Bachelor of Computer Science (Hons) Year-2 Sep 2023

Welcome to Intelligent Systems

CAI3204N

Learning Objectives

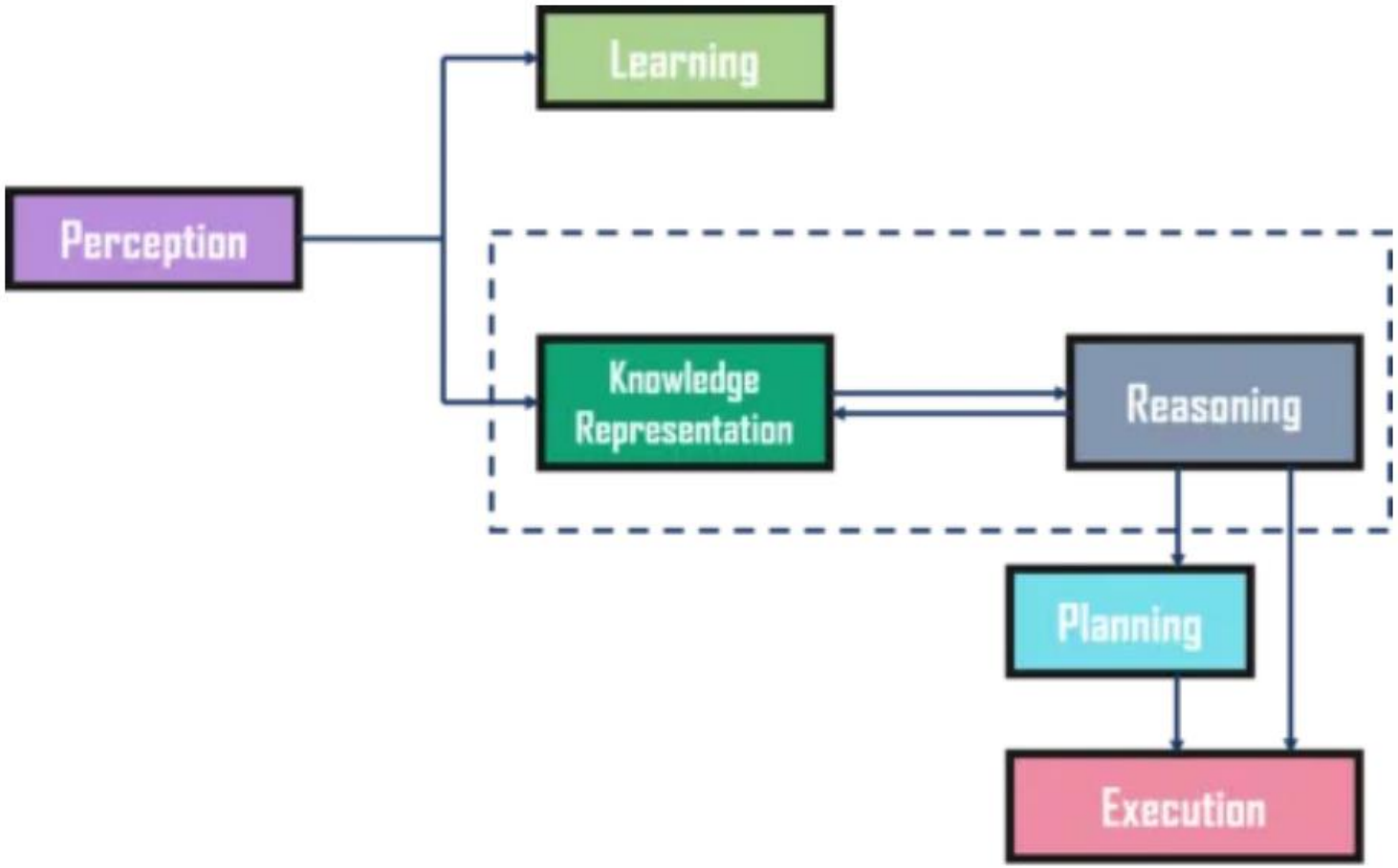
- At the end of the course, students will be able to:
  - CO1: Identify the types of problem that are amenable to "intelligent" solutions.
  - CO2: Compare and contrast the various intelligent system techniques to solve such problems.
  - CO3: Select and apply appropriate intelligent techniques to a given problem.
  - CO4: Critically discuss intelligent system research issues and their applications.

Knowledge Representation

Learning Objectives

- At the end of the course, students will be able to:
  - CO1: Identify the types of problem that are amenable to "intelligent" solutions.
  - CO2: Compare and contrast the various intelligent system techniques to solve such problems.
  - CO3: Select and apply appropriate intelligent techniques to a given problem.
  - CO4: Critically discuss intelligent system research issues and their applications.

What is Knowledge Representation in AI? Techniques You Need To Know



Rule-based expert systems

The most popular choice for building knowledge-based systems: rule-based expert systems.

Introduction, or what is knowledge?

- **In the 1970s, it was finally accepted that to make a machine solve an intellectual problem one had to know the solution. In other words, one has to have knowledge, 'know-how', in some specific domain.**

What is knowledge?

- Knowledge is a theoretical or practical understanding of a subject or a domain.

- Knowledge is also the sum of what is currently known, and apparently knowledge is power.

- **Those who possess knowledge are called experts.**
- **They are the most powerful and important people in their organisations.**
- **Any successful company has at least a few first-class experts and it cannot remain in business without them.**

Who is generally acknowledged as an expert?

- Anyone can be considered a domain expert if he or she has deep knowledge (of both facts and rules) and strong practical experience in a particular domain. The area of the domain may be limited.

- **Experts in electrical machines may have only general knowledge about transformers, while experts in life insurance marketing might have limited understanding of a real estate insurance policy.**

- **In general, an expert is a skillful person who can do things other people cannot.**

How do experts think?

- The human mental process is internal, and it is too complex to be represented as in algorithm.
- However, most experts are capable of expressing their knowledge in the form of rules for problem solving.

Consider a simple example.

Imagine, you meet an alien! He wants to cross a road. Can you help him?

You are an expert in crossing roads - you've been on this job for several years. Thus you are able to teach the alien.

How would you do this?

- **You explain to the alien that he can cross the road safely when the traffic light is green, and he must stop when the traffic light is red**

These are the basic rules. Your knowledge can be formulated as the following simple statements:

IF the 'traffic light' is green

THEN the action is go

IF the 'traffic light' is red

THEN the action is stop

These statements represented in the IF-THEN form are called production rules or just rules.

The term 'rule' in AI, which is the most commonly used type of knowledge representation, can be defined as an IF-THEN structure that relates given information or facts in the IF part to some action in the THEN part.

A rule provides some description of how to solve a problem.

Rules are relatively easy to create and understand.

Rules as a knowledge representation technique

Any rule consists of two parts: the IF part, called the antecedent (premise or condition) and the THEN part called the consequent (conclusion or action). The basic syntax of a rule is:

IF <antecedent>

THEN <consequent>

In general, a rule can have multiple antecedents joined by the keywords AND (conjunction), OR (disjunction) or a combination of both. However, it is a good habit to avoid mixing conjunctions and disjunctions in the same rule.

-

IF <antecedent1>

AND <antecedent2>

.

.

.

AND <antecedent n>

THEN <consequent>

IF <antecedent1>

OR <antecedent2>

.

.

.

OR <antecedent n>

THEN <consequent>

The consequent of a rule can also have multiple clauses:

IF <antecedent>

THEN <consequent 1>

<consequent 2>

……

<consequent m>

The antecedent of a rule incorporates two parts: an object (linguistic object) and its value.

In the road crossing example, the linguistic object 'traffic light' can take either the value green or the value red.

The object and its value are linked by an operator. The operator identifies the object and assigns the value.

- **Operators such as *is, are, is not, are not* are used to assign a symbolic value to a linguistic object. But expert systems can also use mathematical operators to define an object as numerical and assign it to the numerical value.**

For example,

IF 'age of the customer' < 18

AND 'cash withdrawal' > 1000

THEN 'signature of the parent' is required

Similar to a rule antecedent, a consequent combines an object and a value connected by an operator.

The operator assigns the value to the linguistic object.

In the road crossing example, if the value of traffic light is green, the first rule sets the linguistic object action to the value go.

Numerical objects and even simple arithmetical expression can also be used in a rule consequent.

IF 'taxable income' > 16283

THEN 'Medicare levy' = 'taxable income' * 1.5 / 100

Rules can represent relations, recommendations, directives, strategies and heuristics (Durkin, 1994).

Relation

IF the 'fuel tank' is empty **THEN** the car is dead

Recommendation

IF the season is autumn

AND the sky is cloudy

AND the forecast is drizzle

THEN the advice is 'take an umbrella'

Directive

IF the car is dead

AND the 'fuel tank' is empty

THEN the action is 'refuel the car'

Strategy

IF the car is dead
THEN the action is 'check the fuel tank';
stepl is complete

IF stepl is complete
AND the 'fuel tank' is full
THEN the action is 'check the battery';
step2 is complete

Heuristic

IF the spill is liquid

AND the 'spill pH' < 6

AND the 'spill smell' is vinegar

THEN the 'spill material' is 'acetic acid'

The main players in the expert system development team

As soon as knowledge is provided by a human expert, we can input it into a computer.

