
An Introduction to Artificial Intelligence

Michaelmas 2000, 4 lectures
4 lectures

Prof D W Murray
dwm@robots

Contents

- 1) Introduction
- 2) Search – weak methods
- 3) Using Logic for Representation
- 4) Situations, Events and Planning

Books

Recommended:

Artificial Intelligence (2nd ed) by Elaine Rich and Kevin Knight, McGraw Hill (1991). ISBN 0-07-100894-2

Artificial Intelligence (3rd ed) by P H Winston, Addison-Wesley (1992), ISBN 0-201-53377-4

Artificial Intelligence – a modern approach by Russell and Norvig. Prentice Hall Int 1995 ISBN 0-13-360124-2

Lecture 1

The essentials of AI

Reading

Rich and Knight (1991) Chapter 1

Winston (1992): Chapters 1 and 2.

Also Russell and Norvig: Chap 2 on Intelligent Agents

1.1 Background

The design and implementation of systems that possess, reason with, and acquire knowledge is arguably the ultimate intellectual challenge. So why then, when we open almost any book on Artificial Intelligence, does it open with a painstaking, almost defensive, definition of what AI is and what AI is not?

Such caution is understandable, for during its short history AI has been dogged by periods of scientific over-optimism followed by disappointment, echoed by similar cycles of media hype' and disenchantment. It is a young and developing discipline: one which has already achieved much, but — if we believe the first sentence — has much to achieve. It is a subject too which raises people's inherent fear of the subjugation of their own intellect by machines, machines lacking in human frailty. Perhaps little wonder that AI is often misunderstood and its current capabilities at once over-estimated by some and berated by others.

As with anything, the nature of AI is best understood by looking at the subject's achievements; but, for what they are worth, here are two definitions from opposite ends of a spectrum of 'one liner's':

- AI is the attempt to build computational models of cognitive processes.
- AI is about generating representations and procedures that allow machines to perform tasks that would be considered intelligent if performed by a human.

The first points both to AI's lofty goal and to its roots in psychology, philosophy and linguistics. Artificial intelligence complements the traditional perspectives on intelligence gained from those subjects because computer models force clarity of thought and precision, and

provide generally useful, patient, but uncompromising experimental punch-bags. Creating AI allows us to learn about “real” intelligence. The second one-liner is much more practically based and indicates why engineers should be interested in AI. If machines are able to act in an human-like intelligent manner, they will be more *useful*. The lofty goal of engineering, after all, is to put natural science usefully to work for mankind. Another one liner which appeals particularly to those involved in understanding perception is that

- AI is the the link between sensing of the world, perception of the world, and intelligent action in the world.

1.1.1 Application domains

It is possible to distinguish three rather different areas of application, expert task solving, natural task solving and formal task solving.

Expert Tasks

The most commercially advanced is the first, that of rule-based or expert systems. These are most often used as expert assistants in a wide range of fields. There are systems which would help with for example

- Designing VLSI chips
- Architectural design
- Diagnosing infections
- Advising on agriculture
- Prospecting for oil and minerals
- Solving symbolic mathematics
- Diagnosing faults in aero engines
- Analysis (science, finance etc)

Although these are specialist or expert problems, one could pick similar problems which confront an average householder from day to day

- Designing a DIY wardrobe
- Planning a herbaceous border to provide year-round colour
- Deciding whether a child has *roseola infantum* or *german measles*
- Figuring out what is up with car

and so on.

When we think of these everyday problems, we realize that computers have made little impact on everyday decision making, and what enormous potential exists. The computers and computer output that most people rub up against routinely are of the cash-register variety

— the sort that add up you gas bill but don't even bother to check that it is sensible ("unable to gain access ... your bill for £73157.93 is an estimate"), or that lie as embedded processors inside washing machines, microwave cookers and the like. If a general purpose computer exists in the home, it is most likely to be used merely as a glorified typewriter.

