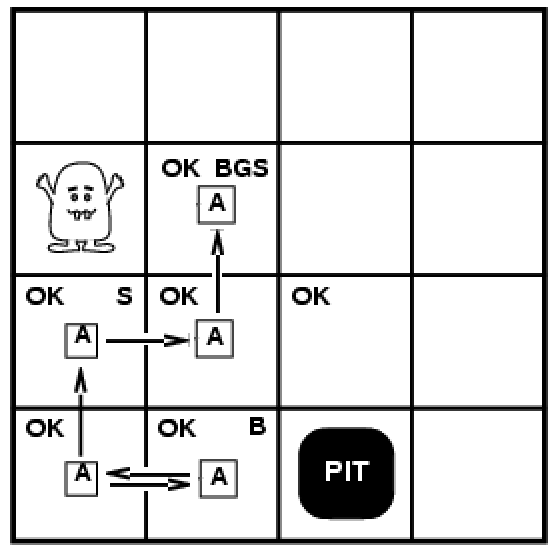
:: Conclusion:

- necessary conditions for successful agent/player:
  internal world representation and ability to perform inference.

:: Progress of the agents, examples:



RA – step 4



IA – step 5

Motivation example 3 – Does Zuzana lay eggs?

:: If I tell you that:

- (S1) Platypus and echidna are the only mammals that lay eggs.
- (S2) Only birds and mammals are warm-blooded.
- (S3) Zuzana, my armadillo, is warm-blooded and has no feathers.
- (S4) Every bird has feathers.

:: and ask: (D) Does Zuzana lay eggs?

Motivation example 3 – Does Zuzana lay eggs?

:: If I tell you that:

- (S1) Platypus and echidna are the only mammals that lay eggs.
- (S2) Only birds and mammals are warm-blooded.
- (S3) Zuzana, my armadillo, is warm-blooded and has no feathers.
- (S4) Every bird has feathers.

:: and ask: (D) Does Zuzana lay eggs?

:: Inference in natural language:

- Zuzana has no feathers and thus it is not a bird.
- Zuzana is warm-blooded and it is not a bird, it must be a mammal.
- Zuzana is mammal and armadillo, not platypus/echidna, it cannot lay eggs.

:: How to implement automatically? Strings difficult for inference ...

Motivation example 4 – IBM Watson

:: System for answering questions posed in natural language

- in 2011 overcame two former Jeopardy (Riskuj) human winners,
- general knowledge quiz, puns and wordplays, the answer is a question in natural language,
- integrates natural language processing, information retrieval, knowledge representation and machine learning.

:: Apparently more difficult than DeepBlue chess system, IBM as well

- answering natural language questions cannot be easily bounded.

:: KR perspective

- heuristic solution different from the "classic" logical one with formal proofs,
- coverage gets more important than precision,
- more than statistical information retrieval without a structured representation.

Motivation example 4 – IBM Watson

:: Two riddles in two categories with the desired solutions/questions

- literary figures: The name of this hat is elementary, my dear contestant.
- solution: What sort of hat does Sherlock Holmes wear?
- US cities: Its largest airport is named for a World War II hero, its second largest for a World War II battle.
- solution: What is Chicago? (O'Hare and Midway)

:: Abilities needed to solve the riddles

- decompose it into subtasks

  (American city, heroes, battles, airports),
- collect the necessary information (facts),
- compile the partial pieces of knowledge and pose the correct question.

Knowledge

:: Very general and frequently used term

- appearance from epistemology to knowledge engineering,
- a set of sentences in a KR language, assertions about the world,
- a content of knowledge base (axioms vs inferred sentences, facts vs rules),
- related terms: data $\rightarrow$ information $\rightarrow$ knowledge (increasing abstraction).

:: Other ways of understanding the term knowledge

- for a person – expertise and skills acquired through learning or from experience,
- for a field of study – all the known facts and pieces of generalized information,
- common sense – reliable understanding of an issue, ability to employ it.

:: Which **cognitive** abilities needed to gain knowledge

- perception, learning, communication, association, inference.

:: Key related terms: representation and reasoning.