We expect the computer to act as an intelligent assistant in some specific domain of expertise or to solve a problem that would otherwise have to be solved by an expert.

We also would like the computer to be able to integrate new knowledge and to show its knowledge in a form that is easy to read and understand, and to deal with simple sentences in a natural language rather than an *artificial programming language.*

Finally, we want our computer to explain how it reaches a particular conclusion. In other words, we have to build an expert system, a computer program capable of performing at the level of a human expert in a narrow problem area.

Finally, we want our computer to explain how it reaches a particular conclusion. In other words, we have to build an expert system, a computer program capable of performing at the level of a human expert in a narrow problem area.

- **The most popular expert systems are rule-based systems.**

- **A great number have been built and successfully applied in such areas as business and engineering, medicine and geology, power systems and mining.**

- **A large number of companies produce and market software for rule-based expert system development - expert system shells for personal computers.**

Expert system shells are becoming particularly popular for developing rule-based systems.

Their main advantage is that the system builder can now concentrate on the knowledge itself rather than on learning a programming language.

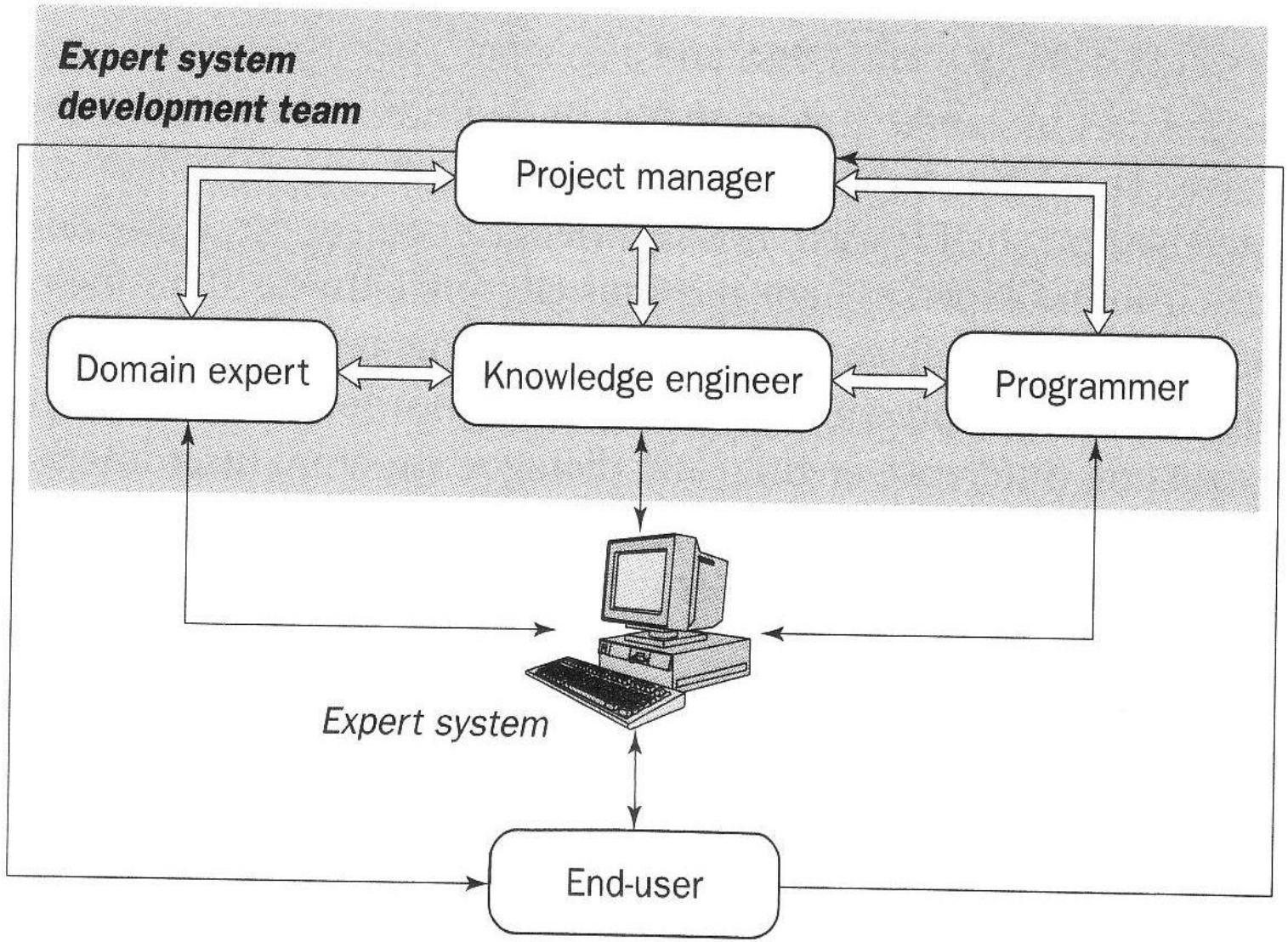
What is an expert system shell?

- An expert system shell can be considered as an expert system with the knowledge removed. Therefore, all the user has to do is to add the knowledge in the form of rules and provide relevant data to solve a problem.

Let us now look at who need to develop an expert system and what skills are needed.

- **In general, there are five members of the expert system development team:**
- **the domain expert, the knowledge engineer, the programmer, the project manager and the end-user.**

The success of their expert system entirely depends on how well the members work together.



- **The domain expert** is a knowledgeable and skilled person capable *of* solving problems in a specific area or **domain.**

The domain expert is the most important player in the expert system development team.

- **The knowledge engineer is someone who is capable of designing, building and testing an expert system.**

This person is responsible for selecting an appropriate task for the expert system. He or she interviews the domain experts to find out how a particular problem is solved.

The knowledge engineer is responsible for testing, revising and integrating the expert system into the workplace.

- **The programmer is the person responsible for the actual programming.**

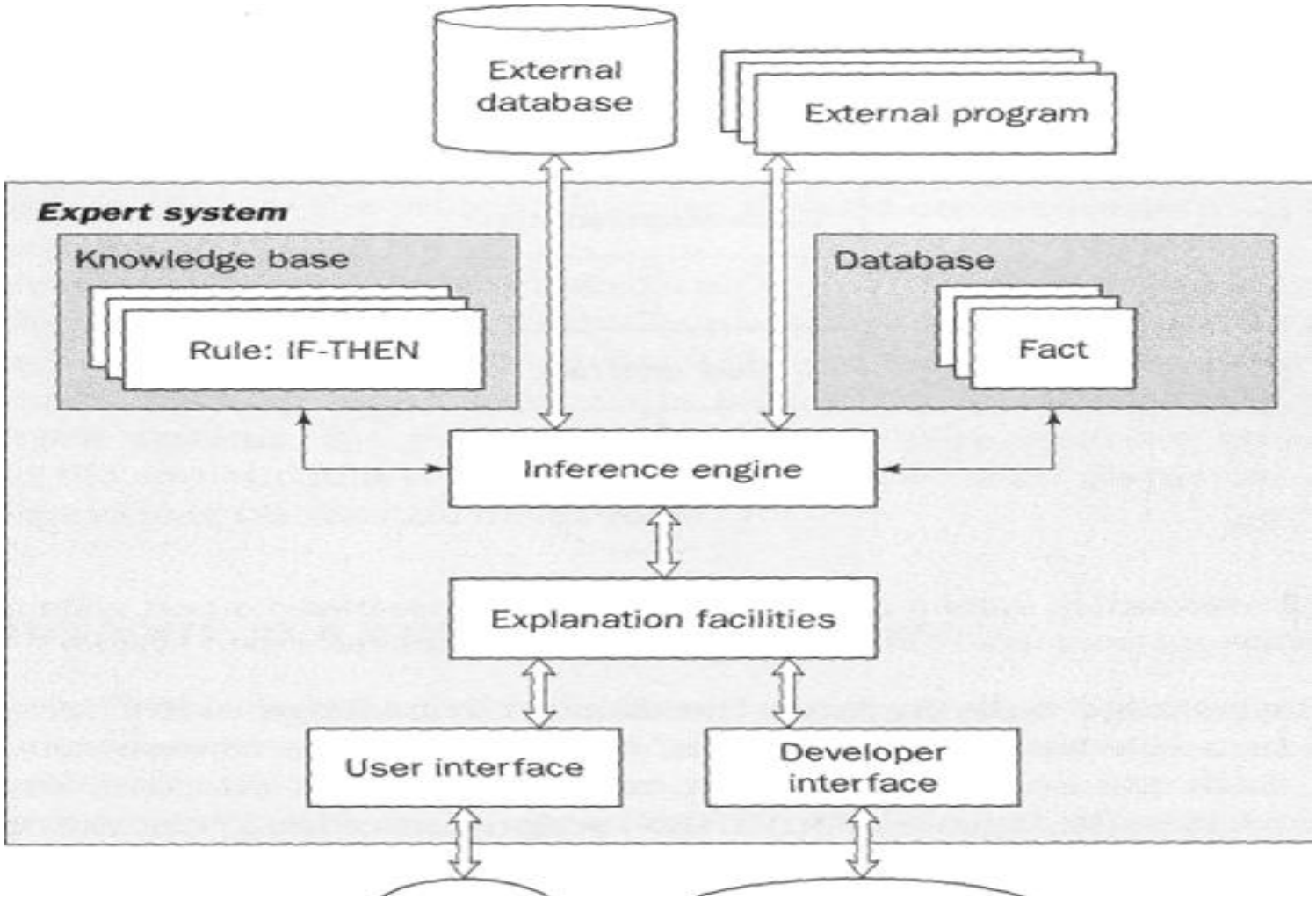
- **The project manager** is the leader of the expert system development team, responsible for keeping the project on track. He or she makes sure that all deliverables and milestones are met, interacts with the expert, knowledge engineer, programmer and end-user.

