Computing

Artificial Intelligence

[INTERMEDIATE 2]

Learning
+Teaching
Scotland

The Scottish Qualifications Authority regularly reviews the arrangements for National Qualifications. Users of all NQ support materials, whether published by LT Scotland or others, are reminded that it is their responsibility to check that the support materials correspond to the requirements of the current arrangements.

Acknowledgement
Learning and Teaching Scotland gratefully acknowledge this contribution to the National Qualifications support programme for Computing.

First published 2004

ISBN 1 84399 024 5

CONTENTS

INTRODUCTION

This unit is designed to provide support material for the teaching of Artificial Intelligence at Intermediate 2 level.

It may be studied as a stand-alone unit or combined with other units as part of the Computing course at Intermediate 2 level. It is also possible for the unit to contribute towards a Scottish Group Award.

There are two outcomes in the unit which require candidates to demonstrate:

1. Knowledge and understanding of a range of facts, ideas and terminology related to the principles, features and purpose of artificial intelligence.

2. Practical skills in the context of Artificial Intelligence using contemporary software and hardware.

Outcome 1 is covered throughout the unit. Satisfactory performance in this outcome will be achieved when the candidate has passed the unit assessment for this outcome.

The practical skills required are:

1. Construction of a knowledge base of facts and simple rules from a semantic net.

2. Creation of simple queries to elicit information from a knowledge base.

3. Testing a knowledge base.

4. Consulting a simple expert system.

Outcome 2 requires the candidate to exhibit each of the skills listed above. Completion of the knowledge representation section of this unit will provide sufficient evidence as identified in the arrangements document to cover practical skills 1 to 3. The observational checklist can be completed whilst candidates work through each of the tasks.

Notes for tutors

Throughout the unit references are made to resources on the Internet. All links were active at the time of writing. However, all Internet links are volatile and tutors are advised to check that they are still live before teaching the unit. The support materials are not dependent on the Internet links and can be used in isolation.

A web-based source has been identified to allow students to experience a conversation with Eliza. Tutors may prefer to download a version of Eliza and install it on a network or stand-alone machine. There are several free versions available on the Internet.

Although the topics of artificial neural systems, handwriting recognition, expert systems, etc. have been discretely identified in this unit, candidates should be aware that they are not treated in isolation, e.g. handwriting recognition systems use fuzzy logic which is a feature of Neural Networks. Fuzzy logic is outside the scope of this course and is covered in Higher Computing. Also vision systems may employ some form of Neural System to interpret images.

The knowledge base section is based on the use of Edinburgh Syntax. This is a standard for representing knowledge in procedural languages and is not platform specific. If the centre is using some other form of prolog (e.g. LPAProlog), the notes can easily be edited.

The technique used for the trace is an attempt to provide a standard method for students when dealing with any knowledge base. Although the content statement indicates that a manual trace should be performed using one rule and one level, the support notes go slightly beyond this limit to provide candidates with a deeper understanding.

Candidates can be given the practical task which will cover skills 1 to 3. In addition, skills 2 and 3 may be covered while working through the rest of the unit.

Skill 4: consulting an expert system cannot be covered in the support notes as it is not possible to supply an expert system. The centre must provide a suitable (simple) expert system for candidates to consult. The URLs below have some free expert systems listed.

http://tiger.coe.missouri.edu/~jonassen/courses/mindtool/ExpertExamples.html
http://cnrit.tamu.edu/rsg/exsel/ (This is an online agricultural KB.)

Alternatively, the ‘whale’ expert system mentioned in the notes may be used.

The practical tasks included in this pack are not prescriptive and centres are free to use their own task to produce evidence of satisfactory completion of Outcome 2.

A folder of freeware resources has been included on the DVD/CD. It contains sample programs which help to exemplify the notes. Acknowledgement should be made to the authors of the programs for allowing the free distribution of their software.

SECTION 1

What is intelligence?

Before we can attempt to look at artificial intelligence (AI) we really need to consider what human intelligence is. It may seem a strange question because everyone knows what intelligence is, or at least we think we know! I'm sure you can think of an example of intelligence (such as solving a crossword) or can think of a person whom you consider to be intelligent (perhaps even yourself!), but when it comes to providing one all-encompassing definition of what intelligence is, we are stumped.

Here are a few of the definitions which have surfaced through the ages:

- 'The ability to carry out abstract thinking.' (Terman, 1921)
- 'The capacity for knowledge, and knowledge possessed.'(Henmon, 1921)
- 'Intelligence is a general factor that runs through all types of performance.' (unknown)
- 'A global concept that involves an individual's ability to act purposefully, think rationally, and deal effectively with the environment.' (Wechsler, 1958)
- 'The capacity to learn or to profit by experience.'(Dearborn, 1921)
- 'Intelligent activity consists of grasping the essentials in a given situation and responding appropriately to them.' (unknown)
- 'Intelligence is a hypothetical idea which we have defined as being reflected by certain types of behaviour.' (unknown)
- 'The capacity to acquire capacity.' (Woodrow, 1921)
- 'Intelligence is what is measured by intelligence tests.' (Boring, 1923)
- 'Intelligence is that faculty of mind by which order is perceived in a situation previously considered disordered.' (R W Young, cited in Kurzweil, 1999)

Some of these definitions of intelligence are trite, e.g. 'Intelligence is what is measured by intelligence tests.' You don't have to be intelligent to figure that one out! Some of the other definitions given above aren't much better and these people were regarded as experts!

Here are some more down-to-earth examples of what we recognise as intelligence:[1]

1. to respond to situations very flexibly;
   Driving a car and someone walks out in front

2. to take advantage of fortuitous circumstances;
   While waiting in a supermarket queue a teller returns from lunch, and you move fast!

3. to make sense out of ambiguous and contradictory messages;
   'The cat sat on the fat' is unclear but you know where the error is

4. to recognise the relative importance of different elements of a situation;
   You have English homework due in tomorrow and a NAB test in Maths. Which one do you work on?

5. to find similarities between situations despite differences which may separate them;
   Changing the tyre on your own car follows the same principle as doing it on another

6. to create new concepts by taking old concepts and putting them together in new ways;
   You are familiar with the base 10 counting system…after studying Computer Systems you can apply your knowledge to base 2 counting systems

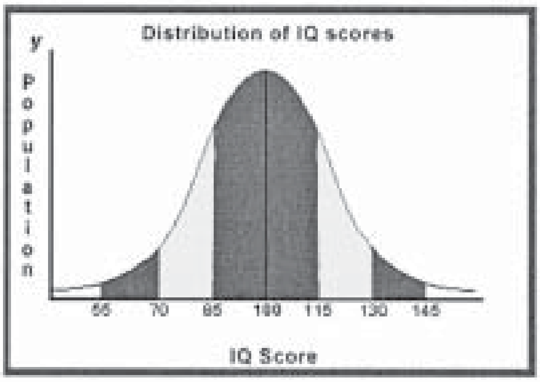
7. to come up with ideas which are novel;
   Invent a joke or write a poem

Without a proper understanding of what intelligence is we have some problems when it comes to measuring it. After all, if we don't know what it is then how can we measure it? Surprisingly enough this has *not* proved a major obstacle to scientists.

[1] Richard Hofstadter, *Godel, Escher, Bach*, Vintage, 1980

IQ (intelligence quotient) tests

The basic idea behind the IQ test is that there exists something called general intelligence which can be quantified, at least relatively.
If an individual takes a properly designed collection of tests, a single number can be generated representing that person's 'intelligence quotient' or IQ. This number is normalised so that the average member of the population has an IQ of 100. The distribution of the population around the mean forms the so-called bell curve. Roughly 68% of the population have an IQ between 85 and 115.



The website http://home8.swipnet.se/~w-80790/Index.htm has some interesting information about the IQ test which you can check out if you wish to find out more.

Grade	IQ range	% of population
Genius	>144	0.13%
Gifted	130–144	2.14%
Above average	115–129	13.59%
Higher average	100–114	34.13%
Lower average	85–99	34.13%
Below average	70–84	13.59%
Borderline low	55–69	2.14%
Low	<55	0.13%

Put simply, your IQ is calculated based on the answers given to a host of 'intelligence testing' questions and your score is compared to that of the general population.

Critics of IQ tests believe that they are too restrictive and that it is not possible to measure intelligence with a single number.

One thing is clear; we cannot agree on a single definition of intelligence. Perhaps the reason is that intelligence itself is so diverse that whenever we think of a new definition there always seems to be some form of intelligence which is not included.

Psychologists have struggled and argued over this for years. One approach is to examine behaviour which we consider to be intelligent and restrict our examination to that type of behaviour. This is called the **behaviorist** approach.

An example of the behaviorist approach in common language might be: ‘If it walks like a duck and it quacks like a duck, then it’s a duck.’ In terms of intelligence: ‘If it behaves in an intelligent manner then we’ll call it intelligent.’

This approach avoids the problem inherent in trying to define the meaning of intelligence.

Interesting facts about your brain and intelligence

- Your brain has about 100 billion neurons. A typical brain cell has from 1,000 to 10,000 connections to other brain cells.

- Studies have shown that children who are breast fed display an IQ up to 10 points higher by the age of 3.

- Your brain is full of nerve cells, but it has no pain receptors. Doctors can operate on your brain while you’re awake and you won’t feel a thing.

