8 Strong Method Problem Solving

Additional references for the slides:
Robert Wilensky's CS188 slides:
www.cs.berkeley.edu/~wilensky/cs188/lectures/index.html
Russell and Norvig's AI book, 2003

Chapter Objectives

- Learn about knowledge-intensive AI applications
- Learn about the issues in building Expert Systems: knowledge engineering, inference, providing explanations
- Learn about the issues in building Planning Systems: writing operators, plan generation, monitoring execution
- The agent models:
  can perform "expert quality" problem solving;
  can generate and monitor plans

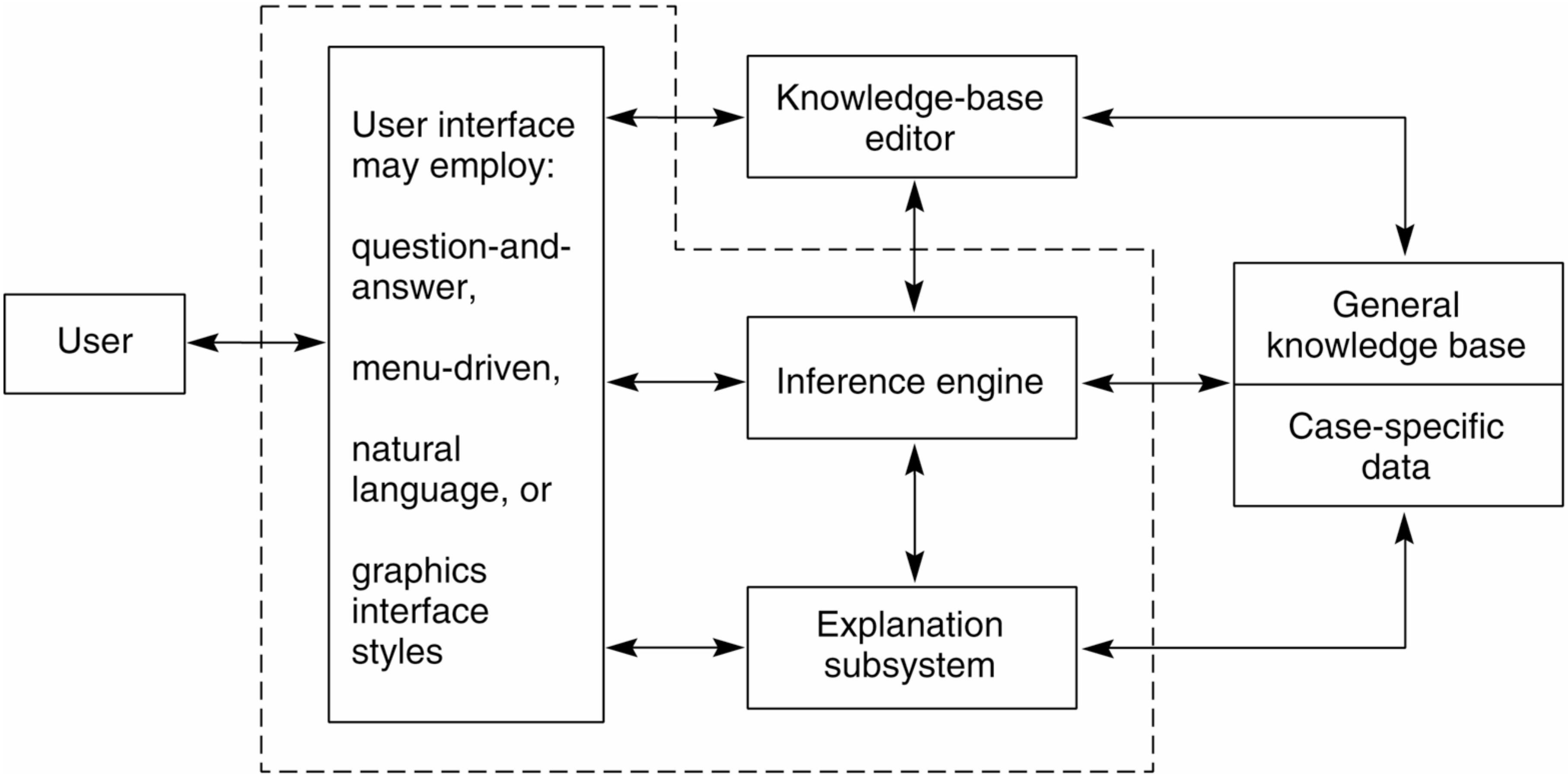
Expert systems (ESs) - motivations

- Experts usually have a lot of knowledge, why not build a system that incorporates a lot of knowledge in a specific area.
- Will attempt to solve a problem that is
  - non-trivial
  - complex
  - poorly understood
- The resulting system will be
  - fast
  - reliable
  - cheap
  - transportable
  - usable in remote sites

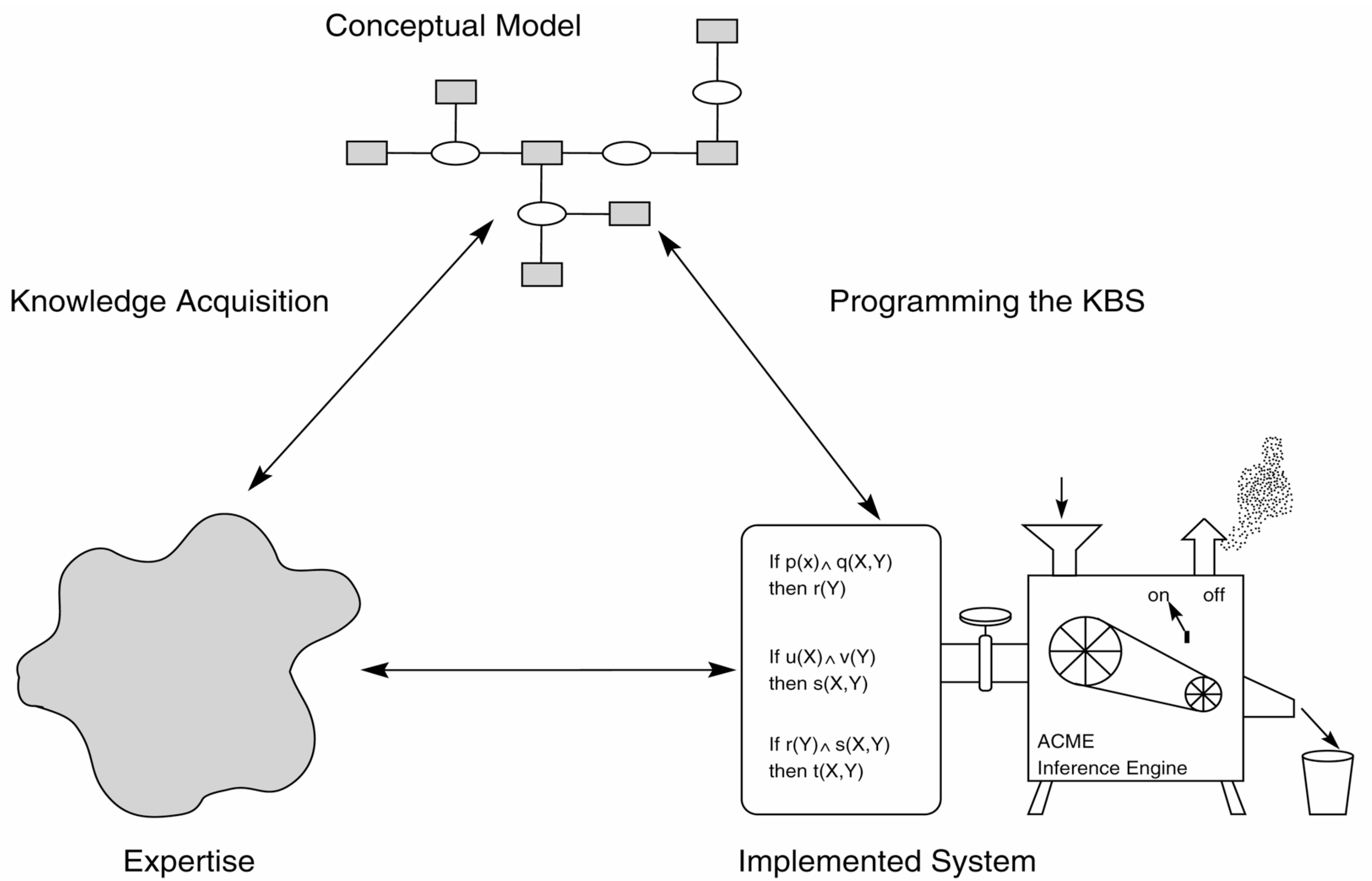
What is in an expert system?

