Knowledge-based Systems

Philippe Morignot

philippe.morignot@vedecom.fr

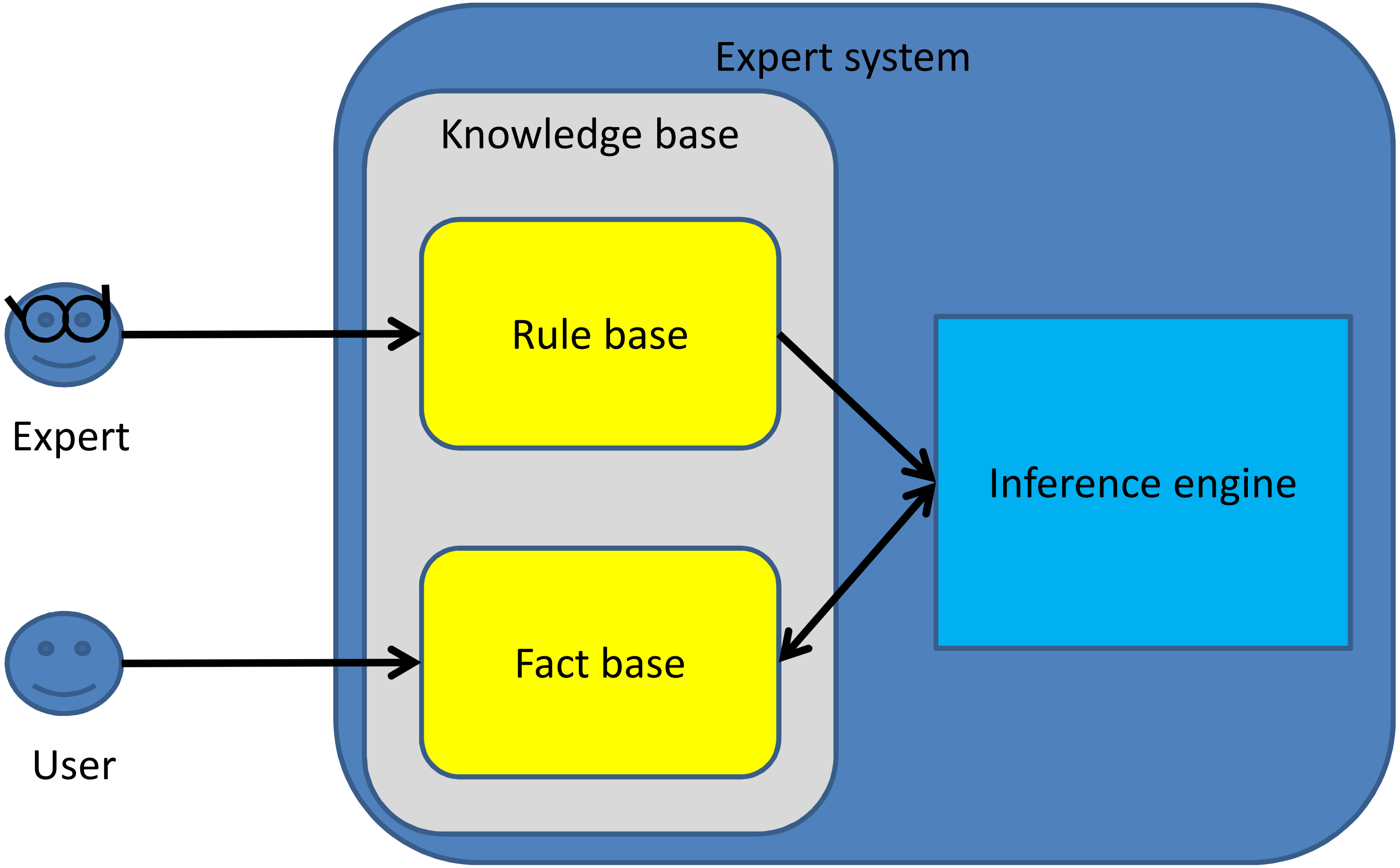
Motivation

- Decision aid:
  - Mimic human reasoning, based on the notion of knowledge.
  - Try to analyze a problem as an expert in a domain would do.
- Examples:
  - Medical domain: diagnosis of infectious disease (MYCIN).
  - Fault detection.
  - Automation of procedures (law, …).