- **The end-user,** often called just the **user,** is a person who uses the expert system when it is developed.
- **The user might be an analytical chemist determining the molecular structure of soil from Mars (Feigenbaum *et al.,* 1971), a junior doctor diagnosing an infectious blood disease (Shortliffe, 1976), an exploration geologist trying to discover a new mineral deposit (Duda *et al.,* 1979), or a power system operator needing advice in an emergency (Negnevitsky, 1996).**

- **Each of these users of expert systems has different needs, which the system must meet: the system's final acceptance will depend on the user's satisfaction.**
- **The user must not only be confident in the expert system performance but also feel comfortable using it. Therefore, the design of the user interface of the expert system is also vital for the project's success; the end-user's contribution here can be crucial.**

Rule-based expert system

- A rule-based expert system has five components: the knowledge base, the database, the inference engine, the explanation facilities, and the user interface.



Fundamental characteristics of an expert system

- The most important characteristic of an expert system is its high-quality performance.

- The speed of reaching a solution is also very important. Even the most accurate decision or diagnosis may not be useful if it is too late to apply, for instance, in an emergency, when a patient dies or a nuclear power plant explodes.

- A unique feature of an expert system is its explanation capability. This enables the expert system to review its own reasoning and explain its decisions.

- **Expert systems employ symbolic reasoning when solving a problem. Symbols are used to represent different types of knowledge such as facts, concepts and rules.**
- **Unlike conventional programs written for numerical data processing, expert systems are built for knowledge processing and can easily deal with qualitative data.**

Can expert systems make mistakes?

- Even a brilliant expert is only a human and thus can **make mistakes**. This suggests that an expert system built to perform at a human expert level also should be allowed to make mistakes.

Forward chaining and backward chaining inference techniques

- **In a rule-based expert system, the domain knowledge is represented by a set of IF-THEN production rules and data is represented by a set of facts about the current situation.**

- The inference engine compares each rule stored in the knowledge base with facts contained in the database. When the IF (condition) part of the rule matches a fact, the rule is fired and its THEN (action) part is executed.

- **The firing rule may change the set of facts by adding a new fact**

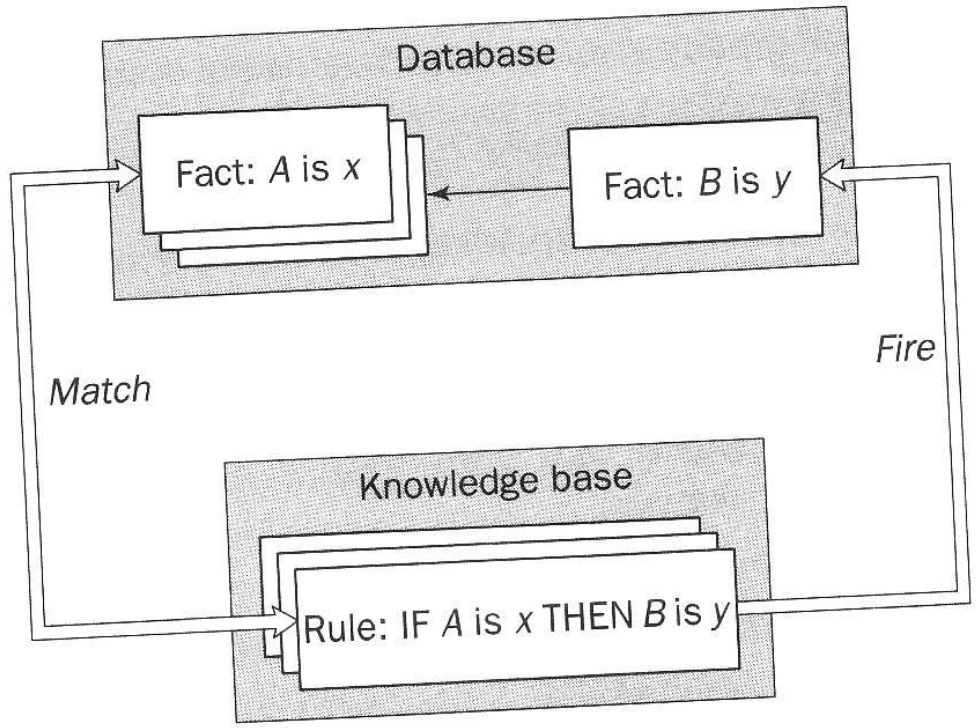


Figure 2.4 The inference engine cycles via a match-fire procedure

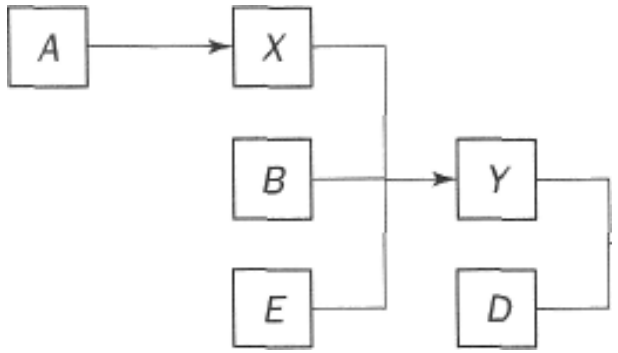
- **The matching of the rule IF parts to the facts produces inference chains. The inference chain indicates how an expert system applies the rules to reach a conclusion.**

- **Suppose the database initially includes facts A, *B*, *C*, *D* and *E*, and the knowledge base contains only three rules:**

- **Rule 1**: **IF** Y is true
- **AND** D is true
- **THEN** Z is true

- **Rule 2**: **IF** *X* is true
- **AND** *B* is true
  AND E is true
- **THEN** *Y* is true

- **Rule 3**: **IF** *A* is true
- **THEN** X is true



- The inference engine must decide when the rules have *to* be fired.
- There are two principal ways in which rules are executed.
- One is called forward chaining and the other backward chaining (Waterman and Hayes-Roth, 1978).

Rule chaining

Logical connection between output from invocation of one rule and input to next. Can operate in two directions:

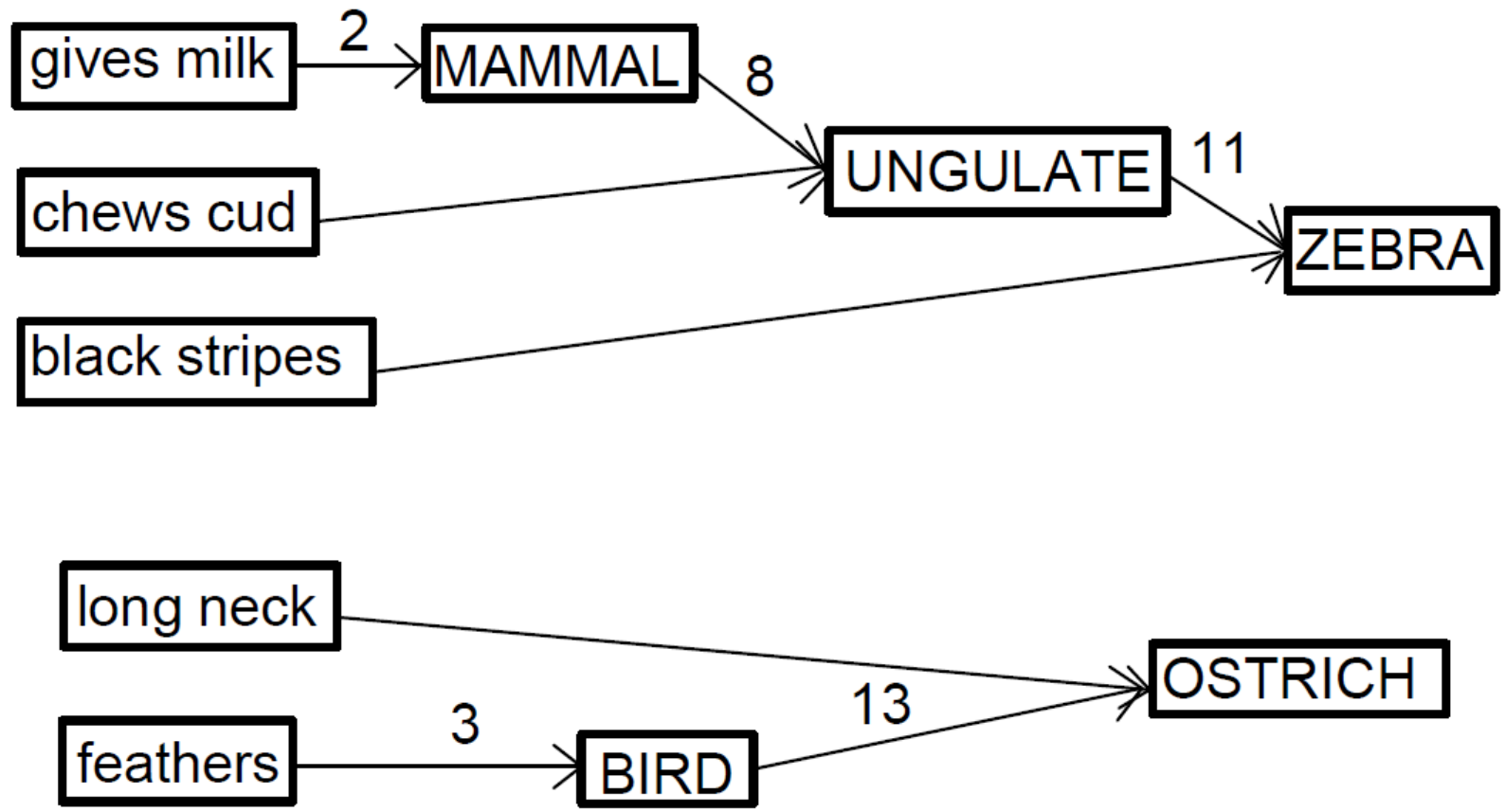
– **Forward chaining** (data driven) - proceed from given data to conclusions