- In 1984 the political scientist James Flynn reported that Americans had gained about 13.8 IQ points in forty-six years. If people taking an IQ test today were scored in the same way as people fifty years ago, then 90% of them would be classified in the genius level.

Intelligent behaviour

We will restrict our behaviorist view of intelligence to the following areas.

- the ability to communicate
- the ability to retain knowledge
- the ability to solve problems

The ability to communicate:
This means all sorts of communication, e.g. speech, the written word and visual communication, but the main factor is that the person can express their ideas in a manner which can be understood by others. An example might be the ability to take part in a debate putting forward a point of view and giving good reasons for that point of view. Another might be writing a book which captures the imagination of its readers.

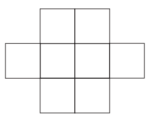
The ability to retain knowledge:
You will have come across this whenever you have had to study for a test. The more raw knowledge you possess the larger the information base which you can draw upon when trying to solve a problem.

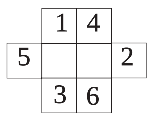
The ability to solve problems:
You do this all the time in your mathematics class when the tutor gives out a whole stack of equations and leaves you to work out the value of X or Y. In fact being good at mathematics is seen as a ‘sign of intelligence’ but we have had enough definitions so we’re not going there!

Try these interesting problems:

- On a cold winter night, in the middle on nowhere, the door lock on your car is frozen; can you think of a way to unfreeze the lock?

- In the grid below there are eight boxes and your task is to place the digits 1–8 in each box but consecutive numbers must not appear in boxes which are touching (even at the corners). Do this within 4 minutes and you are quite smart!





No good, cannot place the 7

- Find the missing letters in the sentence in the box below. Each * is one letter and they are in the same order each time; i.e. the first is one word, the next is two words and the last is two words.

> The ******* surgeon was ***/**** to perform the operation as there was **/*****.

In general we could say that a person (or machine) displays intelligence if it can:

(a) communicate
(b) retain knowledge
(c) solve problems.

Artificial intelligence

Our perceptions of artificial intelligence tend to be clouded by science fiction movies in which machines ‘act’ like humans. However, there is some value in the fact that we instinctively compare the intelligence of a machine to the intelligence of a human because this is precisely the approach which is taken in the scientific community. Here are a few definitions of **artificial** intelligence:

- the study of how to build and/or program computers to enable them to do the sorts of things that minds can do
- making computers do things that would require intelligence if done by people
- the development of computers whose observable performance has features which in humans we would attribute to mental processes
- the science of intelligence in general
- the intellectual core of cognitive science.

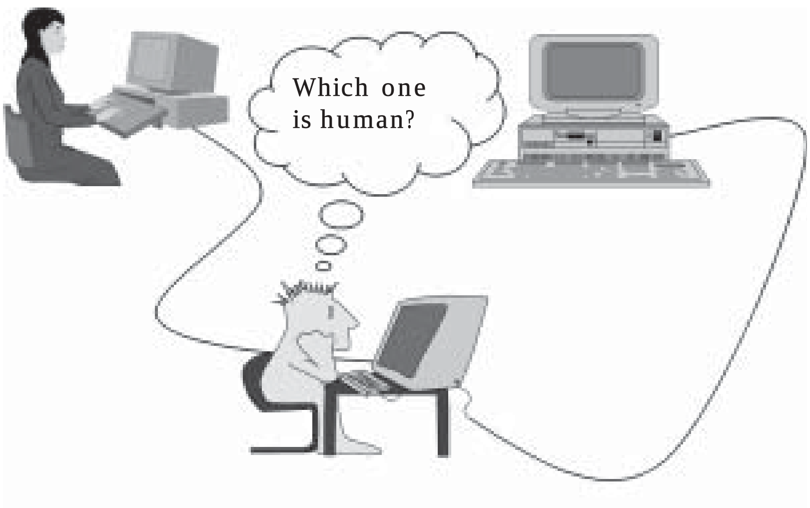
Note that most of these definitions make a direct comparison with human intelligence.

We have discussed some attempts to define or measure intelligence. However interesting these are, they do not provide much guidance to people attempting to construct and verify machine intelligence.

Complete Exercise 1 Questions 1–6 in Section 5

The Turing Test

The first person to discuss criteria for machine intelligence was the British computer science pioneer, Alan Turing. He wished to sidestep the whole issue of defining intelligence, which he considered futile, and to do so he invented what he called the 'imitation game', which today we refer to as the Turing Test.



The interrogator is connected to both a computer and a human and she asks a series of questions. As the reply appears on her screen, she is unaware of whether the computer or the human is answering. If the interrogator cannot distinguish between the human and the computer, the computer has passed the Turing Test.

Put more formally:
The interrogator is connected to one person and one machine via a terminal, therefore can't see her counterparts. Her task is to find out which of the two candidates is the machine, and which is the human only by asking them questions. If the machine can 'fool' the interrogator, it is intelligent.

In the last fifty years, the Turing Test has been the target of several types of criticism and has been at the heart of many discussions about AI, philosophy and cognitive science. One of the goals of AI programmers is to write a program that can pass the Turing Test; to date none have been successful. An added incentive is the 'Loebner Prize', a jackpot of

$100,000 up for grabs to the first person to write a program that passes the Turing Test. There is also an annual award for the best effort which attracts entries from all around the globe. For more detail check out the Loebner website and find out about previous winners of the best-effort prize.

http://www.loebner.net/Prizef/loebner-prize.html

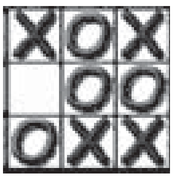
Why is the Turing Test important to AI?
The Turing Test set a goal or challenge for programmers in the field of AI. The first person to create a program which passes the Turing Test will enter the history books and have a highly commercially viable product.

Game playing

Early attempts to create artificial intelligence were based on writing programs which could play games. It was felt that if a program could play a game and possibly even beat a human, it would show signs of intelligence. Some of the games included OXO, draughts and even chess. These games (particularly chess) were viewed as requiring a degree of logic, reasoning and imagination.

Many of the early game-playing programs achieved their goal inasmuch as they were capable of playing against a human opponent and sometimes winning. One draughts-playing program could not only play but learn from previous games to improve on its performance.

Noughts and crosses *(tic-tac-toe)*



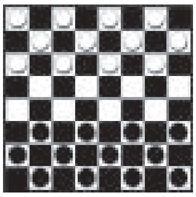
The first software program designed to play noughts and crosses was written by A S Douglas in 1949. The EDSAC machine which ran the program was the first true programable computer as we would understand it today.

There are 255,168 possible games of noughts and crosses and as many as 2,201 different non-winning board positions in the first four moves. Computers are ideal for handling this type of number-crunching task and by 1959 a version of noughts and crosses called MENACE was capable of learning and regularly beating a human opponent.

The hyperlink below has a visual basic version of MENACE which can be downloaded.

http://www.adit.co.uk/html/menace_simulation.html

Draughts *(checkers)*



Chinook is a draughts program which uses search techniques and also has a database to allow it to play a perfect end game. In 1992 Chinook won the US Open and subsequently challenged for the world championship. Dr Marion Tinsley had been the world champion for over forty years. In that time she only lost three games. Playing Chinook she lost her fourth and fifth game but ultimately won the match by 21.5 points to Chinook's 18.5 points. In August 1994 there was a re-match but it ended prematurely when Dr Tinsley had to withdraw for health reasons. As a result of this Chinook become the official world champion. Schaeffer (1996, p.447) claimed that Chinook was rated at 2814. The best human players are rated at 2632 and 2625. Chinook did not include any learning mechanisms. This was the first time in history that a computer had won a world championship. Fancy your chances against Chinook? Try the link below:

http://www.cs.ualberta.ca/~chinook/

More recently Kumar (2000) developed a checkers program that 'learnt' how to play a good game of checkers. The program started knowing just the rules of the game so that it could make legal moves. The program was allowed to evolve by creating a **population** of games that competed against one another, with the best games **surviving** and being adapted in some way before competing again. The adaptation was done using a neural network with the weights on the synapses being changed by an **evolutionary strategy**. The best program was allowed to compete against a commercial version of checkers and it beat it 6-0.

Chess

Skill at chess has always been considered a sign of intelligence because of the need to plan ahead and devise strategies. Early programs used the power of the computer to work out the permutations of every possible move.

Chess has 10^{120} unique games (with an average of 40 moves – the average length of a master game). Working at 200 million positions per second, **Deep Blue** would require 10^{100} years to evaluate all possible games. To put this into some sort of perspective, the universe is only about 10^{10} years old and 10^{120} is larger than the number of atoms in the universe.
New strategies were developed and in 1997 'Deep Blue' beat the best chess player in history, Garry Kasparov. This was seen at the time as the great challenge of man against machine and raised the whole issue of what intelligence really means. Even the AI experts at the time were surprised as they did not expect a computer to beat a human at chess for at least another ten years.

The chess grandmaster does not calculate every permutation on the board but instead relies on his knowledge and experience to generate a few promising moves for each game situation (irrelevant moves are never considered). In contrast, when selecting the best move, the game-playing program exploits brute-force computational speed to explore as many alternative moves and consequences as possible. As the computational speed of modern computers increases, the contest of knowledge vs speed is tilting more and more in the computer's favour, accounting for recent triumphs such as Deep Blue's win over Kasparov. Deep Blue can perform over 200,000,000 move sequences per second.

