Human Expert

A human *expert* is a specialist for a specific differentiated application field who creates solutions to customer problems in this respective field and supports them by applying these solutions.

Requirements

- Formulate precise problem scenarios from customer inquiries
- Find correct and complete solution
- Understandable answers
- Explanation of solution
- Support the deployment of solution

Knowledge Based Systems (2)

“Intelligent” System

An intelligent system is a program that models the knowledge and inference methods of a human expert of a specific field of application.

Requirements for construction:

- Knowledge Representation
- Knowledge Acquisition
- Knowledge Modification

Qualities of Knowledge

In most cases our knowledge about the present world is

incomplete/missing (knowledge is not comprehensive)

- e. g. "I don't know the bus departure times for public holidays because I only take the bus on working days."

vague/fuzzy/imprecise (knowledge is not exact)

- e. g. "The bus departs roughly every full hour."

uncertain (knowledge is unreliable)

- e. g. "The bus departs probably at 12 o'clock."

We have to decide nonetheless!

Reasoning under Vagueness

Reasoning with Probabilities

. . . and Cost/Benefit

Example

Objective: *Be at the university at 9:15 to attend a lecture.*

There are several plans to reach this goal:

- P_1: Get up at 8:00, leave at 8:55, take the bus at 9:00 …
- P_2: Get up at 7:30, leave at 8:25, take the bus at 8:30 …
- …

All plans are *correct*, but

- they imply different *costs* and different *probabilities* to *actually* reach that goal.
- P_2 would be the plan of choice as the lecture is important and the success rate of P_1 is only about 80–95%.

Question: *Is a computer capable of solving these problems involving uncertainty?*

Uncertainty and Facts

Example:

We would like to support a robot's localization by fixed landmarks.
From the presence of a landmark we may infer the location.

Problem:

Sensors are imprecise!

- We cannot conclude definitely a location simply because there was a landmark detected by the sensors.
- The same holds true for undetected landmarks.
- Only probabilities are being increased or decreased.

Degrees of Belief

We (or other agents) are only believing facts or rules to some extent.

One possibility to express this *partial belief* is by using *probability theory*.

"The agent believes the sensor information to 0.9" means:
In 9 out of 10 cases the agent trusts in the correctness of the sensor output.

Probabilities gather the "uncertainty" that originates due to ignorance.

Probabilities $\neq$ Vagueness/Fuzziness!

- The predicate "large" is fuzzy whereas "This might be Peter's watch." is uncertain.

Rational Decisions under Uncertainty

Choice of several *actions* or *plans*

These may lead to different results with different *probabilities*.

The *actions* cause different (possibly subjective) *costs*.

The *results* yield different (possibly subjective) *benefits*.

It would be rational to choose that action that yields the largest total benefit.

$$\text{Decision Theory} = \text{Utility Theory} + \text{Probability Theory}$$

Decision-theoretic Agent

input perception
output action

$K \leftarrow$ a set of probabilistic beliefs about the state of the world
calculate updated probabilities for current state based on available evidence including current percept and previous action
calculate outcome probabilities for actions, given action descriptions and probabilities of current states
select action A with highest expected utility given probabilities of outcomes and utility information
return A

Decision Theory: An agent is rational if and only if it chooses the action yielding the largest utility averaged over all possible outcomes of all actions.

Rule-based Systems

Rule-based Systems

Modi of usage:

Query:	Facts are retrieved from database or user is interrogated
Explanation:	System answers questions how a decision was concluded

