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:

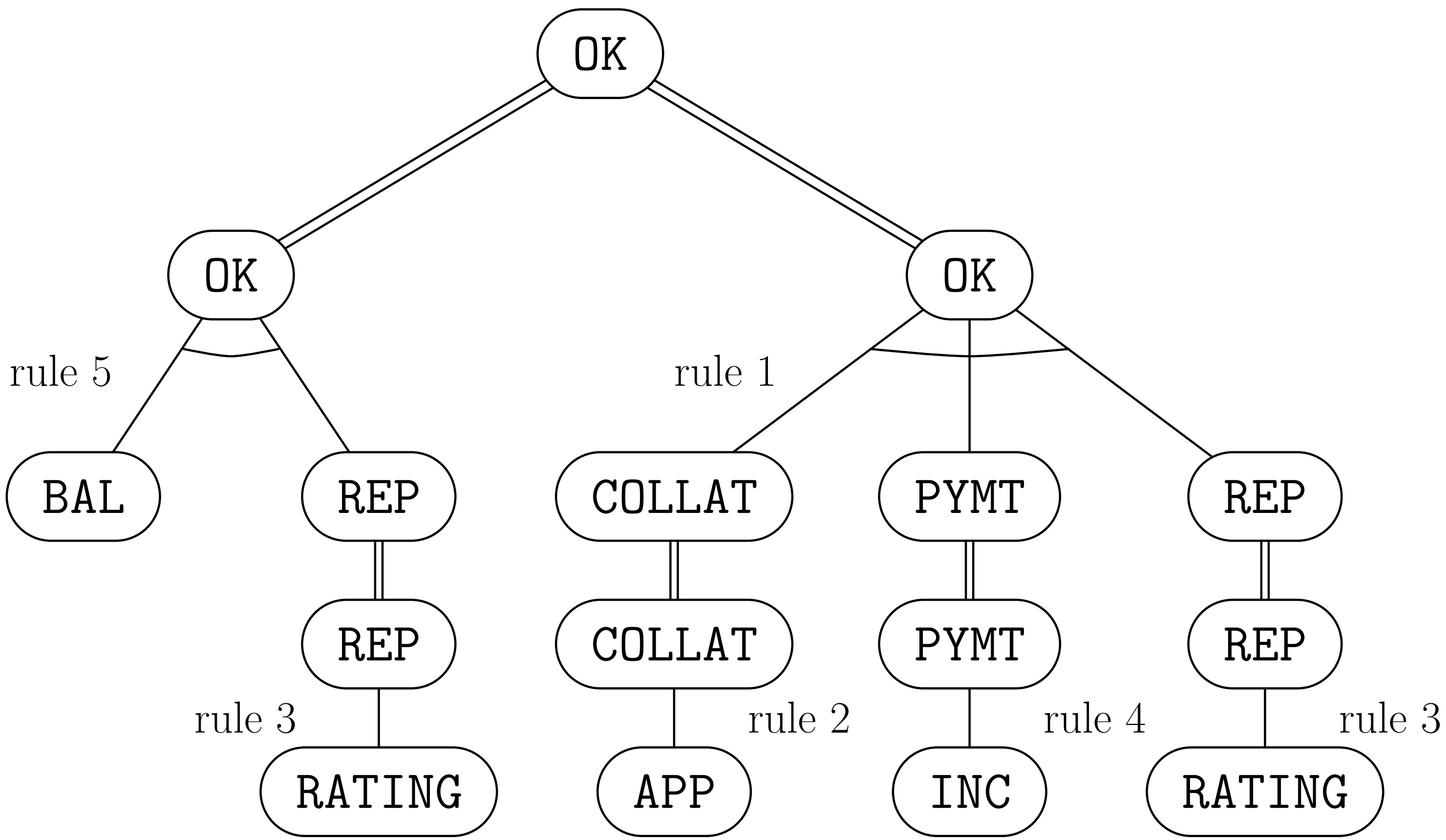
1. `COLLAT` $\wedge$ `PYMT` $\wedge$ `REP` $\Rightarrow$ `OK`
2. `APP` $\Rightarrow$ `COLLAT`
3. `RATING` $\Rightarrow$ `REP`
4. `INC` $\Rightarrow$ `PYMT`
5. `BAL` $\wedge$ `REP` $\Rightarrow$ `OK`

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

Decision Rule Tree

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*.

$$\boxed{\text{If } \textit{condition} \text{ then } \textit{conclusion}}$$

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:

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

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)
```

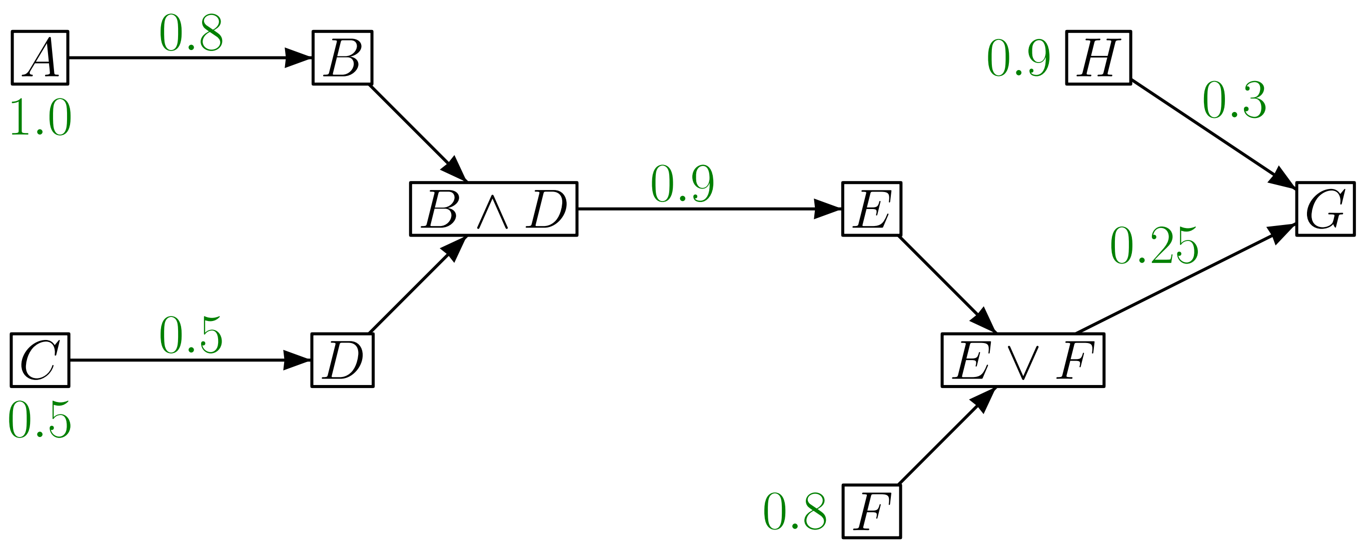
If 1) the *gram stain* of the organism is *gramneg*, and
2) the *morphology* of the organism is *rod*, and
3) the *aerobicity* of the organism is *anaerobic*

then there is suggestive evidence (0.6) that the *identity* of the organism is *bacteroides*

Example

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

$$\begin{aligned} A\,&[1.00] \\ C\,&[0.50] \\ F\,&[0.80] \\ H\,&[0.90] \end{aligned}$$



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

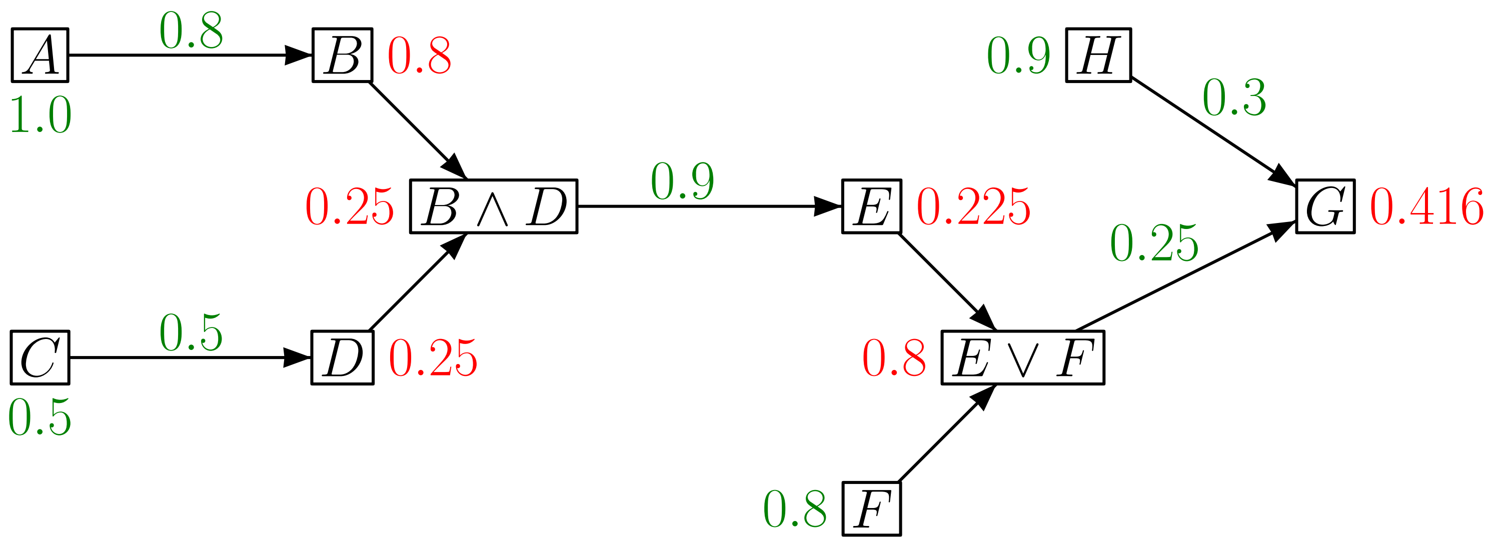
Parallel Combination: for $n > 1$:

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

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.)



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

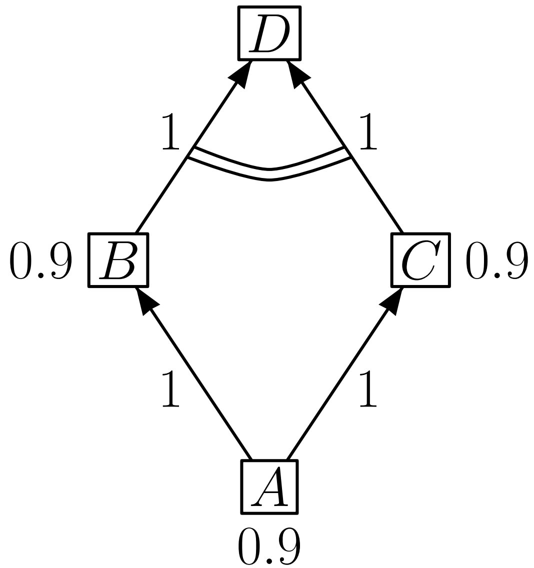
Was Mycin a failure?

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

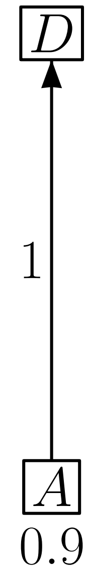
It can be shown that the rule combination scheme is inconsistent in general.

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.

Elements of Graph Theory