Natural Tasks

The second application domain for AI is the one related with linking sensing to action in every day tasks. Here, as well as wanting to make an intelligent machine, we want it to use its intelligent to move around in and interact with its environment. We want intelligent robots which can

- Balance on less than three legs
- Avoid lampposts
- Recognize acquaintances
- Grasp tools
- Hold hands, and
- Understand conversations
- Perform commonsense Reasoning

Programming such *natural* tasks (tasks which we have taken for granted since the age of two) turns out to be hard. Indeed tasks which traditionally are thought of demanding the highest intelligence — triple integration or whatever — are relatively straightforward to program — there are maths libraries each with several routines to do just that. Such tasks are already well-formalized, and their domain knowledge is complete.

By contrast, the tasks towards which AI is focussed are difficult because they involve large amounts of knowledge which may be uncertain and rather difficult to represent formally. The reasoning may involve intuitive processes, or at least processes whose underpinning reasoning is not at all obvious.

Formal Tasks

The final domain is that of formal tasks such as

- Logic
- Geometrical reasoning
- Theorem proving
- Game Playing

AI as a driving force

In addition, one should add that AI has consistently probed problems which raise new challenges in systems design, in terms of both hardware and software.

1.1.2 The development of AI

AI first flowered as a separate field of study in the 1950s, but one might look back a hundred years or so to see that the seeds were already planted. It was in the 1840s that Charles Babbage built his analytical engine, of which Lady Ada Lovelace, his patron, said: “The operating mechanism ... might act on other things besides numbers were objects found whose mutual relations could be expressed by those of the abstract science of operations ...”. Babbage was sadly too far ahead of his time to see anything come of his ideas.

There were several fields of study which contributed to the initial growth of AI.

First there was the work of mathematicians, in particular the logicians such as Godel, Church, Post and Turing, who devised formal methods such as propositional and predicate calculus by which facts and ideas expressed in understandable language could be manipulated. These ideas extended into linguistics, where formal grammars were devised. Second, a new area of study call cybernetics brought together several areas where parallels could be drawn between human and machine performance. Thirdly, work was being undertaken in the area of information theory in which Shannon was a prominent figure. Shannon also pioneered the possibility of playing chess by computer, and indeed described much of the background to game playing theory. Fourthly there was increasing sophistication in neural function and modelling. Finally, the crucial catalyst was the development of the electronic computer. The essential advance here had been the invention of the valve in the 1920's, and the first machines (eg the Mark I Harvard relay machine, and the U. Penn's ENIAC) appear in the mid 1940s. By the 1950s machines (eg the UNIVAC) were becoming commercially viable and more widely available.

Perhaps the germination of AI was the publication in 1950 of Alan Turing's article *Computing Machinery and Intelligence* in which he proposed that machines could be programmed to exhibit intelligence and in which he described the famous Turing Test (although, he had been nurturing the idea for some time). The first flowering (and apparently the first use of the term AI) is widely held to be at the 1956 workshop held at Dartmouth College. Its participants, among them Claude Shannon, John McCarthy, Herb Simon, John Newell, Marvin Minsky — were to dominate the subject for the next two an more decades.

As mentioned earlier, in these early days there was considerable over-optimism about the scientific growth of AI, and it was not uncommon for the idea that machines would be more capable than humans in a decade to be expressed. Given the illustrious nature of the key researchers, one can reasonably suppose that the reason for the optimism was not stupidity. Rather, it seems there were a number of significant early successes and rather too few setbacks to indicate just how difficult some of the problems (perception, learning, language understanding) would turn out.

Amongst these early successes were

- The Logic Theorist, one of the first programs for automatic theorem proving (Newell, Shaw and Simon, 1957)
- Generative Grammars (Chomsky)
- Perceptrons (Rosenblatt, 1958)

- Lisp programming language (McCarthy, 1958)
- A Geometry Prove (Gelernter)
- A Masters Level Draughts player (Samuel 1965)
- Dendral (a molecular structure analyser) (Lederberg, Feigenbaum and Djerassi 1965)
- The first version of Macsyma (Engelman, Martin and Moses 1968)
- Object recognition system, which could recognize simple wedge and cube shapes from imagery (Roberts 1965).
- Resolution proofs in logic (Robinson 1965)

The next advances came in the early seventies with the introduction of a major infection diagnosis system, MYCIN, the forerunner of rule-based systems, which began to take off commercially in the early 1980s. The 1980s were years of considerable expansion of research into every aspect of AI, of considerable commercialization. Unfortunately there was considerable over-selling of just one AI product, the so called expert-system — and “1st generation” expert systems at that, systems that only involve shallow knowledge. The skeptic might have asked whether the easy problems and solutions had all been creamed off, leaving the door open for another cycle of scientific disillusionment.