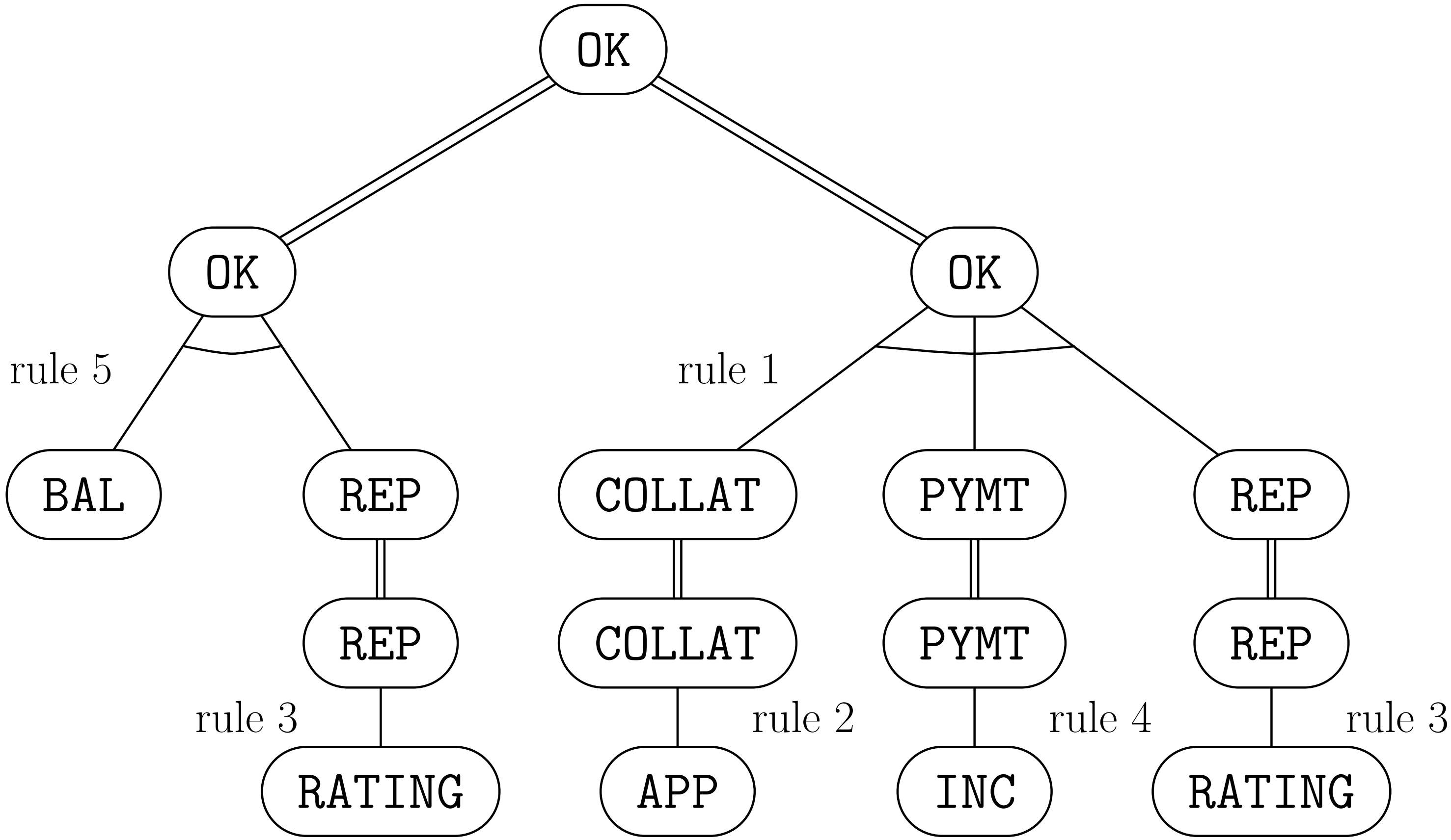
Example rule base:

1. `COLLAT` $\wedge$ `PYMT` $\wedge$ `REP` $\Rightarrow$ `OK`
2. `APP` $\Rightarrow$ `COLLAT`
3. `RATING` $\Rightarrow$ `REP`
4. `INC` $\Rightarrow$ `PYMT`
5. `BAL` $\wedge$ `REP` $\Rightarrow$ `OK`

Description:

`COLLAT`	satisfactory collateral
`PYMT`	payments undisputed
`REP`	good reputation
`APP`	high appraisal
`RATING`	good rating
`INC`	positive income
`BAL`	excellent balance
`OK`	loan will be approved

Corresponding `and/or`-Tree:



Example Queries

User: Why do you believe that the loan should be approved?

System: Because the collateral for the loan is satisfactory, and the applicant is able to make the loan payments, and the applicant has a good financial reputation.

User: Why did you ask me about whether or not the applicant's income exceeds his/her expenses?

System: Because I was trying to establish whether or not the applicant is able to make the loan payments.