When Deep Blue defeated Kasparov this was viewed as a milestone in the field of artificial intelligence, but some have claimed that Deep Blue is simply a machine which is good at playing chess; it has no conscience, unlike humans, in fact it can't even recognise a chess piece let alone move one.

The same techniques used to program Deep Blue have been applied to other games and also proved to be successful.

All of the game-playing programs had three characteristics in common: they were played in a **restricted environment**; they had a clearly defined **set of rules**, and the **criterion for success** was straightforward (i.e. someone wins). This made programming games a lot easier but the question still arises as to whether it is really displaying intelligence.

There are a number of contemporary learning-type games on the Internet. Some are part of research projects and others are just for fun. The first one listed below is called 1001 questions; the researchers are trying to gather knowledge on just about anything.

http://teach-computers.org/learner.html	MIT research trying to give a computer common sense
http://y.20q.net:8095/btest	You think of an object and the computer will try to guess within twenty questions
http://www.press12.freeserve.co.uk/reaper.html	AI war games for insects?
http://www-2.cs.cmu.edu/~trb/soccer/	Play ASCII football

Our behaviourist approach to intelligence identified three criteria for intelligent behaviour. How well do they apply to Deep Blue?

The ability to communicate: Deep Blue can communicate moves and accept the moves of an opponent. So this criterion seems to be met.

The ability to retain knowledge: Before playing Kasparov every one of the world champion's tournament games was input to Deep Blue so there is definitely retention of knowledge.

The ability to solve problems: This is Deep Blue's strongest asset.

Does this mean that Deep Blue really is intelligent? Or perhaps our definition is too limited?

Complete Exercise 1 Questions 7–13 in Section 5

Natural language processing (NLP)

The goal of natural language processing is to reach the position where a human can talk to a computer, be understood and get a response in the same way that they would from another human. We will examine some of the attempts to create programs that could hold a conversation with humans.

Language processing can be divided into two tasks:

1. **Processing spoken language**: involves taking words and sentences inputted via a microphone and using the knowledge of grammar and structure to interpret the input. Put simply, the user 'converses' with the machine.

2. **Processing written text**: handwriting is inputted via a stylus pen (or scanned), the software attempts to make sense of the writing and convert it to text as if typed.

Practical applications of NLP

Many companies would make a lot of money if they could use computer programs that understood text or speech. Just imagine if a computer could be used for:

- answering the phone and directing a caller to the right person
- understanding the text on a web page to decide who it might be of interest to
- translating a daily newspaper from German to English (an attempt is made to do this already)
- understanding text in journals/books and building an expert system based on that understanding
- giving people advice on almost any subject.

The attraction of such possible systems drives commercial research into NLP. The complexity of language and its use holds back success.

We will now look at some historical and modern attempts to create programs which can communicate with humans.

Communication between man and machine

Eliza

This was one of the first attempts to write a program which could hold a conversation with a human. It was written by Joseph Weizenbaum in 1966 and, amazingly, is still popular today. The intention was to create a 'computer psychologist' that could respond to people's problems by talking them through.

There are many Eliza spin-offs with different personalities, mainly because people quite enjoy the novelty of having a conversation with a computer.

Typical conversation with Eliza
[Eliza's part of the conversation is in capitals]

Men are all alike.

IN WHAT WAY?

They're always bugging us about something or other.

CAN YOU THINK OF A SPECIFIC EXAMPLE?

Well, my boyfriend made me come here.

YOUR BOYFRIEND MADE YOU COME HERE?

He says I'm depressed much of the time.

I AM SORRY TO HEAR YOU ARE DEPRESSED

It's true. I am unhappy.

DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY?

I need some help, that much seems certain.

WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP?

Perhaps I could learn to get along with my mother.

TELL ME MORE ABOUT YOUR FAMILY

My mother takes care of me.

WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU?

My father.

YOUR FATHER

You are like my father in some ways.

WHAT RESEMBLANCE DO YOU SEE?

You are not very aggressive but I think you don't want me to notice that.

WHAT MAKES YOU THINK I AM NOT VERY AGGRESSIVE?

You don't argue with me.

WHY DO YOU THINK I DON'T ARGUE WITH YOU?

You are afraid of me.

DOES IT PLEASE YOU TO BELIEVE I AM AFRAID OF YOU?

My father is afraid of everybody.

WHAT ELSE COMES TO YOUR MIND WHEN YOU THINK OF YOUR FATHER?

Bullies.

DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE?

Although this might seem like a fairly normal conversation, on further inspection you can see that Eliza is simply picking out keywords from the 'patient' and inserting them into a response. Bear in mind that keywords to a psychiatrist include *father*, *mother*, *dream*, *hate*, *love*.

In some versions of the program, when it can't find a suitable keyword, Eliza falls back to the response 'tell me more'. This open approach was responsible for the popularity of Eliza in that it could provide a response to most questions, but it was also the main drawback because it showed no understanding whatsoever and could easily be led astray in a conversation.

Try a web-based version of Eliza at http://www-ai.ijs.si/eliza/eliza.html

Or try a conversation with a different version of Eliza at the site on the web at the URL below:

http://www.cybermecha.com/

Chatterbots

There has been a resurgence of interest in programs which can hold a conversation with humans partly due to the expansion of the Internet. Such programs are now commonly referred to as *chatterbots* or just *bots*.

A chatterbot is a computer program for simulating conversation between a human and a machine. You input a question or statement of any kind, and the chatterbot replies, just as a person would (using its own version of logic!). Chatterbots try to create the illusion that an authentic exchange is taking place between two thinking, living entities. Sometimes you have to pinch yourself to remember that you are not talking to a real person. At other times, it's all too obvious.

Here are a few examples of chatterbots:

Alex	http://jurist.law.pitt.edu/alex.htm	Gives legal advice – it can talk!
jabberwacky	http://www.jabberwacky.com/	This one can talk to itself
Elizabeth	http://www.etext.leeds.ac.uk/elizabeth/	Update of Eliza
Arty	http://www.ww7.com/looney-bin/Arty_Fishal/index.html	This one is insane
Ebot	http://www.elbot.com/	A grump bot
Interface bot	http://www.maybot.com/	Professional bots
John Lennon	http://triumphpc.com/johnlennon/	John Lennon in a bot!
Dr. Werner Webowitz's	http://www.parnasse.com/drwww.shtml	With a name like that he had to be a shrink!
Commercial	http://www.conversive.com/	Have a good chat with this one

These are some of the 'faces' given to bots on the Internet.

Some people say that chatterbots have struggled to go beyond the level of Eliza. But if we consider the question in terms of developing applications, rather than robotic human beings, we can see that chatterbots increasingly have a role to play in humanising the Internet. With the explosion of the web, more and more chatterbots are making their appearance online.

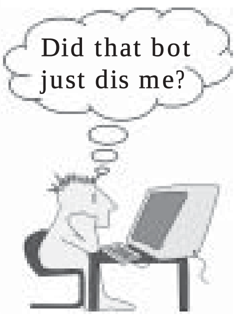
A.L.I.C.E.

This bot is worth a special mention. Read the extract from an article below:

> ## **ALICE** wins Loebner Prize again
>
> 13 October 2001 – London UK
>
> Today A.L.I.C.E. was awarded the bronze Loebner Prize medal for the second year in a row, at the annual competition based on the Turing Test, this year held at the London Science Museum. Once again, A.L.I.C.E. was ranked 'most human' computer. In fact, one contest judge ranked A.L.I.C.E. higher than one of the human 'confederates'.
>
> The Loebner Prize is an annual competition that challenges computer programs to convince human judges that the computer programs are actually people. The contest was established in 1990 by Dr Hugh Loebner, a renowned philanthropist.

This program seems to contain a large database on a whole range of subjects. The responses are quite amusing and it can often pick up the user's slang or modern vernacular. See the example below.

You can try A.L.I.C.E. online at the URL http://www.pandorabots.com/pandora/talk?botid=f5d922d97e345aa1



A conversation with A.L.I.C.E.

Here is a full transcript of a conversation with A.L.I.C.E.

[User] I'm interested in Hope College.
[ALICE] What would you like to know about Hope College?
[User] I want to know about computer science.
[ALICE] CS, great! I'm obviously very interested in the subject myself.
[User] Who teaches it?
[ALICE] Drs. Dershem, Jipping and McFall teach computer science.
[User] Who is Dr. Dershem?
[ALICE] Dr. Dershem is a professor of computer science, and chair of the department. You can find out more about him on his homepage: http://www.cs.hope.edu/~dershem
[User] What sort of courses are offered?
[ALICE] Courses are displayed in the window I just opened for you.
[User] What sort of degrees are offered?
[ALICE] The department offers both a Bachelor of Arts in computer science (30 credit hours) and a Bachelor of Science in computer science (36 credit hours). The department also offers a minor in computer science (minimum 18 credit hours).
[User] What's so special about the CS program?
[ALICE] Hope College is one of nineteen colleges, universities, and research labs in the country to be part of the National Science Foundation Research Experience for Undergraduates program...
[User] Who is president of the college?
[ALICE] Dr. James Bultman is president of Hope College.
[User] Do you know who David Myers is?
[ALICE] I don't know who David Myers is. You might search the Hope College homepage.
[User] Who invented Linux?
[ALICE] Linus Torvalds, along with many open source developers, and the Free Software Foundation GNU.
[User] Why is the sky blue?
[ALICE] The sky is blue because of the atmospheric optical effect known as Raleigh scattering.
[User] What is your favorite movie?
[ALICE] My favorite movie is Sneakers. Have you seen Sneakers?
[User] Yes, I enjoyed it.
[ALICE] That's great. Tell me what you liked about it.
[User] I have to go.
[ALICE] See you later. Nice chatting with you.