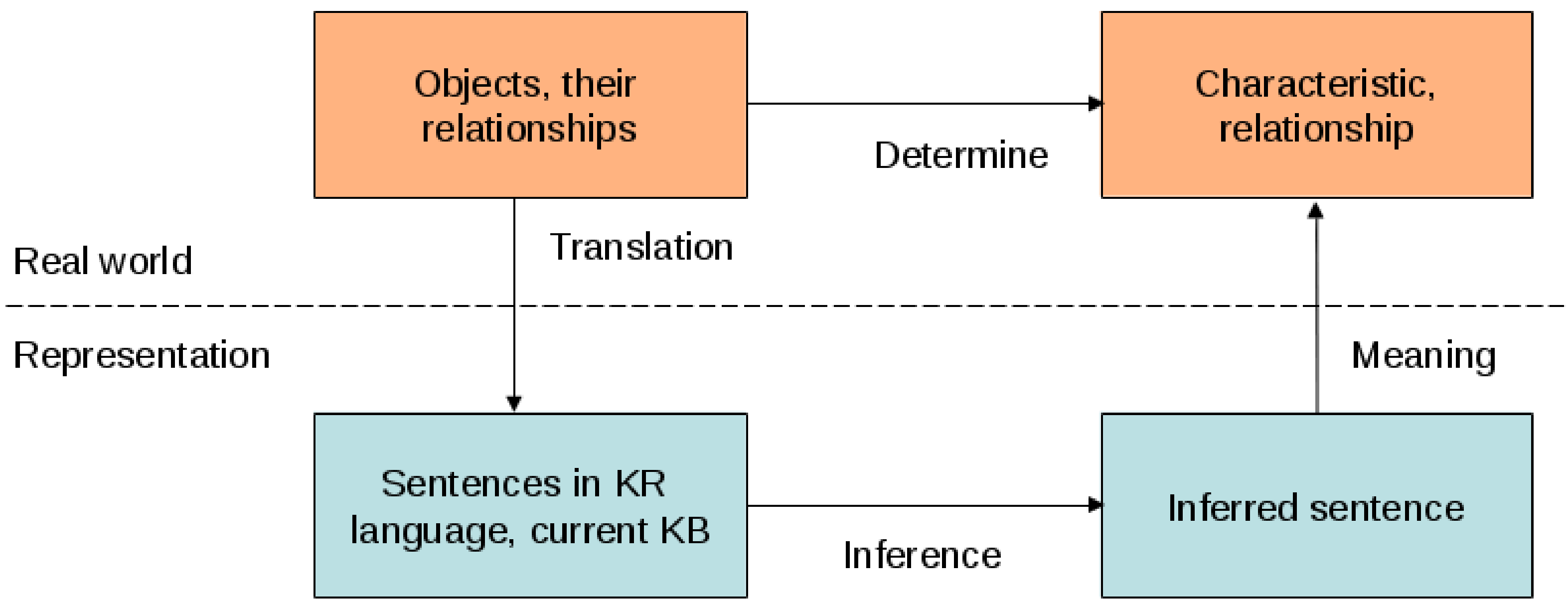
Knowledge representation

:: What is its role?

- surrogate
  - reasoning about consequences of actions without their real performing,
  - internal substitution of external real objects, necessarily imperfect,
- medium for efficient computation
  - mechanistic viewpoint, emphasis on efficiency of inference,
  - some procedures get easy other difficult,
- set of ontological commitments = decisions how and what to see in the world
  - opportunity to focus on aspects of the world we believe to be relevant,
  - unavoidable because of the inevitable imperfections of representations,
- formalization of (human) communication
  - in which form to share information with other humans or machines.

Knowledge representation – fundamental schema

- KR language must have clear **syntax** and **semantics**,
- it is necessary to interpret the meaning of sentences in the real world.



Knowledge representation language – basic characteristics

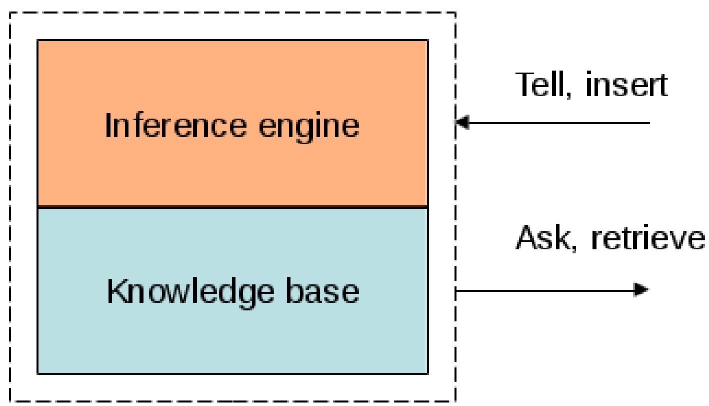
- Understandability/naturalness
  - compare bird(suzanne) and ydbk!op for: Suzanne is a bird.
- Expressiveness and its adequacy
  - represents the required scale of objects, characteristics and relationships,
  - e.g., propositional logic is not enough expressive for Aristotle's syllogisms,
- Suitability for reasoning
  - **soundness** – all the inferred sentences truly hold in the real world,
  - efficiency – inference in reasonable time within a reasonable memory,
  - **completeness** – all the sentences that follow from KB can be derived,

:: Natural language (Czech) is natural, expressive, but entirely improper for automated reasoning. Btw. it is ambiguous (I saw the man with the binoculars.) and context dependent (I was on my way to the bank.)