- lots of knowledge
- a production system architecture
- inference techniques
- advanced features for the user
  - should make their job easier
  - explanations

Architecture of a typical expert system



The role of mental or conceptual models in problem solving



A small ES for diagnosing automotive problems

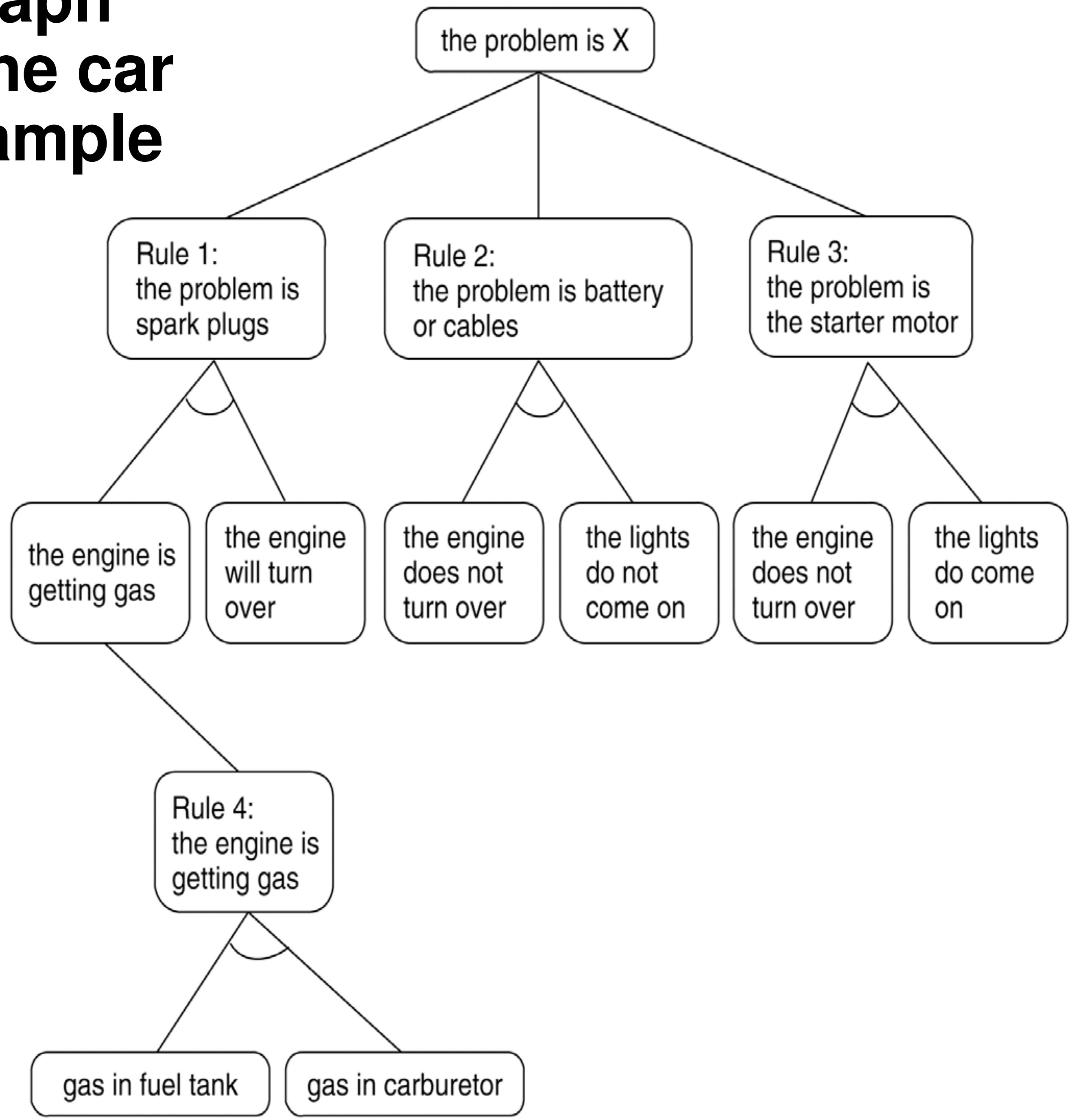
If
the engine is getting gas**, and**
the engine will turn over**,**
then
the problem is spark plugs
1

If
the engine does not turn over**, and**
the lights do not come on
then
the problem is battery or cables
2

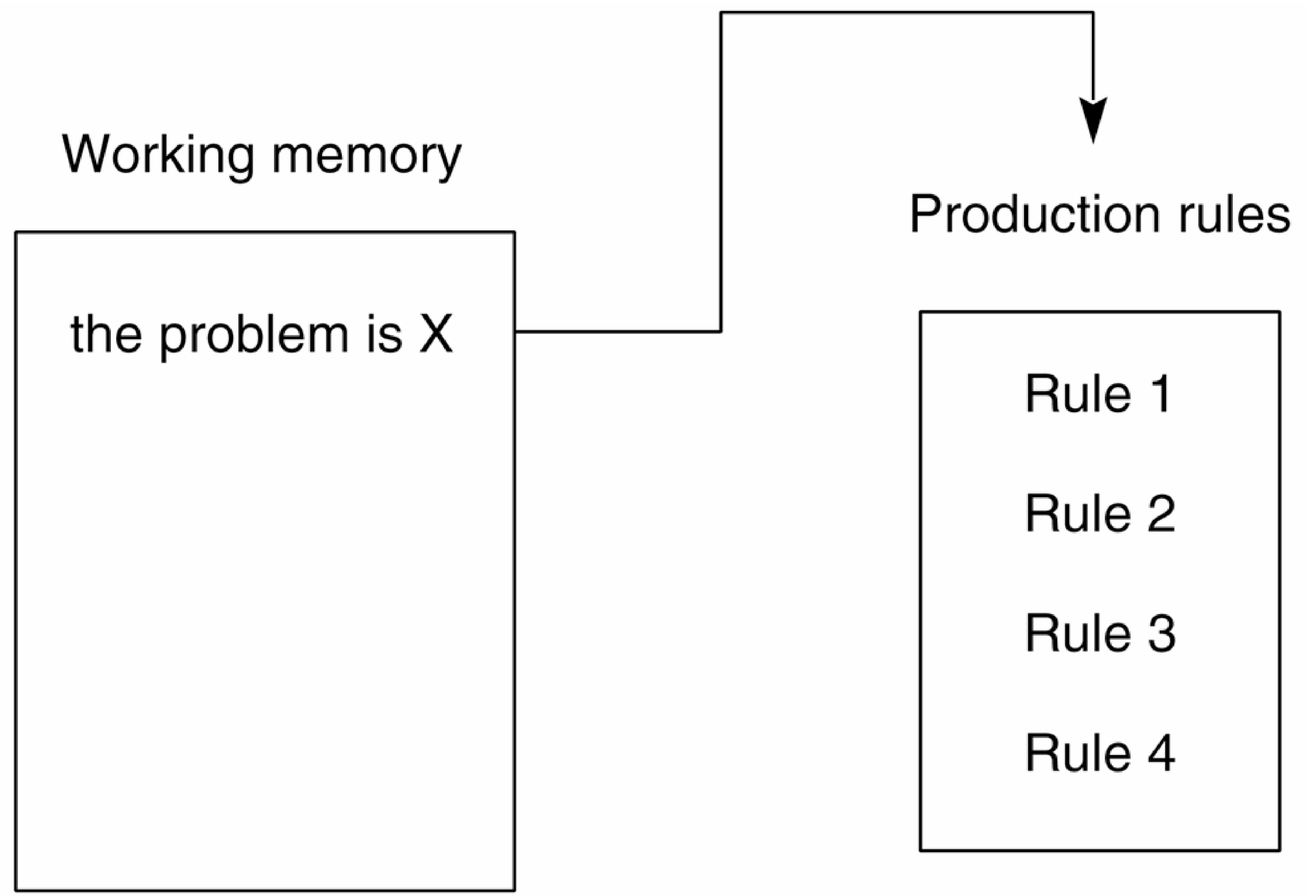
If
the engine does not turn over**, and**
the lights do come on**,**
then
the problem is the starter motor
3

If
there is gas in the fuel tank**, and**
there is gas in the carburetor
then
the engine is getting gas
4