There would be something in this, were it not for the considerable advances which have been made in the last decade in the difficult areas of natural processing. In vision, for example, there now exist stereo and motion systems, that deliver useful 3D descriptions of the surrounding in real time, whereas at the beginning of the 1980s it was not clear whether such processing could be performed at all. There has been enormous growth too research into non-symbolic, or sub-symbolic, processing, the sort carried on in neural networks. At first, it seemed that neural-networks were to suffer the same over-enthusiastic treatment that was given to expert systems, but it now seems that there is a reasonable and reasoned debate about the role and place of such systems vis-a-vis symbolic computation.

1.2 The essential ingredients

Whatever ‘definition’ of AI one prefers, an essential tenet of the subject must be that

- human-like reasoning *is* a form of computation that may be identified, formalized and consequently automated.

Newell and Simon say something rather stronger in their Physical Symbol System hypothesis. A PSS consists of a set of entities called symbols which are physical patterns that can occur as components of another type of entity called an expression or symbol structure. At any instant the system is a collection of symbol structures. Besides these, the system has a library of processes at its disposal which operate on expressions by creation, modification, reproduction and destruction. A PSS is one which contains a time-evolving collection of

symbol structures. Such a system exists in a world of objects wider than just these symbolic structures themselves. The hypothesis itself says: *A physical symbol structures has the necessary and sufficient means for general intelligent action.* There is support for it from across AI's spectrum of problems — for example, from game playing, where one might expect support; to vision, where one might expect there to be non-symbolic processing.

The *need* for symbolic processing for intelligence is contentious, but for our purposes all we need to accept is that much of AI's progress *to date* has been based on symbolic processing. The received experience from the past 40 years can be summed up in four messages:

1. Intelligence needs knowledge, and knowledge needs representation

Knowledge, or facts about the world and problem domain, are essential. In the early days of AI there was an emphasis on general problem solving, but it was found that such solvers only worked when supplied with a large amount of domain-specific knowledge: no longer were they general.

Knowledge, then, is crucial — but it has less endearing characteristics:

- First there is usually a lot of it: indeed, it appears that human expertise in an area comes in chunks of $7 \pm 2 \times 10^4$ items (eg the number of board positions known by chess masters, the number of words in a graduate's vocabulary, etc.).
- Secondly, knowledge is not static. It is forever changing as a consequence of events.
- Thirdly, in its raw state it is neither succinct nor uniform. For example, both words and pixels in the aphorism "one picture is worth a thousand words" portray the same world situation, but each is quite different. Is each word of equal importance? Is each image pixel of equal importance? Of course not. Because of this non-uniformity, we require formalized (and uniformized) internal representations — ones which make the essential distinct from the trivial. Representations are sets of conventions about *how* to describe a thing or class of things. A description uses the convention to describe a particular instance of object. If all knowledge were uniformly pertinent, then the role of representation would be unimportant, but it turns out that the choice of representation can often make the difference between finding a solution to a problem and failing utterly. Good representations make problem solving easier by exposing the essential.

A *representation* has four parts to it:

- Lexical part — this determines what symbols are allowed.
- Structural part — this determines how the symbols can be arranged
- Semantic part — this attaches meanings to the descriptions.
- Procedural part — this gives access functions that allow you to write and read descriptions into the structure, to modify them and to interrogate them.

2. Intra- and inter-representational changes require reasoning processes.

It is perhaps more correct to say that reasoning processes are such those which create intra- and inter-representational changes. It is rare for knowledge of a particular domain to be such that reasoning proceeds inexorably. More often you will be faced with

choices which have to be made without full knowledge of the consequences, in which case you will need to search for possible solutions.