Comparing this conversation with the Eliza you can see major differences. Where Eliza just takes a key phrase or word and throws it back at the user, Alice appears to be searching a database of knowledge and providing the user with useful answers.

Note the first line 'I'm interested in HOPE College'. Any Eliza-type program would focus on the word HOPE and be sidetracked but Alice takes it in context. Also, in the line 'who is the president of the college?' Alice has used the fact that the topic of conversation is Hope College and realised that the question is directed within that framework, i.e. otherwise it would come up with the name of the President of the United States.

Impressive and amusing are apt terms to describe Alice but this is still a long way from our goal of creating an intelligent machine. The bronze Loebner Prize meant it was good, but not good enough to pass the Turing Test.

Applications of chatterbots

Chatterbots are becoming very useful in the commercial world, with the growth of electronic commerce on the world wide web. Ordinarily, there is no one to answer the customers' questions at a website. Chatterbots can remedy that problem. More and more companies are placing chatterbots on their websites to interact and converse with the user. Chatterbots make websites more 'user-friendly'.

There are three main applications of chatterbots:

1. Internet search tools
2. Interactive website
3. Shopping bots

Internet search bots

Internet search engines work by employing web 'spiders' which trawl the Internet and add URLs to a database which is then searched by the user. It is now possible to release your own spider to find and gather information on your behalf whilst you are doing another task. They work particularly well if the search is quite specific, e.g. 'find me the data protection act' as opposed to 'find me information on the second world war'. In essence, these cyberbots have been created to serve as personal research assistants for their respective users. 'Ask jeeves' is probably the most famous use of a search bot.

Interactive conversational bots

The Extempo company has created a number of chatterbots, which it calls Imp Characters, and which have a variety of applications according to the company website. ‘Imp Characters can play many roles, for example as a spokesperson in an online product showroom, a tour guide on a company website, or a bartender in an online pub’, Extempo promotional materials read. ‘Interacting through actions, gestures, facial expressions, and conversation, each Imp Character engages and delights users with its distinctive personality and individual style.’

Shopping bots

Shopping bots don’t really shop for the user, but rather they engage in price comparisons – i.e. find me a Smashing Pumpkins CD for less than £10 – and then the shopping bot reports back with a list of likely websites carrying the item you requested. An example is Junglee, which shops for electronics, music, books and clothing.

NativeMinds has developed **vReps** (virtual representatives) to interact with web surfers. The website hosts its own vRep, named Nicole. Unlike most other chatterbots, Nicole is made to look human. Some of NativeMinds’ customers are Oracle, Ford Motor Company, Nissan, The Coca-Cola Company, American Express, FannieMae, Convergys, Deutsche Telekom’s One 2 One, and Misys.
http://www.nativeminds.com/

Despite the growth and popularity of these cyberbots we have to remember that they are not really intelligent. None of them could pass the Turing Test, but who knows what advances may be made in years to come?

Complete Exercise 2 in Section 5

Expert systems

Early attempts at producing an intelligent program were hampered by the traditional approach in which programmers tried to 'code' a program which was intelligent. Despite numerous attempts none were truly successful but programmers did begin to realise that the scope of the problem was too large and that enormous amounts of data had to be stored in order to solve the simplest of problems. As a result AI stagnated throughout the 1970s.

During the 1990s processor speeds increased dramatically as did the amount of RAM and backing store. All of these factors contributed to computers which operated much more quickly and could store a lot of data. Combined with a fresh approach to the programming problem the conditions were right for resurgence in the field of artificial intelligence.

Conventional computer programs perform tasks using decision-making logic – containing little knowledge other than the basic algorithm for solving that specific problem and the necessary boundary conditions. This program knowledge is often embedded as part of the programming code, so that as the knowledge changes, the program has to be changed and then rebuilt.

Expert systems use a different approach; they collect the small fragments of human know-how into a knowledge-base which is used to reason through a problem, using the knowledge that is appropriate. A different problem, within the domain of the knowledge-base, can be solved using the same program without reprogramming. The ability of these systems to explain the reasoning process through back-traces and to handle levels of confidence and uncertainty provides an additional feature that conventional programming doesn't handle.

Instead of attempting to create an intelligent program, research focused on creating a means of representing and accessing knowledge. The result was expert systems, computer programs which could offer advice in a restricted subject where it was possible to create facts and rules representing knowledge. An expert system is an attempt to replace the human expert and to make their knowledge available in a cost-effective and non-perishable form.

Advantages of expert systems

Availability

In remote areas an expert might not be readily accessible or may be hours away from an emergency. However, with the knowledge and experience of a consultant housed in a computer, distance is not an issue. The expert system is also 'on call' 24/7 and can be consulted at any time.

Elf Aquitaine's drilling adviser saves time on diagnosing oil-rig faults. A lost day's drilling can cost $250,000. A BP fault diagnosis system on offshore oil rigs can solve in a short time problems that previously required flying the company expert there.

Reduced wages/cost

Training a doctor takes several years and costs a great deal of money. This cost can only be recouped through many years of service in the profession. Human experts can command large fees for their services, but once an expert system is set up the company wage bill can be reduced by employing fewer people. There may always be a need at some point to have human experts on hand but not quite so many.

Credit assessment at American Express is now done with fewer staff per customer than it was when the decisions were not filtered through an expert system.

Combined expertise

A single human expert has only his own knowledge gained from years of experience to guide him when faced with a problem. An expert system can contain the combined knowledge of many experts in the same field. Human experts can disagree on a course of action or the same expert might even come up with different advice in different situations. But an expert system will always produce the same advice when faced with the same input.

An airline uses an expert system to help reschedule aircraft movements when one breaks down. The human planner would take the first solution he could find – the expert system explores many more, and presents the best three.

Non-permanent

After money has been spent training the human expert he/she may move to another job with some other company; they may even decide on a career move. One thing is certain, he/she will retire or die (not a pleasant thought but we are mortal!). The computerised expert is not mortal and doesn't even require a holiday.

Reliability

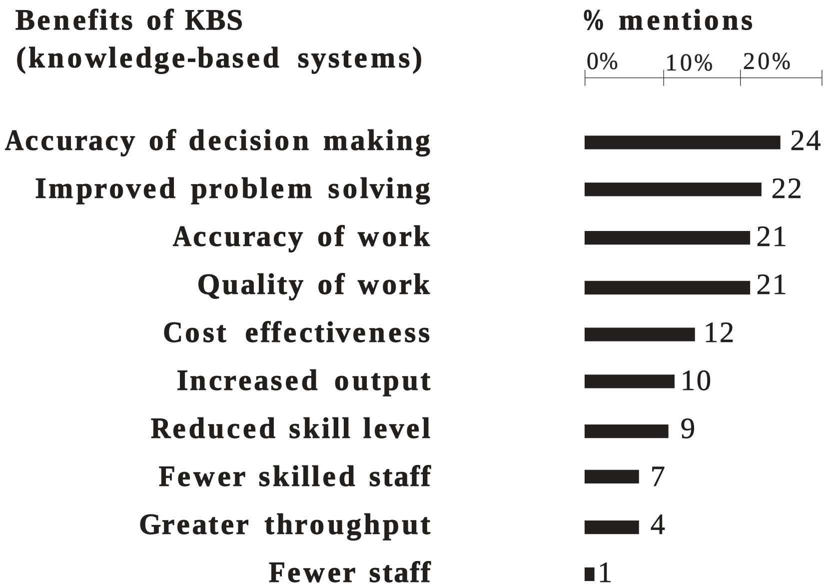
Despite the best of intentions human experts are prone to error. This may be caused by indecision, lack of information, emotional factors (home life) or just being tired on the job. There are a host of reasons why human experts are imperfect: knowledge can be forgotten as the years pass and the health of the expert may be an issue; humans are emotional creatures and this can produce a conscious or even unconscious bias in their reasoning; even the most learned man in his subject does not know everything; anyone can make an honest mistake in judgment; it is possible that the expert may lie or conceal information deliberately. Whatever the reason, human experts are vulnerable to outside influences that can affect their performance.

Expert systems have a restricted domain; they are only aware of the task for which they were designed and are not prone to interference from external factors. This makes them more reliable.

An expert system would not pass the Turing Test. It would perform well if you asked it questions about its domain, but as soon as you asked about some other area it would be unable to respond sensibly.

Do expert systems bring benefits?

The table below shows the results of a survey indicating the perceived benefits of expert systems by companies who were actually using them.