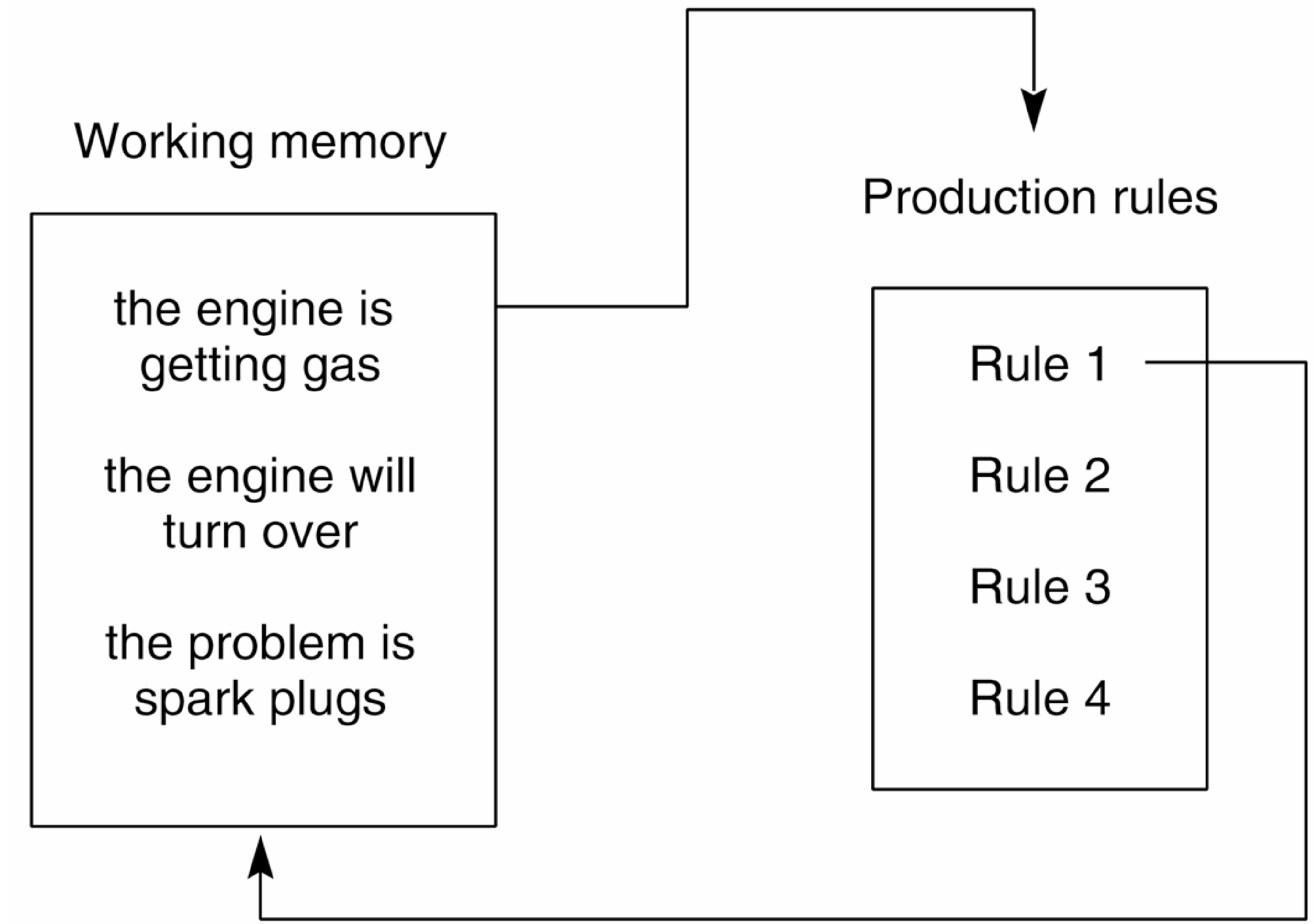
The and/or graph searched in the car diagnosis example



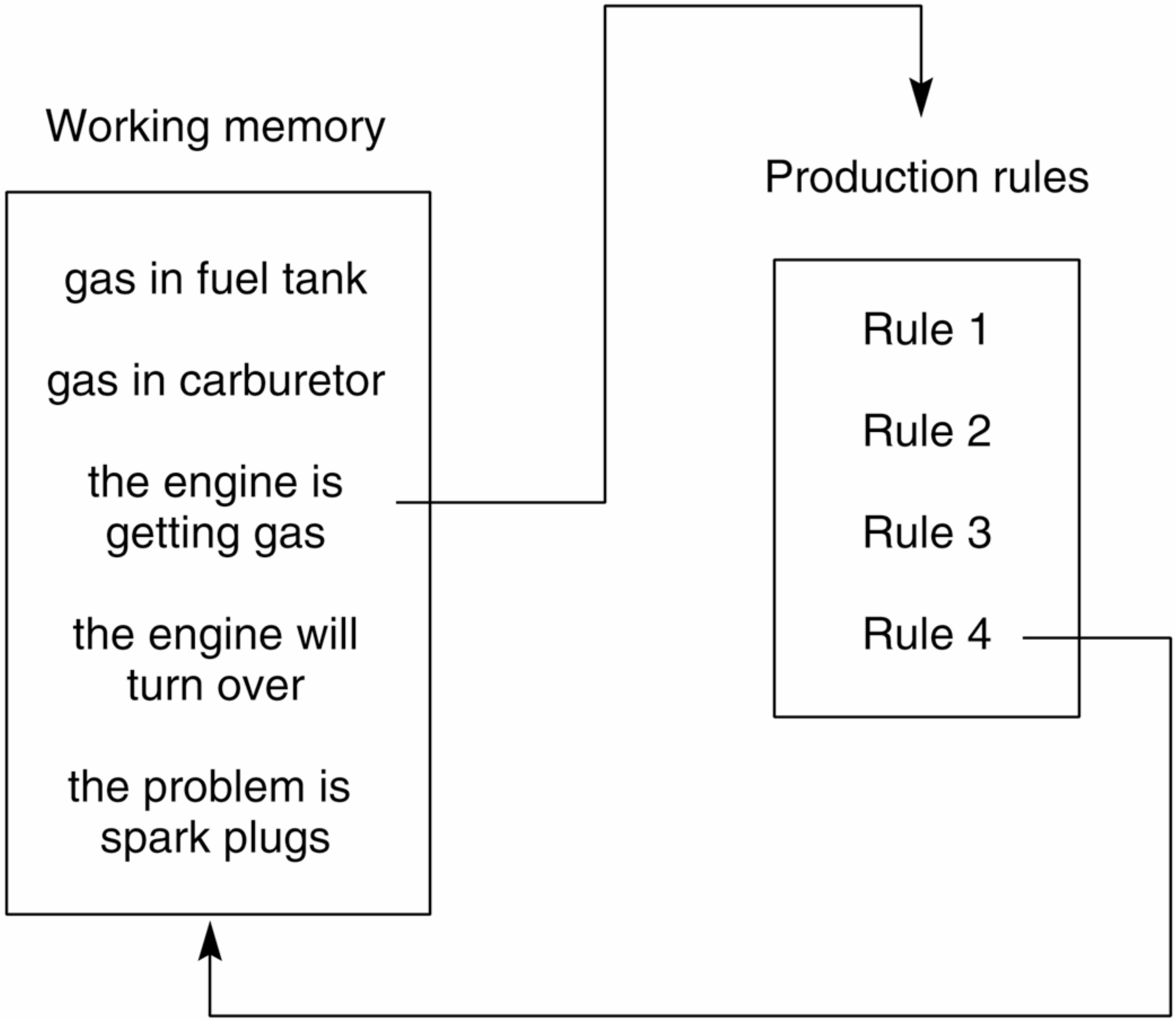
The production system at the start of a consultation (it will be DFS)



The production system after Rule 1 has fired



The system after Rule 4 has fired. Note the stack-based approach to goal reduction



Explanation: Answering “why” questions

ES - Gas in fuel tank?
U- yes
ES- Gas in carburetor?
U- yes
ES- Engine will turn over?
U- why

ES- It has been established that
1. The engine is getting gas,
therefore if
2. The engine will turn over,
then the problem is spark plugs.

The user is asking
“why is this information important?”

Explanation: Answering "how" questions

U- how the engine is getting gas

ES- This follows from rule 4:
if
gas in fuel tank, and
there is gas in the carburetor
then
the engine is getting gas