User: Why do you believe that the applicant has a good financial reputation?

System: Because the credit rating service said that the applicant has a good credit rating.

Rules

A rule in general is a *if-then*-construct consisting of a *condition* and an *action*.

If *condition* then *conclusion*

These two parts may be interpreted differently according to the context:

- **Inference rules**: If *premise* then *conclusion*
- **Hypotheses:** If *evidence* then *hypothesis*
- **Productions:** If *condition* then *action*

Rules are often referred to as *productions* or *production rules*.

Rules

A rule in the ideal case represents a unit of knowledge.

A set of rules together with an execution/evaluation strategy comprises a program to find solutions to specific problem classes.

Prolog program: rule-based system

Rule-based systems are historically the first types of AI systems and were for a long time considered prototypical expert systems.

Nowadays, not every expert systems uses rules as its core inference mechanism.

Rising importance in the field of business process rules.

Rule Evaluation

Forward chaining

Expansion of knowledge base: as soon as new facts are inserted the system also calculates the conclusions/consequences.

Data-driven behavior

Premises-oriented reasoning: the chaining is determined by the left parts of the rules.

Backward chaining

Answering queries

Demand-driven behavior

Conclusion-oriented reasoning: the chaining is determined by the right parts of the rules.

Components of a Rules-based System

Data base

Set of structured data objects

Current state of modeled part of world

Rule base

Set of rules

Application of a rule will alter the data base

Rule interpreter

Inference machine

Controls the program flow of the system

Rule Interpretation

Main scheme forward chaining

- Select and apply rules from the set of rules with valid antecedences. This will lead to a modified data base and the possibility to apply further rules.

Run this cycle as long as possible.

The process terminates, if

- there is no rule left with valid antecendence
- a solution criterion is satisfied
- a stop criterion is satisfied (e. g. maximum number of steps)

Following tasks have to be solved:

- Identify those rules with a valid condition
  $\Rightarrow$ **Instantiation** or **Matching**
- Select rules to be executed
  $\Rightarrow$ need for **conflict resolution**
  (e. g. via partial or total orderings on the rules)

Certainty Factors

Mycin (1970)

Objective: Development of a system that supports physicians in diagnosing bacterial infections and suggesting antibiotics.

Features: Uncertain knowledge was represented and processed via *uncertainty factors*.

Knowledge: 500 (uncertain) decision rules as static knowledge base.

Case-specific knowledge:

- static: patients' data
- dynamic: intermediate results (facts)

Strengths:

- diagnosis-oriented interrogation
- hypotheses generation
- finding notification
- therapy recommendation
- explanation of inference path

Uncertainty Factors

Uncertainty factor $\quad \mathrm{CF} \in [-1, 1] \quad \approx \quad$ degree of belief.

Rules:

$$\mathrm{CF}(A \rightarrow B) \begin{cases} = 1 & B \text{ is certainly true given } A \\ > 0 & A \text{ supports } B \\ = 0 & A \text{ has no influence on } B \\ < 0 & A \text{ provides evidence against } B \\ = -1 & B \text{ is certainly false given } A \end{cases}$$