Further applications of expert systems

The commercial application of expert systems has grown considerably over the years. Here are a few examples of uses:

- Medicine — MYCIN was the first medical application of an expert system. There are now systems which can give advice to doctors on specialised areas such as kidney disease, cancer and blood conditions.
- DHSS — The laws and regulations governing the payment system have been rationalised by an expert system which will advise on the benefit due depending on circumstances.
- Legal — The laws in Scotland are vast and new laws are added each year. By asking an expert system solicitors can check that they are giving the right advice to a client. *(This is very controversial.)*
- British Gas — Expert systems can be used to predict future events based on previous trends and empirical data. British Gas has an expert system which is used to calculate the most likely place where corrosion will occur in a gas pipe.
- SEM Diagnosis — Developed to find faults in a Scanning Electron Microscope. Part of the problem is the wide geographical spread of SEM equipment and the lack of availability of human experts. This system is resident on a web server and can be accessed from anywhere.
- Aircraft — The maintenance of hydraulic landing gear systems usually involves a tedious process of manual look-ups to match the technicians' observations. This expert system would drastically reduce the amount of time the expert has to spend verifying the observations made by less experienced personnel for mundane tasks.
- Power Stations — During the 3 Mile Island accident, there were so many alarms going off, and so many gauges to check that the operators were confused. In some

French nuclear power stations, expert systems filter the alarm signals to present the critical ones to the operators.

- AFIS — Automated Fingerprint Identification System – provides automated searches of 10-print cards and latent fingerprints and generates a ranked candidate list. Unsolved latents can be entered and automatically searched against new 10-print cards.

Whale watcher is a program which is intended to identify a whale given its characteristics. You can download it from http://www.aiinc.ca/demos/whale.html Why not give it a try?

Social, legal and ethical issues

The rapid advances in artificial intelligence and in particular expert systems, have raised questions about where this technology will lead and what effects it might have on society.

For example, with the introduction of complex medical expert systems, it is becoming increasingly likely that in the future we will be able to predict what diseases people might have or might be at risk from later in life. Many of our present health problems have possible solutions through the use of artificial intelligence in studies at universities, hospitals and in research groups. For example, AI has made possible the implant of artificial corneas into the eyes of blind people, enabling them to see.

In aircraft control and nuclear power systems, AI is increasingly involved in carrying out safety-critical tasks with absolute precision, thus minimising, say, the possibility of the loss of life due to a slight error on the human operator's part.

Another future application of AI is in advancements in education. Researchers are developing new techniques in the field of intelligent tutoring systems (ITSs). ITSs will offer considerable flexibility in the presentation of material and a greater ability to respond to student needs. Such systems claim to advance a student's ability level by as much as six months with regular use.

Examples such as these are clearly beneficial to mankind as they are fulfilling the worthy goal of improving the quality of life. However, with any advances we seem to make there always appear to be some drawbacks.

Perhaps the single most obvious drawback is the amount of funding that has gone into AI research. Government funding is crucially important in establishing new disciplines because it can sustain long-term, high-risk research areas and nurture a critical mass of technical and human resources. AI is no exception. The big question is whether such funding is justifiable. Indeed, in an age and a world in which poverty, disease and hunger still plague the less fortunate, wouldn’t the billions of pounds be better spent helping those in need? Put it another way, why worry about the possibility of artificial life when we still haven’t conquered our own more immediate problems?

Effects on employment

One of the advantages of expert systems for employers is that of reduced cost due to the fact that we do not require so many human experts. Rather ironically, it was the lack of experts in the first place that created the demand for expert systems; surely we can’t have it both ways?

If we decide to fully embrace expert systems within society the initial impact must be to reduce the number of human experts in the field. There are two major impacts in the notion of replacing human experts with machines:

1. Expert systems were initially created from the knowledge that was extracted from human experts. One of the advantages was the fact that the experiences of a large number of experts could be incorporated into the system. If we reduce the number of human experts or even have none, then how do we check that the advice is correct? Every expert system will arrive at the same conclusion and give the same advice!

2. Creativity is seen as one aspect of intelligence but where is the creativity when the expert system is simply searching a database of knowledge? Where and how do we create new ideas? It is the human experts that make progress in their field by proposing theories and doing hard research. Without human experts the body of human knowledge will stagnate as new ideas are not formulated.

Perhaps we will always need human experts after all.

What happens when we allow expert systems to invade others areas of society?

Many fear that using AI in industry will mean a decrease in jobs in general, and for good reasons. One current area of development, which could mean a loss of jobs, is that of the on-line sales clerk. With the introduction of intelligent agents (chatterbots) to the Internet, we are now able to shop for things online. Consequently, as time progresses, the role of a sales clerk in a high-street shop may not be needed any more.

The job of a sales clerk can hardly be described as that of an 'expert' in the same way as a doctor or engineer but remember, even what we consider to be the simplest of tasks can require a degree of intelligence. If we do reach the stage when machines which can hold a meaningful conversation with a human it's not going to be only the 'experts' whose jobs are at risk.

Ethical considerations

As computers are programmed to act more like people, several social and ethical concerns come into focus. For example: are there ethical limits to what computers should be programmed to do?

It was pointed out that one major benefit of expert systems is in the area of safety-critical systems in a nuclear power station. But suppose a machine does something wrong, or chooses an inappropriate route, which then leads to loss of life, who should be held responsible? We cannot punish the machine itself, so should we blame the programmers, the experts, or the people who decided to put the system together in the first place?

In the 1980s there was a dramatic fall in the stock market and share prices worldwide plummeted. The problem escalated very quickly from one market to the next until traders were dreading the opening of business the next day. No one seemed to be in control, and despite assurance from government officials markets continued to fall. This had a lasting effect on jobs, economies, businesses, pensions and investments for years to come. When the smoke cleared the blame was levelled at computerised trading systems which were designed to react to market forces. If the general trend was for share prices to fall then the systems would sell, feeding the fall.

The question of blame is currently perceived to be in the hands of the person who used the expert system in the first place. The program is regarded as a resource which the expert can choose to use or ignore at his own discretion. When the expert takes advice from an expert system and it is shown later to be flawed, he may have a case against the company that produced the system. But ultimately the human expert is responsible as he is supposed to be in charge.

Moral issues

A less dramatic but equally valid scenario is where the expert system has been developed to provide the percentage likelihood of success in a particular operation. The patient is 60 years old and the chances of survival are 20%. The operation will cost £50,000. Our expert system advises us not to go ahead with the operation. This scenario goes to the very heart of the debate about man being ruled by machines.

It is the gradual handing over of our independence and decision making to computers that the public are increasingly concerned about. A doctor using an expert system to help make a diagnosis is one thing, but the thought of a machine treating a patient on a one-to-one basis is not only impersonal but repulsive to the public in general.

The claimant at the social security office is told that they will not receive a particular benefit because the 'system' will not allow it. Regardless of the fact that it is humans who make the laws governing the running of the country, it's the machine that is seen to be the culprit.

How many times has a machine sent a bill to someone for £0.10, even though the cost of the postage is more? Or worse still, the computer sends out a bill to someone who has died? This is a scenario that highlights the dehumanising effects that machines can have on man.

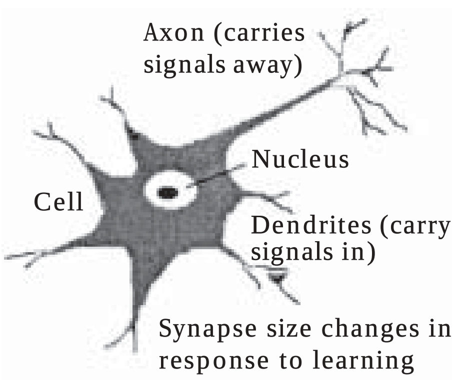
Complete Exercise 3 in Section 5

Artificial neural systems

After making numerous attempts to write programs in the conventional manner, researchers began to believe that an intelligent program would require a completely new approach. Neural networks take a different approach to problem solving from that of conventional computers. Conventional computers use an algorithmic approach, i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known, the computer cannot solve the problem. That restricts the problem-solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do.

Neural networks are a very different type of computation	
Conventional computation	**Neural networks**
• single processor • fault intolerant • serial • general to any task	• many processors • fault tolerant • parallel • designed per task

Researchers looked at how the human brain operates and decided to try to create a system which could emulate the brain.

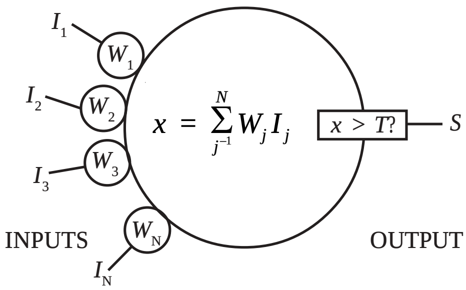


The most basic element of the human brain is a specific type of cell, which provides us with the ability to remember, think, and apply previous experiences to our every action. These cells are known as **neurons**, each of which may connect with up to 200,000 other neurons and the power of the brain comes from the huge numbers of these basic components and the multiple connections between them.

The brain consists of millions of interconnected records. An artificial neural system consists of hundreds of interconnected artificial neurons, so it is based on the same model as the human brain, but with far fewer neurons.

The artificial neural system is the result of attempts to copy the way the human brain works. In a biological system, learning involves adjustments to the synaptic connections between neurons. The same is true for artificial neural systems.

An artificial neuron



In contrast to conventional computers, which are programmed to perform a specific task, neural networks must be taught, or trained. They can learn new associations and new patterns which, once learned, allow the neural systems to recognise features or characteristics, e.g. learning to read English; reading postcodes.

Artificial neural systems process information much as the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural systems learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted, or even worse the network might function incorrectly. The disadvantage is that, because the network finds out how to solve the problem by itself, its operation can be unpredictable.

Just like people, neural systems learn from experience, not from programming. They are good at pattern recognition, generalisation, and trend prediction. They are fast, tolerant of imperfect data, and do not need formulas or rules. Neural systems are trained by repeatedly presenting examples to the network.