– **Backward chaining** (goal driven) - attempt to prove hypothesis, requesting data as required

Backward chaining used for most operational expert systems

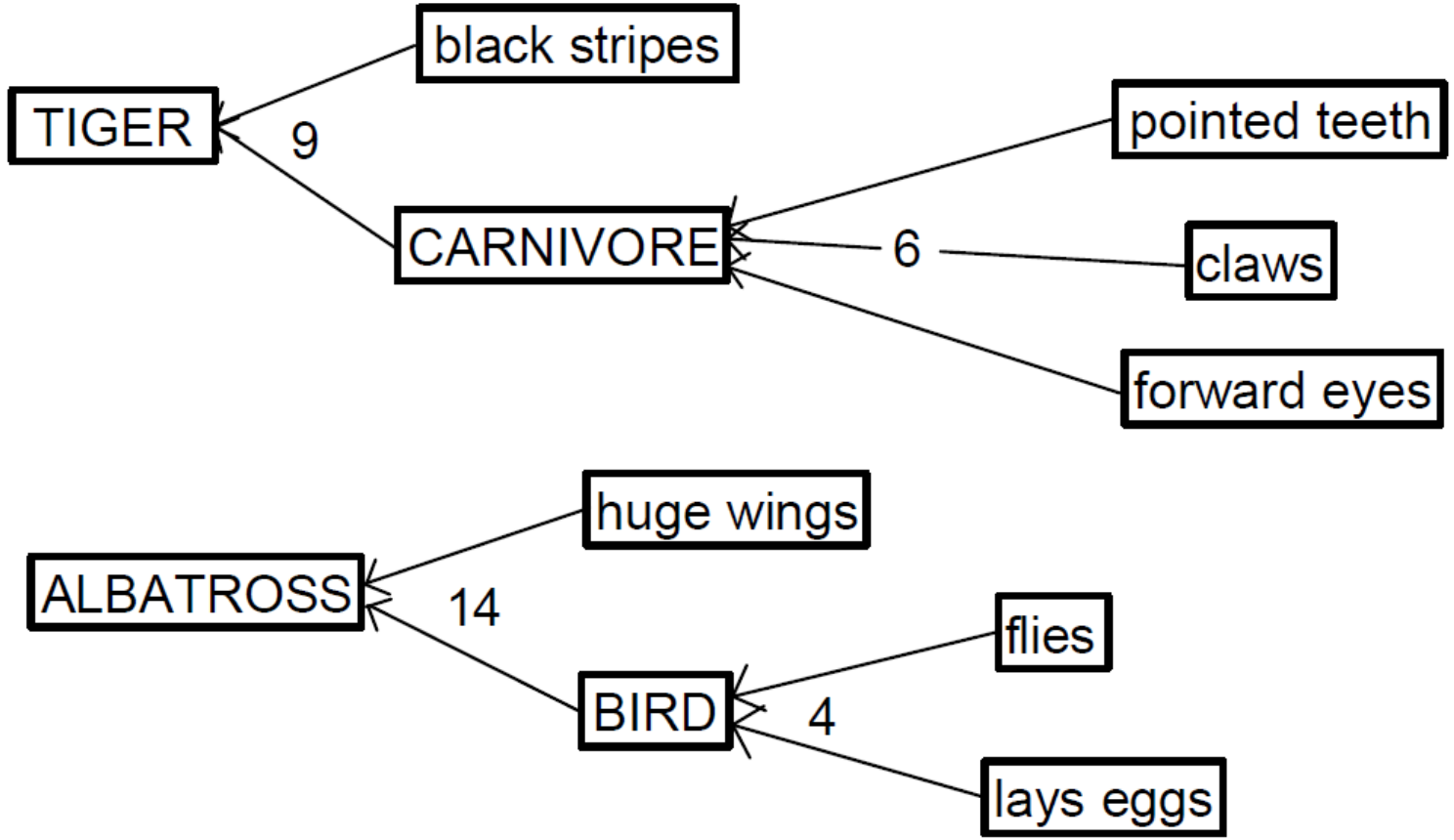
Forward chaining

Proceeds FROM start data TO conclusions., e.g.



Backward chaining

Proceeds FROM starting hypothesis TO data which will support or refute it, e.g.

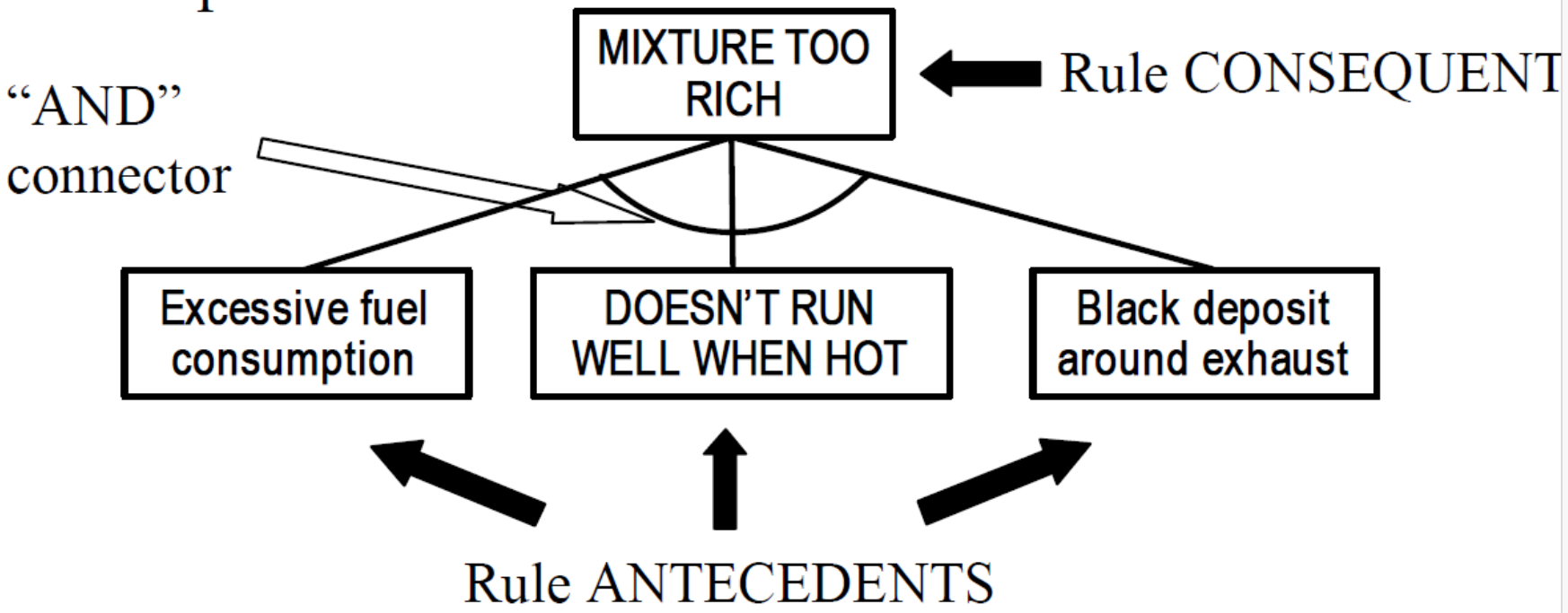


Representing a rule base - 1

Most straightforward way is to represent rules in form of ***goal trees,*** e.g. the statement:

"The fuel/air mixture in a car engine is too rich if fuel consumption is excessive, the car doesn't run well when hot, and there is a black deposit around the exhaust"

can be represented as:

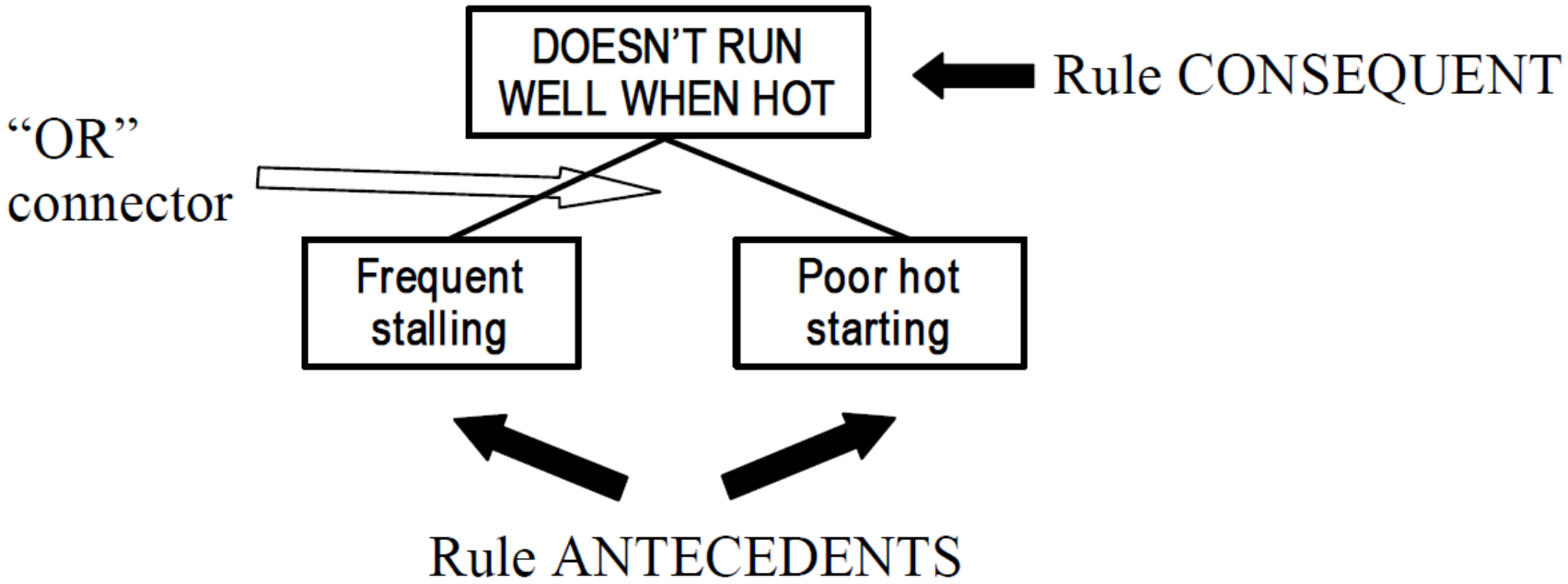


Representing a rule base - 2

Similarly, the statement :

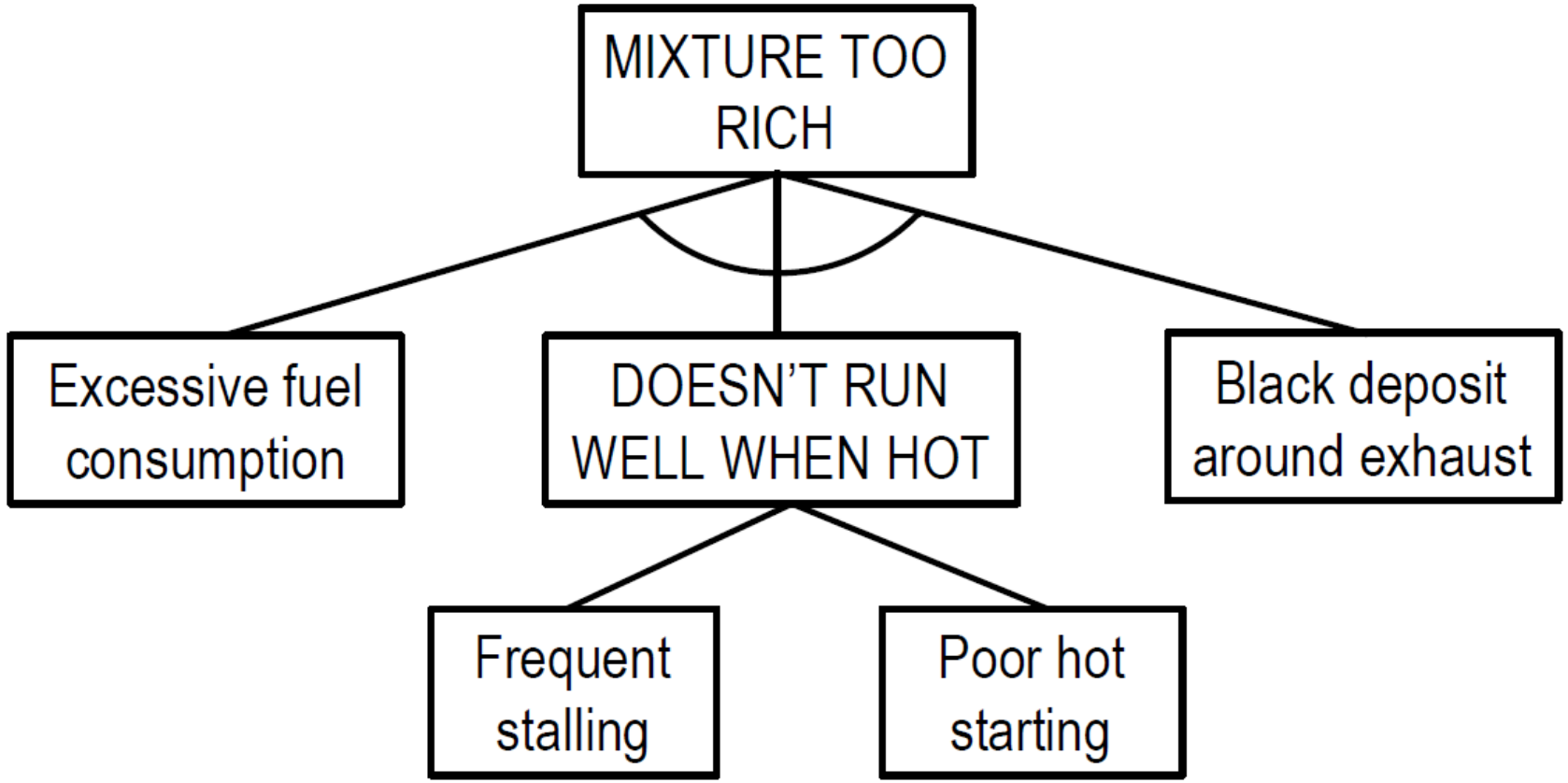
"Symptoms of a car not running well when hot might be frequent stalling or poor starting when hot"

can be represented as:



Combining rules

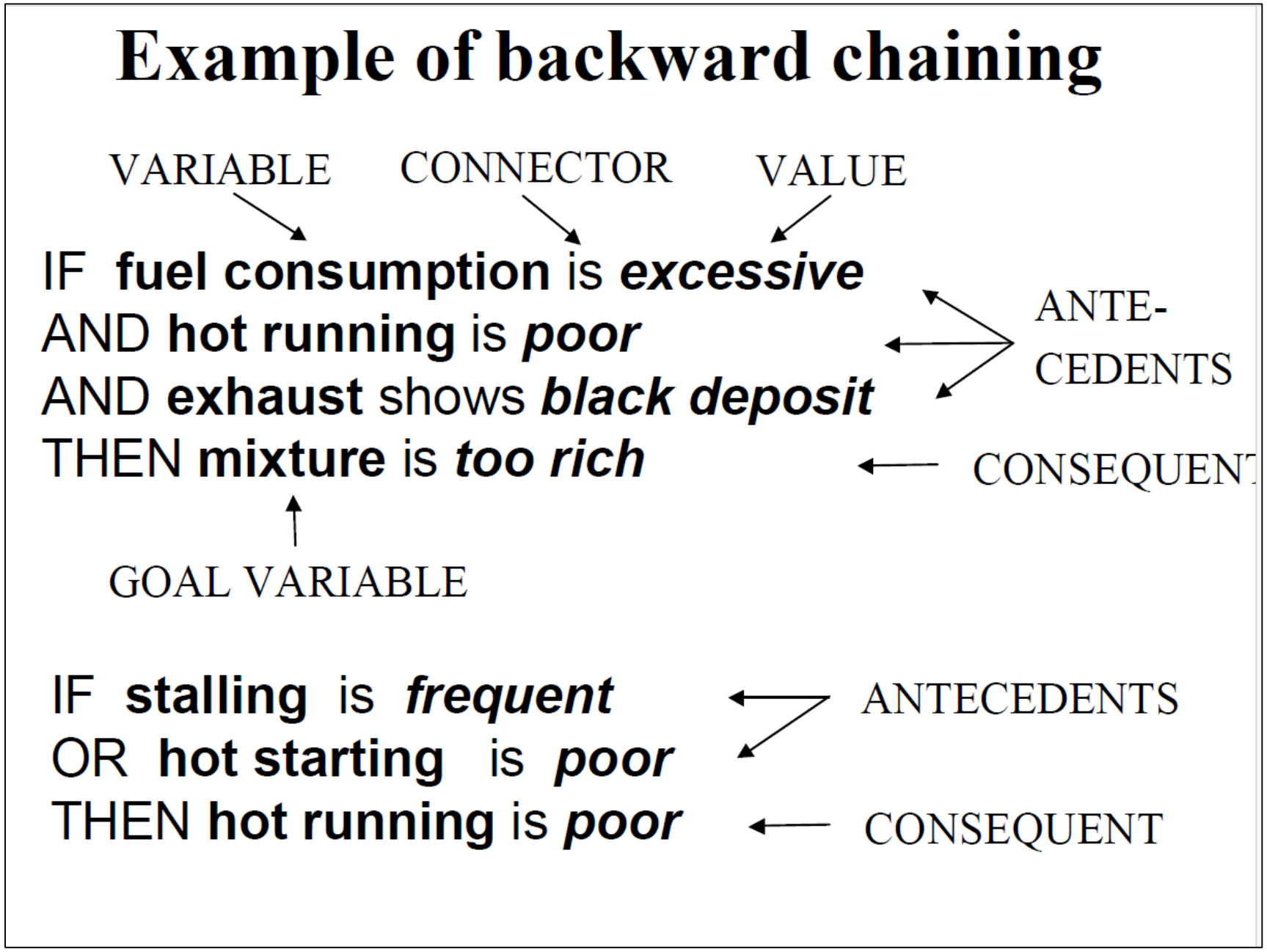
Rules whose goals form antecedents for other rules can then be combined into a multi-level goal tree, e.g:



Production rules: a more detailed look

1. Rules consist of **antecedent** (conditions) and **consequent** (actions) parts
2. Both halves of a rule involve **variables** which may be single- or multi-valued
3. Variables are initially undefined, and acquire values as inference proceeds
4. Forward chaining is triggered by **volunteering** values of starting variables in rule antecedents
5. Backward chaining is triggered by seeking a value for a **goal** variable in a rule consequent

Example of backward chaining



Example of forward chaining

VOLUNTEERED VARIABLE

IF **key feature** is ***durability*** ← ANTECEDENT

THEN **make** includes ***volkswagen*** ← CONSEQUENT

MULTI-VALUED CONNECTOR

IF **make** includes ***volkswagen***

AND **main use** is ***load-carrying***

THEN **model** includes ***VW Passat Estate***

NEW VALUE OF CONSEQUENT VARIABLE

Rule ordering

Standard order of rule firing:

– apply first matching rule in sequence

– activate clauses in order specified in rule

Can over-ride using **meta-rules**, specifying order in which goals are tested or questions asked, e.g:

IF **choice** includes ***bmw***
THEN ask **performance rating**

IF **influenza** is ***eliminated***
THEN seek **mumps**

Class Activity

Designing a rule base for a simplified expert system application

Consider the following scenario -

"Three factors all need to be favourable before a heat pump can be considered suitable. Firstly, the condenser temperature must not be too high, to avoid any risk of the heat transfer fluid exploding. 120°C is a safe maximum temperature. Secondly, the difference in temperature between the condenser and evaporator must be within the right limits. If it is too high, the process will not run efficiently. If it is negative (i.e. the evaporator is warmer than the condenser) then the process will not work at all. A temperature difference between 0 and 60° is normally recommended. Finally, there must be no significant adverse factors - a very dusty or very corrosive atmosphere normally rules out the use of a heat pump".

Goal tree design - 1

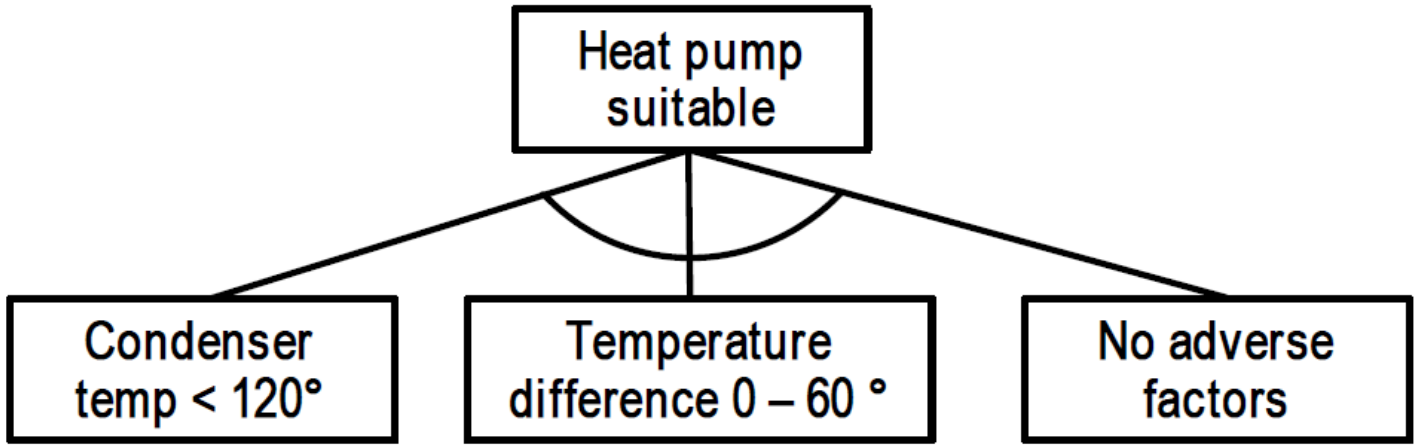
Overall purpose of system:

- *Is a heat pump suitable?*

Antecedents to main goal:

- *Condenser temperature < 120°*
- *Temperature difference 0 - 60°*
- *No adverse factors*

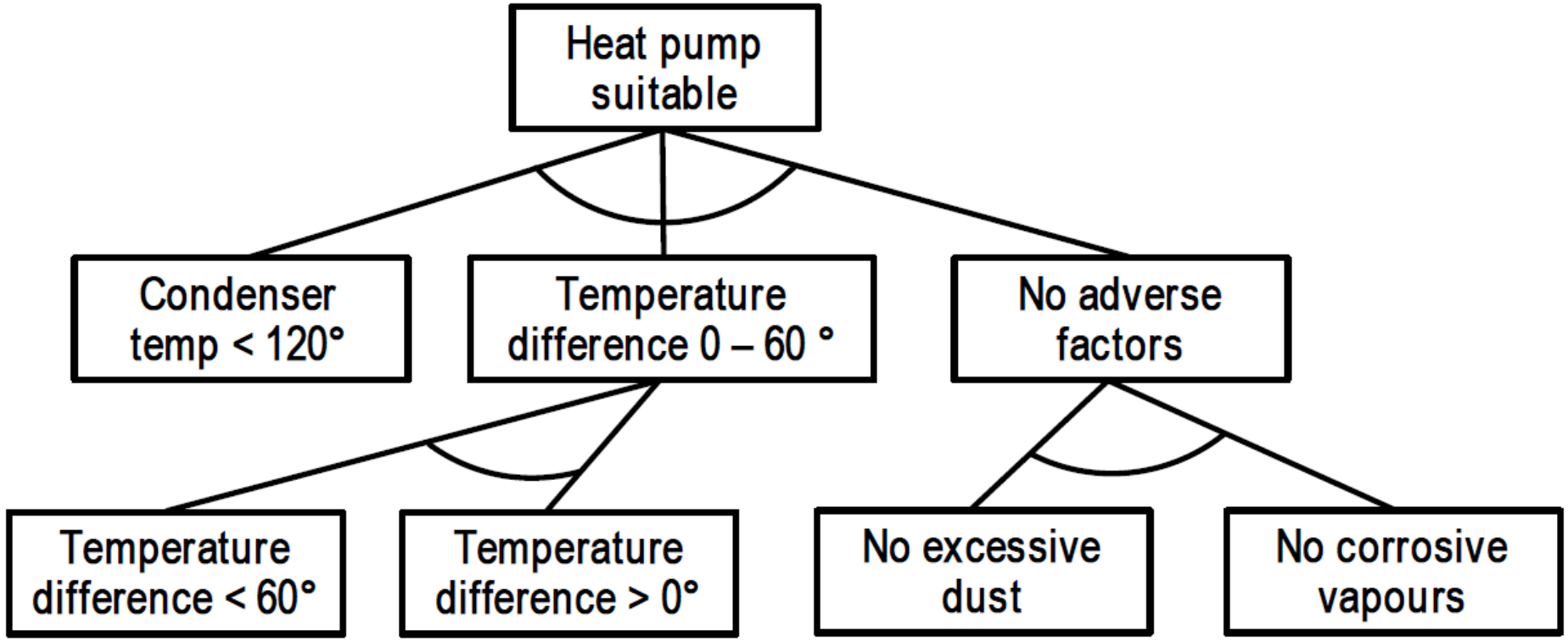
Hence top-level design could be:



Goal tree design - 2

Subsidiary goals *no adverse factors* and *temperature difference 0-60°* can also be broken down further.