:: There is no universal language, different problems ask for different languages.

Knowledge – representation and use

- Knowledge base (KB)
  - a set of sentences in a KR language,
  - knowledge specific for the domain under study, both user defined and automatically derived,
  - working memory – gives the current state of the solved problem.
- Inference engine
  - implements reasoning – derives new knowledge and controls KB,
  - problem independent content (general inference rules, conflict resolution mechanism).



KR languages – basic approaches

- Logical schemas
  - declarative KB in the form of logical assertions + classic logic inference,
  - formalism: first-order logic, tool: PROLOG language.
- Procedural schemas
  - KB in the form of instructions,
  - typical case rule-based production systems – if ... then ... rules
- Network-based models
  - knowledge in the form of graph,
  - objects = concepts = nodes, relationships = edges,
  - example: semantic networks.
- Structured models
  - extension of network-based models, graph nodes correspond to complex structures,
  - example: scripts, frames and objects.

Logic

- Formal language with well-defined:
  - syntax – how to build permissible (well-formed) formulae,
  - semantics – what is the meaning of individual sentences,
  - axioms and deduction rules – reasoning tools.

:: The most simple is **propositional logic**

- Syntax – elements of language
  - a non-empty set of symbols, symbol $=$ atomic propositional formula,
  - logical operators $\Rightarrow$ a $\neg$, parentheses (),
  - efficiency of representation increased by $\wedge$, $\vee$ and $\Leftrightarrow$ (e.g. $A \vee B$ instead of $\neg A \Rightarrow B$),
- Syntax – definition of well-formed propositional formula (WFF)
  - every atomic propositonal formula is WFF,
  - provided that A and B are WFFs, then both $(\neg A)$ and $(A \Rightarrow B)$ are WFFs.

Propositional logic

- Semantics – meaning of symbols, logical connectives and formulae
  - symbols = elementary statements, e.g. P for “Peter likes chocolate.”
  - each statement can either be true or false,
  - **interpretation** = assigns all the symbols truth values,
  - **model** = interpretation for which formula is true,
  - every operator corresponds to a truth table.
- Axioms – basic and evident premises, accepted as true without controversy
  - (A1) $A \Rightarrow (B \Rightarrow A)$,
  - (A2) $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$,
  - (A3) $(\neg B \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)$.
- Inference rules – to derive formulae and carry out proofs
  - the basic rule is Modus Ponens (implication elimination): $A, A \Rightarrow B \models B$,
  - the rules for extended set of operators: excluded middle $\models A \vee \neg A$, introduction of disjunction: $A \models A \vee B$, etc.

Logical entailment and inference

- Logical entailment
  - $\alpha \models \beta$ – sentence α **entails** sentence β,
  - $\alpha \models \beta$ iff all the models of α also models for β,
- Logical inference
  - KB $\vdash_i \alpha$ – α is **derivable** from KB (a sentence as well!) by procedure i,
  - **soundness** – i is sound iff: KB $\vdash_i \alpha \Rightarrow$ KB $\models \alpha$,
  - **completeness** – i is complete iff: KB $\models \alpha \Rightarrow$ KB $\vdash_i \alpha$.

Propositional logic – example

:: On an island we can only meet truth tellers and liars. Both types are compulsive, liars never tell the truth and vice versa. We meet A and he/she says: "I am a liar but B is not." Decide the characters of A and B. Employ propositional logic.

Propositional logic – example

:: On an island we can only meet truth tellers and liars. Both types are compulsive, liars never tell the truth and vice versa. We meet A and he/she says: "I am a liar but B is not." Decide the characters of A and B. Employ propositional logic.

- Task formalization:
  - atomic propositions – a: "A is a liar." b: "B is a liar.",
  - (F1) $a \Rightarrow \neg(a \wedge \neg b)$,
  - (F2) $\neg a \Rightarrow a \wedge \neg b$,
- Solution 1 through truth table (model checking)
  - search for model(s) satisfying both (F1) and (F2).

a	b	¬a	¬b	(a ∧ ¬b)	F1	F2
F	F	T	T	F	F	T
F	T	T	F	F	F	T
T	F	F	T	T	T	F
T	**T**	F	F	F	**T**	**T**

Propositional logic – example (cont.)

- Deductive reasoning – gradually employ arbitrary inference rules
  - (F1) $a \Rightarrow \neg(a \wedge \neg b)$,
  - (F2) $\neg a \Rightarrow a \wedge \neg b$,
  - (F3) $\neg a \vee \neg(a \wedge \neg b)$ (implication removal in (F1)),
  - (F4) $\neg\neg a \vee (a \wedge \neg b)$ (implication removal in (F2)),
  - (F5) $\neg a \vee \neg a \vee b$ (De Morgan's law in (F3)),
  - (F6) $(a \vee a) \wedge (a \vee \neg b)$ (double negation and distributivity in (F4)),
  - (F7) $\neg a \vee b$ (idempotency (F5)),
  - (F8) $a \wedge (a \vee \neg b)$ (idempotency (F6)),
  - (F9) a (absorption (F8)),
  - (F10) $a \wedge b$ (negation absorption (F7) $\wedge$ (F9)),
  - (F11) b (conjunction elimination (F10)).