You can download a free simulation of a neural net from:
http://rainbow.mimuw.edu.pl/~mwojnar/ltfcim/

Applications of artificial neural systems

Neural networks have broad applicability to real-world business problems. In fact, they have already been successfully applied in many industries. Because neural networks are best at identifying patterns or trends in data, they are well suited to prediction or forecasting needs including:

- recognition of speakers in communications;
- diagnosis of hepatitis;

- three-dimensional object recognition;
- hand-written word recognition;
- facial recognition (used by police forces).

Credit scoring

The HNC company has developed several neural systems. One of them is the Credit Scoring System which increases the profitability of the existing model by up to 27%. The HNC neural systems were also applied to mortgage screening. A neural network automated mortgage insurance underwriting system was developed by the Nestor Company. This system was trained with 5,048 applications of which 2,597 were certified. The data related to property and borrower qualifications. In a conservative model the system agreed with the underwriters on 97% of cases. In the liberal model the system agreed with 84% of cases. This system can process a case file in approximately one second.

Debt risk assessment

Loan granting is one area in which neural networks can aid humans, as it is not based on predetermined criteria, but instead answers are vague. Banks want to make as much money as they can, and one way to do this is to lower the failure rate by using neural networks to decide whether the bank should approve a loan.

Neural networks are particularly useful in this area because no process will guarantee 100% accuracy. Even an 85–90% accuracy would be an improvement on the methods humans use. In fact, in some banks, the failure rate of loans approved using neural networks is lower than that of some of their best traditional methods. Some credit card companies are now beginning to use neural networks in deciding whether to grant an application.

The process works by analysing past failures and making current decisions based upon past experience. Nonetheless, this creates its own problems. For example, the bank or credit card company must justify its decision to the applicant. The process of explaining how the network learned and on what characteristics the neural network made its decision is difficult; you can't look at the code and follow the data path. Imagine the reaction from the applicant if they were told: 'You can't have the loan because my neural network computer recommended against it.'

Stock-market prediction
Neural systems have been touted as all-powerful tools in stock-market prediction. Some companies have claimed an amazing 199.2% return over a two-year period using their neural network prediction methods. The idea of stock-market prediction is not new, of course. Business people often attempt to anticipate the market by interpreting external parameters, such as economic indicators, public opinion, and the current political climate. The question is, though, can neural networks discover trends in data that humans might not notice, and successfully use these trends in their predictions? For some strange reason the precise details of how such neural systems operate is a well guarded secret!

Here are some details on a commerical stock-market predictor which uses a neural network called 'Brainmaker'.

The system uses Brainmaker to determine the average discount the market is currently allocating to particular industries as a whole (e.g. oil, power, etc.), and then uses that standard to compare different types of industries to find out those stocks which are trading below their market value. In plain English, the system will work out what the value of stock is worth and compare that to its actual value on the stock market.

How well does this system perform in stock price prediction? From January 1996 to February 1996, the system's 20 most undervalued stocks have risen by 44.40%. *This system appears to be beating the experts at their own game.*

Postal services
Siemens and IBM are two companies interested in the development of off-line character-recognition and document-analysis technologies. Off-line recognition is applied to handwritten documents and is used, for example, to help post offices automate the sorting of mail.

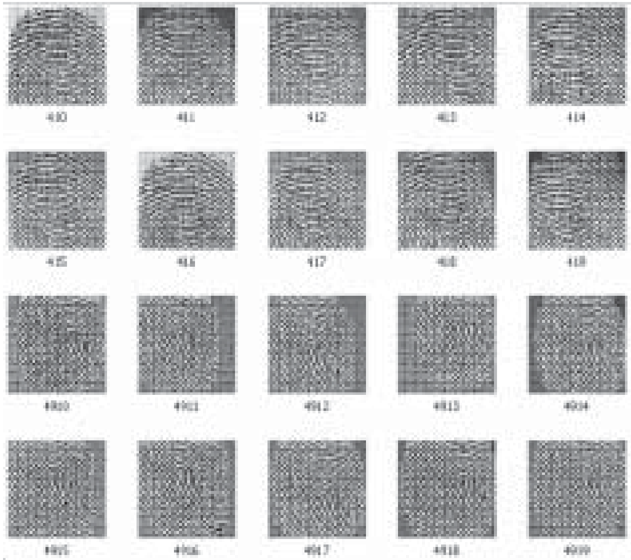
Postal automation represents a fertile area for the application of image processing and neural network techniques. The postal service in the UK alone processes over a million pieces of letter mail per day and about 10–15% of these are handwritten.

There is a growing need to automate mail sorting worldwide, particularly at post offices where mail is presently sorted largely by hand. Currently, the US Post Office is the only one in the world that uses automated mail-sorting machines. Australia, the United Kingdom and Canada all plan to move to automated sorting to cut costs, which will create a considerable global market for mail-sorting equipment.

Fingerprint recognition

Several fingerprints of the same finger are taken and digital images of each are created. This is done using a scanner. The images are then 'enrolled' as a group into the software recognition package. The package examines each of the images and extracts general features which are common to them all. If more images are taken, the accuracy of the matching process is improved.

Common features identified



Once the enrolment process is complete the grouped images are stored in a database. This database is the source file which will be searched whenever a matching fingerprint is being sought.

The suspect's fingerprint is scanned and loaded into the software package, then the identification/search feature is activated.

At this stage the fingerprint is compared to the ones in the database and all matches are reported.

As the images will not be physically 100% identical (even if they are from the same finger), there is an artificial neural system within the software which carries out a 'pattern matching' exercise based on the common features which were identified during the enrolment process. Users can adjust the percentage match to widen the search criteria.

All matches are reported after the database has been fully searched.

Applications:

- This is how the police catch suspects when a fingerprint is taken from the scene of a crime and checked against a database of convicted criminals.
- Activate modern tablet PCs instead of logging on with a password.
- Door locks can be equipped with fingerprint recognition for secure entry.

Face recognition

This is a similar process to that of fingerprint recognition. Several images are taken of the same person and 'enrolled' together into the software package. Once again, common characteristics are taken from the images and stored in a template of the person. The more images that are taken, the greater will be the accuracy of the match.

Common features extracted

When a 'suspect' is found, an image is taken face on and loaded into the software package. The image is then compared to the templates in the database. As with fingerprints there will never be a 100% match so the user has to set the 'matching threshold'. All images in the database which fall within the matching threshold will be reported to the user. Manufacturers of face recognition systems offer the following advice when taking both the initial database images and the image to be matched.

When taking the image ensure that the following features are not present:

- a smile where the inside of the mouth is exposed (jaw open)
- raised eyebrows
- closed eyes
- eyes looking away from the camera
- squinting
- frowning
- hair covering eyes
- rim of glasses covering part of the eye.

The software contains an artificial neural network which is responsible for controlling the matching process. Systems like this are increasingly used by police forces to identify suspects who have been caught on CCTV and may have been involved in criminal activities. One was even used to identify football hooligans who invaded the pitch during a match.

Complete Exercise 4 in Section 5

Vision systems

To fully understand the workings of machine vision a good starting point is to compare it to our own vision system with which we are all familiar.

Characteristic	Human vision	Machine vision
Image capture	People see images by receiving light that is transmitted or reflected by objects on to the retina.	In machine vision systems this sensor is normally a CCD chip in an industrial video camera.
Image processing	Our eyes send the images to our brains to be processed and interpreted.	A camera sends its images to a computer, which is programmed to do the analysis, interpretation and decision making.
Image Interpretation	The human brain has had lots of experience of seeing images and can identify from previous knowledge.	In machine vision the computer compares the images of items passing on a production line, with a 'learned' image of an ideal item of the same type. Systems use artificial intelligence to improve their flexibility and to increase their 'prior' knowledge.
Speed of the process	This entire processing is done so fast (about 100 billion complex operations per second) that we are normally not aware of it happening alongside all the other things we do in our day-to-day lives.	As technology is progressing, machines are rapidly becoming faster. Nowadays, specialised hardware can also manage 100 billion complex operations per second of basic calculations.

Some (but not all) vision systems use artificial neural systems for image interpretation.

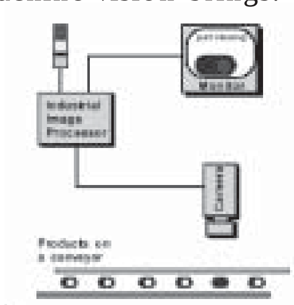
Making sense of the image is where the neural system comes into play. With its ability to pattern-match, the neural system will compare characteristics of the object with those in its memory and try to 'match' the object. Without some form of analysis of the object to make it meaningful the whole process would be pointless.

Applications of vision systems

Industrial use

There are a number of advantages that machine vision systems have over people for checking items on production lines. First of all, a machine always does the same thing, with the same accuracy, over and over again, not being troubled with fatigue or illness, not requiring any pause, leisure time, holiday, or wages. So machine vision brings:

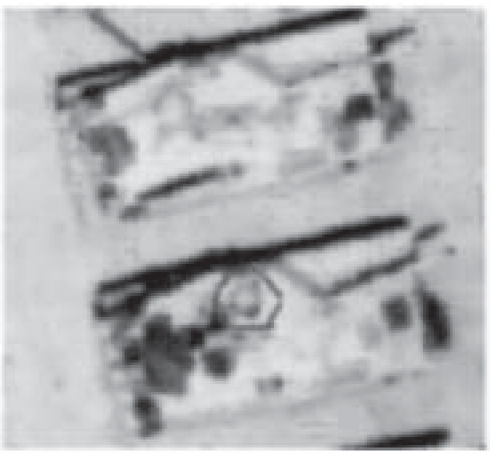
- consistency
- objectivity
- constant high accuracy
- at low cost.

