Topic 1: Introduction to Knowledge-Based Systems (KBSs)

Contents

1.1 What is a KBS?
1.2 Separation of Knowledge & Control
1.3 Algorithmic vs. Heuristic Solutions
1.4 Characteristics of KBSs
1.5 Introduction to Prolog
• a las 15:00 pondré una hoja por grupo de prácticas en la puerta del laboratorio B-208, para que los alumnos se apunten en el grupo que más les convenga.

• Si ya les habéis dicho que las pondré en el B-303 (mi despacho), no problem, porque dejaré una nota en la puerta diciendo dónde están finalmente.

Las hojas las retiraré el viernes a las 18:00.

Hay un grupo (el Tc, que finalmente es el lunes de 15:00 a 17:00), que está reservado SOLO a alumnos de informática + matemáticas.

1.1 What is a Knowledge-Based System?

“A computer program that uses knowledge of the application domain to solve problems in that domain, obtaining essentially the same solutions that a person with experience in the same domain would obtain.”

• This definition would (falsely) include conventional programs that calculate current in a circuit, or forces and tensions in a structure, etc.

• A KBS is something more than a program which just copies the algorithm/formula used by the expert. It must be able to use the information in an “intelligent” way
1.1 What is a Knowledge-Based System?

Mycin sample dialogue (simulated):

System: What is the patient's name?
User: George Smith
System: What is the patient's age?
User: 52 years
System: Is the patient male or female?
User: Male
System: What are the symptoms?
User: Headache dizziness upset stomach high fever
System: When did these symptoms first appear?
User: 23/03/2002
System: I conclude that the patient has a viral infection, my recommendation is that he should take two aspirin, drink plenty of fluid, get lots of rest.

3 fundamental differences between KBS and other types of software systems:

1. Separation between knowledge and the use of knowledge
2. Utilisation of knowledge specific to a domain
3. Heuristic nature (as opposed to algorithmic)
1.1 What is a Knowledge-based System?

3 fundamental differences between KBS and other types of software systems:

1. Separation between knowledge and the use of knowledge

First developed in the GPS system (early 60s): a system that applies means-ends analysis (uninformed, blind search) to reduce differences between the current problem state and the solution, by applying operators that change a state into another, eliminating some differences.

2. Utilisation of knowledge specific to a domain

Originated with DENDRAL (early 70s), a system capable of inferring the unknown molecular structure of some chemical products, by observing their mass spectrum. Part of its success came from the use of rules specifically used by human experts to interpret mass spectrum and propose fragments of structure from the spectrum.

...
1.1 What is a Knowledge-based System?

3 fundamental differences between KBS and other types of software systems:

- ... 

3. Heuristic nature

Also present in DENDRAL, but was more obvious in MYCIN (1976), a system capable of diagnosing and recommending treatment for infectious blood diseases, after extracting information about the patient's condition by asking questions about his/her characteristics, lab. test results, etc.

1.1 What is a Knowledge-Based System?

KBS: History

- The first Expert Systems (now called KBSs):
  - DENDRAL (1969): Prediction of molecular structure; this signalled a transition: search -> knowledge
  - MYCIN (1976): Diagnosis of diseases, blood infections, etc.
  - PROSPECTOR: (1978) Evaluates of the mineral potential of site
  - XCON (1982): System for configuring computer hardware purchases (for DEC's VAX), using several thousands rules, early commercial success of KBS.

- Their good results in practice contributed to their increasing popularity.
- They became the first commercial (profitable) products of AI
- Sometimes KBS and AI were (mistakenly) identified as one
- But the truth is that IA - KBS != { }
1.1 What is a Knowledge-based System?

A Short History:
- 1960s: Search (main AI paradigm): GPS, General Problem Solver
- 1970s: Use of knowledge specific to a given application domain
- 1980s: First ‘Expert Systems’ (later called KBS)
- 1990s: Shift of focus towards automating knowledge acquisition

Developments:
- from slow computers to fast processors
- from simple problems (polynomial) to complex problems (NP)
- from closed solutions to approximate solutions

May Include:
- Neural Networks (some include it)

Does not include:
- Data Mining (knowledge discovery)
Topic 1: Introduction to Knowledge-Based Systems (KBSs)

1.2 Separating Knowledge and Control

- Classical view (N. Wirth): Algorithm + Data Structure = Program
- KBS view: Program + Knowledge = KBS

- The fundamental characteristic of KBS: Separating knowledge from the application of knowledge
- The KBS embodies a program whose goal is to apply (interpret) the given knowledge
- Example: In your AI course:
  - Predicate Logic is a form of representing knowledge
  - Unification+resolution is an algorithm to apply knowledge
1.2 Separating Knowledge and Control

Algorithms which know how to apply knowledge

Knowledge about the domain
1.2 Separating Knowledge and Control

Benefits of separating Knowledge and Use

- KBSs become easier to develop and maintain.
- We can focus on the knowledge, independently of the program

(Although in practice, there is no total independence)
1.2 Separating Knowledge and Control

The Inference Engine can be Generic (reusable)