Hence second-level design could be:



Generating rules from goal tree - 1

- Rule ***consequent*** (goal) forms THEN clause
- Rule ***antecedents*** form IF (plus AND or OR) clauses
- Rules must all follow VARIABLE – CONNECTOR – VALUE syntax

So top level rule might be:

IF **condenser temperature** < ***120***
AND **temperature difference** is ***small enough***
AND **adverse factors** are ***absent***
THEN **heat pump** is ***suitable***

Generating rules from goal tree - 2

and lower level rules might be:

IF **(condenser temp – evaporator temp)** < *60*
AND **(condenser temp – evaporator temp)** > *0*
THEN **temperature difference** is ***small enough***

IF **dust level** is not *excessive*
AND **corrosive vapours** are not *present*
THEN **adverse factors** are ***absent***

NB - exact rule syntax differs from one expert system shell to another

How do we choose between forward and backward chaining?

- The answer is to study how a domain expert solves a problem. If an expert first needs to gather some information and then tries to infer from it whatever can be inferred, choose the forward chaining inference engine.
- However, if your expert begins with a hypothetical solution and then attempts to find facts to prove it, choose the backward chaining inference engine.

- **Most backward chaining expert systems are used for diagnostic purposes. For instance, MYCIN, a medical expert system for diagnosing infectious blood diseases (Shortliffe, 1976), uses backward chaining.**

Can we combine forward and backward chaining?

- Many expert system shells use a combination of forward and backward chaining inference techniques, so the knowledge engineer does not have to choose between them.

- **However, the basic inference mechanism is usually backward chaining. Only when a new fact is established is forward chaining employed to maximise the use of the new data.**

MEDIA ADVISOR: a demonstration rule-based expert system

- **Rule: 1**
- **if** the environment is papers
- **or** the environment is manuals
- **or** the environment is documents
- **or** the environment is textbooks
  - **then** the stimulus_situation is verbal

- **Rule: 2**
- **if** the environment is pictures
- **or** the environment is illustrations
- **or** the environment is photographs
- **or** the environment is diagrams
  - **then** the stimulus_situation is visual

- **Rule: 3**
- **if** the environment is machines
- **or** the environment is buildings
- **or** the environment is tools
  - **then** the stimulus_situation is 'physical object'

- **Rule: 4**
- **if** the environment is numbers
- **or** the environment is formulas
- **or** the environment is 'computer programs'
  - **then** the stimulus_situation is symbolic

- **Rule: 5**
- **if** the job is lecturing
- **or** the job is advising
- **or** the job is counselling
  - **then** the stimulus_response is oral

- **Rule: 6**
- **if** the job is building
- **or** the job is repairing
- **or** the job is troubleshooting
  - **then** the stimulus_response is 'hands-on'

- **Rule: 7**
- **if** the job is writing
- **or** the job is typing
- **or** the job is drawing
  - **then** the stimulus_response is documented

- **Rule: 8**
- **if** the job is evaluating
- **or** the job is reasoning
- **or** the job is investigating
  - **then** the stimulus_response is analytical

- **Rule: 8**
- **if** the job is evaluating
- **or** the job is reasoning
- **or** the job is investigating
  - **then** the stimulus_response is analytical

- **Rule: 10**
- **if** the stimulus_situation is symbolic
- **and** the stimulus_response is analytical
- **and** feedback is required
  - **then** medium is 'lecture - tutorial'

- **Rule: 11**
- **if** the stimulus_situation is visual
- **and** the stimulus_response is documented
- **and** feedback is not required
  - **then** medium is videocassette

- **Rule: 12**
- **if** the stimulus_situation is visual
- **and** the stimulusjesponse is oral
- **and** feedback is required
  - **then** medium is 'lecture - tutorial'

- **Rule: 13**
- **if** the stimulus_situation is verbal
- **and** the stimulus_response is analytical
- **and** feedback is required
  - **then** medium is 'lecture - tutorial'

- **Rule: 14**
- **if** the stimulus_situation is verbal
- **and** the stimulus_response is oral
- **and** feedback is required
  - **then** medium is 'role-play exercises'

Objects

MEDIA ADVISOR uses six linguistic objects*: environment, stimulus_situation, job, stimulus_response, feedback* and *medium*.

Each object can take one of the allowed values (for example, object *environment* can take the value of *papers, manuals, documents, textbooks, pictures, illustrations, photographs, diagrams, machines, buildings, tools, numbers, formulas, computer programs*).

An object and its value constitute a fact (for instance, the *environment is machines*, and the *job* is *repairing*). All facts are placed in the database.

•

Object	Allowed values	Object	Allowed values
environment	*papers*	*job*	*lecturing*
	manuals		*advising*
	documents		*counselling*
	textbooks		*building*
	pictures		*repairing*
	illustrations		*troubleshooting*
	photographs		*writing*
	diagrams		*typing*
	machines		*drawing*
	buildings		*evaluating*
	tools		*reasoning*
	numbers		*investigating*
	formulas		
	computer programs	*stimulus_ response*	*oral*
			hands-on
			documented
			analytical
stimulus_situation	*verbal*		
	visual		
	physical object	*feedback*	*required*
	symbolic		*not required*

Options

The final goal of the rule-based expert system is to produce a solution to the problem based on input data. In MEDIA ADVISOR, the solution is a medium selected from the list of four options:

- medium is workshop
- medium is 'lecture - tutorial'
- medium is videocassette
- medium is 'role-play exercises'

Dialogue

- What sort of environment is a trainee dealing with on the job! => machines
- **Rule: 3**
- **if** the environment is machines
- **or** the environment is buildings
- **or** the environment is tools
  - **then** the **stimulus_situation** is **'physical object'**

- In what way is a trainee expected to act or respond on the job? => repairing
- **Rule: 6**
- **if** the job is building
- **or** the job is repairing
- **or** the job is troubleshooting
  - **then** **the stimulus_response** is **'hands-on'**

- Is feedback on the trainee's progress required during training? => **required**
- **Rule: 9**
- **if** the stimulus_situation is 'physical object'
- **and** the stimulus_response is 'hands-on'
- **and** feedback is required
  - **then** **medium** is **workshop**

medium is workshop

Inference techniques

- **The standard inference technique in Leonardo is backward chaining with opportunistic forward chaining, which is the most efficient way to deal with the available information.**