Propositional logic – example (cont.)

- Axiomatic proof – only axioms and modus ponens (MP)
  - (F1) $a \Rightarrow \neg(a \wedge \neg b)$,
  - (F2) $\neg a \Rightarrow a \wedge \neg b$,
  - (F3) $a \Rightarrow \neg a \vee b \Leftrightarrow a \Rightarrow (a \Rightarrow b)$ (F1 with implications only),
  - (F4) $\neg a \Rightarrow \neg\neg(a \wedge \neg b) \Leftrightarrow \neg a \Rightarrow \neg(a \Rightarrow b)$ (F2 with implications only),
  - (A1) $A \Rightarrow (B \Rightarrow A)$,
  - (A2) $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$,
  - (A3) $(\neg B \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)$,
  - (F5) $(a \Rightarrow (a \Rightarrow b)) \Rightarrow ((a \Rightarrow a) \Rightarrow (a \Rightarrow b))$ ((A2) A/a, B/a, C/b),
  - (F6) $(a \Rightarrow a) \Rightarrow (a \Rightarrow b)$ (MP (F3), (F5)),
  - (F7) $(a \Rightarrow b)$ (MP (F6), **validity of $(a \Rightarrow a)$ without a proof, HW**),
  - (F8) $(\neg a \Rightarrow \neg(a \Rightarrow b)) \Rightarrow ((a \Rightarrow b) \Rightarrow b)$ (from (A3) $A/(a \Rightarrow b)$, B/a),
  - (F9) $(a \Rightarrow b) \Rightarrow a$ (MP (F4), (F8)),
  - (F10) a (MP (F7), (F9)),
  - (F11) b (MP (F7), (F10)).

Resolution

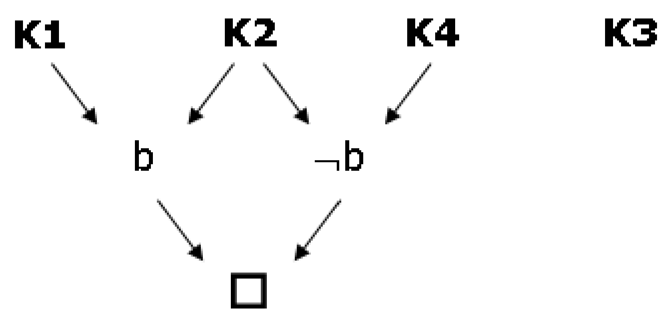
- A **complete** inference mechanism often needed,
- deductive rule application?
  - selection of an axiom, inference rule or sentence depends on intuition,
  - proof is difficult to automate,
  - completeness efforts lead to combinatorial explosion.
- model checking?
  - sound and complete, but $O(2^n)$, where n is the number of symbols.
- **resolution** is sound and in combination with a complete search algorithm complete
  - refutation complete
    - the given goal confirms or refutes, but does not have to finish,
    - does not generate a list of true statements (confirmation only)!

Resolution

- reasons for “efficiency”:
  - complete conjunctive normal form
    * minimizes the number of applicable rules,
    * literal – atomic formula or its negation, clause – disjunction of literals,
    * resolution rule: $A \vee \psi, \neg A \vee \phi \models \psi \vee \phi$,
    * from a pair of clauses derive their resolvent,
    * A and $\neg A$ are complementary literals,
  - proof by contradiction
    * better control through search space, goal oriented search,
    * $T \models \phi$ iff $T \cup \{\neg\phi\} \models \Box$,
    * contradiction equals to an empty clause, denoted by $\Box$.

Resolution – example (cont.)

- Resolution proof – transform to clausal form first
  - (F1) $a \Rightarrow \neg(a \wedge \neg b)$,
  - (F2) $\neg a \Rightarrow a \wedge \neg b$,
  - (K1) $\neg a \vee b$ (transform (F1), see the deductive proof before),
  - (K2,3) $a \wedge (a \vee \neg b)$ (transform (F2), see the deductive proof before).
- Resolution proof – extend with the negation of the goal formula
  - (D1) $a \wedge b$,
  - $(\neg D1) \equiv$ (K4) $\neg a \vee \neg b$.
- Resolution proof tree *