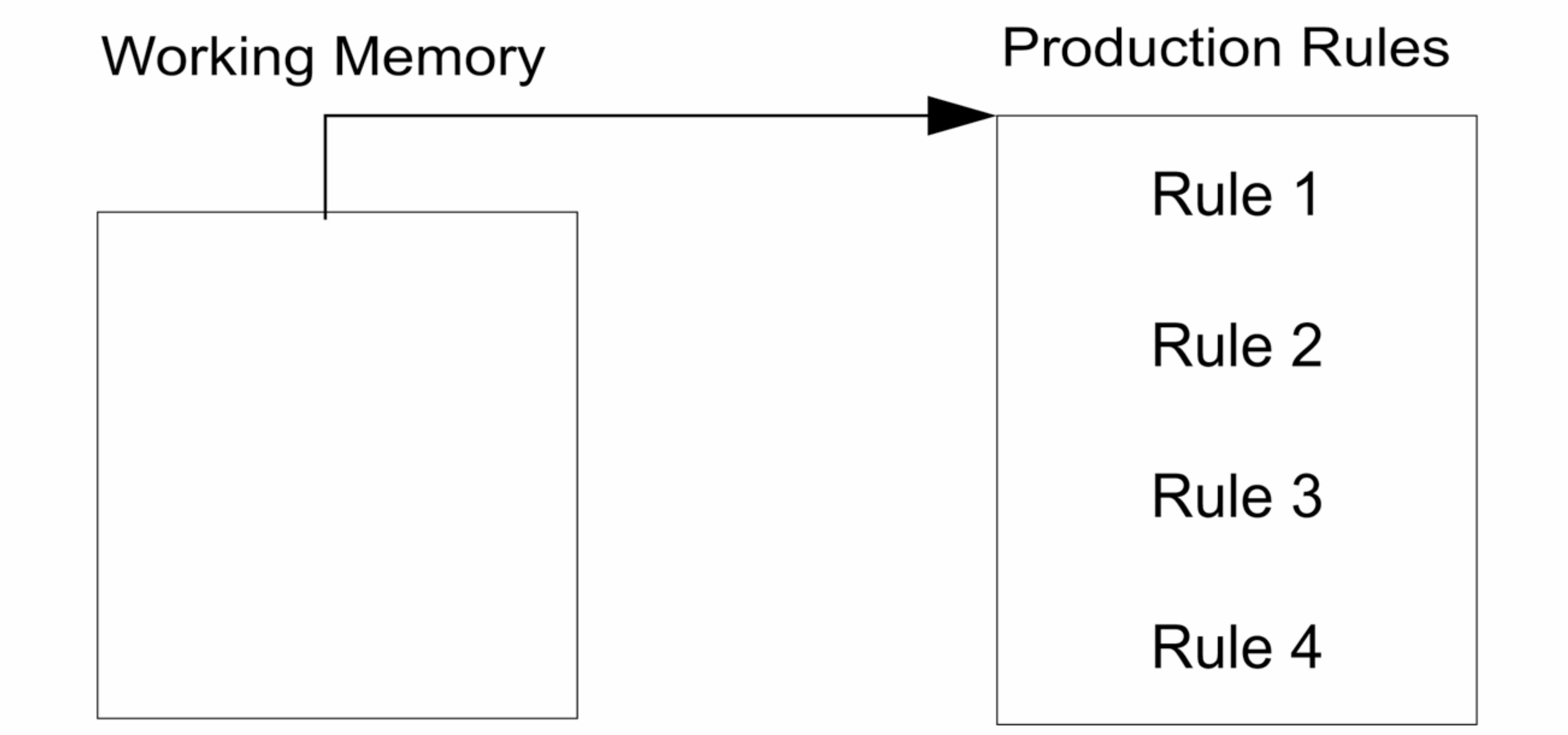
gas in fuel tank was given by the user
gas in the carburetor was given by the user

The user is asking
"How did the system come up with this conclusion?"

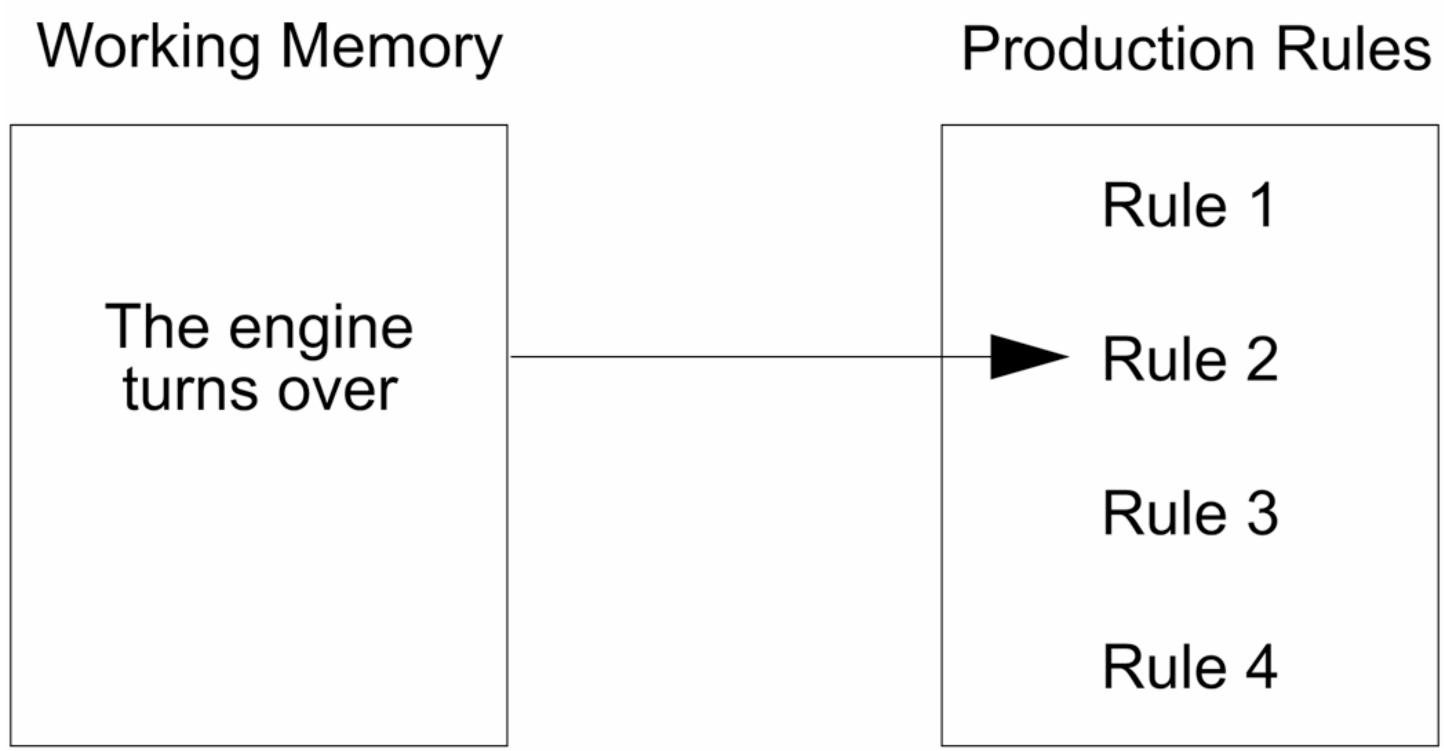
Data-driven reasoning in ESs

- **Use breadth-first search**
- **Algorithm:**
  - **Do the next step until the working memory does not change anymore**
  - **For each rule:**
    - Compare the contents of the working memory with the conditions of each rule in the rule base using the ordering of the rule base.
    - If the data in working memory supports a rule's firing place the result in working memory

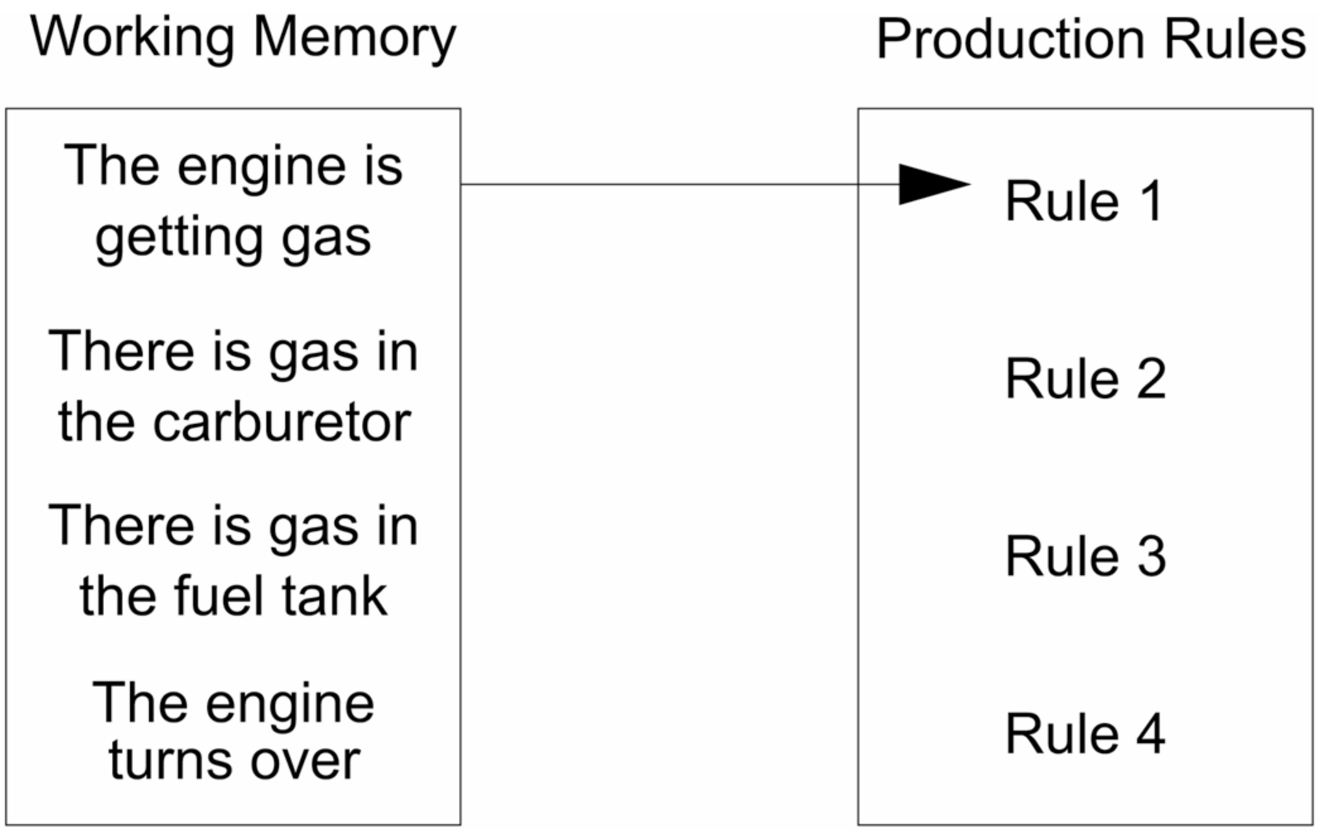
At the start of a consultation for data-driven reasoning (Fig. 7.9)



