Intelligent Decision Support Systems

Sesi 13

Dosen Pembina :

Danang Junaedi

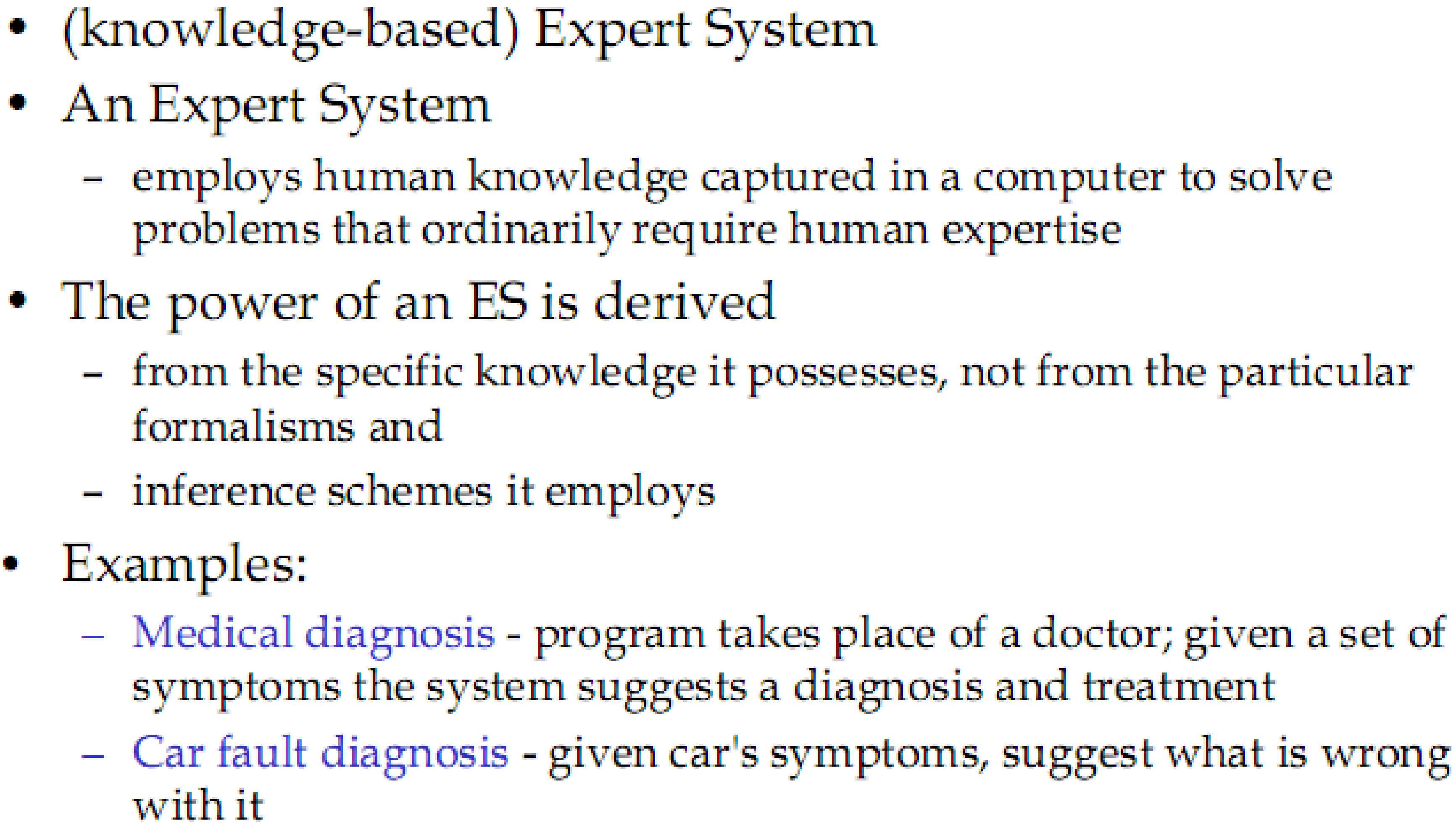
Introduction

- Intelligent DSS = Artificially Intelligent DSS
- Artificial Intelligence (AI) ➔Endeavors to make machines such as computers capable of displaying intelligent behavior
- Artificially intelligent DSS is one that uses AI mechanisms
  - May not be identical to human mechanisms
  - Results comparable
- Why do artificially intelligent DSSs exist?
  - Technological advances in AI make them feasible
  - They yield potential benefits to decision makers and organizations

AI Research Topics

1. Reasoning systems/Expert System
2. Natural language processing
3. Knowledge representation
4. Machine learning
5. Automatic programming
6. Pattern recognition
7. Any can furnish mechanisms for artificially intelligent DSSs

1. Reasoning System/Expert System



- (knowledge-based) Expert System
- An Expert System
  - employs human knowledge captured in a computer to solve problems that ordinarily require human expertise
- The power of an ES is derived
  - from the specific knowledge it possesses, not from the particular formalisms and
  - inference schemes it employs
- Examples:
  - Medical diagnosis - program takes place of a doctor; given a set of symptoms the system suggests a diagnosis and treatment
  - Car fault diagnosis - given car's symptoms, suggest what is wrong with it

Reasoning Systems

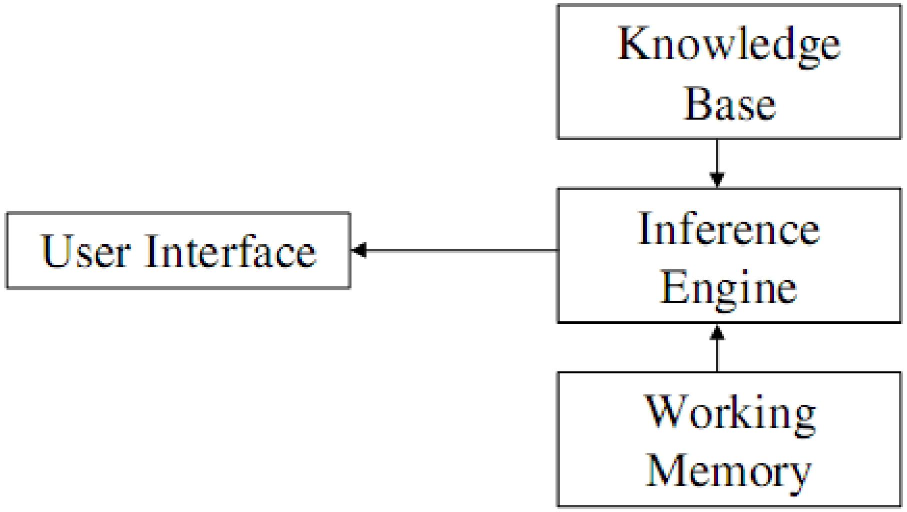
- **Components :**
  - Knowledge System
    - Holds facts and assertions about a problem area.
    - Is not a database
    - Knowledge representation: Array structures, semantic networks, property hierarchies, list structures, predicate calculus expression sets, rules.
  - Language System
    - For stating specific problems to be solved
    - Often rudimentary. Developer may be able to set interface behaviors.
  - Problem Processing System

    Uses knowledge in the knowledge system to infer solutions to problems stated with language system.
    - What knowledge is relevant?
    - Sequence of examination?
  - Presentation System
    - For presenting responses
    - Often rudimentary

Components & Structure of Expert System

- Knowledge Acquisition Subsystem
- Knowledge Base
- Inference Engine
- User Interface
- Working memory
- Explanation Subsystem (Justifier)
- Knowledge Refining System
- User

- Most ES do not have a Knowledge Refinement Component



Knowledge Base

- The knowledge base contains the knowledge necessary for understanding, formulating, and solving problems
- Two Basic Knowledge Base Elements
  - Facts
  - Procedures (Usually Rules)
- Rules
  - IF-THEN-ELSE

Inference Engine

- The terms "inference" and "reasoning" are generally used to cover any process by which conclusions are reached.
- Logical inference ≡ deduction
- Inference is performed by the Inference Engine