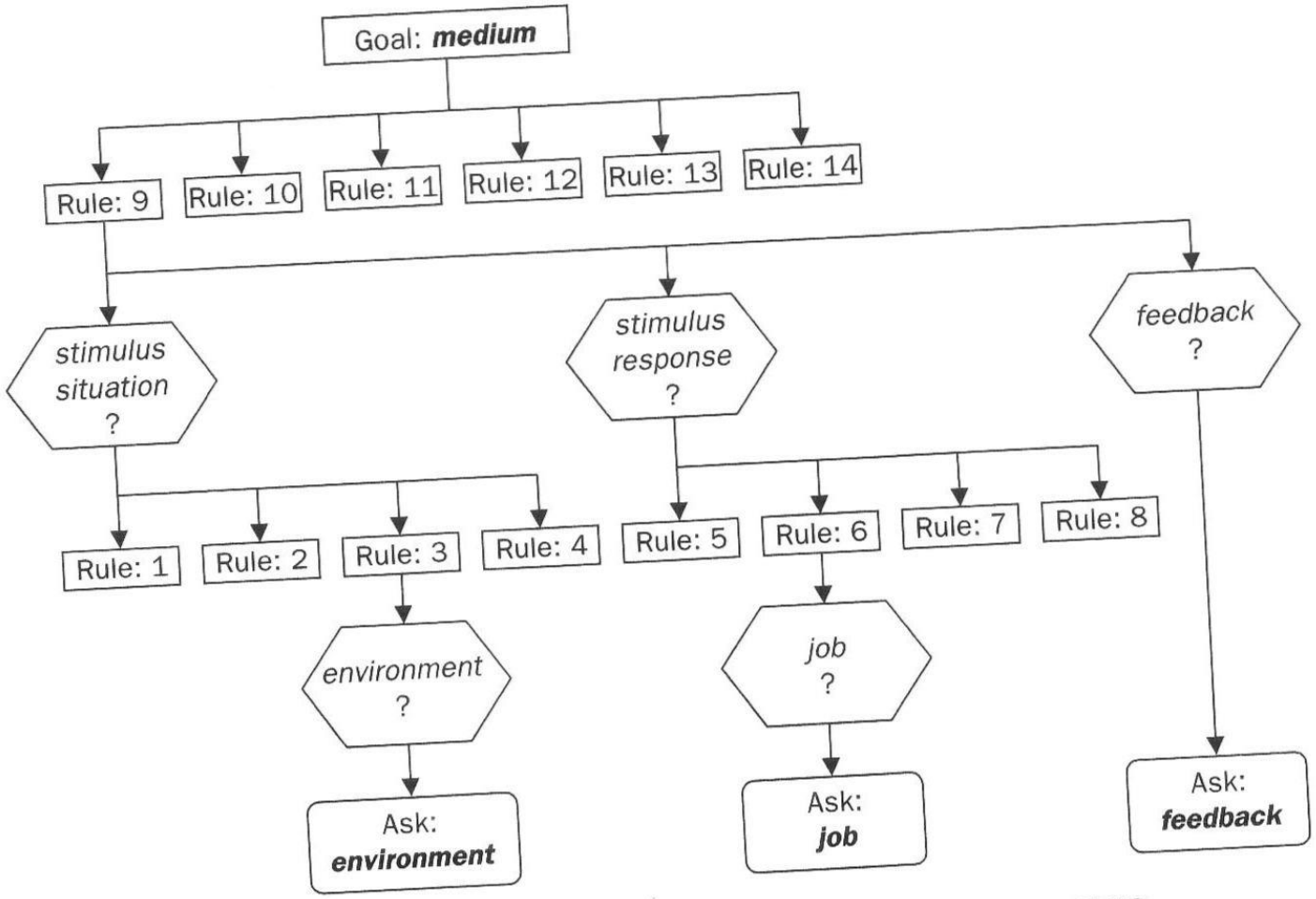
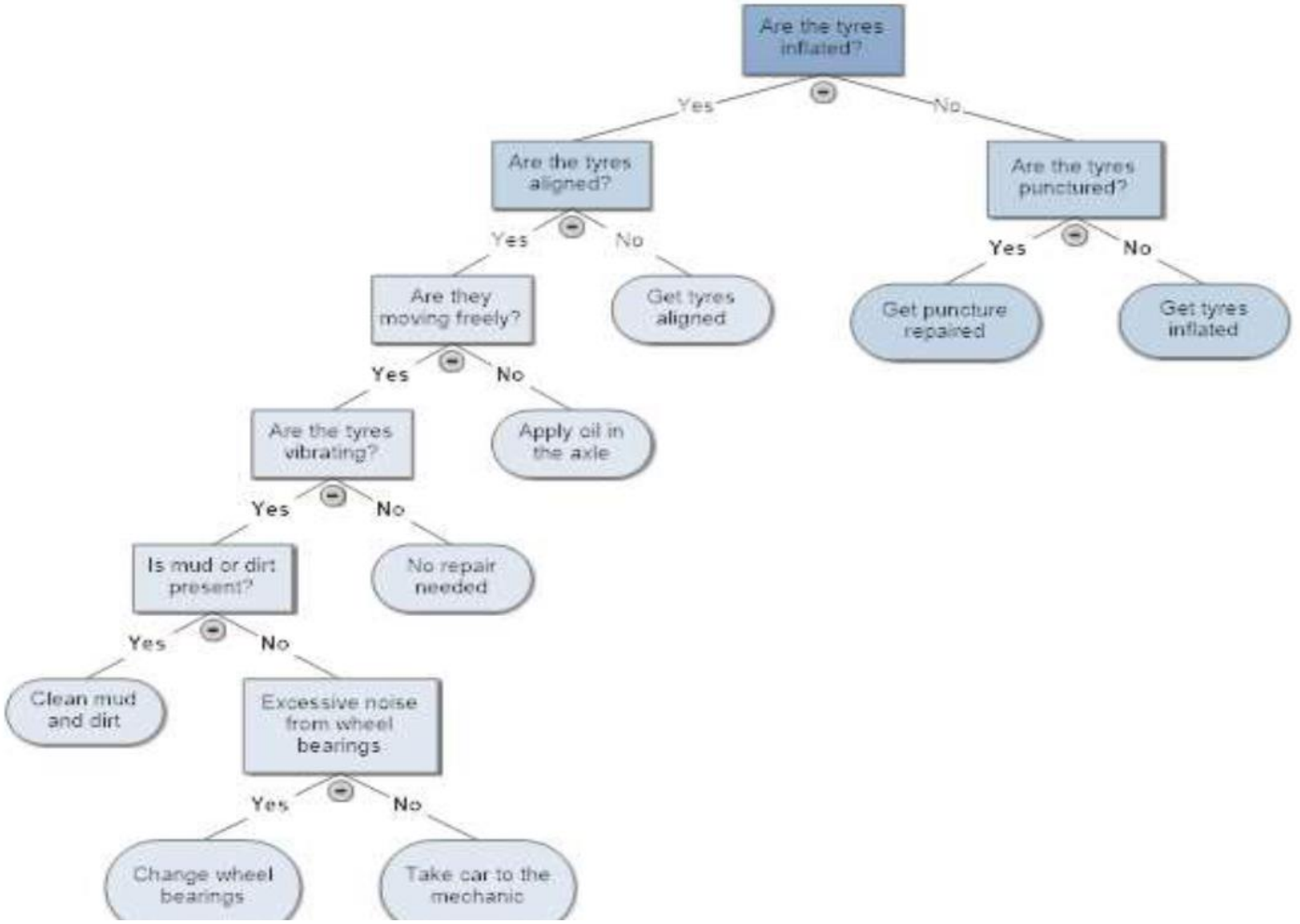


Figure 2.8 Tree diagram for the rule-based expert system MEDIA ADVISOR

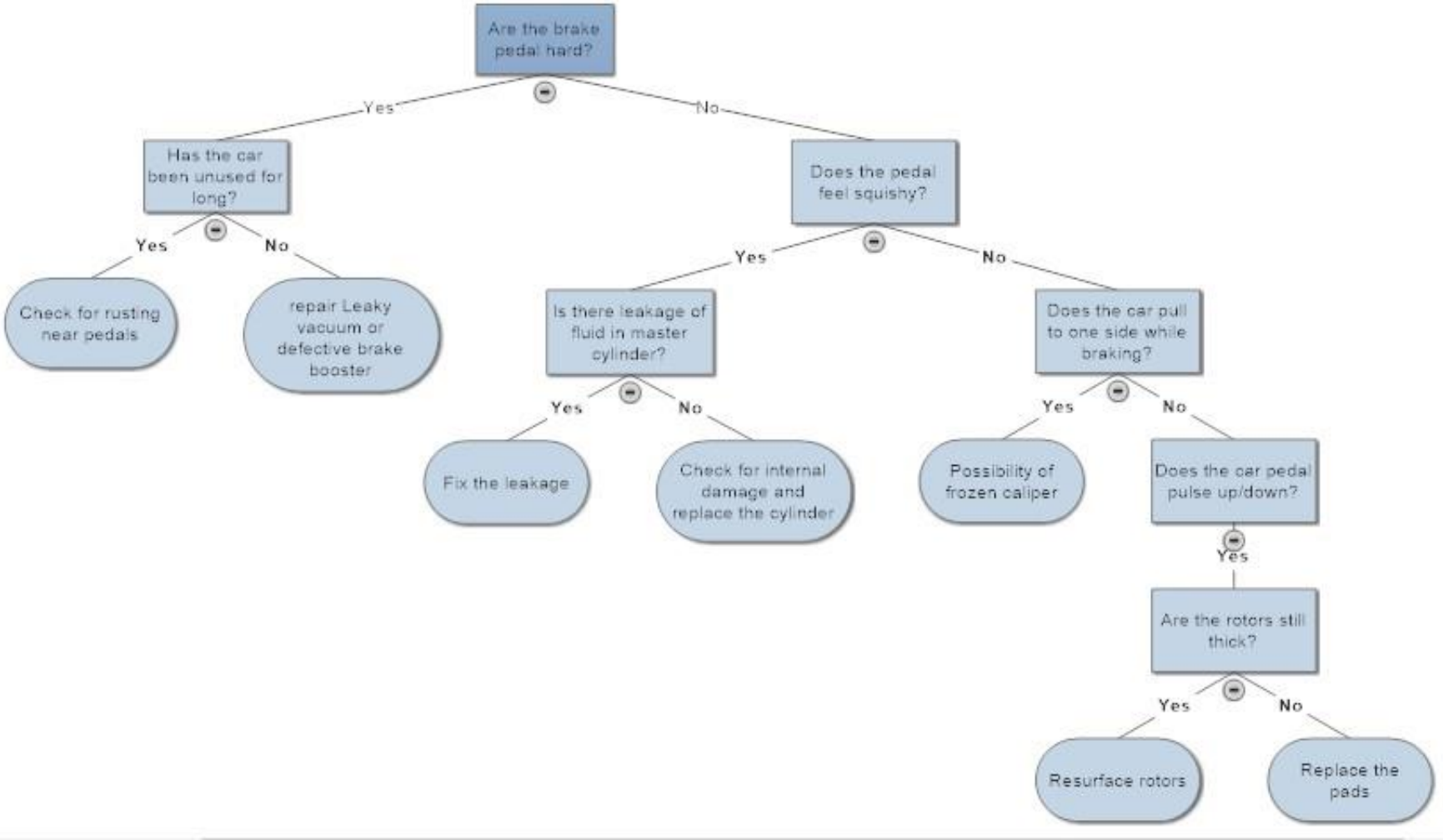
DEMO

- CLIPS ES- Shell

Tyre Problems- Goal Tree



Brake Problems –Goal Tree



Stretch Break!

Thank you

Seminar

- Applications of Artificial Intelligence – upload your Presentation (Week 1 …Pending Task from you guys)
- Write a ES-Shell script for CLIPS