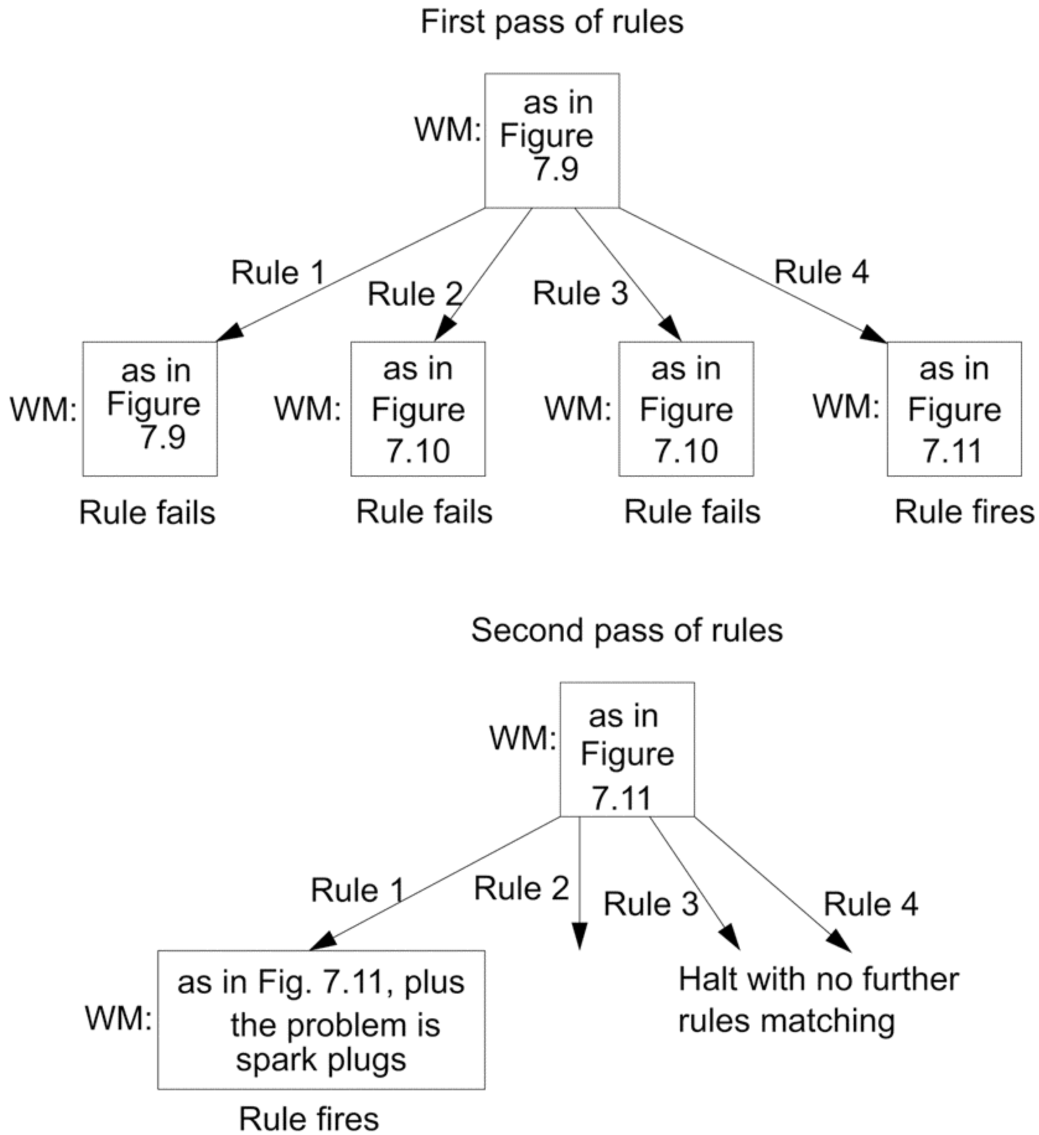
After evaluating the first premise of Rule 2, which then fails (Fig. 7.10)



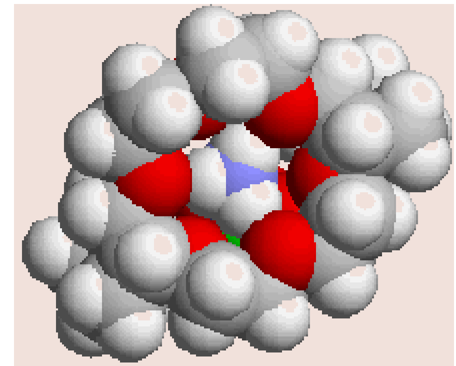
After considering Rule 4, beginning its second pass through the rules (Fig. 7.11)



The search graph as described by the contents of WM data-driven BFS



ES examples – DENDRAL (Russell & Norvig, 2003)



DENDRAL is the earliest ES
(project 1965- 1980)

Developed at Stanford by Ed Feigenbaum, Bruce Buchanan, Joshua Lederberg, G.L. Sutherland, Carl Djerassi.

Problem solved: inferring molecular structure from the information provided by a mass spectrometer. This is an important problem because the chemical and physical properties of compounds are determined not just by their constituent atoms, but by the arrangement of these atoms as well.

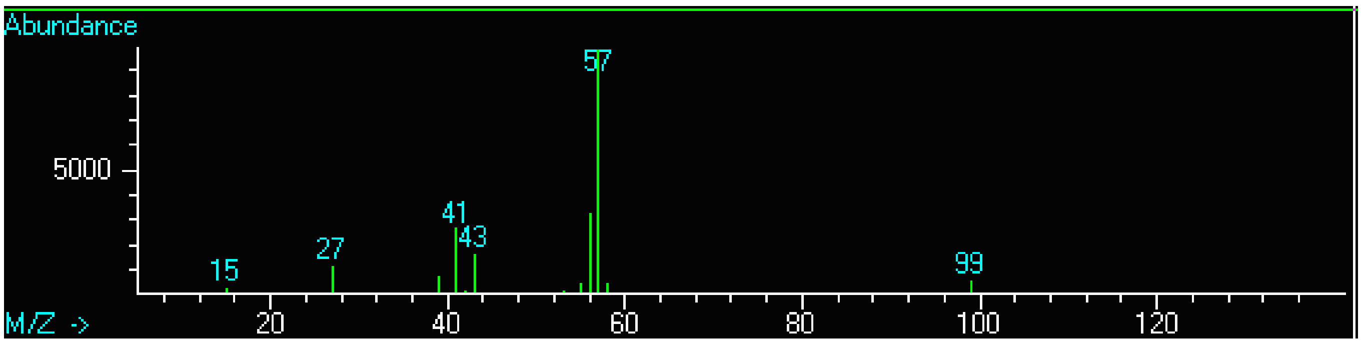
ES examples – DENDRAL (cont'd)

Inputs:

- **elementary formula of the molecule**
  e.g., $C_6H_{13}NO_2$

- **the mass spectrum giving the masses of the various fragments of the molecule generated when it is bombarded by an electron beam**
  e.g., the mass spectrum might contain a peak at m=15, corresponding to the mass of a methyl (CH_3) fragment.

Mass spectrum



Shows the distribution of ions

Y axis: signal intensity

X axis: atomic weight (amu – atomic mass unit)

ES examples - DENDRAL (cont'd)

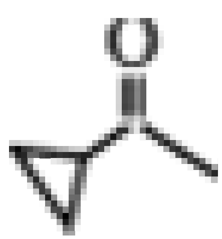
Naïve version: DENDRAL stands for DENDritic Algorithm: a procedure to exhaustively and nonredundantly enumerate all the topologically distinct arrangements of any given set of atoms. Generate all the possible structures consistent with the formula, predict what mass spectrum would be observed for each, compare this with the actual spectrum.
This is intractable for large molecules!

Improved version: look for well-known patterns of peaks in the spectrum that suggested common substructures in the molecule. This reduces the number of possible candidates enormously.

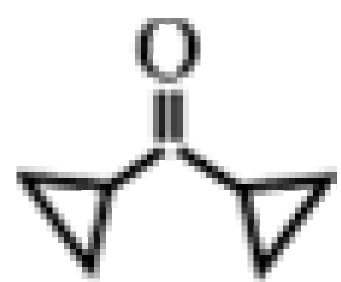
ES examples - DENDRAL (cont'd)

A rule to recognize a ketone (C=0) subgroup (weighs 28)

if there are two peaks at x1 and x2 such that
(a) x1 + x2 = M + 28 (M is the mass of the whole molecule);
(b) x1 - 28 is a high peak
(c) x2 - 28 is a high peak
(d) at least one of x1 and x2 is high
then there is a ketone subgroup



Cyclopropyl-methyl-ketone



Dicyclopropyl-methyl-ketone

ES examples - MYCIN

MYCIN is another well known ES.