Search is a key process in AI, and is inextricably linked with knowledge. There are several points:

- *Search compensates for lack of knowledge.* When faced with a puzzle we haven't seen before we don't give up: we engage in trial-and-error behaviour, usually until a solution is found. Note that search is often required even where the domain knowledge is complete: we often cannot make a decision without exploring several options until they turn into dead-ends.
- *More knowledge reduces search.* One can easily think of examples where a specific extra piece of knowledge can cut out wasteful search. But another aspect is turning cognitive actions into skills. Eg, by practicing and learning a piece of music.
- *There is a knowledge/search tradeoff (within bounds!)*

For example, a human chess Master rarely searches more than 200 potential moves, but can recognize 50000 board patterns. By contrast, the Hitech chess machine searched 20 million positions, but has only 200 rules.

Again, early versions of expert systems had 10-20 rules and engaged in modest search. Nowadays, expert systems typically use far more rules and little search.

- *Search can be optimal or sub-optimal.*

One of the fundamental result of Simon's research into decision making in organizations was that *bounded rationality requires opportunistic search*. Computational constraints on human thinking lead people to be satisfied with "good enough" solutions. Prior to Simon's work it had been assumed that a person given all the facts would weigh each of them and arrive at the globally optimum solution. On the contrary, when a person is faced with a problem that begins to overload his thinking capacity, he uses a strategy which finds the shortest computational route to a goal, given the constraints of limited memory, bandwidth or time.

3. **AI systems need to learn.** AI systems need methods of acquiring new knowledge autonomously. Learning was in large measure neglected in the early years of AI but its importance is now accepted. There seem to be two different sorts of learning. What is learning? Is it making facts explicit using existing knowledge, or is it discovering something completely new?
4. **We are just scatching.** Perhaps the strongest message is that the scale of the problems we would like to tackle is enormous, but most work has been on small tractable problems. One should not expect problems to scale linearly.

1.2.1 Criteria for intelligence?

When making one of the artificial intelligentia it would be useful to have criterion for success.

Turing devised a test for intelligence. The essence is that in one room you place your computer, in another you place a human being and in a third yourself. You communicate to

both rooms via a VDU which can be connected to either room via a T-switch. You don't know initially which position of the switch corresponds to the machine or the other human. You start dialogues with both, switching you switch whenever you want. If you are unable to determine which position of the switch is the machine, then the system is intelligent. (Actually, Turing wrote about distinguishing between a man pretending to be a woman and a computer imitating a man pretending to be a woman — this extra layer of subterfuge baffles me!) Before the test, one is allowed to tell the machine that it is being tested, so that it can moderate its abilities in certain areas — eg in computing π^{π} to 9 significant figure in under a millisecond! Turing suggested that an intelligent machine would spend a few minutes working out the product of two 6-digit numbers and then given the wrong answer. Note too that Turing does not try to distinguish between a computer and a human. It tries to distinguish between a computer and human *mediated by a computer interface*.

The Turing Test though straightforward is just a bit extreme. Can we say anything about the hallmarks of an AI system? Newell suggested the following characteristics:

- real time operation
- exploitation of vast amounts of knowledge
- robustness to errorful, unknown and unexpected inputs
- the use of symbols and abstractions
- communication using natural language
- learning from the environment, and
- exhibition of adaptive goal-oriented behaviours.

There are only some systems that do some of these some of the time.

1.3 Representation and search mixed: some examples

A difficulty studying AI techniques is that each problem appears to present a different cocktail of knowledge representation issues and search issues, and it is therefore difficult to see where the power of a procedure lies.

In general however it is the knowledge and its representation that is key, and having chosen a good representation following sensible procedures helps. In the first few lectures we will concentrate more on rather more general procedures, called “weak methods” which form the backbone of many of the knowledge-rich “strong” techniques.

1.3.1 The power of representation: example

First though, to illustrate the power of representation, consider the problem sketched in Figure ?? . You wish to tile a patio with 2×1 slabs. The patio is 8×8 , but has two 1×1 drains at opposite corners which should not be covered. Is it possible to tile the patio without cutting a tile?

One approach would be to try all possible arrangements of tile. Let us *estimate* the size of the search space. Neglecting edge effects, we can cover a 1×1 square in two ways — the longer axis of the tile can lie NS or WE. (Note there are not four ways. The other two ways are equivalent to tiling neighbouring 1×1 squares.) We have to place 31 tiles to do it perfectly, or at least 32 to do it imperfectly. Ignoring edge effects, the search space would have a size around $4.31!$ or 10^{34} . This is an upper bound — but whatever, we are talking big.