Resolution – search strategy, complexity

- How to search clauses to be resolved?
  - controlled by a search **strategy**,
  - it creates the derivation graph (DG) – leaves are clauses, internal nodes are resolvents,
  - resolution proof tree – subgraph of DG whose root is $\Box$.
- Complete strategies
  - breadth-first search
    - derive all the 1. order resolvents first (1 aplication of the resolution rule on the input set of clauses),
    - then all the resolvents of the 2. order (at least 1 parent is of the 1. order), etc.,
    - empty clause always found at the lowest possible tree depth,
    - danger of combinatorial explosion,

Resolution – search strategy, complexity

- set of support
  - theory is contradiction free, conflict from the negated goal formula only,
  - derive resolvents having at least one of the ancestors (parents, grand-parents, ...) the goal formula,
  - the number of resolvents at each depth of proof grows slower,
- linear
  - the last generated resolvent is the nearest parent,
  - reduces the number of resolvents again.

- Incomplete strategies
  - unit – one of the parent clauses is always a unit clause (one literal only),
  - input – one parent clause always comes from theory or negated goal,
  - filtration – input strategy with affinity relationship
    - it is allowed to resolve a clause with its ascendant,
  - combined
    - e.g., linear-input, complete for Horn clauses only (see later).

Resolution – search strategy, complexity

- Removal of redundant clauses from the set of resolvents
  - remove tautologies ($P \vee \neg P$),
  - remove subsumptions (logical consequences) of an existing clause (P, $P \vee \neg Q$),
  - test validity of literals (namely in FOL: greater_than(3,4)).
- Complexity of resolution
  - belongs to the exponential family of algorithms,
  - still, in many cases "much more efficient" than model checking,
  - efficient in tasks with a small or large rate between the number of clauses and symbols,
  - alternative for propositional logic makes Davis-Putnam algorithm for CNF logical formula satisfiability testing.

Forward and backward chaining

- Limited to Horn clauses with one positive literal at most
  - $(P \vee \neg Q \vee \neg R) \Leftrightarrow (Q \wedge R) \Rightarrow P$,
  - we cannot formalize e.g. $(\neg P \vee Q \vee R) \Leftrightarrow P \Rightarrow (Q \vee R)$,
  - i.e., only simple facts and implications with the only consequent.
- Reward is inference in $O(n)$, where n is the size of knowledge base (the number of clauses)
  - forward or backward chaining represent the inference methods,
  - both approaches are sound and complete,
  - namely the complexity of backward chaining often sublinear.

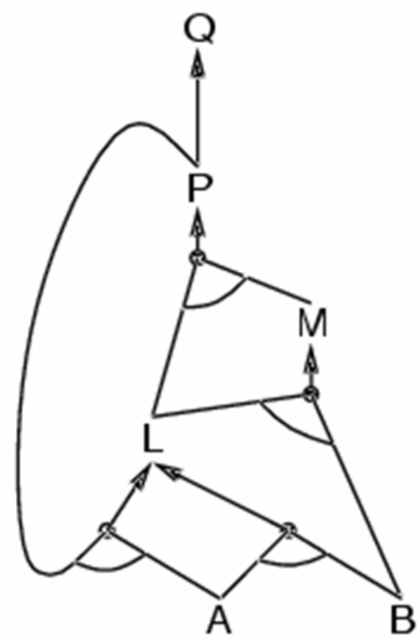
Forward and backward chaining

- Forward chaining
  - keeps *agenda* of symbols known to be true (and so far unprocessed),
  - for each implication *counter* of unknown premises,
  - process an agenda symbol, reduce counts for implications that have the symbol in its premises,
  - if 0, the conclusion of implication added into the agenda,
  - stops when agenda is empty or the query is proven to be true/gets processed,
  - data driven – from known facts towards goal or exhaustion of all the possible operations.
- Backward chaining
  - finds premises for validity of current query,
  - tries to gradually verify them = check they are true,
  - goal oriented – proceeds from the query towards its premises.

Forward and backward chaining – example

:: query: Q?

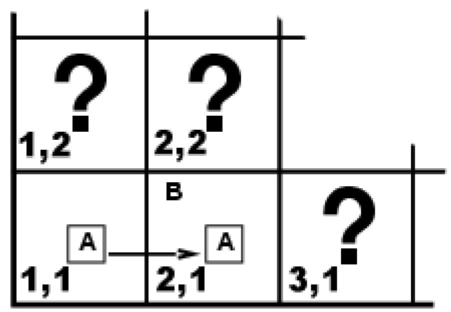
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Knowledge base and AND-OR graph