Developed at Stanford by Ed Feigenbaum, Bruce Buchanan, Dr. Edward Shortliffe.

Problem solved: diagnose blood infections. This is an important problem because physicians usually must begin antibiotic treatment without knowing what the organism is (laboratory cultures take time). They have two choices:

(1) prescribe a broad spectrum drug
(2) prescribe a disease-specific drug (better)

.

ES examples - MYCIN (cont'd)

Differences from DENDRAL:

- **No general theoretical model existed from which MYCIN rules could be deduced. They had to be acquired from extensive interviewing of experts, who in turn acquired them from textbooks, other experts, and direct experience of cases.**
- **The rules reflected uncertainty associated with medical knowledge: *certainty factor* (not a sound theory)**

ES examples - MYCIN (cont'd)

About 450 rules. One example is:

If
the site of the culture is blood
the gram of the organism is neg
the morphology of the organism is rod
the burn of the patient is serious
then
there is weakly suggestive evidence (0.4) that the identity of the organism is pseudomonas.

ES examples - MYCIN (cont'd)

If

the infection which requires therapy is meningitis
only circumstantial evidence is available for this case
the type of the infection is bacterial
the patient is receiving corticosteroids

then

there is evidence that the organisms which might be causing the infection are e.coli(0.4), klebsiella-pneumonia(0.2), or pseudomonas-aeruginosa(0.1).

ES examples - MYCIN (cont'd)

Starting rule: "If there is an organism requiring therapy, then, compute the possible therapies and pick the best one."

It first tries to see if the disease is known. Otherwise, tries to find it out.

ES examples - MYCIN (cont'd)

Can ask questions during the process:

> **What is the patient's name?**
John Doe.
> **Male or female?**
Male.
> **Age?**
He is 55.
> **Have you obtained positive cultures indicating general type?**
Yes.
> **What type of infection is it?**
Primary bacteremia.

ES examples - MYCIN (cont'd)

> Let's call the first significant organism from this culture U1. Do you know the identity of U1?

No.

> Is U1 a rod or a coccus or something else?

Rod.

> What is the Gram stain of U1?

Gram-negative.

In the last two questions, it is trying to ask the most general question possible, so that repeated questions of the same type do not annoy the user. The format of the KB should make the questions reasonable.

ES examples - MYCIN (cont'd)

Studies about its performance showed its recommendations were as well as some experts, and considerably better than junior doctors.

Could calculate drug dosages very precisely.

Dealt well with drug interactions.

Had good explanation features and rule acquisition systems.

Was narrow in scope (not a large set of diseases). Another expert system, INTERNIST, knows about internal medicine.

Issues in usability, doctors' egos, legal aspects.

Asking questions to the user

- Which questions should be asked and in what order?

- Try to ask questions to make facilitate a more comfortable dialogue. For instance, ask related questions rather than bouncing between unrelated topics (e.g., zipcode as part of an address or to relate the evidence to the area the patient lives).

ES examples - R1 (or XCON)

The first commercial expert system (~1982).

Developed at Digital Equipment Corporation (DEC).

Problem solved: Configure orders for new computer systems. Each customer order was generally a variety of computer products not guaranteed to be compatible with one another (conversion cards, cabling, support software…)

By 1986, it was saving the company $40 million a year. Previously, each customer shipment had to be tested for compatibility as an assembly before being shipped. By 1988, DEC's AI group had 40 expert systems deployed.

ES examples - R1 (or XCON) (cont'd)

Rules to match computers and their peripherals:

"If the Stockman 800 printer and DPK202 computer have been selected, add a printer conversion card, because they are not compatible."

Being able to change the rule base easily was an important issue because the products were always changing.

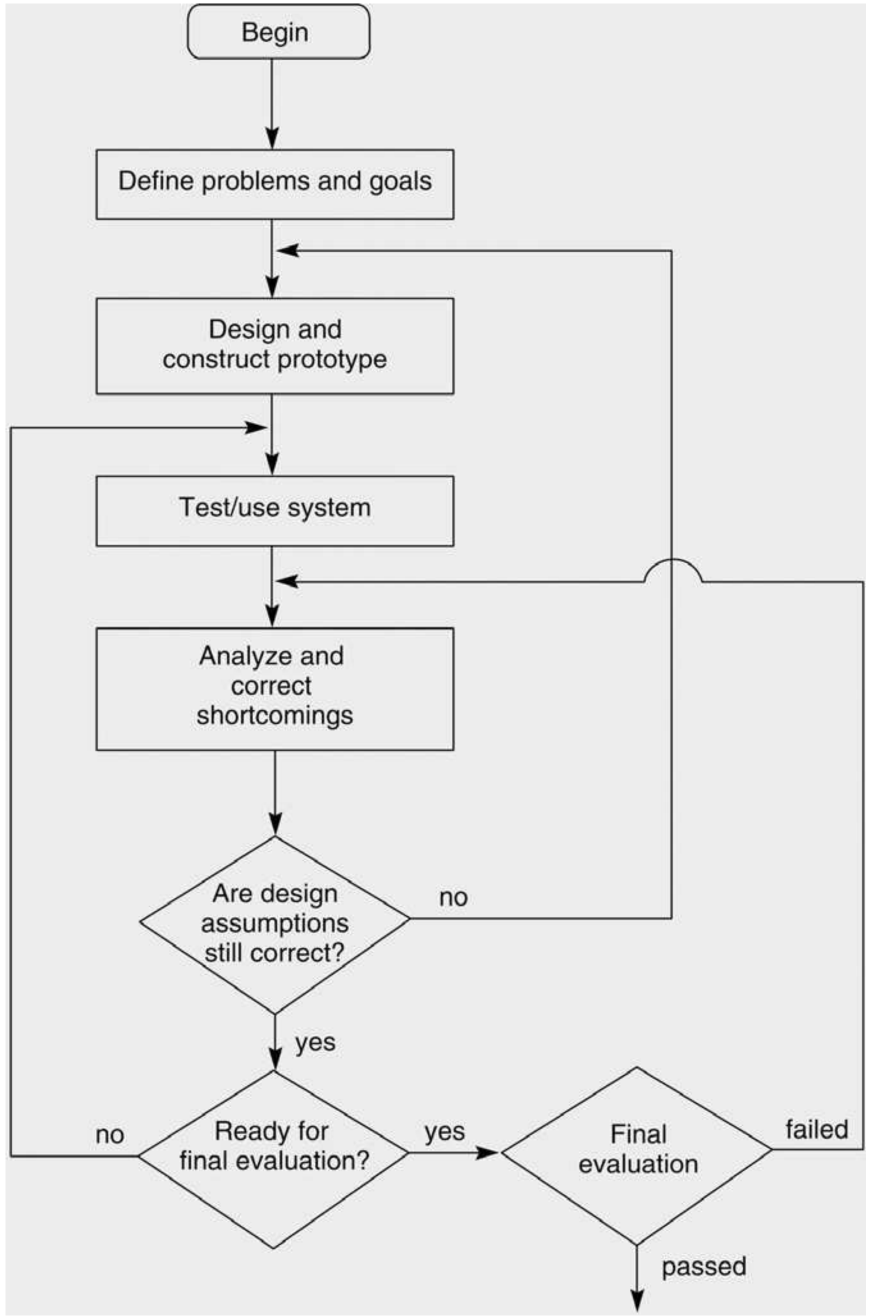
Over 99% of the configurations were reported to be accurate. Errors were due to lack of product information on recent products (easily correctible.) Like MYCIN, performs as well as or better than most experts.

6,000 - 10,000 rules.

Is an Expert System the right solution?

- The need for the solution justifies the cost and effort of building an expert system.
- Human expertise is not available in all situations where it is needed.
- The problem may be solved using symbolic reasoning.
- The problem domain is well structured and does not require commonsense reasoning.
- The problem may not be solved using traditional computing methods.
- Cooperative and articulate experts exist.
- The problem is of proper size and scope.

Exploratory development cycle



Expert Systems: then and now

- The AI industry boomed from a few million dollars in 1980 to billions of dollars in 1988.
- Nearly every major U.S. corporation had its own AI group and was either using or investigating expert systems.
- For instance, Du Pont had 100 ESs in use and 500 in development, saving an estimated $10 million per year.
- AAAI had 15,000 members during the "expert systems craze."
- Soon a period called the "AI Winter" came …BIRRR...

Expert Systems: then and now (cont'd)

- The AI industry has shifted focus and stabilized (AAAI members 5500- 7000)
- Expert systems continue to save companies money
    - IBM's San Jose facility has an ES that diagnoses problems on disk drives
    - Pac Bell's diagnoses computer network problems
    - Boeing's tells workers how to assemble electrical connectors
    - American Express Co's helps in card application approvals
    - Met Life's processes mortgage applications
- ***Expert Sytem Shells***: abstract away the details to produce an inference engine that might be useful for other tasks. Many are available.