Now think about representation. Is there a way of describing the patio that makes the answer obvious?

Imagine the patio as a chess board, with alternate black and white squares. The two drains are at diagonal corners, and so will be coloured the same, say white. Thus we have 62 squares to tile, 32 black and 30 white. But a 2×1 tile must always cover one black and one white square. So the best we can do is to lay 30 tiles, and then we have 2 black squares over, which we cannot tile. Thus the tiling can not be done perfectly.

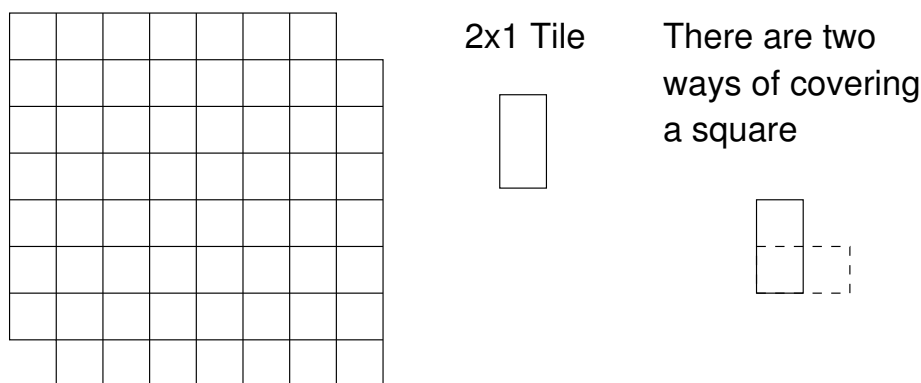


Figure 1.1: A tiling problem indicating how representation

1.3.2 A more generalizable solution method: example

The tiling example is possibly extreme, in that with the right representation the problem all but solves itself. The representation is a *strong* one — that is, it uses highly specific domain knowledge and is not generalizable. By contrast, let us consider a so-called *weak* method, where the representation and search method are quite general. Again you should note the importance of representation.

Lord Lovely had arranged a picnic for his daughter, Lavinia, and two of the local talent, viz, one Victor Vile and one Hon. Simon Simple, all to be chaperoned by Dr Gertrude Gooseberry. Now Vile has designs on Ms Lovely, while she fancies Simple, who himself is oblivious to everything except the propriety of his dress. All went well until the party, arriving at the bush-lined river bank of the River Cherwell, found only a two-man person punt to take them across to the meadow beyond, the problem being that only Gertrude could punt, leaving space for only one passenger. But if left alone together, Vile would make advances to Lavinia, while if Lavinia is left with Simple, goodness knows what might occur. Golly, what is Gooseberry to do?

The English description grips the imagination (well, slightly!) but most of it is irrelevant detail. Do we care that it is a meadow and not a turnip field, or that it is the Cherwell and not the Ouse? We do not.

Gertrude cuts to the heart of the problem. She represents start and goal states as GLSV= and =GLSV, where = represents the river. She notes that the ordering of people on the same bank is irrelevant and quickly computes that there are $1+4+6+4+1 = 16$ states, 4 of which are forbidden, and 2 of which are redundant. She notes that any change in state must involve her switching bank, and that when punting forward there should always be two people in the boat, but on return there may be one or two.

GLSV	≈	
SV	≈	GL
GSV	≈	L
V	≈	GLS
GLV	≈	S
L	≈	GSV
GL	≈	SV
	≈	GLSV

Figure 1.2: The Chaperone's plan. ≈ denotes the river

What Gertrude Gooseberry did was to set the problem up a journey through the problem's state space. The state space (all 16 states) is given in figure ??.

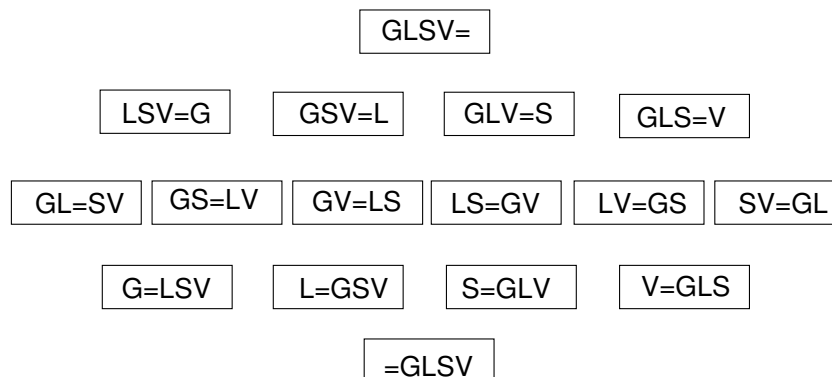


Figure 1.3: The state space for the chaperone's problem.