Definitions

- A knowledge-based system (expert system) is composed of:
  - A knowledge base:
    - A **rule base**: knowledge of a domain.
    - A **fact base**: data about the case under study.
  - An inference engine:
    - Algorithm which activates knowledge to make a reasoning and to produce a result.
    - Makes inferences.
  - A user interface: the knowledge base changes over time.

Structure of
a knowledge-based system



Expert system

Knowledge base

Rule base

Fact base

Inference engine

Expert

User

Logic

- 0 order logic: proposition logic, boolean variables, propositions.
  - Example of rule: IF « patient is sick » AND « patient is over 40 years old » THEN …
- 0+ order logic: multi valued variables.
  - Example of rule: IF symptom = pain AND age $\geq$ 40 THEN …
- 1 order logic: 1st-order predicate logic.
  - Example of rule: IF (?x has pain) AND (?x has age ?n) AND (?n $\geq$ 40) THEN …

Knowledge Representation
Fact base

- The fact base is the working memory of the knowledge based system. It changes over time.
- Initially, it includes the initial facts.
- During execution, it is augmented by the facts inferred by the inference engine.
- Example: diagnosis in medical domain.

Knowledge representation
Rule base

- Knowledge and know-how of an expert.
- An expert can change the rule base at execution time; a user cannot.
- Uses 0-order, 0+ -order or 1-order logic.
- An inference rule is composed of:
  - Its left-hand side: terms which must be satisfied for the rule to fire; activation conditions.
  - Its right-hand side: actions to execute.

Inference rules

- Rule base:

  *(**defrule** is-uncle*

  (<x> ^father-of <y>) (<z> ^sex male) (<z> ^brother-of <x>)

  => { <z> ^uncle-of <y> }

)

- Fact base:

  *(**fact** Louis ^sex male)*

  *(**fact** Louis ^brother-of Alexandre)*

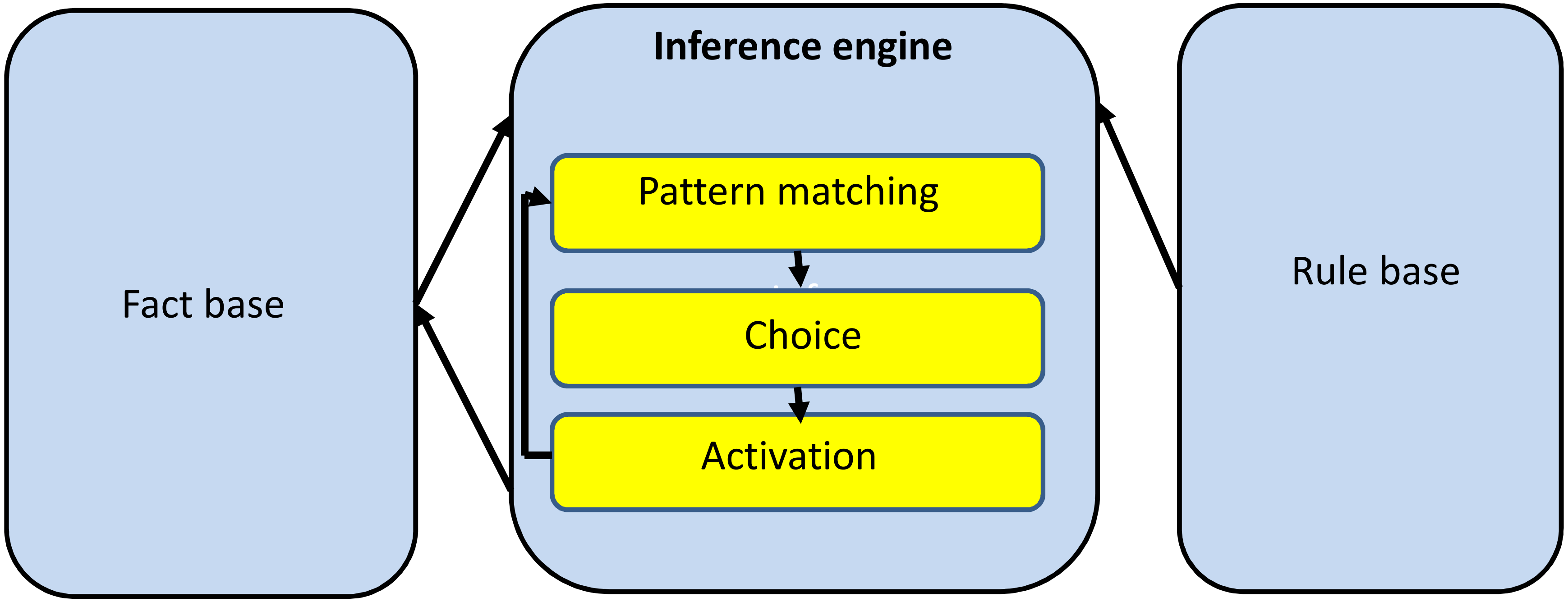
  *(**fact** Alexandre ^father-of Jean)*

