

# MYCIN

AN EXPERT SYSTEM

# MYCIN

- **MYCIN** is a **computer program** designed to provide attending physicians with **advice** comparable to that which they would otherwise get from a consulting physician in the diagnosis and **treatment of meningitis and bacteraemia infections.**
- written in **LISP**
- the name derived from the antibiotics themselves, as many antibiotics have the suffix "mycin".
- the first large expert system to perform at the level of a human expert
- MYCIN was developed at Stanford University in the mid-1970s

# HOW IS MYCIN USED?

1. The attending physician must sit in front of a computer terminal that is connected to a DEC-20 (one of Digital Equipment Corporation's mainframe computers) where the MYCIN program is stored.
2. When the MYCIN program is evoked, it initiates a dialogue.
3. The physician types answers in response to various questions.

**OUTPUT:** MYCIN provides a diagnosis and a detailed drug therapy recommendation.

# Some features of MYCIN

- MYCIN accommodates situations where the physician does not have results of some tests by accepting **UNKNOWN** as a response.
- Like a human specialist, MYCIN will reason with **incomplete information**.
- To accommodate uncertainty, all information given to MYCIN may be qualified by a **certainty factor**- a number between - 1 and + 1 that indicates the doctor's degree of confidence in the answer to a question.
- Users can ask "**WHY?**" when MYCIN is asking for data, and MYCIN will explain what hypotheses it is considering.
- **MYCIN can make a prescription**, explain its reasoning, and propose additional tests and procedures.

# HOW DOES MYCIN WORK?

- It uses **production rules** to reason **backwards** from goals to clinical observations.
- To solve a top-level diagnostic goal, it looks for rules whose right sides suggest disease.
- It also looks for the left side of these rules, sets sub-goals to check if the left side holds true, which in turn would invoke the corresponding rule.
- It learns from the input given by users and avoids asking redundant or already discovered observations.
- MYCIN uses **backward chaining** to discover what organisms are present.
- It uses **forward chaining** to reason from organisms-causing the infection to treatment regime.

# MYCIN'S FRONT END - TEIRESIAS

- Teiresias was first to support MYCIN's explanation and knowledge acquisition.
- Behaviour of the program is explained by referring to a trace in program's execution.
- As the user keeps giving inputs and asks for answers, Teiresias is responsible for interpreting the user's answers efficiently. MYCIN's job is to maintain the knowledge base and learn from the given facts.



Is the patient's illness with ORGANISM-1 a hospital-acquired infection?

**\*\* WHY**

[i.e., WHY is it important to determine whether or not the infection with ORGANISM-1 was acquired while the patient was hospitalized?]

[2.0] This will aid in determining the category of ORGANISM-1. It has already been established that

[2.1] the gram stain of ORGANISM-1 is gramneg, and

[2.2] the morphology of ORGANISM-1 is rod, and

[2.3] the aerobicity of ORGANISM-1 is facultative

Therefore, if

[2.4] the infection with ORGANISM-1 was not acquired while the patient was hospitalized, then

there is strongly suggestive evidence (.8) that the category of ORGANISM-1 is enterobacteriaceae

[ALSO: there is weakly suggestive evidence (.1) that the identity of ORGANISM-1 is pseudomonas-aeruginosa]

[RULE037]

**\*\* HOW [2.3]**

[i.e., HOW was it established that the aerobicity of ORGANISM-1 is facultative?]

The following were used:

[3.1] RULE027 indicated there is strongly suggestive evidence (.8) that the aerobicity of ORGANISM-1 is facultative.

[3.2] RULE033 indicated there is weakly suggestive evidence (.2) that the aerobicity of ORGANISM-1 is not facultative.

Since this gave a cumulative CF of (.6) for facultative, the evidence indicates that the aerobicity of ORGANISM-1 is facultative.

# EMYCIN

- Several expert systems had a lot in common since the systems were constructed as a set of declarative representations combined with some interpreter for these representations, the interpreter could be separated from domain-specific knowledge.
- Example : EMYCIN ( empty MYCIN )
- These are called shells.
- Could be used to build new expert systems to avoid repetitive work.



# REFERENCES

- <https://en.wikipedia.org/wiki/Mycin>
- Research paper with mycin and a physician's conversation:  
<http://psy.haifa.ac.il/~ep/Lecture%20Files/AI/Secure/Download/Introduction%20to%20expert%20systems%20-%20MYCIN.pdf>
- Artificial Intelligence-3<sup>rd</sup> edition by Elaine Rich, Kevin Knight and Shivashankar Nair.