Finding a solution involved:

- Consideration of the state-space containing all possible configurations of objects. (We shall see later that this need not be explicit.)
- Specifying initial and goal states. (Here we have one of each, but other problems can have multiple initial and goal states.)
- Making up rules describing operations (Eg G always changes state, and forward journeys always involve G + ANOther.)

- Matching rules that can be applied to a state (this will be more obvious in the next example).
- Applying the rules in some order in the search for a goal.

In general terms, the approach required

- specific problem-dependent rules, with the
- general techniques of search.

1.3.3 Semantic net representation

If we linked related states in the state space of the Chaperone's problem example of a *semantic net*, one of the more common representations in AI. The net consists of nodes denoting objects, links denoting relationships between objects, and labelled links denoting specific relationships. Structurally, nodes are connected by links.

Winston (1992) (p21) gives a diagram of the some of the family of representations that fall under the description.

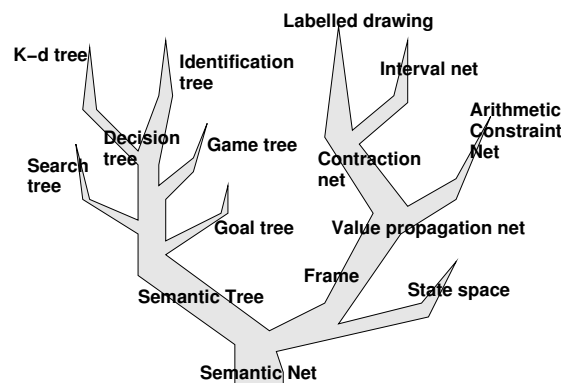


Figure 1.4: Part of the family tree of representations (from Winston).

Another member of the family is the search tree, which explores paths from the start to the final state as they are built up serially.

The four parts of the semantic net representation consist of

1. Lexical part — (i) Nodes, (ii) Links and (iii) Link labels.
2. Structural part — The Links link nodes.
3. Procedural part — Allow you to create nodes, create links, and to traverse the net.
4. Semantic part — depends on the application

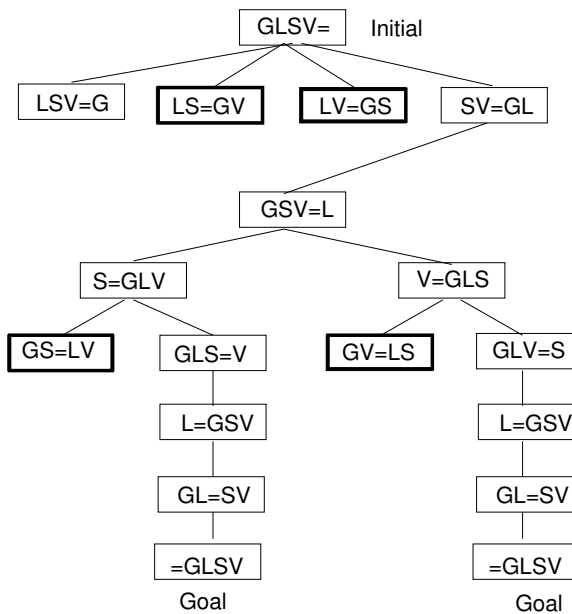


Figure 1.5: Chaperone's solution as a search tree.