A Mycin Rule

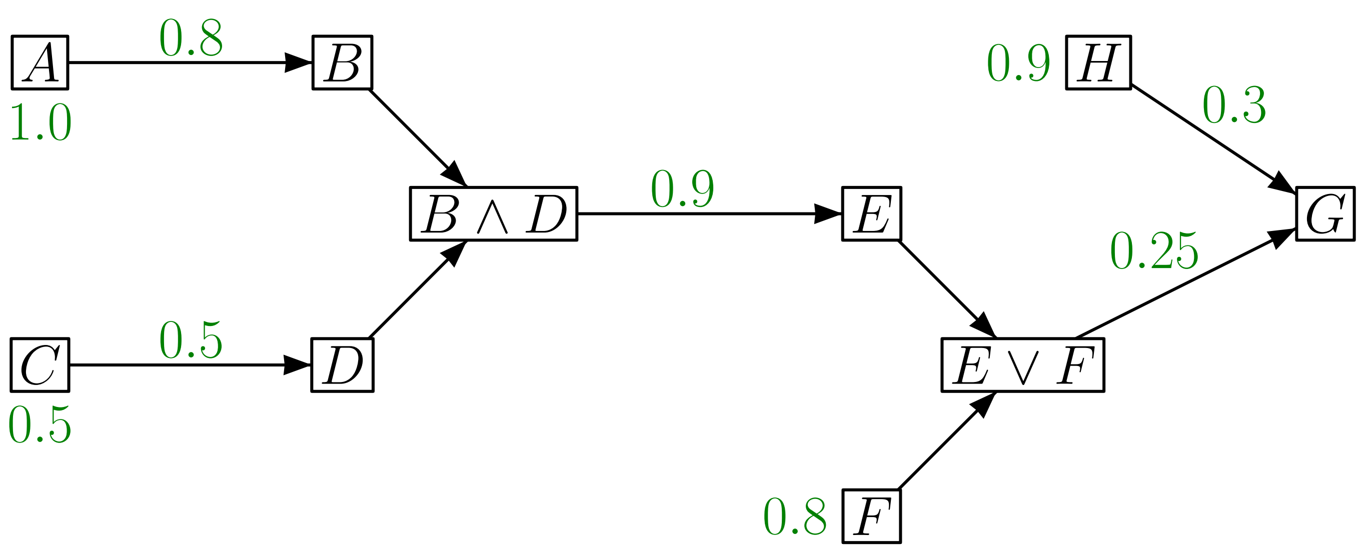
```
RULE035

PREMISE:   ($AND    (SAME CNTXT GRAM GRAMNEG)
                    (SAME CNTXT MORPH ROD)
                    (SAME CNTXT AIR ANAEROBIC))
ACTION:    (CONCL.CNTXT IDENTITY BACTEROIDES TALLY .6)
```

If	1) the *gram stain* of the organism is *gramneg*, and 2) the *morphology* of the organism is *rod*, and 3) the *aerobicity* of the organism is *anaerobic*
then	there is suggestive evidence (0.6) that the *identity* of the organism is *bacteroides*

Example

$$\begin{aligned} A &\to B\ [0.80] \\ C &\to D\ [0.50] \\ B \wedge D &\to E\ [0.90] \\ E \vee F &\to G\ [0.25] \\ H &\to G\ [0.30] \end{aligned} \qquad \begin{aligned} A\ &[1.00] \\ C\ &[0.50] \\ F\ &[0.80] \\ H\ &[0.90] \end{aligned}$$



Propagation Rules

Conjunction: $\mathrm{CF}(A \wedge B) = \min\{\mathrm{CF}(A), \mathrm{CF}(B)\}$

Disjunction: $\mathrm{CF}(A \vee B) = \max\{\mathrm{CF}(A), \mathrm{CF}(B)\}$

Serial Combination: $\mathrm{CF}(B, \{A\}) = \mathrm{CF}(A \to B) \cdot \max\{0, \mathrm{CF}(A)\}$