- Inferred fact: *(**add-fact** Louis ^uncle-of Jean)*

Inference engine (1/2)

- Inference cycle:
  1. Pattern matching: find the rules which could be fired.
  2. Choice + conflict resolution: determine the rule to fire among the possible ones.
  3. Inference + activation: fire the chosen rule.
- Update the fact base.
- Infinite cycle: once a rule has fired it cannot fire again in the same cycle.
- Forward chaining or backward chaining.

Inference engine (2/2)

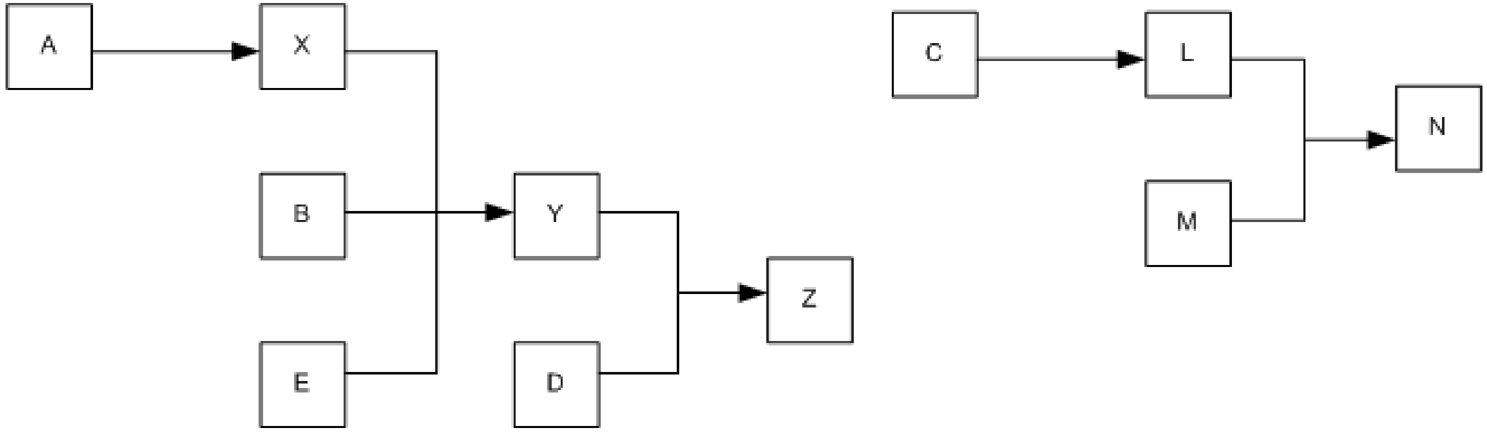


Forward chaining (1 / 2)

- Data-driven reasoning:
  - Uses the known facts
  - Fire all the possible firable rules
  - Infers facts which are added to the fact base
  - Keep reasoning

Forward chaining (2 / 3)
Example

- Rule base:



- Fact base: { A, B, C, D, E }
- Which facts are inferred?