1.3.4 The Describe and Match Paradigm

A classic IQ test is that of geometrical analogy, as in figure ??, where we ask *A is to B as C is to which X?* Again we will see that the problem is easily solved by thinking about the right

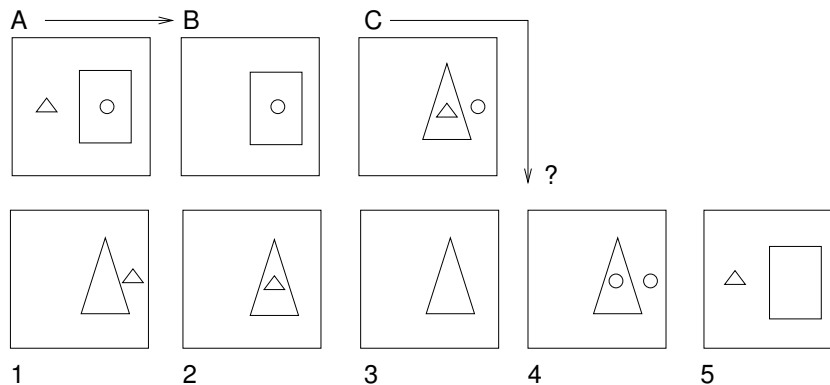


Figure 1.6: A geometrical analogy problem.

representation. In addition the method, described in Evans' (1964) program ANALOGY, uses one of AI's classic paradigm's — *describe and match*.

Looking at the problem however indicates that it is not quite a simple as describing a person and matching, say, to a library of photographs, because we are not matching object to object. Evans' recipe was

1. Describe the rules that *transform* A to B and C to each of (1,2,3,4,5).
2. Evaluate a matching similarity between the *transformation* A to B and those of C to (1,2,3,4,5).

3. Select the most similar.

So you don't describe and match figures per se, but rather describe and match transformation rules.

Creating a net descriptions of the figure pairs proceeds in four parts.

- Label subshapes
- Describe figure A
- Describe figure B
- Describe linking subshape changes

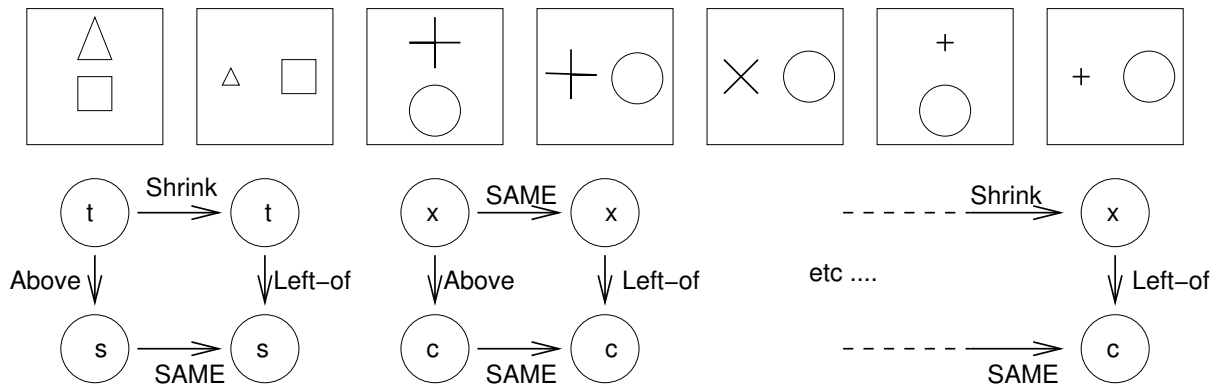


Figure 1.7: A semantic net for the ANALOGY problem.

If the descriptions match perfectly then there is obviously no difficulty choosing the best. This may not always be possible, and then we need to devise a measure of similarity. One similarity measure would be

$$\begin{aligned} \text{Similarity} = & (\text{Number of links labels shared in both nets}) \\ & - \alpha(\text{Number in AB net not shared}) \\ & - \beta(\text{Number in CX net not shared}) . \end{aligned}$$

Refinements would be:

- (i) to weight the links describing spatial relationships (above, inside, left-of) more heavily than subshape changes (shrinking, rotating) etc,
- (ii) to weight different subshape transformations differently

For example, one might try the following ordering:

Most similar transformation	Unchanged
	magnified
	rotated
	rotated and magnified
↓	reflected
	reflected and magnified
	reflected and rotated
	reflected, magnified and rotated
Least similar	Deleted

Such similarity ordering can also help in reducing ambiguity in forming descriptions, as in figure ?? where it is uncertain whether the outer has been deleted or the inner deleted and the outer shrunk. The ordering would prefer the former description.