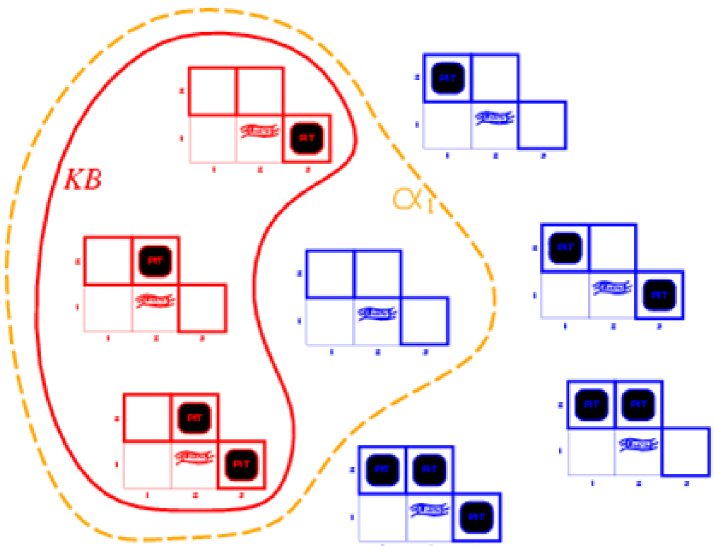
Propositional logic in wumpus world

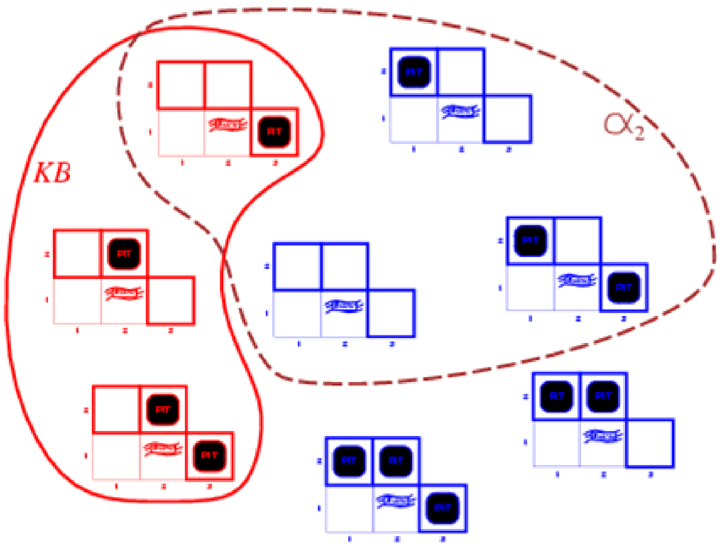
- A simplified example
  - only pits, a restricted number of cave rooms,
  - KB stems from observations in rooms [1,1] a [2,1]
    - $\neg B_{1,1}$ – no breeze at [1,1], $B_{2,1}$ – breezy at [2,1], $\neg P_{1,1}$ – no pit at [1,1],
  - KB contains local inference rules
    - $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$, $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$,
  - the goal is to find out occurrence of pits at [1,2], [2,2] and [3,1].