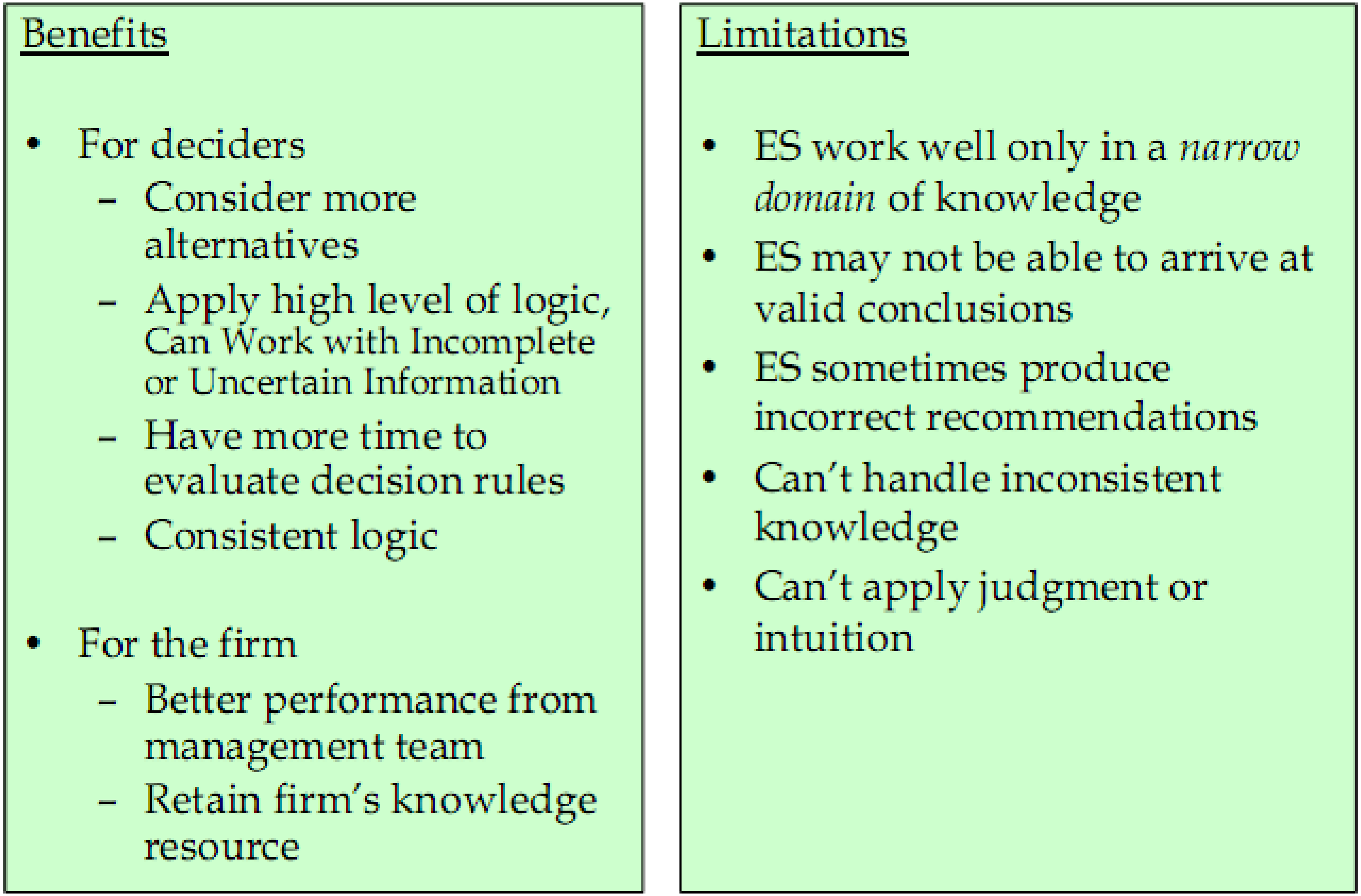
User interface

- Language processor for friendly, problem-oriented communication
- Natural Language Processor, or menus and graphics

Working Memory

- The contents of the working memory are constantly compared to the production rules
- When the contents match the condition of a rule, that rule is fired, and its action is executed
- More than one production rule may match the working memory

Expert System Benefit & Limitation



Benefits

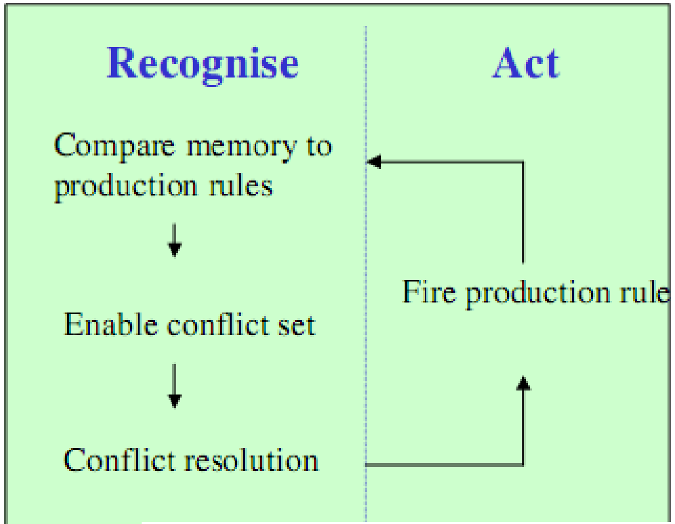
- For deciders
  - Consider more alternatives
  - Apply high level of logic, Can Work with Incomplete or Uncertain Information
  - Have more time to evaluate decision rules
  - Consistent logic
- For the firm
  - Better performance from management team
  - Retain firm's knowledge resource

Limitations

- ES work well only in a *narrow domain* of knowledge
- ES may not be able to arrive at valid conclusions
- ES sometimes produce incorrect recommendations
- Can't handle inconsistent knowledge
- Can't apply judgment or intuition

Expert System Recognise Cycle

- The expert system cycles around in the recognise-act cycle
- Whenever a condition is matched, it is added to the conflict set – all the rules which are currently matched
- The system must then decide which rule within the conflict set to fire – conflict resolution



Recognise

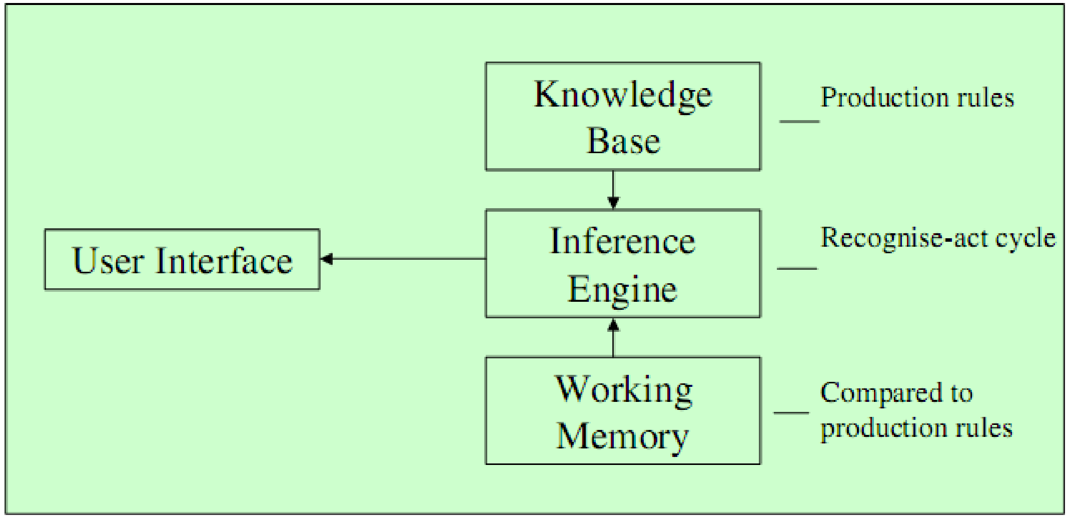
Compare memory to production rules

Enable conflict set

Conflict resolution

Act

Fire production rule



Knowledge Base

Production rules

Inference Engine

Recognise-act cycle

Working Memory

Compared to production rules

User Interface

Expert System - Structure Revisited

Reasoning Systems

- **Two kinds of reasoning :**
  - Forward reasoning -- begins with basic knowledge about problem area. Examines knowledge in a sequence and keeps track of implications until they are discovered to provide a solution.
  - Reverse reasoning -- begins with original problem statement. Decomposes problem into smaller and smaller subproblems. Solves subproblems in an attempt to solve original problem.
- **Types of problem processors**

  Problem processing systems could be general or specific
  - General -- all application specific knowledge stored in knowledge system.
  - Specific -- application specific reasoning knowledge incorporated into problem processor.
- **Types of tools for developing an expert system**
  - Programming languages
  - Shells
    - rule set builder
    - inference engine
  - Integrated environments