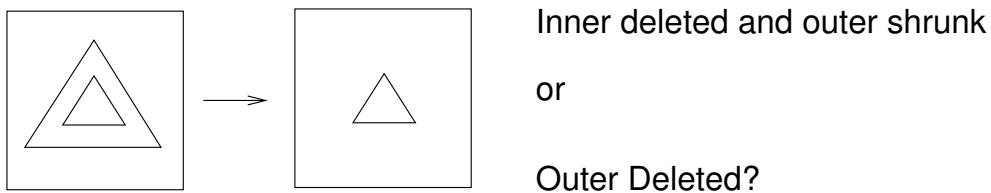


Figure 1.8:

1.3.5 Knowledge as rules: example

How do we cause movement from one state to another in the Chaperone's problem? One way we have already explored would be to set up an ordered search which explored possibilities thoroughly in turn. Another commonly used method is to write a few if-then rules describing what to do in various circumstances. We could describe playing chess in the same sort of way, or composing music to the rules of classical harmony or to the unwritten rule of jazz. We then apply the rules in some order to seeking out the the goal state. This sort of system is formally called a production system, and it forms a useful structure.

The Water Cans Problem. You are given two cans, one holding 3 pints, the other 4 pints. Neither has level markers. How can you exactly half fill the 4 pint can?

The state space is described by the pair (m, n) , the volumes in the 4pt and 3pt can respectively. The initial state is $(0, 0)$, and the final state is $(2, *)$ where $*$ is a "don't care" wildcard. Here then we already see an example of multiple goal states.

A set of production rules might be:

No	Applicability	After	Comment
1	$(m, n : m < 4)$	$\rightarrow (4, n)$	Fill 4 pint can
2	$(m, n : n < 3)$	$\rightarrow (m, 3)$	Fill 3 pint can
3	$(m, n : m > 0)$	$\rightarrow (m - d, n)$	Pour volume d out of 4pt
4	$(m, n : n > 0)$	$\rightarrow (m, n - d)$	Pour volume d out of 3pt
5	$(m, n : m > 0)$	$\rightarrow (0, n)$	Drain 4pt
6	$(m, n : n > 0)$	$\rightarrow (m, 0)$	Drain 3pt
7	$(m, n : m + n \geq 4 \wedge n > 0)$	$\rightarrow (4, m + n - 4)$	From 3 into 4 until full
8	$(m, n : m + n \geq 3 \wedge m > 0)$	$\rightarrow (m + n - 3, 3)$	From 4 into 3 until full
9	$(m, n : m + n \leq 4 \wedge n > 0)$	$\rightarrow (m + n, 0)$	Empty 3 into 4
10	$(m, n : m + n \leq 3 \wedge m > 0)$	$\rightarrow (0, m + n)$	Empty 4 into 3

So the production system comprises:

- a set of rules, each rule has a LHS that determines its applicability and a RHS that says what action is performed
- a database containing pertinent knowledge (here (m, n))
- A *matching procedure* that determines which rules can be applied to the current state
- A search control strategy that decides which rule to follow if multiple rules apply to the current state. The search control strategy must also stop when the goal is found.

One possible solution is

m	n	Then Rule	Comment
0	0	1	Start; Fill 4
4	0	8	From 4 to 3 until full
1	3	6	Drain 3
1	0	10	Empty 4 into 3
0	1	1	Fill 4
4	1	8	From 4 to 3 until full
2	2		Goal

Look again at rules 3 and 4. The fact that we have 3 and 4pt cans tells *us* that the smallest quantum of water we can measure exactly is 1pt. So what place for some arbitrary quantity d ? Of course there is none: those rules will never be used by our system in reaching the goal, and so they will always result in dead-ends in our search. The question is, do we remove them and face the accusation of “fixing” the problem or do we let the system *tell us* they are redundant by never using them?

1.4 Summary of lecture 1

We have indicated that two important ingredients in AI are search and knowledge representation. The two appear, often inextricably entwined, in one guise or another in every AI problem. Whilst search is important as a *technique*, it is not the key issue. This is the adequacy or otherwise of our representation and exploitation of knowledge. The third important ingredient in AI is how to acquire new knowledge.