Forward chaining (3 / 3) Example

- *Rule base:*

R1 :	$a \vee b$	$\Rightarrow$	*add(e)*
R2 :	$b \wedge d$	$\Rightarrow$	*add(f)*
R3 :	$c \wedge d \wedge g$	$\Rightarrow$	*add(a)*
R4 :	$g \vee (d \wedge f)$	$\Rightarrow$	*add(c), remove(b)*
R5 :	$a \wedge d \wedge e$	$\Rightarrow$	*add($\neg$b)*

- *Fact base:* ***{ b, d, a }***

- *Conflict resolution:* $R_4 > R_3 > R_5 > R_1 > R_2$

Which facts are included in the knowledge base in the end?

Backward chaining (1 / 3)

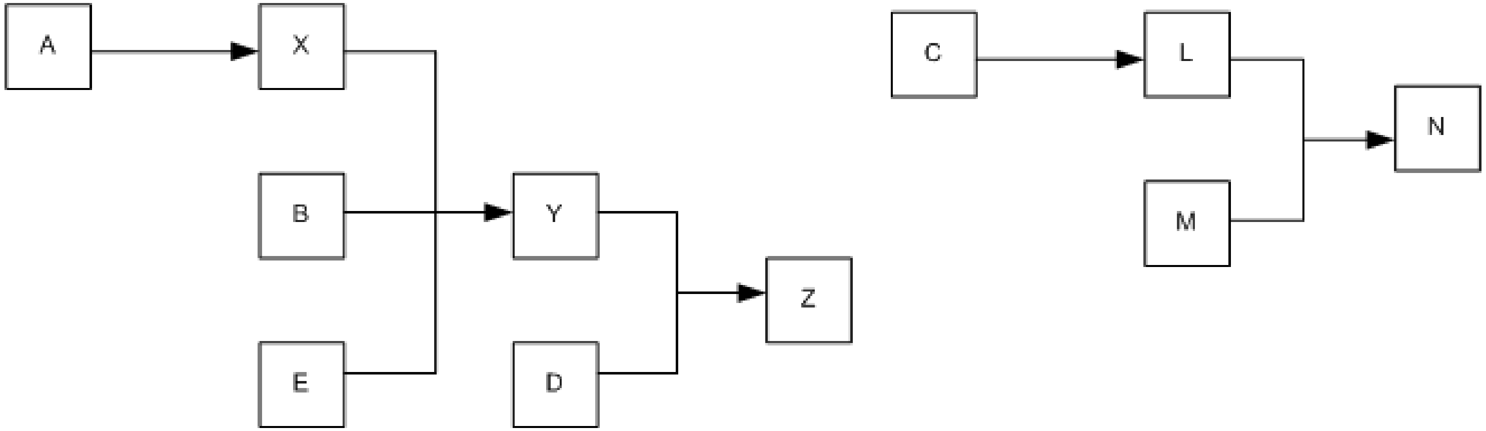
- Goal-driven reasoning.
- Starts from the goal to be proved.
- Goes back to the causes, in order to prove the (sub-)goals.
- When a goal is in the fact base, this goal is proved.

Backward chaining (2 / 3)
Algorithme

- In order to prove goal f:
  1. Find the rules which can prove f.
  2. Find the facts to be proved in order to fire these rules.
  3. For each one of these rules:
     a) If that rule can be fired, the goal f is proved.
     b) If that rule cannot be fired, recursively call that procedure on the fact of the LHS which are not satisfied.
  4. UNTIL goal f is proved OR there are unsatisfied goals and no rule to prove them.

Backward chaining (3 / 3) Example

- Rule base:



- Fact base: { A, B, C, D, E }

Can fact Z be proved? What about fact N?