    Shell capabilities integrated with other computing

    capabilities into a single tool

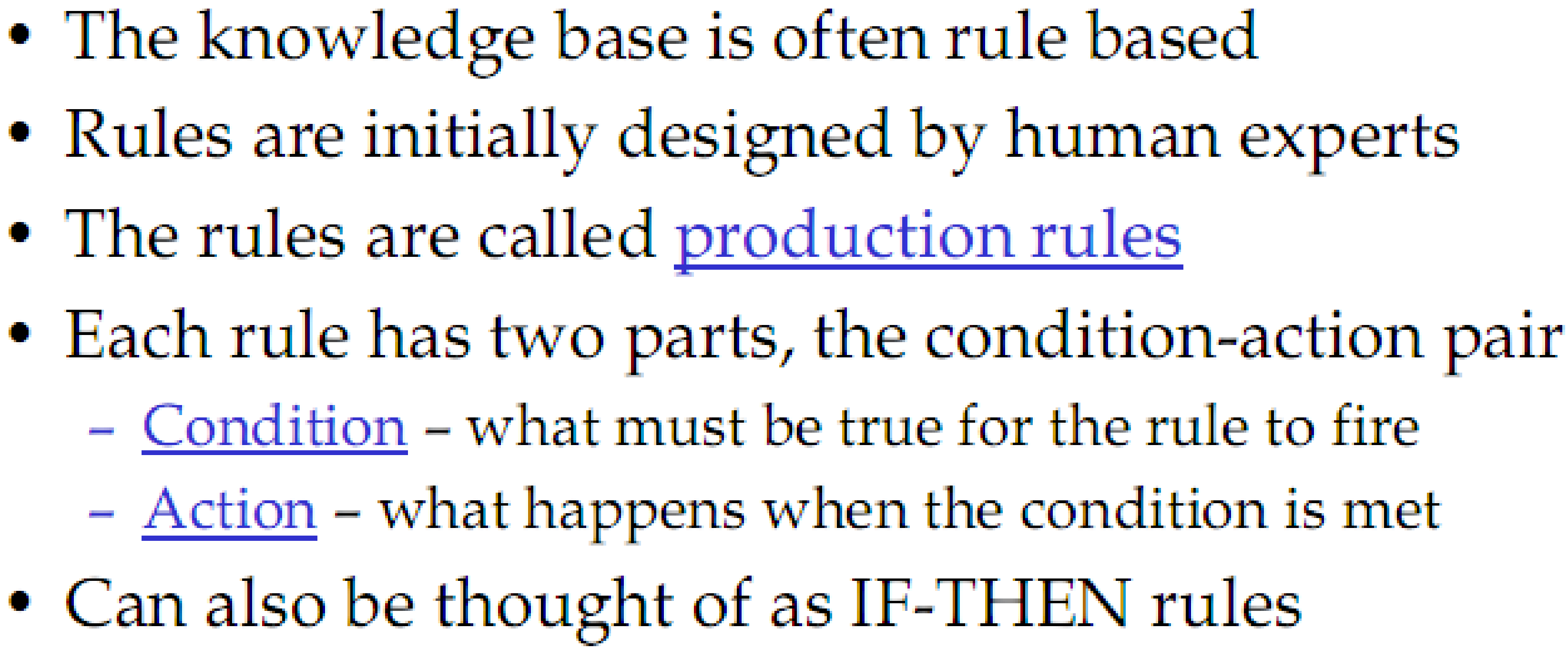
Reasoning Systems

- **Types of tools for developing an expert system (contd)**
  - Rule set builder: (software for building, maintaining, and compiling rule sets)
    - Building: specifying rules and specifying knowledge about usage
    - Maintenance: changing specifications as new reasoning expertise becomes known
    - Compiling: as a consequence of building and maintenance
      - check validity and report errors
      - if specification is valid, generate new version of rule set that saves memory space and solution time
      - optional
  - Inference engine
    - Reasons with any rule constructed via rule set manager
    - Some inference engines support a single rigid user interface
      - gives developer little control over nature of user interfaces. All systems created have essentially the same interface.
    - Others integrate inference engine with familiar I/O capabilities, such as:
      - control over prompt positionings
      - use of form-oriented interaction
      - color and intensity selection
      - customized menus

Reasoning Systems

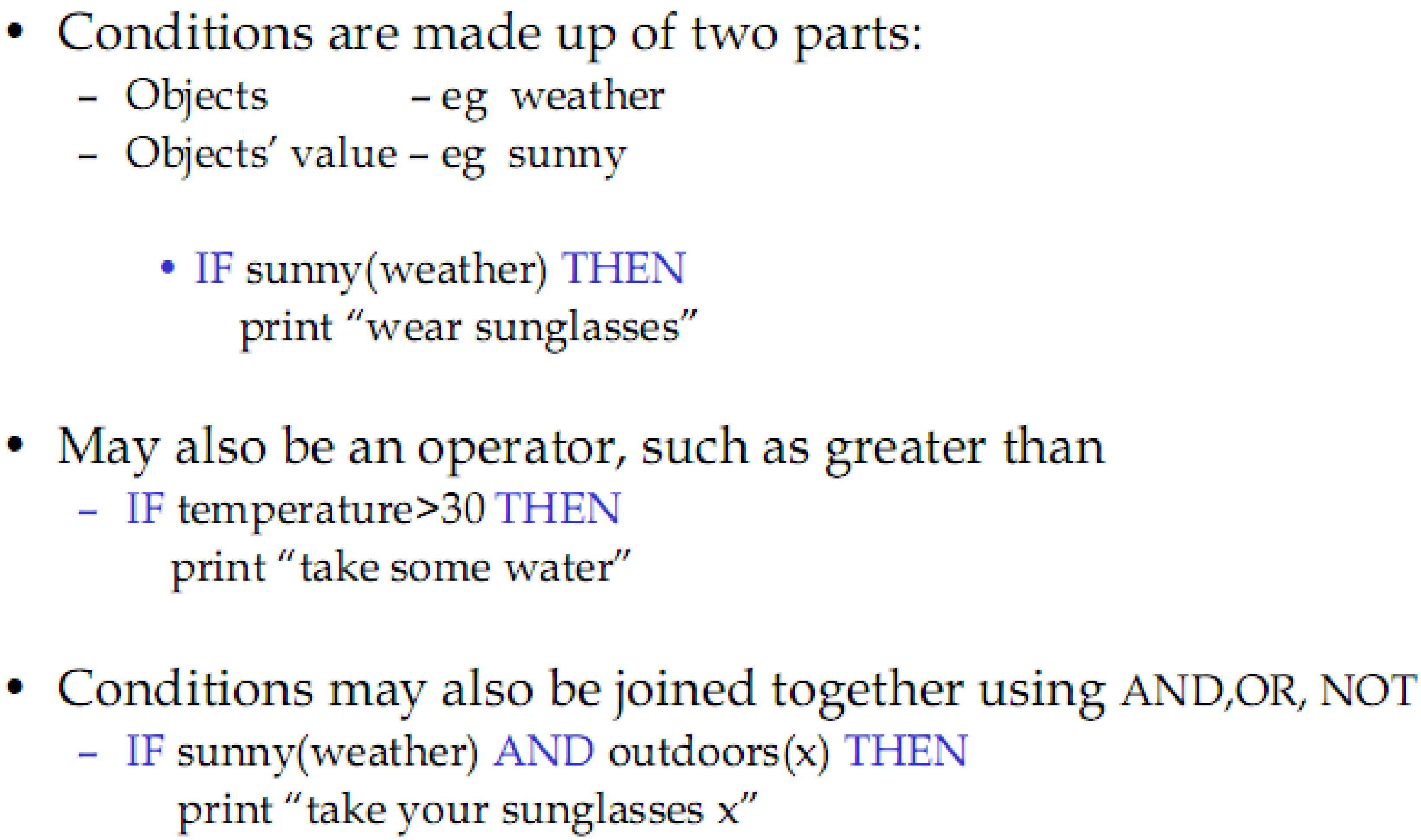
- **Types of tools for developing an expert system (contd)**
  - Inference engine (contd)
    - Power: Related to kinds of rules that can be processed
      - Lower power handles “rudimentary” rules only
      - High power handles “sophisticated” and “rudimentary” rules
    - Kind of reasoning
      - forward reasoning (chaining)
      - reverse reasoning (chaining)
      - both (more versatile)
    - Ability to deal with uncertainty
      - kinds of uncertainties handled
      - control over how these are factored into a reasoning process
  - Integrated environment
    - stand-alone shell: the inference engine is an isolated program
    - integrated environment: the inference engine can be invoked wherever desired
      - within a spreadsheet computation
      - within a procedural model
      - within the midst of text processing
      - etc.