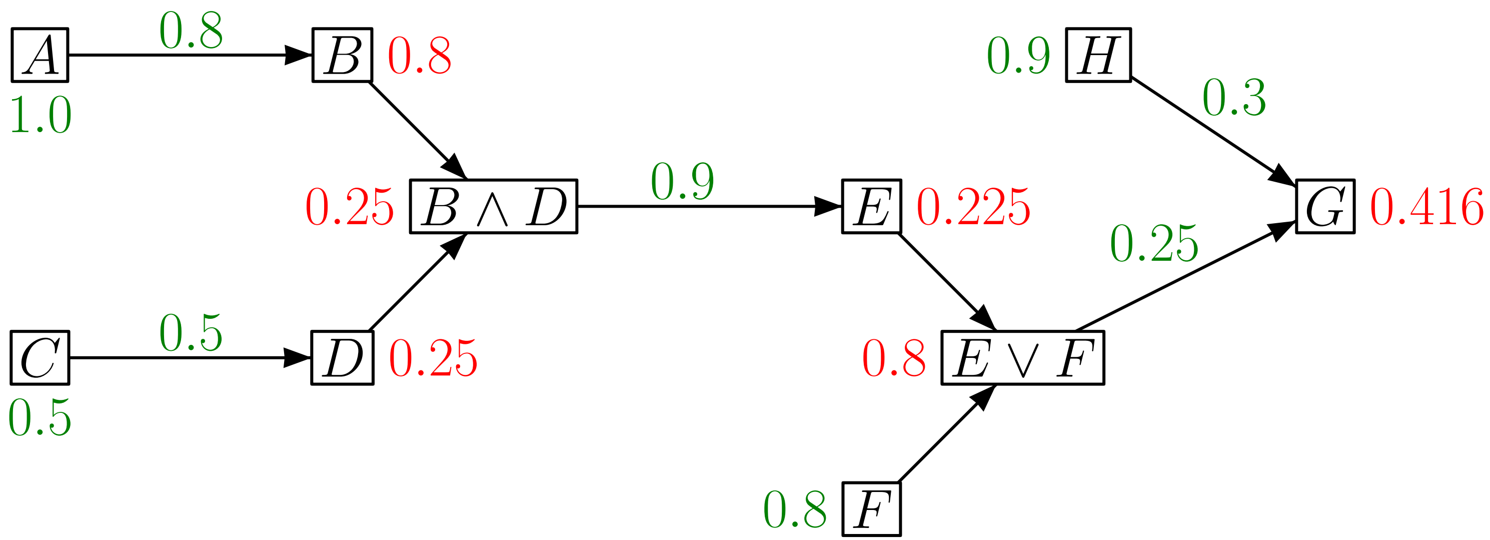
Parallel Combination: for $n > 1$:

$$\mathrm{CF}(B, \{A_1, \ldots, A_n\}) = f(\mathrm{CF}(B, \{A_1, \ldots, A_{n-1}\}), \mathrm{CF}(B, \{A_n\}))$$

with

$$f(x, y) = \begin{cases} x + y - xy & \text{if} \quad x, y > 0 \\ x + y + xy & \text{if} \quad x, y < 0 \\ \dfrac{x + y}{1 - \min\{|x|, |y|\}} & \text{otherwise} \end{cases}$$

Example (cont.)



$$f(0.3 \cdot 0.9, 0.25 \cdot 0.8) \; = \; 0.27 + 0.2 - 0.27 \cdot 0.2 = 0.416$$

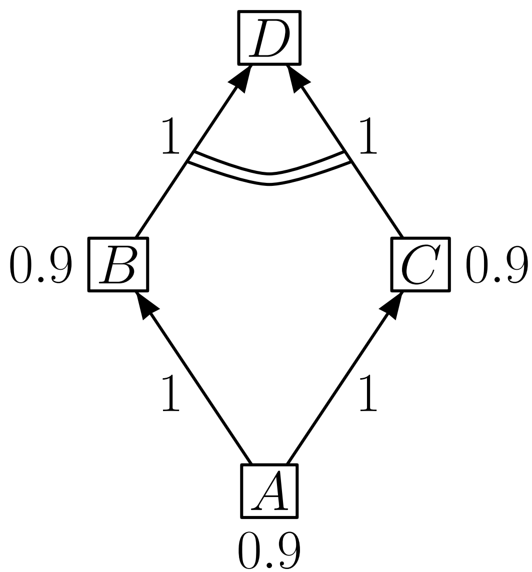
Was Mycin a failure?

It worked in the Mycin case because the rules had tree-like structure.

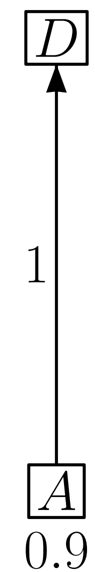
It can be shown that the rule combination scheme is inconsistent in general.

Example: $\mathrm{CF}(A) = 0.9$, $\mathrm{CF}(D) = ?$

$\mathrm{CF}(D) = 0.9 + 0.9 - 0.9 \cdot 0.9 = 0.99$ $\qquad$ $\mathrm{CF}(D) = 0.9$



vs.



Certainty factor is increased just because (the same) evidence is transferred over different (parallel) paths!

Was Mycin a failure?

Mycin was never used for its intended purpose, because

- physicians were distrustful and not willing to accept Mycin's recommendations.
- Mycin was too good.

However,

- Mycin was a milestone for the development of expert systems.
- it gave rise to impulses for expert system development in general.