Backward chaining (3 / 3)
Example

- *Rule base:*

R1 : $a \vee (e \wedge b) \Rightarrow f$

R2 : $c \wedge d \Rightarrow f$

R3 : $d \wedge h \Rightarrow c$

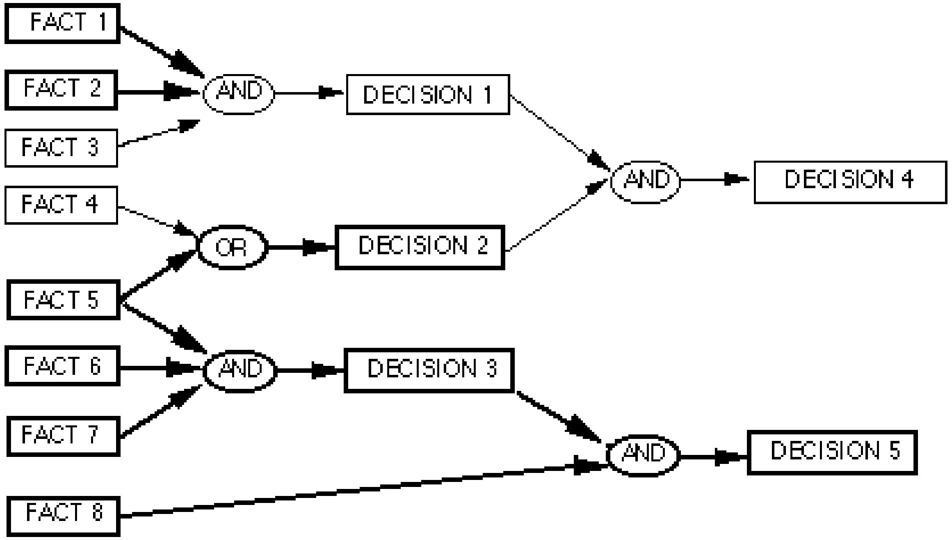
R4 : $c \wedge g \Rightarrow e$

- *Fact base:* ***{ d, g, h }***

Is fact f true?

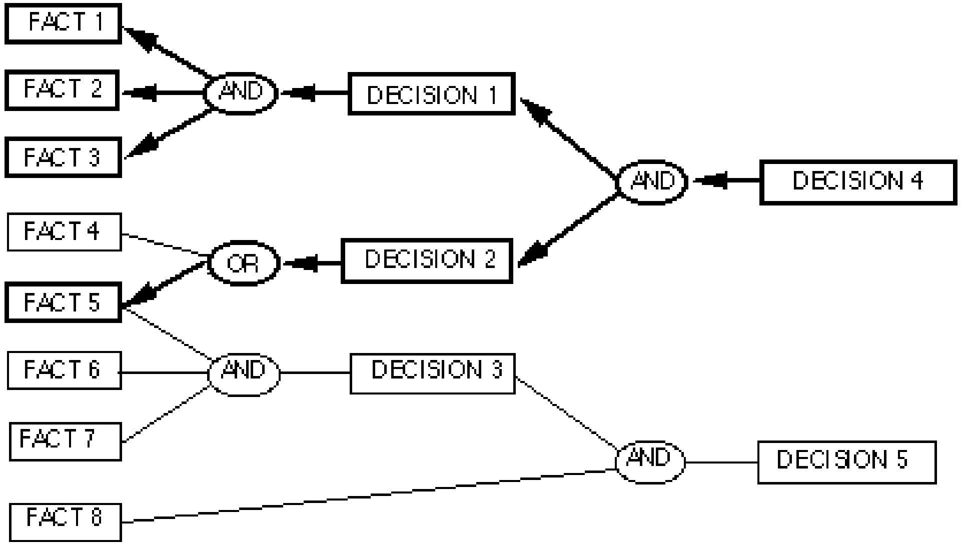
Use of forward chaining

- Starts from facts and infers as many facts as possible.
- Useful when one wants to discover which decisions can be taken given some facts.



Use of backward chaining

- Starts from the goal and looks for explanations of it.
- Used when one wants to take a decision, to know whether the conditions of this decision are satisfied.



Conclusion

- Knowledge-based systems (expert system) are composed of a knowledge base (rule base, fact base) and an inference engine.
- An inference engine runs a cycle : pattern matching, choice and activation.
- An inference engine can run in forward or backward chaining.
- Seminal papers:
  - J. Lederberg, E. Feigenbaum. Mechanization of Inductive Inference in Organic Chemistry. In Formal Representation of Human Judgment, B. Kleinmuntz (ed.), Wiley, 1968. [DENDRAL]
  - B. Buchanan, E. Shortliffe. [MYCIN]