World fragment – 3 rooms – 8 models

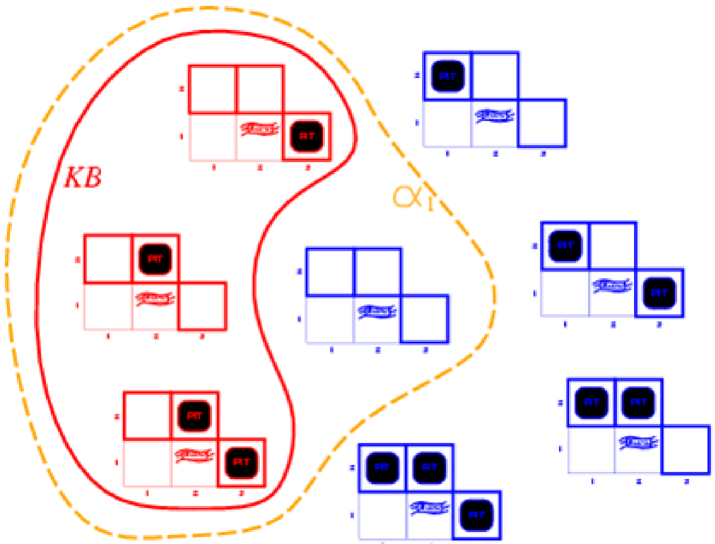
Model checking in wumpus world

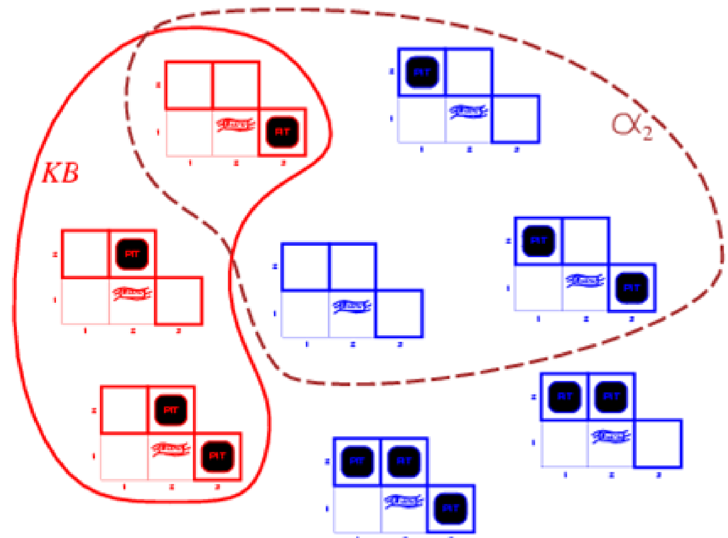




$\alpha_1 \equiv \neg P_{1,2}$, M(KB) $\subseteq$ M(α_1) $\Rightarrow$ KB $\models \alpha_1$ $\alpha_2 \equiv \neg P_{2,2}$, M(KB) $\not\subseteq$ M(α_2) $\Rightarrow$KB $\not\models \alpha_2$

Model checking in wumpus world





$\alpha_1 \equiv \neg P_{1,2}$, $\mathsf{M}(\mathsf{KB}) \subseteq \mathsf{M}(\alpha_1) \Rightarrow \mathsf{KB} \models \alpha_1$ $\quad$ $\alpha_2 \equiv \neg P_{2,2}$, $\mathsf{M}(\mathsf{KB}) \not\subseteq \mathsf{M}(\alpha_2) \Rightarrow \mathsf{KB} \not\models \alpha_2$

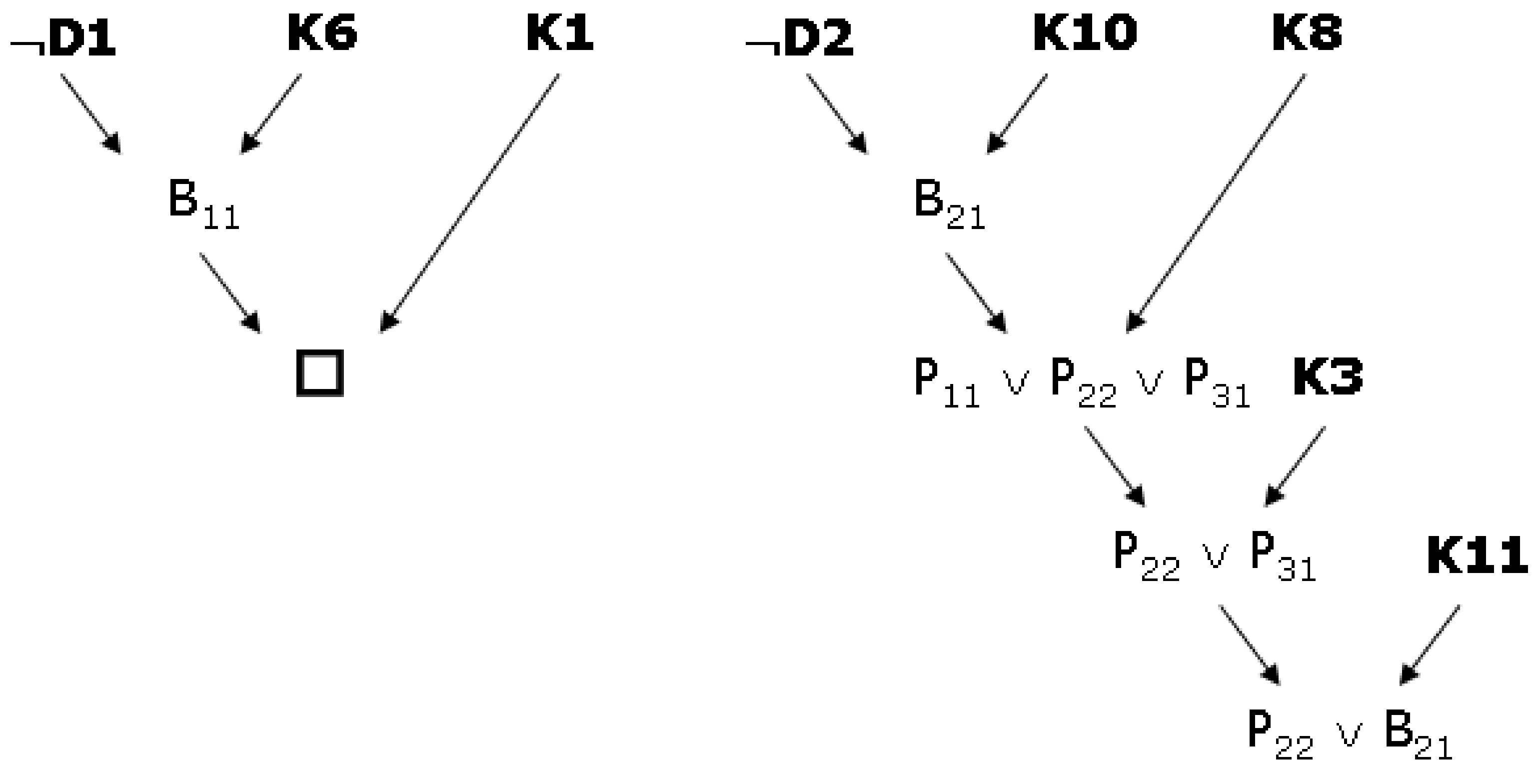
$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	α_1	α_2
F	F	F	F	F	F	F	F	T	T
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
F	T	F	F	F	F	T	**T**	**T**	**T**
F	T	F	F	F	T	F	**T**	**T**	**F**
F	T	F	F	F	T	T	**T**	**T**	**F**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T	T	T	T	T	T	T	F	F	F