Heuristics and control in expert systems

- organization of a rule's premises
- rule order
- costs of different tests
- which rules to select:
  - refraction
  - recency
  - specificity
- restrict potentially usable rules

Model-based reasoning

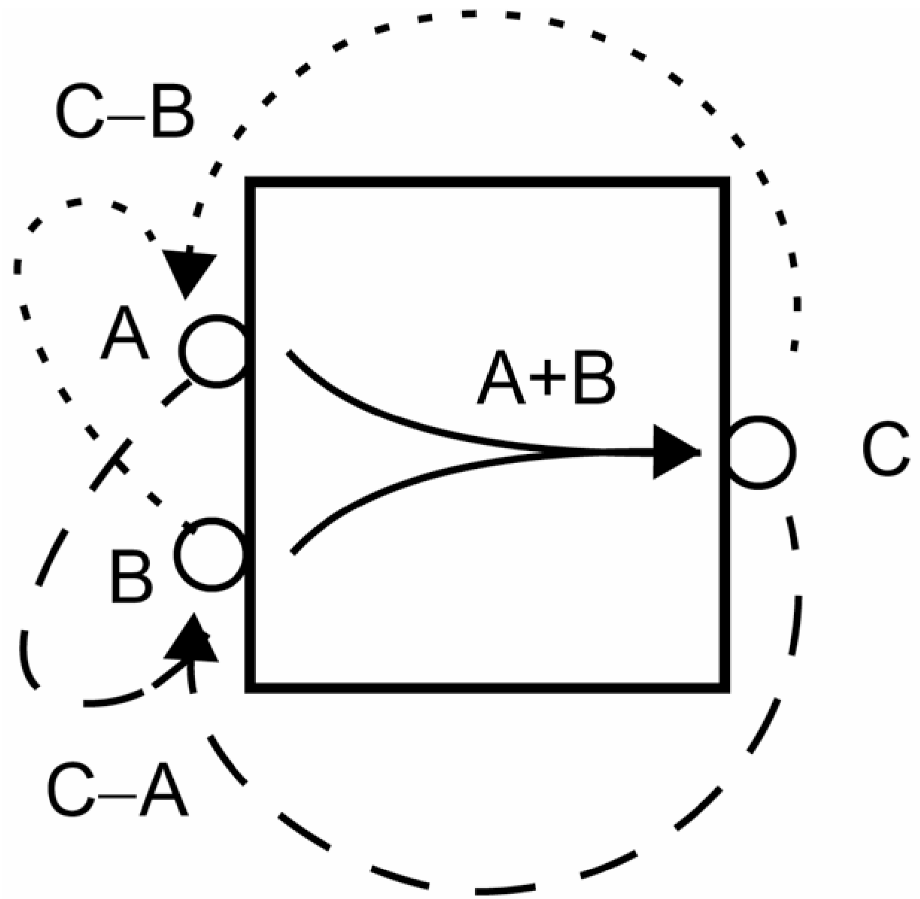
Attempt to describe the “inner details” of the system.

This way, the expert system (or any other knowledge-intensive program) can revert to first principles, and can still make inferences if rules summarizing the situation are not present.

Include a description of:

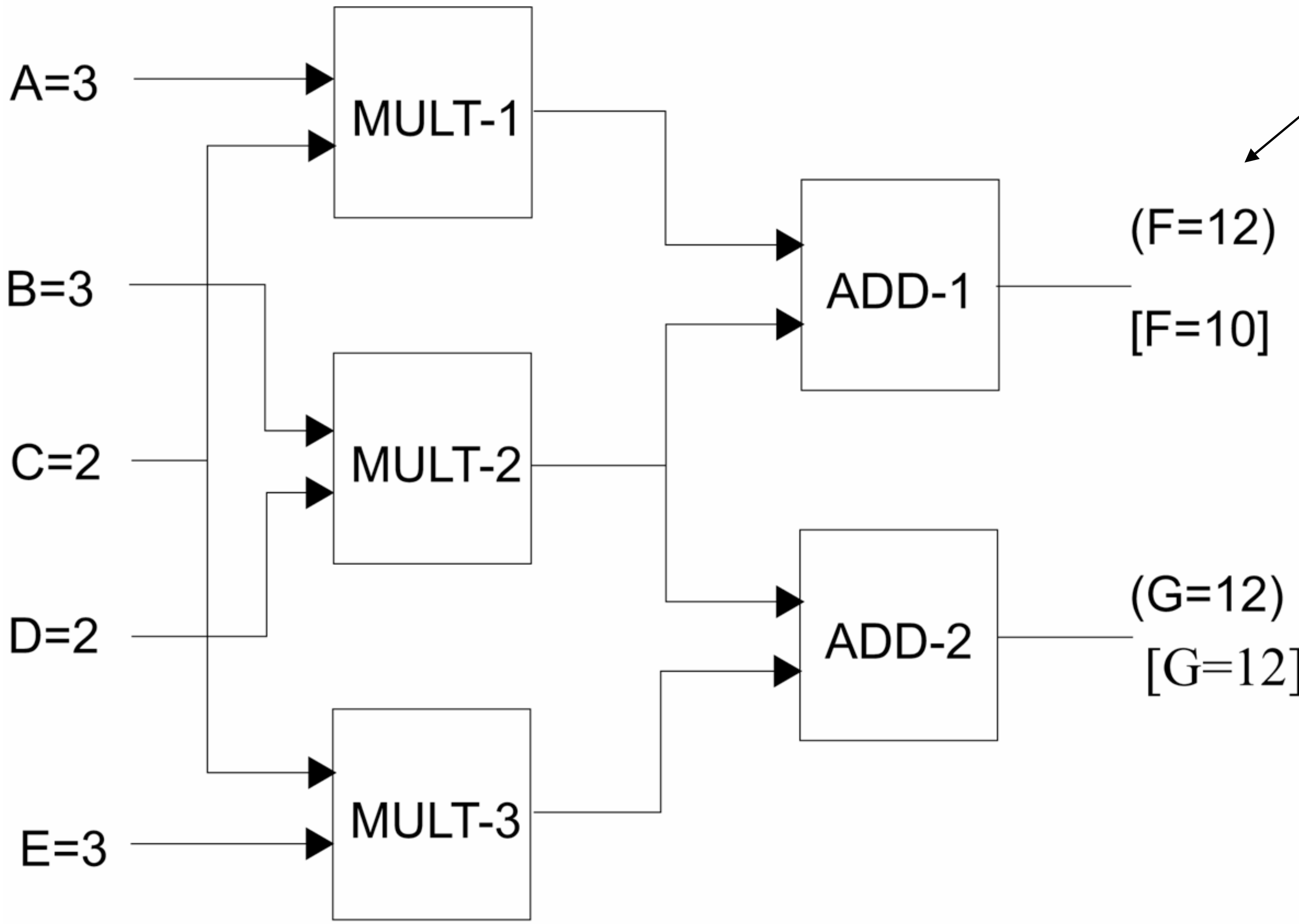
- each component of the device,
- device’s internal structure,
- observations of the device’s actual performance

The behavioral description of an adder
(Davis and Hamscher,1988)



Behavior at the terminals of the device: e.g., C is A+B.

Taking advantage of direction of information flow (Davis and Hamscher, 1988)