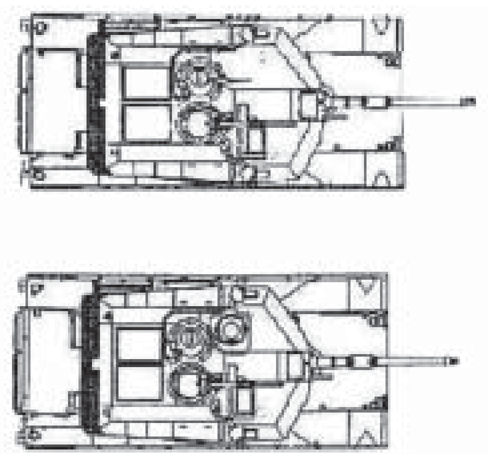


Military use

A great deal of the funding for artificial intelligence comes from the military. The First World War was known as the chemical war, the Second as the nuclear; if there is a third, will it be the war of intelligent machines?

Target recognition is a military application which uses video and/or infrared image data to determine if an enemy target is present. Not only can such systems identify enemy targets, but they can also be trained to recognise specific vehicles – giving a complete breakdown of enemy forces.





However, training can go wrong. A military project apparently trained a neural network to recognise camouflaged tanks in photographs of woodland. Unfortunately it turned out that all the pictures with tanks in were taken in sunny weather so what they had actually created was a neural network for recognising photographs of sunny woods!

Satellite photo interpretation

Satellites collect an enormous amount of data which would normally take researchers and scientists years to analyse and interpret. Once the image is captured, finding areas of interest can be accomplished by experts who are skilled at spotting terrain features which give signs of what lies beneath. By training a neural system to spot the characteristics which are important in the photos it is possible to examine the growth of crops; monitor environmental changes; predict the possibility of oil or gas; and in the example below predict fire patterns.

Images taken by a satellite above Stanford University in California are fed into a neural system which examines the flammable vegetation around populated areas and predicts fire patterns in case of an emergency. This information is then used to provide managers with the best positions in which to deploy firebreaks.

Looking for lesions

Doctors are in the habit of using endoscopes (miniature body cameras) to investigate the insides of their patients. Once the images have been taken, experts can examine the tape, looking for abnormalities some of which are very difficult to spot. By training a neural net the images can be examined by computer for signs of lesions, for example in the esophagus, saving valuable time for both the doctor and the patient.

Furthermore, a camera can be placed in locations where people cannot operate, or even where humans could not survive, such as in a limited space, under extreme temperatures, in dangerous atmospheric conditions, or in other special environments. Finally, a machine’s vision can be extended to see what the human eye cannot observe, such as defects in products that can only be detected with X-rays or with ultraviolet or infrared light.

Complete Exercise 5 in Section 5

Speech recognition

The objective of speech recognition software is to allow the user to communicate with the computer by talking to it. However, before using the system you must first go through a training process; not for you, for the computer!

This consists of reading a pre-defined text into the computer for about twenty minutes. While you are reading the text the computer is sampling your voice and matching it to sounds which are common in all words. The reason everyone has to go through this training process is because of the characteristics of human speech. Everyone's voice is unique and so the computer has to be able to recognise the individual patterns for each person.

The next stage is to dictate to the computer through a microphone, preferably when there is little noise in the background that might distort the sound. The quality of the microphone is important and the computer's processing power is also a crucial factor.

The instructions below should be taken into account before training.

- Speak in a consistent, level tone. Speaking too loudly or too softly makes it difficult for the computer to recognise what you've said.
- Use a consistent rate, without speeding up or slowing down.
- Speak without pausing between words; a phrase is easier for the computer to interpret than just one word. For example, the computer has a hard time understanding phrases such as 'This (pause) is (pause) another (pause) example (pause) sentence.'
- Start by working in a quiet environment so that the computer hears you instead of the sounds around you, and use a good quality of microphone. Keep the microphone in the same position; try not to move it around once it is adjusted.

- Train your computer to recognise your voice by reading aloud the prepared training text in the Voice Training Wizard. Additional training increases speech recognition accuracy.
- As you dictate, do not be concerned if you do not immediately see your words on the screen. Continue speaking and pause at the end of what you are saying. The computer will display the recognised text on the screen after it finishes processing your voice.
- Pronounce words clearly, but do not separate each syllable. For example, sounding out each syllable in 'e-nun-ci-ate,' will make it harder for the computer to recognise what you've said.

Continuous speech recognition (SR) software takes voice input through the user's microphone and uses it to 'type' words into a document displayed on the computer screen. (This document can be saved and used just like any other file.) It is called 'continuous' because the user is expected to dictate in a conversational manner, speaking entire phrases with brief pauses between. This is different from 'discrete' speech recognition software, where words must be uttered individually.

Speech recognition uses a neural net to 'learn' to recognise your voice. As you speak, the voice recognition software remembers the way you say each word. This customisation allows speech recognition, even though everyone speaks with different accents and inflection. In addition to learning how you pronounce words speech recognition also uses grammatical context and frequency of use to predict the word you wish to input. These powerful statistical tools allow the software to cut down the massive language database before you even speak the next word.

The primary use of SR is in word processing. Producers of SR make varying claims of compatibility of SR with other types of applications, including spreadsheets, web browsers and e-mail clients. However, the technology is now working its way into embedded computer systems including mobile phones, PDA devices and cars.

The function of SR software may be divided into three broad areas:

1. dictation
2. correction and editing
3. command and control.

Some SR products perform tasks within these areas better than others, and the extent to which users may use functions and features within these areas exclusively by voice varies from one SR product to another. All will require some keyboard or mouse input to accomplish certain tasks – none is 100% hands-free.

During a dictation session, users simply say what they want to appear in the document. Inadvertent sound such as coughs or 'ahem'-type hesitations are to be avoided, as the SR software will attempt to interpret these and insert them into the document. It cannot 'guess' what the user meant to say and meant not to say. Recognition errors will occur. Users play an extremely important role in preventing misrecognitions by enunciating each word with precision. This doesn't mean that users must dictate more slowly, loudly, or using one word at a time. In fact, any of these would be counter-productive. What it does mean is that users will get better recognition accuracy if they concentrate on pronouncing every syllable of every word clearly, without any slurring or laziness.

Correction is the process of replacing a misrecognised word with the intended word. Each SR product has a specific procedure for accomplishing this task. In contrast, when users change their minds about which words they want to use in their documents, replacing existing words with new words is **editing**. Each SR product also has a specific procedure for editing. Normally words in a document which are suspect will be highlighted. If the user is in command mode and the software is unsure of the command it will prompt for a repeat of the word.

Command and control includes using menus, changing settings in dialog boxes and similar tasks. SR products vary in their ability to accomplish these tasks with voice input. Keystrokes or mouse movements and clicks may be required. There are some issues that are unique to the use of SR software. Although the vocabularies of SR applications are very large, an unusual proper noun might require the user to take the extra step of adding it to the vocabulary for better future recognition.

When used for command and control purposes SR is really acting as an alternative form of HCI (human computer interface) as it is possible to control the computer using command instructions. This has major benefits to users who previously were unable to operate a computer because of disability. The user can switch from dictation mode to command mode easily when working on a document. In many packages if the software is in dictation mode commands can still be used if the command is preceded by a ‘command’ word like ‘computer’. Some typical commands are given below.

Command	**Action**
Computer file	Drop down the file menu
Computer open	Open the directory
Computer close	Close a file
Computer insert table	Insert a table
Computer select word	Highlight current word
Computer select paragraph	Highlight current paragraph

SR software should not be installed on marginal systems. A minimum system for satisfactory use with the native word-processing application would be a processor of not less than 600 MHz, RAM not less than 256 MB, and audio components of high quality. If the intended use includes using SR within other applications, a more powerful computer will be required. Given good hardware, users who develop good enunciation habits should have excellent results.

Is speech recognition software a form of artificial intelligence?

Yes, because the software is not simply recognising sounds; it is using its knowledge of words, grammar and language structure to interpret the input in a flexible way. The use of a neural network to ‘learn’ the user’s speech patterns and even predict the next words puts this software firmly in the field of artificial intelligence.

Applications of speech recognition

As previously mentioned, the most common use of SR is for word processing, but it can also be used for spreadsheets and web browsing. After the initial training session the system will adapt and become more user specific and accurate with continual usage. Further training sessions will also help.

Once the system is working to peak performance and the user is enunciating correctly, documents can be dictated with more than 99% accuracy. During the dictation process the user will wish to insert a comma or a full stop or open a new paragraph, so the software has to have some means of identifying the text which the user wants to dictate and the commands to insert a comma or even to format the text. It really depends very much on the particular software package being used, but some require a *command word* before the actual command (this could be 'computer' or 'mate' or just about anything). Other packages are fairly smart and simply stating the words 'new paragraph' is enough.

Further applications of speech recognition

The use of SR software is not restricted to desktop computers but has many other applications, particularly in embedded computer systems.