Resolution in wumpus world

- Conversion into clausal form
  - (F1-3) $\neg B_{1,1}$, $B_{1,2}$, $\neg P_{1,1}$,
  - (F4) $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$,
  - (F5) $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$,
  - (K1-3) $\neg B_{1,1}$, $B_{1,2}$, $\neg P_{1,1}$,
  - (K5-7) $\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$, $\neg P_{1,2} \vee B_{1,1}$, $\neg P_{2,1} \vee B_{1,1}$ ((F4) modification),
  - (K8-11) $\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$, $\neg P_{1,1} \vee B_{2,1}$, $\neg P_{2,2} \vee B_{2,1}$, $\neg P_{3,1} \vee B_{2,1}$ ((F5) modification).

Resolution in wumpus world

- Resolution proof tree for $D_1 \equiv \neg P_{1,2}$ a $D_2 \equiv \neg P_{2,2}$



Intentional agent in wumpus world

function **PL-wumpus-agent**(sensors=[stench,breeze,glitter]) returns action

if stench **then** Tell(KB,$S_{x,y}$) **else** Tell(KB,$\neg S_{x,y}$) % KB update

if breeze **then** Tell(KB,$B_{x,y}$) **else** Tell(KB,$\neg B_{x,y}$) % x,y is current position

if glitter **then** action $\leftarrow$ grab

else if plan $<>$ {} **then** action $\leftarrow$ Pop(plan) % plan = waiting action stack

else

repeat % search for a safe neighbor room/field

Neighbor-room(x,y,i,j) % first unexplored then explored rooms

until Ask($\neg P_{i,j} \wedge \neg W_{i,j}$) = True

plan $\leftarrow$ Make-plan(x,y,direction,i,j) % last action to reach room i,j

action $\leftarrow$ Pop(plan) % execute first action of the existing plan

Update(direction, room (x,y), visited rooms, action) % procedural, out of KB

return action

Summary (1)

- Knowledge can be represented in the form of **symbols**
  - symbols – general and potentially complex data structures,
  - objects, concepts, facts, rules, strategies,
- intelligent behavior can be reached by **manipulating** these symbols,
- assumptions of symbolic (traditional) AI
  - an alternative is connectionism – neural networks,
- intelligent system then requires
  - **formal language** for knowledge representation,
  - **ability to derive** new knowledge/conclusions,
- no universal language nor inference method applicable to all the problems,
- logic = very general formal representation facilitating consequent reasoning
  - resolution is a method of sound and complete logical inference.

Summary (2)

- Logical agents apply inference to a knowledge base
  - to derive new information and make decisions,
- propositional logic is a simple language
  - based on propositional/primitive symbols and logical operators/connectives,
  - it facilitates comprehensible and efficient solution of many problems,
  - for other problems inappropriate because of
    - efficiency – see spatial rules in the wumpus world NxN,
    - finiteness of the set of propositional symbols – impossible to cope with unbounded problems,
    - insufficient expressive power – see mortal Socrates.
- Next: predicate logic and its utilization for planning in dynamic environment.

Recommended reading, lecture resources

:: Reading

- Russel, Norvig: **AI: A Modern Approach**, Logical Agents, chapter 7
  - KR and intelligent agents,
  - pdf available – `http://aima.cs.berkeley.edu/newchap07.pdf`.
- Brachman, Levesque: **Knowledge Representation and Reasoning**
  - book, The Morgan Kaufmann Series in Artificial Intelligence.
- Mařík a kol. **Umělá inteligence 1**
  - kapitola Reprezentace znalostí: formáty, logika, sémantické sítě, rámce,
  - kapitola Řešení úloh a dokazování vět: predikátová logika a důkazní prostředky.
- Mařík a kol. **Umělá inteligence 2**
  - kapitola Znalostní inženýrství: praktická, znalostní systémy a aplikace.