Introduction to Rules



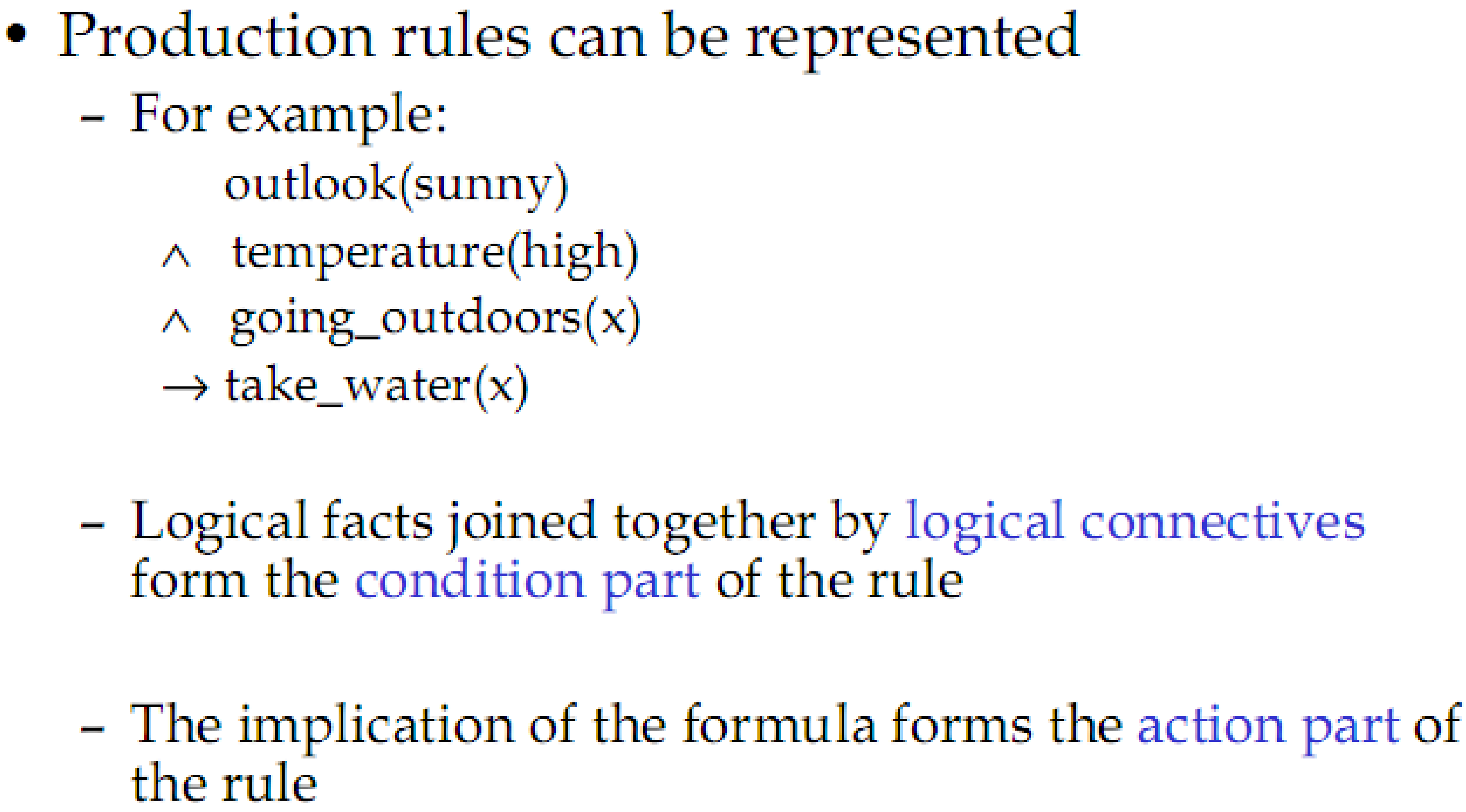
- The knowledge base is often rule based
- Rules are initially designed by human experts
- The rules are called production rules
- Each rule has two parts, the condition-action pair
  - Condition – what must be true for the rule to fire
  - Action – what happens when the condition is met
- Can also be thought of as IF-THEN rules

Rules - Conditions



- Conditions are made up of two parts:
  - Objects – eg weather
  - Objects' value – eg sunny
    - IF sunny(weather) THEN
      print "wear sunglasses"
- May also be an operator, such as greater than
  - IF temperature>30 THEN
    print "take some water"
- Conditions may also be joined together using AND,OR, NOT
  - IF sunny(weather) AND outdoors(x) THEN
    print "take your sunglasses x"

Logic and Expert System

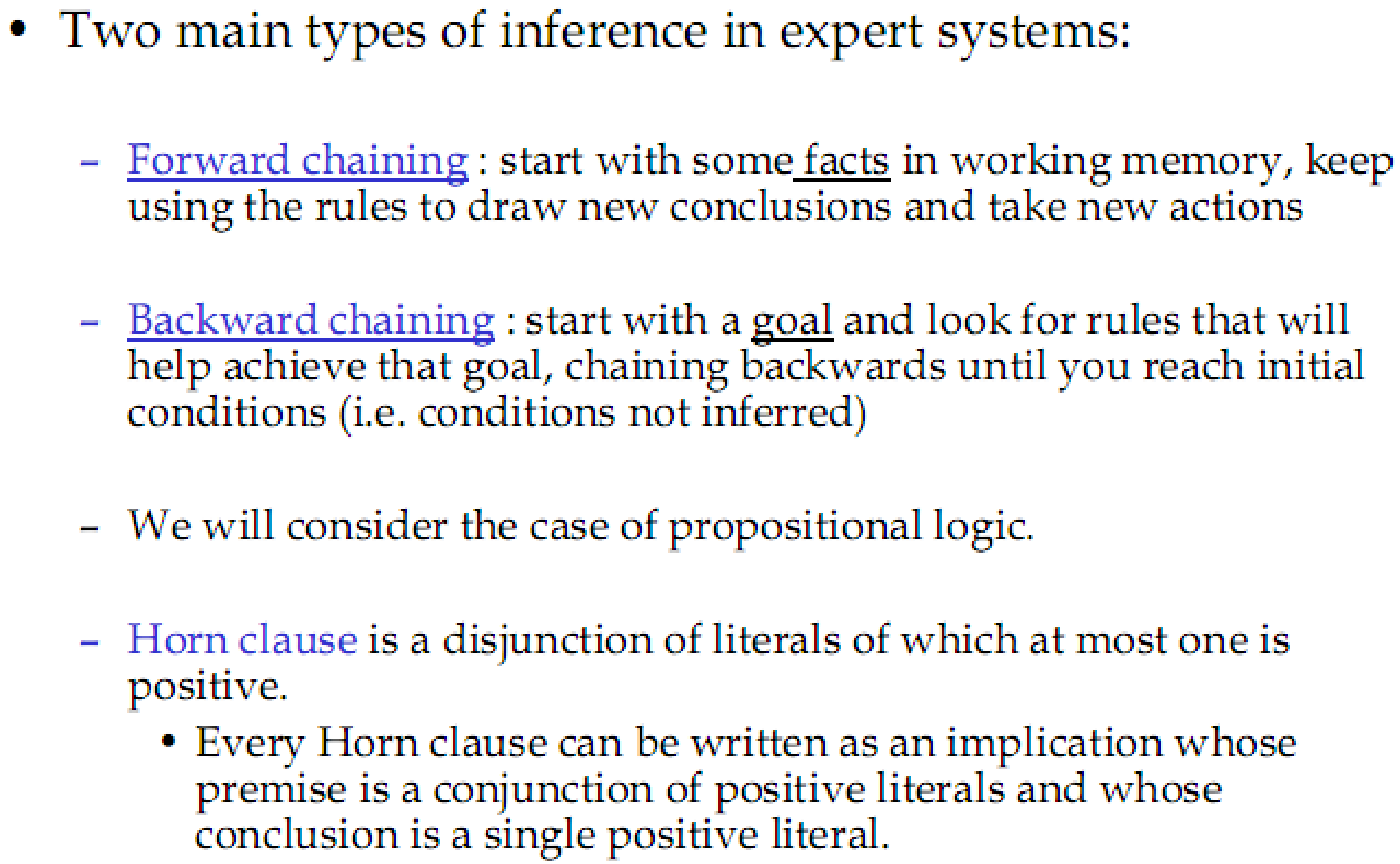


- Production rules can be represented
  - For example:

$$
\begin{aligned}
& \text{outlook(sunny)} \\
\wedge\ & \text{temperature(high)} \\
\wedge\ & \text{going\_outdoors}(x) \\
\rightarrow\ & \text{take\_water}(x)
\end{aligned}
$$