- In the 70s and 80s, many KBS proved that the inference techniques developed for a given domain could be applied to other domains:
  - Medicine (INTERNIST, 1970; TEIRESIAS, 1978)
  - Speech understanding (HEARSAY, 1975)
  - Geology (PROSPECTOR, 1978)
  - Computer configuration (XCON, 1980)

- Going further, MYCIN showed that the same inference engine could be re-used with different knowledge base to become a different “expert system”
- MYCIN was thus the first expert system “shell”

---

**Diagram:**

- **KBS**
  - Inference Engine
  - Knowledge Base

- **Problem Data**
  - Different Data for each problem

- **Generic:** same engine for different problems (medical diagnosis vs car diagnosis)

- **Different KB for each Domain**
1.2 Separating Knowledge and Control

What is knowledge?
- Data (storage, recordings, codification, ...)
- Information (useful, communication, awareness, ...)
- Experience, intuitions, clues, hints, doubts, ...

• Is this equation correct?
  Knowledge = Expert-Average Person

• Knowledge (as experience) sometimes lacks the precision of mathematical formulas, biological taxonomies, ...
• It often has heuristic nature: it is approximate, incomplete.

1.2 Separating Knowledge and Control

Different types of expertise (knowledge)

1. **Associative knowledge**: observations of input/output behaviour, ...

2. **Motor knowledge**: automatic reactions, semi-spontaneous, ... relative to physical behaviour: to walk, ride a bicycle, talk, etc.

3. **Theoretical knowledge**: theoretical models (deep knowledge) that attempt to explain everything regarding the application domain, with precision and detail.
1.3 Algorithmic and Heuristic Solutions

Let us think how we solve some problem with heuristics

1. **The 9 points**: draw 4 contiguous strokes that passes through all points without taking the pen off the paper

2. Use scissors to cut a square of paper in 12 equal parts with the smaller possible number of rectilinear cuts.

3. **Rubick’s Cube** (blind search is out of the question)

4. **Others puzzles**, other activities in which we are expert: walking, riding a bicycle, cooking, programming …

Basically, we find solutions which work, but are not based on theory.
1.3 Algorithmic and Heuristic Solutions

Algorithmic method:
- The programmer has a complete knowledge on the problem and the way to calculate the solution.
- In addition, this is implementable and it is possible to be computed with realistic resources and time.
- The programmer "copies" this algorithm in his program.

Heuristic method:
- The programmer does not have a complete knowledge of the problem, he does not know how to calculate the solution, or it is not implementable with realistic time and resources.
- The programmer constructs a program that uses tricks, knowledge taken from the experience, etc. The program finds approximate solutions: "sufficiently good".

Basic techniques for solving problems:
- **Generate-and-test** (trial and error): generally inefficient by nature, but can be improved by incorporating knowledge at the generation phase
  
  Algorithm:
  1. Generate a solution;
  2. Test if it is correct;
  3. If not, return to 1;

- **Heuristic techniques**, for problems without known algorithmic solution (decidability), or with known complex/inefficient solution (NP, non-deterministic polynomial)

These techniques are both used within KBS
1.4 Knowledge-Based Systems: Characteristics

**Advantages**

1. Allow distribution of ‘expertise’ (which is often scarce);
2. Easy to update (relatively to the equivalent traditional system);
3. Consistent answers (without the anomalies sometimes exhibited by human experts);
4. Availability (24 hours/day, 7 days/week);
5. Preservation of expertise;
6. Capable of "working" with incomplete information;
7. Capable of explaining their solutions (and their reasoning);
8. Encourages organizations to clarify the logic of their decision-making;
9. Never "forgets" to ask a question, as a human might.
1.4 Knowledge-Based Systems: Characteristics

Disadvantages
1. Answers are heuristic, could even be incorrect
2. Limited to a very specific domain (field) of application
3. Hard to extract the knowledge from the expert (introspection is hard)
4. (In general) Lack of commonsense
   - Example: system concluded water in pipes flowing at -12°C
5. Cannot make creative responses as human expert would in unusual circumstances
6. Cannot adapt to changing environments, unless knowledge base is changed

1.4 KBS: Evolution, Development and Future

Development of KBS
• Lifecycle of Software: (Software Engineering)
  – specification, design, implementation, validation, and maintenance
• Lifecycle of KBS:
  – very difficult to specify, as it depends heavily on the expertise of human experts, which need to be extracted via many interactions with them, and this often takes most of a project’s time and resources.
• There are fewer computer tools for implementation and validation (compilers, test generators, ...) sometimes are part of the KBS itself
1.4 KBS: Evolution, Development and Future

Knowledge Engineering (def.):

“The acquisition of knowledge regarding an application domain (from sources of implicit knowledge, such as experts, books, observations, ...) and its conversion into a format that allows its automated use for solving problems typically only solvable by human experts in the domain.”

- This mentions only the extraction and representation of knowledge
- In addition, we should include the development and use of tools suitable for knowledge manipulation.
- This includes tools not only for “representing” knowledge, but also for recognizing relevant knowledge, and classifying/organize knowledge (... ontologist)