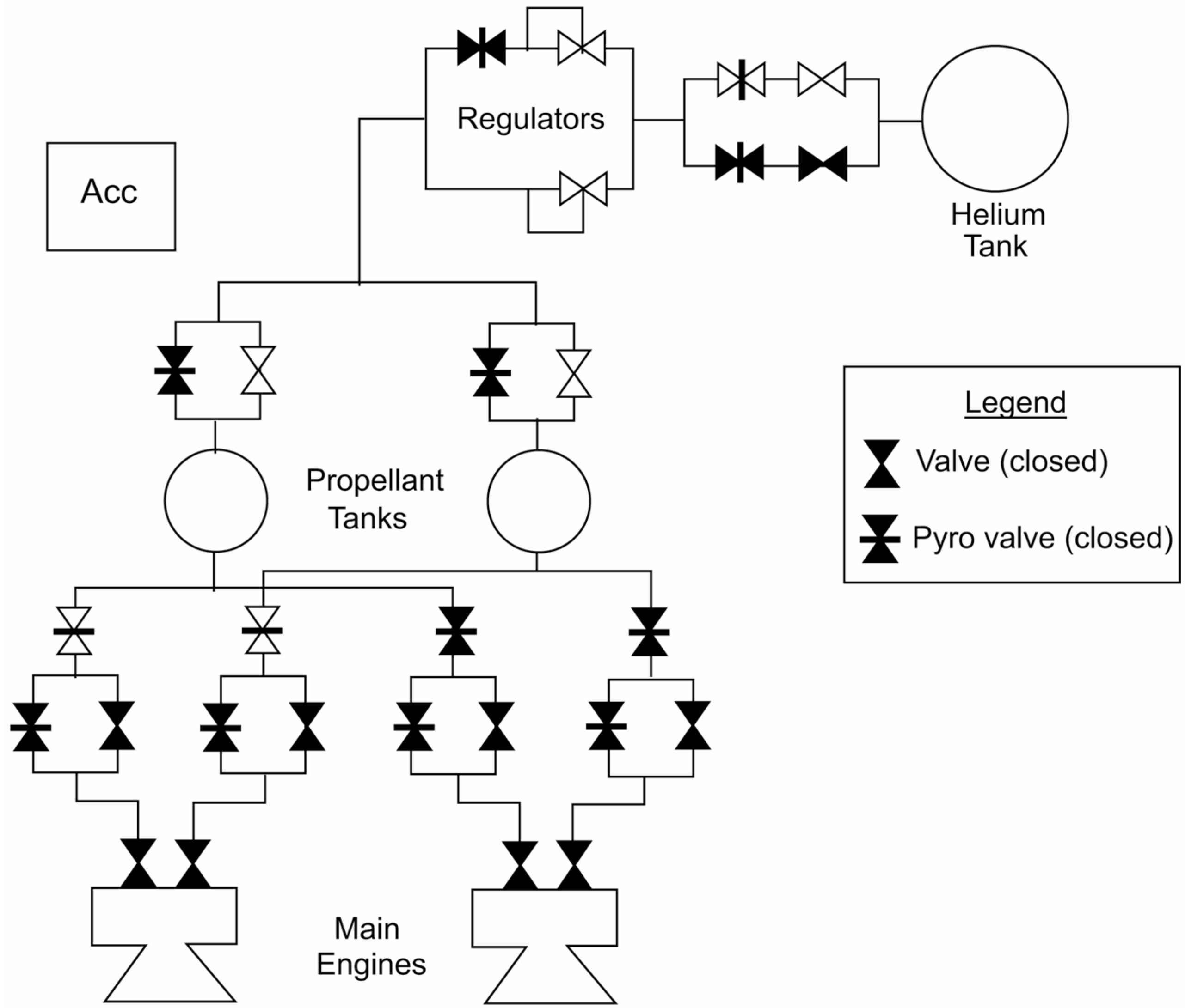


Either ADD-1 is bad, or the inputs are incorrect (MULT-1 or MULT-2 is bad)

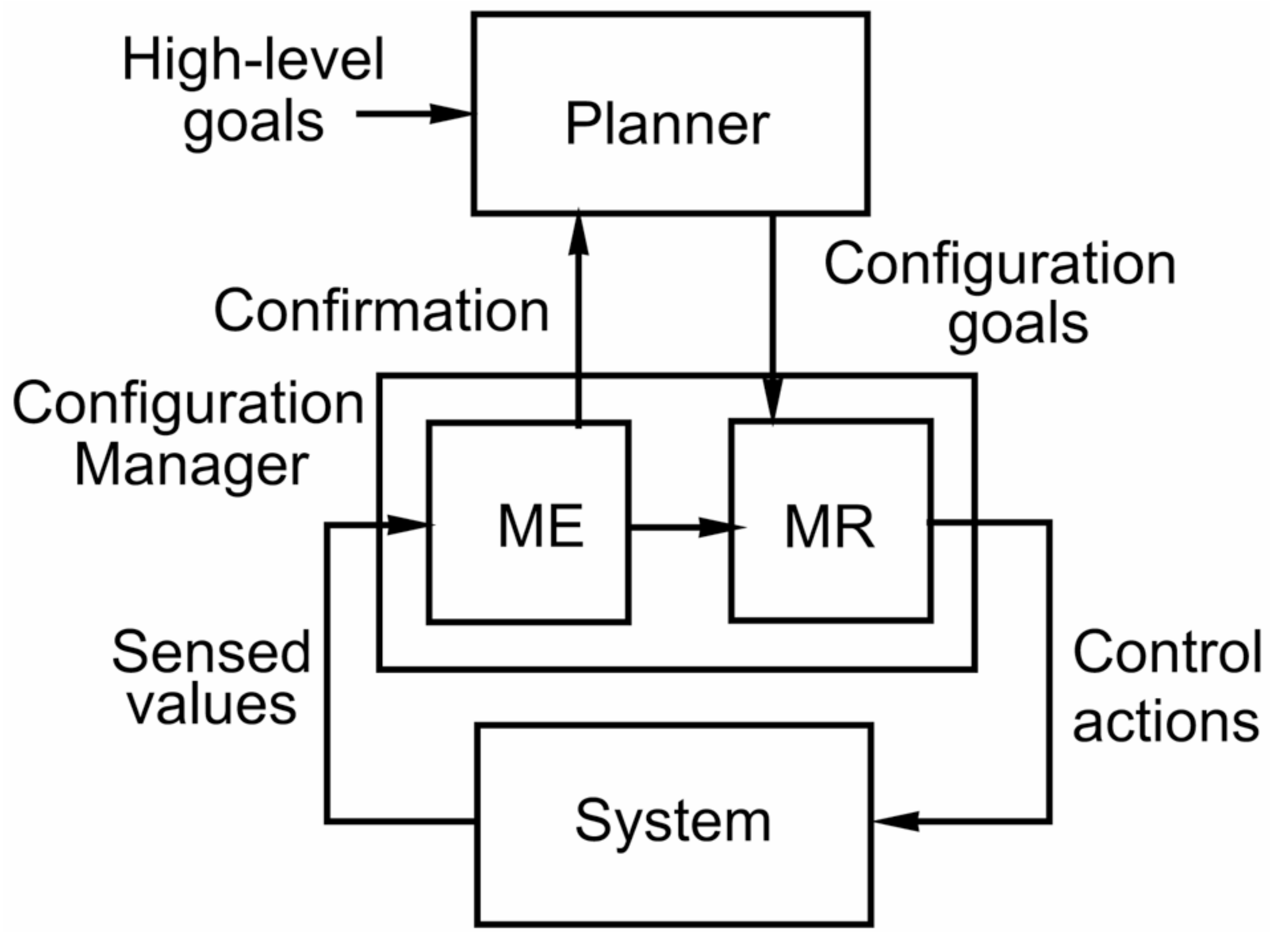
Fault diagnosis procedure

- Generate hypotheses: identify the faulty component(s) , e.g., ADD-1 is not faulty
- Test hypotheses: Can they explain the observed behaviour?
- Discriminate between hypotheses: What additional information is necessary to resolve conflicts?

A schematic of the simplified Livingstone propulsion system (Williams and Nayak ,1996)



A model-based configuration management system (Williams and Nayak, 1996)



Case-based reasoning (CBR)

Allows reference to past "cases" to solve new situations.

Ubiquitous practice: medicine, law, programming, car repairs, …

Common steps performed by a case-based reasoner

- Retrieve appropriate cases from memory
- Modify a retrieved case so that it will apply to the current situation
- Apply the transformed case
- Save the solution, with a record of success or failure, for future use

Preference heuristics to help organize the storage and retrieval cases (Kolodner, 1993)

- **Goal directed preference: Retrieve cases that have the same goal as the current situation**
- **Salient-feature preference: Prefer cases that match the most important features or those matching the largest number of important features**
- **Specify preference: Look for as exact as possible matches of features before considering more general matches**
- **Recency preference: Prefer cases used most recently**
- **Ease of adaptation preference: Use first cases most easily adapted to the currrent situation**

Advantages of a rule-based approach

- Ability to directly use experiential knowledge acquired from human experts
- Mapping of rules to state space search
- Separation of knowledge from control
- Possibility of good performance in limited domains
- Good explanation facilities

Disadvantages of a rule-based approach

- highly heuristic nature of rules not capturing the functional (or model-based) knowledge of the domain
- brittle nature of heuristic rules
- rapid degradation of heuristic rules
- descriptive (rather than theoretical) nature of explanation rules
- highly task dependent knowledge

Advantages of model-based reasoning

- Ability to use functional/structure of the domain
- Robustness due to ability to resort to first principles
- Transferable knowledge
- Aibility to provide causal explanations

Advantages of model-based reasoning

- Lack of experiental (descriptive) knowledge of the domain
- Requirement for an explicit domain model
- High complexity
- Unability to deal with exceptional situations

Advantages of case-based reasoning

- Ability to encode historical knowledge directly
- Achieving speed-up in reasoning using shortcuts
- Avoiding past errors and exploiting past successes
- No (strong) requirement for an extensive analysis of domain knowledge
- Added problems solving power via appropriate indexing strategies

Disadvantages of case-based reasoning

- No deeper knowledge of the domain
- Large storage requirements
- Requirement for good indexing and matching criteria

How about combining those approaches?

Complex!! But nevertheless useful.

- **rule-based + case-based can**
  - **first check among previous cases, then engage in rule-based reasoning**
  - **provide a record of examples and exceptions**
  - **provide a record of searches done**

How about combining those approaches?

- rule-based + model-based can
  - enhance explanations with functional knowledge
  - improve robustness when rules fail
  - add heuristic search to model-based search
- model-based + case-based can
  - give more mature explanations to the situations recorded in cases
  - first check against stored cases before proceeding with model-based reasoning
  - provide a record of examples and exceptions
  - record results of model-based inference

Opportunities are endless!