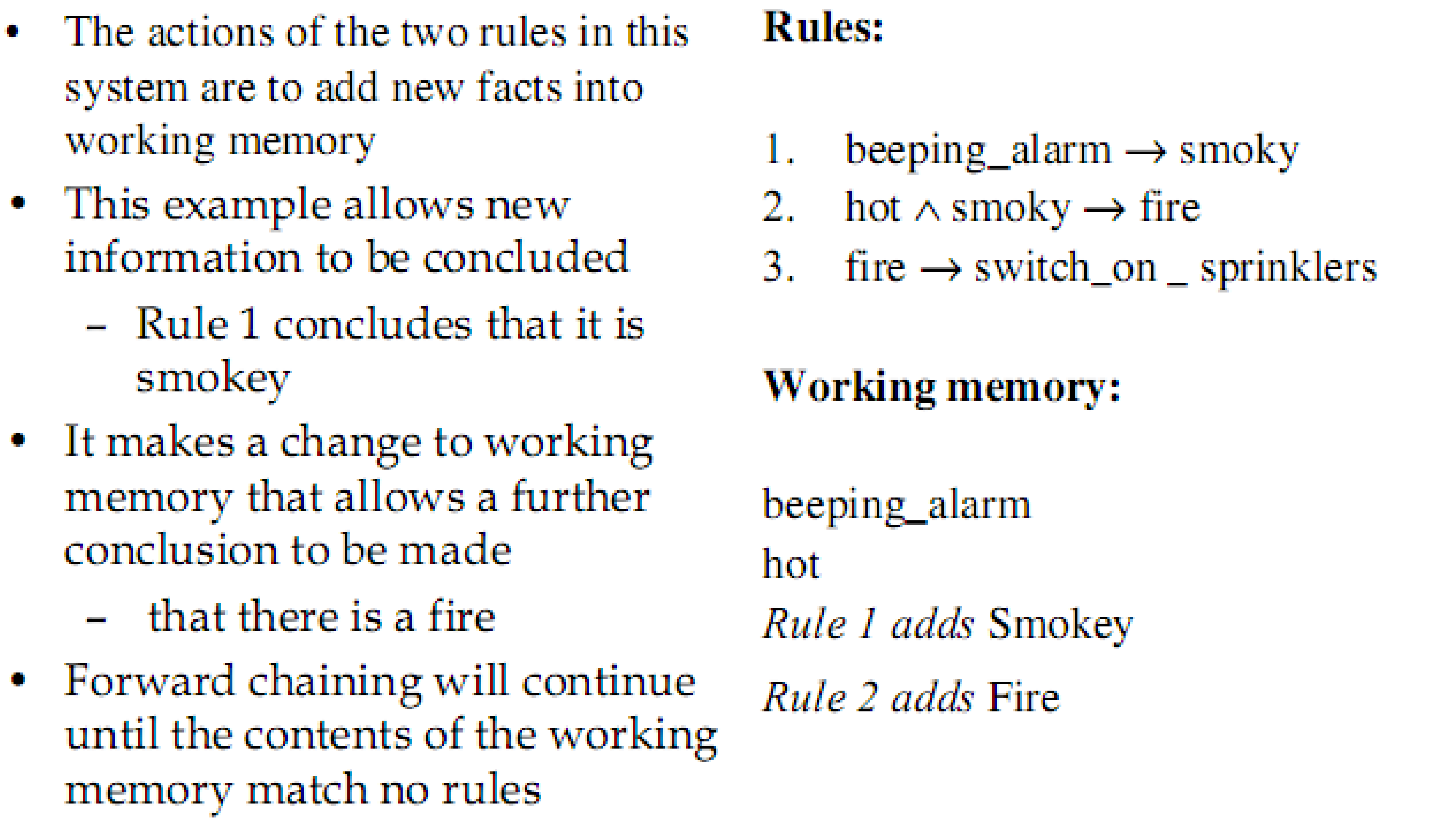
  - Logical facts joined together by logical connectives form the condition part of the rule
  - The implication of the formula forms the action part of the rule

Introduction to Chaining



- Two main types of inference in expert systems:
  - Forward chaining : start with some facts in working memory, keep using the rules to draw new conclusions and take new actions
  - Backward chaining : start with a goal and look for rules that will help achieve that goal, chaining backwards until you reach initial conditions (i.e. conditions not inferred)
  - We will consider the case of propositional logic.
  - Horn clause is a disjunction of literals of which at most one is positive.
    - Every Horn clause can be written as an implication whose premise is a conjunction of positive literals and whose conclusion is a single positive literal.

Forward Chaining



- The actions of the two rules in this system are to add new facts into working memory
- This example allows new information to be concluded
  - Rule 1 concludes that it is smokey
- It makes a change to working memory that allows a further conclusion to be made
  - that there is a fire
- Forward chaining will continue until the contents of the working memory match no rules

Rules:

1. beeping_alarm → smoky
2. hot ∧ smoky → fire
3. fire → switch_on _ sprinklers

Working memory:

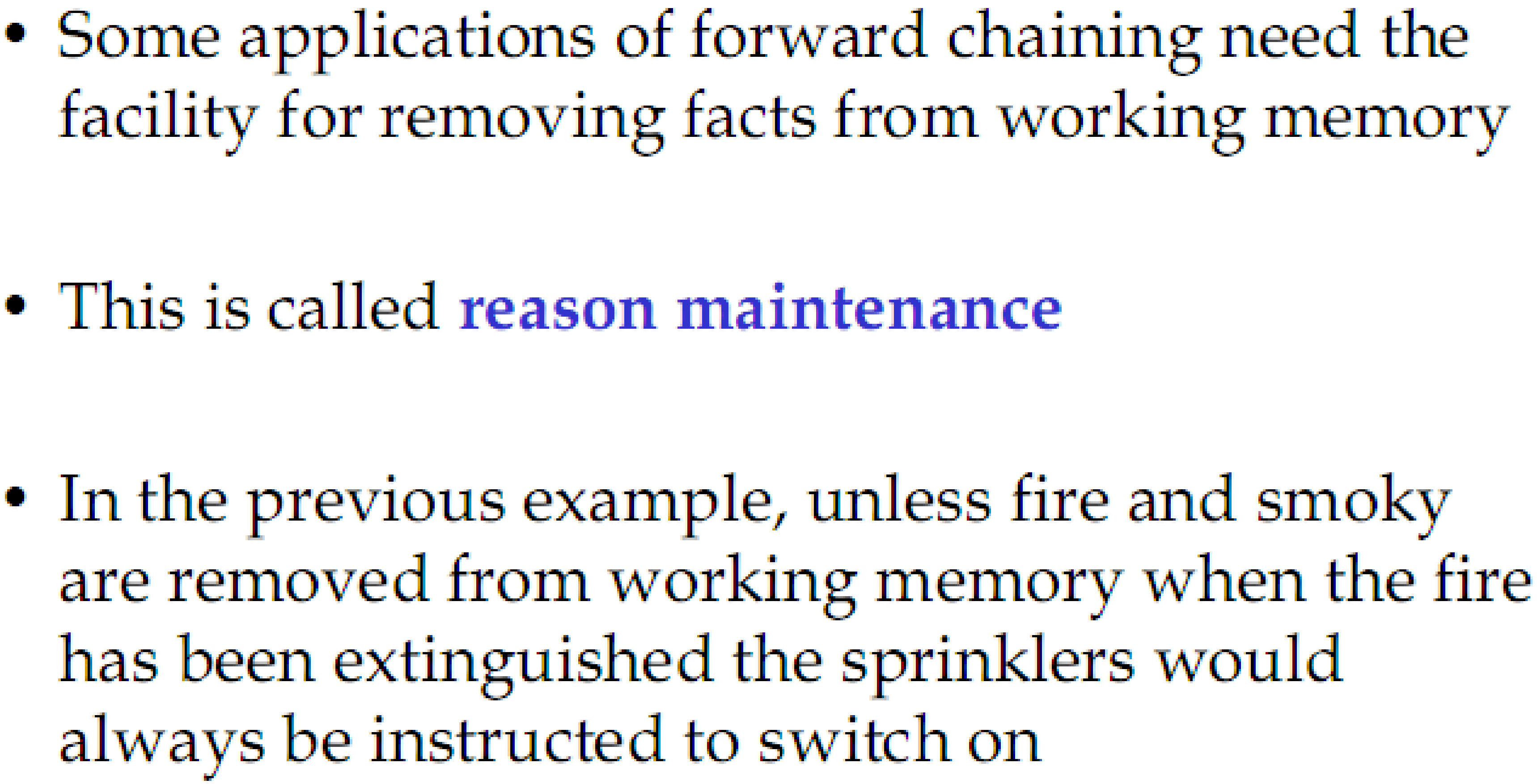
beeping_alarm

hot

Rule 1 adds Smokey

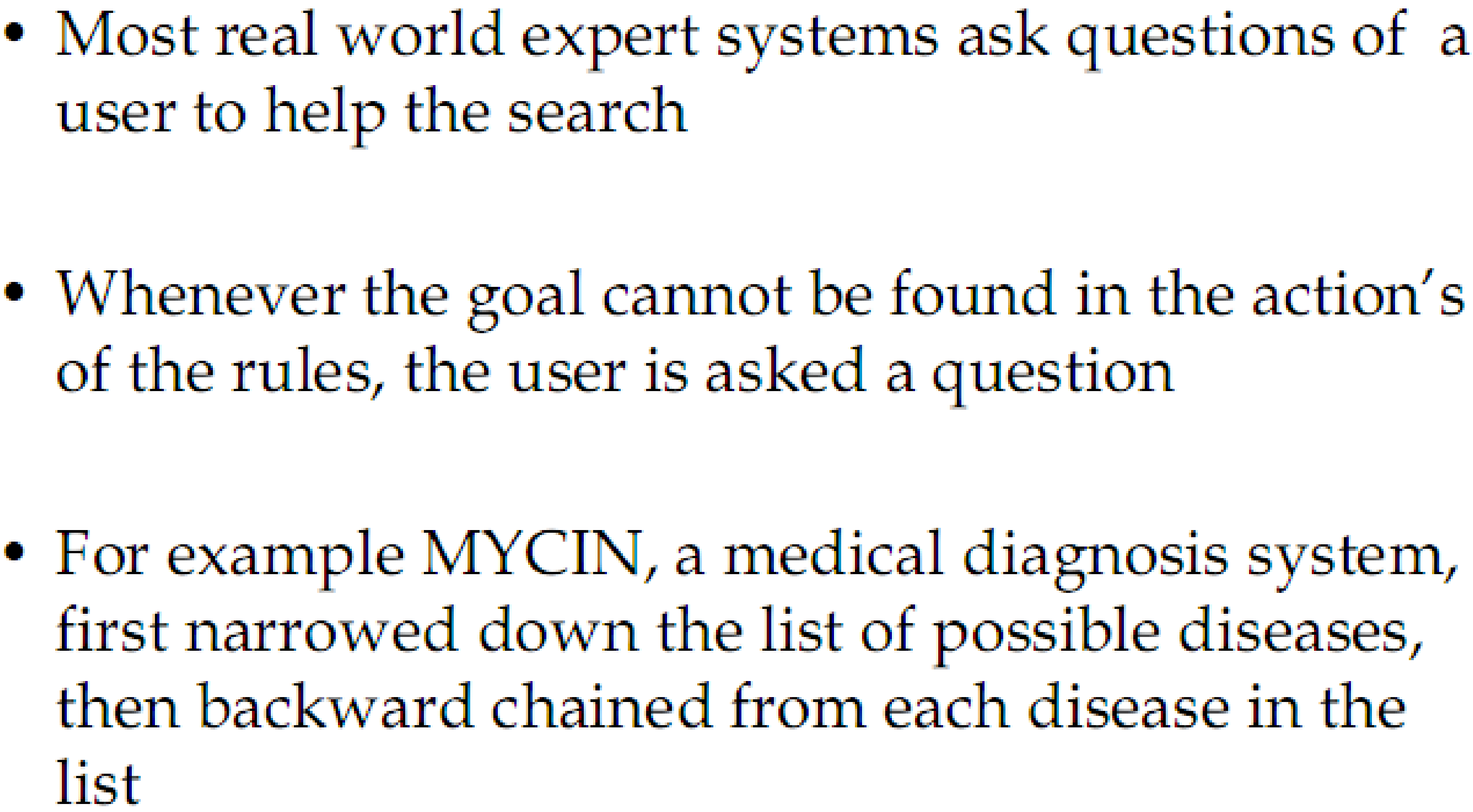
Rule 2 adds Fire

Forward Chaining –Reason Maintenance



- Some applications of forward chaining need the facility for removing facts from working memory
- This is called **reason maintenance**
- In the previous example, unless fire and smoky are removed from working memory when the fire has been extinguished the sprinklers would always be instructed to switch on

Forward Chaining Involving a User



- Most real world expert systems ask questions of a user to help the search

- Whenever the goal cannot be found in the action's of the rules, the user is asked a question

- For example MYCIN, a medical diagnosis system, first narrowed down the list of possible diseases, then backward chained from each disease in the list

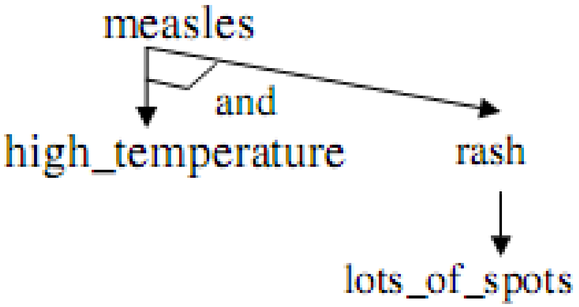
Forward Chaining Involving a User Example

Rule base:

1. high_temperature ∧ rash → measles
2. high_temperature ∧ headache → flu
3. lots_of_spots → rash

Patient complains of high temperature

MYCIN has forward chained and cut the list of possible illnesses to two

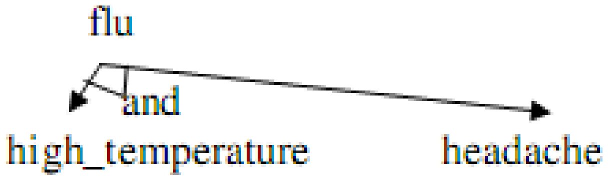


User interface:

Patient complains of high temperature

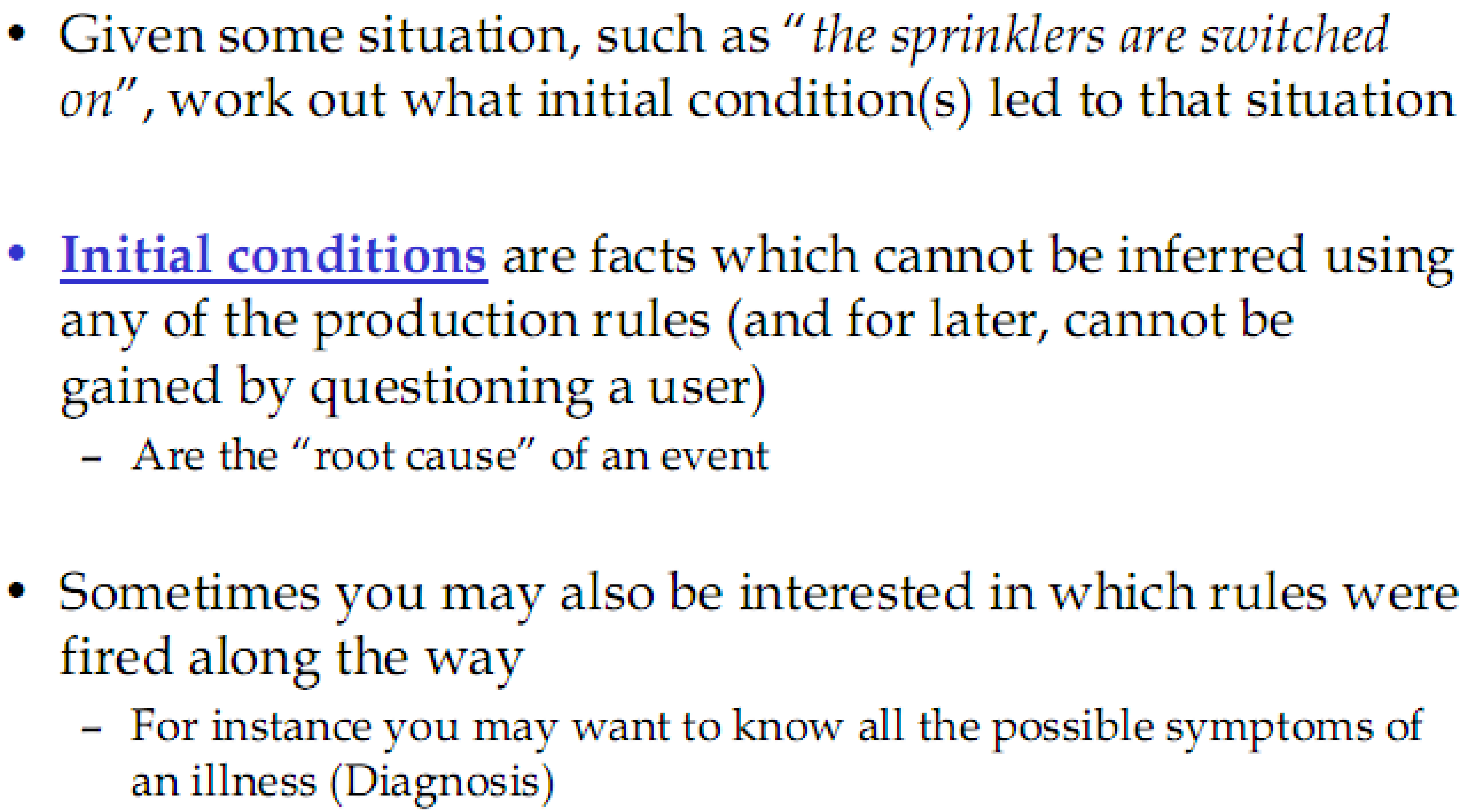
Q: Does the patient have lots of spots?
A: No

Q: Does the patient have a headache?
A: yes



Patient has flu

Backward Chaining

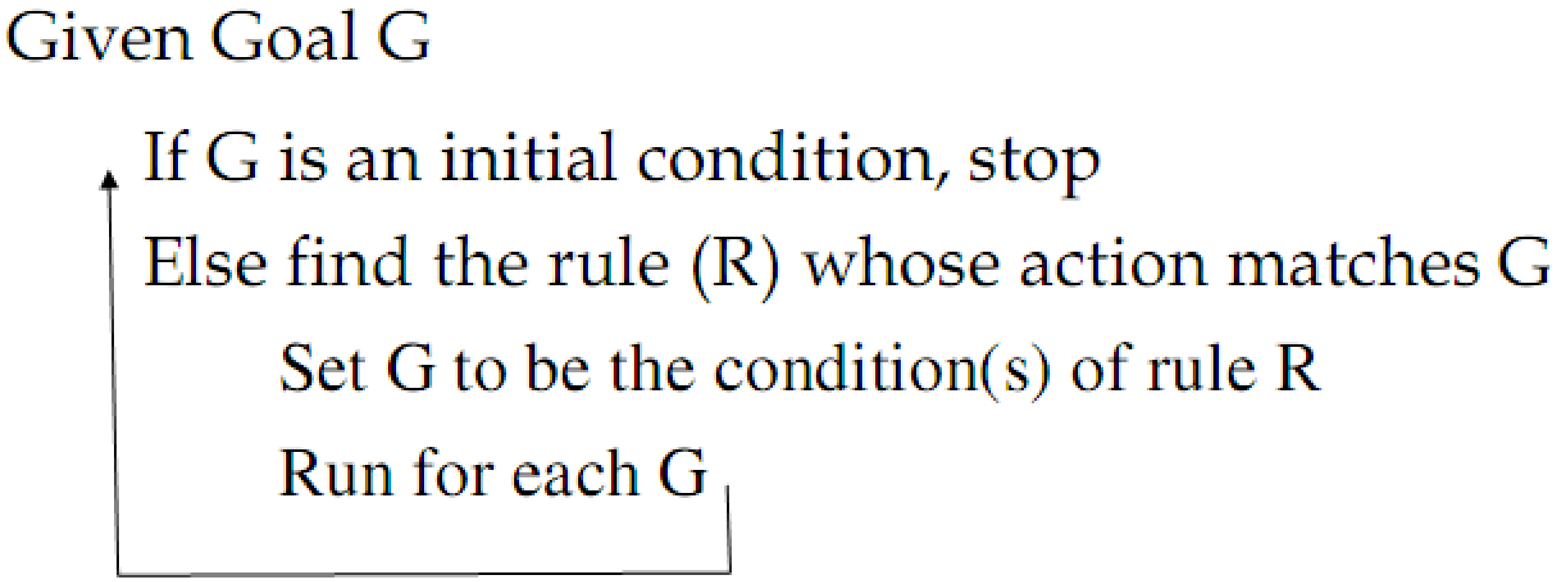


- Given some situation, such as "*the sprinklers are switched on*", work out what initial condition(s) led to that situation

- **Initial conditions** are facts which cannot be inferred using any of the production rules (and for later, cannot be gained by questioning a user)
  - Are the "root cause" of an event

- Sometimes you may also be interested in which rules were fired along the way
  - For instance you may want to know all the possible symptoms of an illness (Diagnosis)

Backward Chaining Algorithm



```
Given Goal G
    If G is an initial condition, stop
    Else find the rule (R) whose action matches G
        Set G to be the condition(s) of rule R
        Run for each G
```

Backward Chaining Example

G = “sprinklers_ on”

R = 3,

G3 = “fire”

G3 is not an initial condition

G3 = “fire”

R = 2,

G2a = “hot”

G2a is an initial condition

G2b = “smokey”

G2b is not an initial condition

G2 = “smokey”

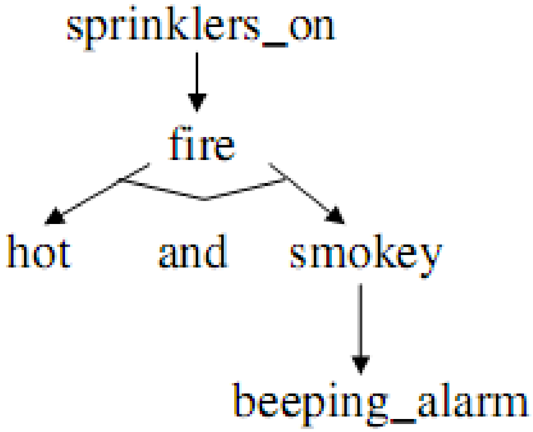
R = 1

G1 = “beeping_alarm”

G1 is an initial condition

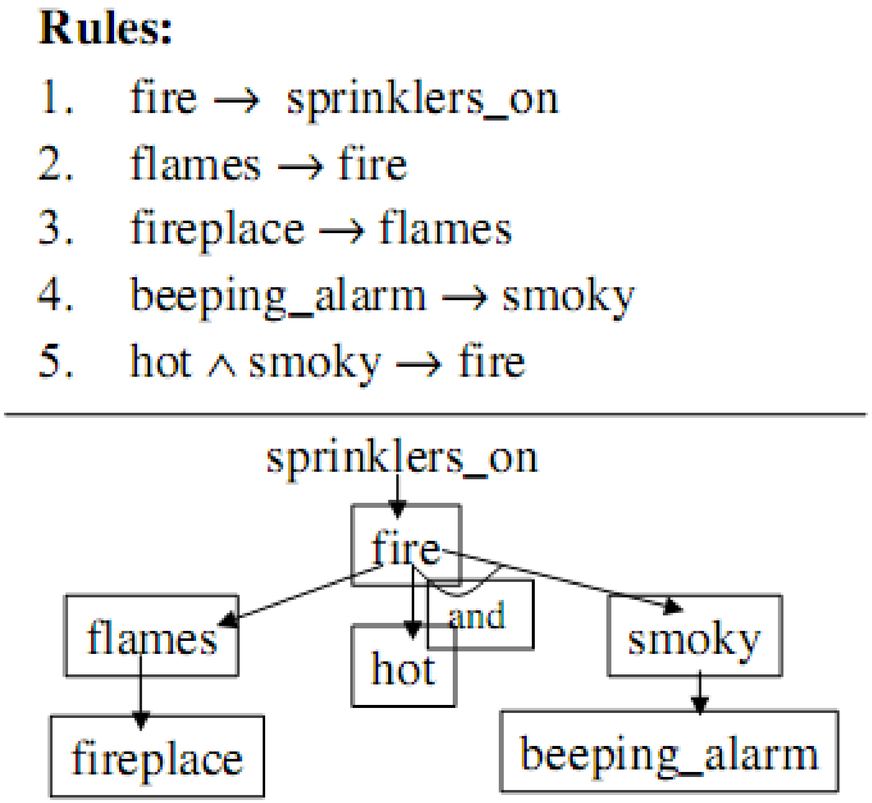
Rules:

1. beeping_alarm → smokey
2. hot ∧ smokey → fire
3. fire → sprinklers_on



Backward Chaining Example

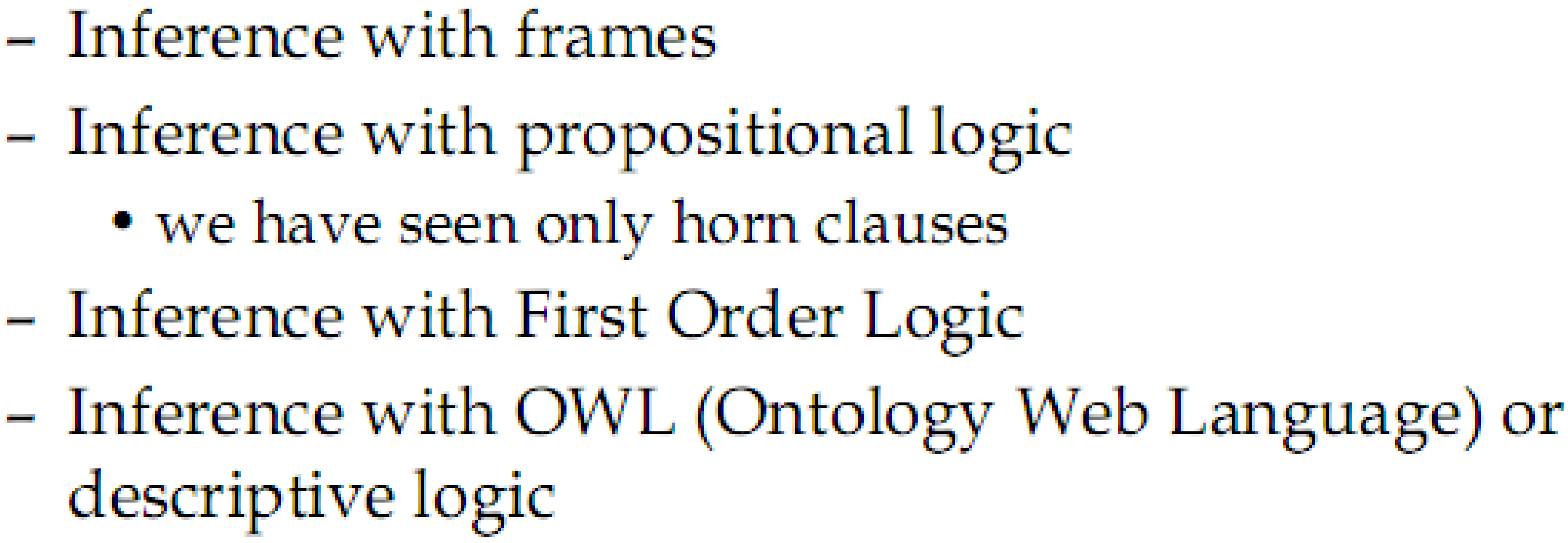
- In the previous example, each goal matched only one rule
- But supposing it matched more than one?
- In this case a tree can be built as before, and depth first search can be used to find the chains of rules



Rules:

1. fire → sprinklers_on
2. flames → fire
3. fireplace → flames
4. beeping_alarm → smoky
5. hot ∧ smoky → fire

More Inference Techniques



- Inference with frames
- Inference with propositional logic
  - we have seen only horn clauses
- Inference with First Order Logic
- Inference with OWL (Ontology Web Language) or descriptive logic

2. Natural Language Processing

- Allow humans to interact with computers in a natural language rather than a computer language.
  - No rigid syntax requirements
  - Conversational interaction with user
  - Can be customized
  - Automatic error detection/correction
  - Interpretation based on context
  - Used for information retrieval, data modification, numeric computation, statistical analyses, graphics generation, expert system consultation, etc.
- Value to
  - Managers
  - Support Staff
- Typically standalone utility operating on a database

3. Knowledge Representation

- How knowledge is stored in the knowledge system.
  - Prevalent method for storing expert system reasoning knowledge is “productions” (i.e., rules).
  - Other representations include array structures, semantic networks, frames, predicate calculus expression sets, etc.
  - Knowledge is not monolithic.

4. Machine Learning

- Deals with the ability of a software system to learn.
  - Issues include learning from experience, learning by examples, learning by analogy, behavior modification, fact accumulation.
  - Neural networks
  - Genetic algorithms

5. Automatic Programming

- Concerned with mechanisms for automatically generating a program to carry out a prescribed task.
  - User describes desired characteristics and behavior only.
  - No need to specify how to build the program.
  - Automatic programming software builds the program.
  - A reasoning system might have program generation as its objective.

6. Pattern Recognition

- The ability of a system to recognize visual and audio patterns.
  - Goes beyond sensing keystrokes or mouse movements: recognizing audio and visual patterns
  - Considerable impact on robotics.

Potential Benefits

- Timely advice
- Replication
- Frees human expert
- Consistent, uniform advice
- Explains itself
- May handle uncertainties
- Evolution
- Formalization of expertise

Competitive Implications

- Increasing Internal Productivity
  - Setting
    - large drug company
    - regional sales managers
  - Strategy
    - enhance competitiveness by increasing sales manager productivity
    - reducing time and effort involved in setting sales quotas
    - increasing effectiveness through replication of expertise
  - Implementation
    - expert system to offer quota advice
    - uses reasoning expertise of an expert quota setter as well as other kinds of knowledge (descriptive, procedural, etc.)
  - Applications
    - operational control
    - management control
    - strategic planning
    - structured to unstructured

Competitive Implications

- Providing Enhanced Services
  - Setting
    - large division of industrial chemicals company
    - many chemical products have multiple uses
    - often several chemicals that could meet customer need
    - customer does not know what amount of what chemical it needs
    - salesperson attempts to provide answers
  - Strategy
    - enhance service provided
    - focus on furnishing solutions rather than merely selling chemicals
    - help customer clarify problem faced, offer expert advice about solving it, justify the advice
  - Implementation
    - sufficient in-house expertise exists, but how can it be delivered
    - more technical training for sales reps (lengthy, costly, infeasible, vulnerable)
    - make technical experts directly accessible to customers (scarce, not trained in sales, contention)
    - expert system for portable or customer computer (effective, controllable asset)

Competitive Implications

- Providing New Services
  - Setting
    - small retail bank
    - cannot afford in-house investment banker
    - inflexible consideration of customer loan requests
  - Strategy
    - increase customer base by offer ing investment banker services
    - provide customized financing arrangement for small applicants
  - Implementation
    - expert system that advises loan officer about customization
    - draws on investment banker expertise, knowledge about the customer, economic forecasts, etc.
    - development cost shared by consortium of banks

Competitive Implications

- Spawning New Industries
  - Publishing
    - expert system as alternative delivery to books, articles, lectures
    - alternative delivery to consulting firms, professionals
    - publishing chunks of reasoning knowledge
    - plug into generalized inference software
    - subscription services
  - Librarian/Teacher
    - compact disk era
    - expert systems to dig out and apply relevant knowledge
  - Artificially Intelligent Business Systems
    - application systems for record-keeping
    - decision support systems
    - natural result of integrated knowledge management

Sample Applications

- Establishing sales quotas
- Conducting trainee orientations
- Recommending acquisition strategies
- Generating project proposals
- Planning advertising spot layouts
- Job shop scheduling
- Facilities maintenance
- Selection of forecasting models
- Determining credit limits

- Selecting transport routes
- Providing investment counseling
- Analyzing market timing situations
- Offering job-costing advice
- Assessing job qualifications
- Performance evaluation
- Requirements planning
- Application of discounting policies
- Responding to customer inquiries

Examples of Vendors of DSS-Related Software

- Acquired Intelligence, Inc. (expert system development tools)
- AskMe (employee knowledge network software)
- ATLAS.ti (knowledge workbench)
- Autonomy (knowledge management software)
- BackWeb Technologies (tools for intranet/extranet knowledge distribution systems)
- Converva (knowledge access software)
- Decision Support Associates, Inc (business simulation systems)
- Entrieva (KM portal software)
- EXSYS (expert systems development software)
- EZ-Xpert (expert system code generation)
- Fuzzy Systems Engineering (fuzzy logic products)
- Higher Level Systems (knowledge management software)
- Hummingbird (BI & knowledge management software)
- Hyperion (knowledge management, DSS software)
- Hyperknowledge (knowledge management software)
- Information Builders (knowledge management tools)
- Inxight Software (KM portal software)
- Livelink Discovery Server (knowledge management software)
- Manifold Net (database, GIS, 3D analysis software)
- Microsoft (knowledge management tools)
- OpenText (knowledge management software)
- Partek (analysis, inference, modeling)
- Rocket Software (business intelligence)
- Service Ware (knowledge management for customer support)
- TEC (Which & Why decision valuation software, demo download)
- Teknowledge (expert system development tools)
- Topiary (self-service automation, online demo)
- Vanguard (decision support analysis and modeling software)
- Vignette (knowledge management system with collaboration, analysis, search features)
- Verity (knowledge selection software)
- Xpert Universe (expertise location software)

Referensi

1. Dr. Mourad YKHLEF,2009,Decision Support System-Intelligent DSS, King Saud University
2. Dr. Clyde W. Holsapple and Dr. Andrew B. Whinston,2001, Decision Support Systems : A Knowledge-Based Approach, Springer(online), available at : http://www.uky.edu/BusinessEconomics/dssakba/ (Accesed : 14 December 2010)