In cars
Embedded and network-based speech recognition plays a vital role in the automobile industry, because a major issue is safety. Drivers want to be informed, connected to resources and be able to make transactions via mobile phone while they are on the road. For many consumers, driving time is considered idle time. Services such as driving directions, business finder, roadside assistance, e-mail, Internet access and corporate access are examples of such requests. Recent laws banning the use of mobile phones in vehicles will make the use of voice-activated handsets increasingly important.

BMW's voice-activated system in their series 7 models has the following functions: calling home, checking voice mail, operating the navigation system, switching among two or three favourite radio stations. To operate the radio the driver has only to issue the command 'Radio on' followed by the station 'FM 108.2'. You can also tell it which CD track to play.

Telephony and telecommunications
Embedded and network-based speech recognition technology is used by a number of companies in the industry. Embedded voice recognition enhances handsets' capabilities, making them more productive, multi-tasking tools. Wireless phones, wireline phones, PDAs and other devices can include many speech features, such as voice activation and voice dialling. Voice portals use network-based speech recognition to help operators perform a range of transactions and access information more efficiently.

Electronic devices
Speech recognition is being used in electronic devices such as phones, computers, TV sets, security and monitoring systems, audio/video equipment, radios and lights. Speech control has a key effect on drivers, the handicapped and the elderly. Many companies are using advanced technologies to increase efficiencies and cut costs in their operations to

remain profitable and competitive. Companies are taking advantage of hands-free system control, which lets employees access data and entry to warehouses or industrial units by using their voices.

Disabled users
There is a large market for the use of SR technology to help the disabled to communicate either with other humans or using a computer. One manufacturer quotes the case of a computer operator who was diagnoised as suffering from Lou Gerhigs disease. This is a neurological disease which attacks the motor functions making the patient unable to move or control their hands. The operator was allowed by his company to work from home, and using the speech recognition software, he was still capable of doing his job.

Military use
Speech technology has many potential applications in the military sphere, from the hands- and eyes-free operation of complex command and control systems (where a speech interface can improve situational awareness), to training simulators where a speech interface can simulate multiple remote operators, removing the need for additional human operators.

There are many ways in which speech technology can be applied. Some examples are:

- automatic speech recognition – for hands-free command of equipment;
- text-to-speech synthesis – for text message playback;
- spoken language dialogue management – to extract information from the spoken language to facilitate meaningful dialogue between humans and machines;
- speech categorisation – to identify a speech signal, its language, the speaker, the message gist and more; intelligence gathering
- speech morphing – to allow communication with divers using helium.

Phraselator
About 200 American and British military personnel in Iraq are communicating with the locals using a hand-held device into which soldiers speak English phrases to have them sounded out in either Arabic or Kurdish.

The device, called a Phraselator, is designed to help compensate for a shortage of linguists, and it has already proven its worth. The device has been used to locate caches of hidden weapons and hostile forces. A similar device was used by the Americans in Afghanistan and was found to be very helpful as that country has four main native languages.

The Phraselator uses speech-recognition technology called Dynaspeak, developed by SRI International. This technology recognises phrases phonetically and then emits the equivalent pre-recorded phrase in Arabic, Kurdish or another foreign language. The devices used in Iraq had between 500 and 1000 standard phrases stored on a 64-megabyte flash card. However, the card can hold an amazing 30,000 possible phrases!

The device does not pretend to be a language translator as such, merely a ‘phrase matcher’. It is not perfect by any means but the use of such systems may well increase and eventually make their way on to the commercial market.

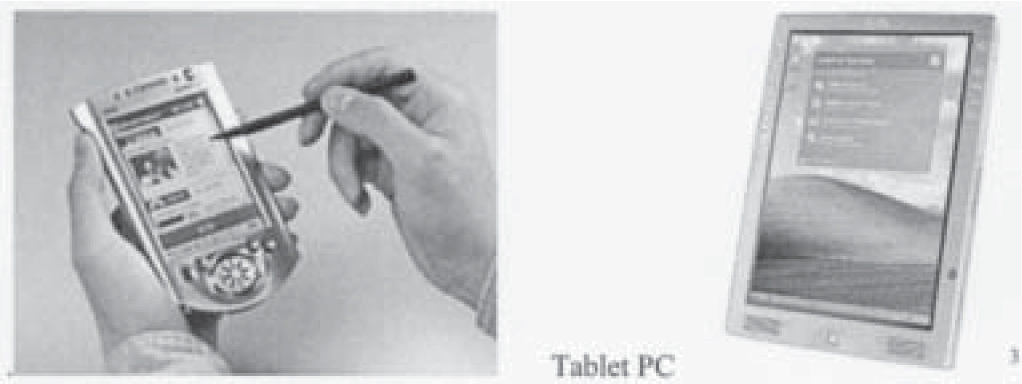
Aircraft

An obvious application of speech technology is in the operation of military aircraft. A commercial pilot is faced with a myriad of controls, lights, panels and devices in the cabin and this is even more true for the military pilot. Research is currently being conducted in both the commercial and military fields to develop a working speech recognition system for an aircraft. The main problems faced by such systems are the background noises. In any aircraft there is a great deal of background noise from engines, fans, electrical equipment, landing gear, etc. In a military environment there also could be explosions, rockets launched and machine-gun fire. Although SR is not yet available in aircraft it is used in simulators for training purposes.

Complete Exercise 6 in Section 5

Handwriting recognition

Systems using handwriting recognition have become very popular in the past few years and there is now a wide range of devices which use this form of input. The obvious advantage compared to the conventional keyboard is in space saving; most systems combine the input and output device in one, that is, an LCD screen in which the user writes directly on the surface using a stylus pen (really just a piece of plastic); then these scribbles are translated into ‘typed’ text and displayed back on the LCD.



Tablet PC [3]

How are characters recognised?

Early handwriting systems required the user to enter the letters of the alphabet and numbers so that they could be stored and matched against any future input. The program would try to match each letter as best it could and display the result on the LCD. Clearly, the important factor is not just for the user to write in a reasonable, legible fashion but to be consistent in their writing; i.e. do not write a seven as 7 one minute then change it to 7 the next.

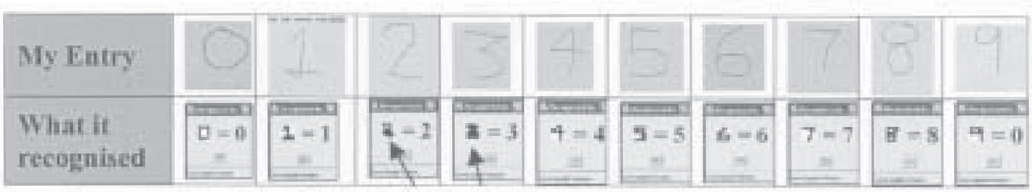
The example below describes an attempt to ‘train’ a java applet on the Internet so it will recognise the digits 0–9 in my handwriting (or rather my scribbles using a mouse). I first had to draw each digit in turn and tell it which digit was a 0,1, 2 etc. up to 9.

This window shows the digits I entered into the system. When I test the system it will attempt to match each number in turn. *Note my number 8 and number 9 are not very well defined (that was deliberate, honest!).*

[3] Kind permission from RM (Research Machines)

After entering each digit above I clicked on the ‘train’ button so that the program would recongise each of the above as each digit indicated.

I then used the mouse to write the numbers 0–9 and got the program to display what number it thought I had entered. The table below shows my entries and the results.



The results are quite interesting; every number was recognised apart from the 9. To explain why this was not recognised look at the first table and see what I told it a ‘9’ looks like. My ‘9’ looks like a zero and that’s just what the machine thought it was.

Actually I’m amazed that it managed to pick out the 2 and 3 because the image I drew in each case seemed OK but the pixel image it grabbed was a complete mess. Look at my drawing of a ‘1’. Doesn’t it look a bit like a ‘2’? But the machine wasn’t fooled.

You can try the above java applet yourself at the following URL:

http://members.aol.com/Trane64/java/JRec.html

Be sure to read and understand the instructions before using the applet.

No training!

There have been major advances in handwriting technology to the extent that in a modern system you do not have to train the software. The new techniques examine the start and end position of the stroke as well as the finished image.

It is now possible to write on the LCD at a reasonable speed and have the input converted to text underneath while you are still writing. The processor speed will be a large factor when it comes to the time delay between writing and the transcribed text appearing, but normally it takes only a few seconds.

Try the following link to see an online example of how it works.
http://www.penreader.com/WinCE/PenReader.html

The link below is another java applet but you don't have to train it.
http://www.thomastannahill.com/tom-ato/applet.html

Accuracy

As with voice recognition, accuracy can depend on the user being consistent. Here are a few recommendations from a manufacturer of handwriting technology.

- You will have the best recognition quality when using the default letter shape settings. Disabling too many shapes will decrease the recognition quality. There is only one reason to disable a certain shape of a certain letter, which is if the calligrapher recognises this letter incorrectly because you wrote its shape similar to another letter. For example, if your 'o' is recognised as 'u' you can try to disable some shapes for 'o' or for 'u' or for both. However, if you disable too many shapes you will decrease the recognition quality.

- Try to write bigger characters. The minimum recommended size of your handwritten characters should be equivalent to size 16 (or bigger) of the Tahoma font. (To get an idea how big it is, run the Pocket Word, select the Tahoma, size 16 font, and write 'something'.)

- Upper-case characters should be at least twice as big as the lower-case characters (or bigger).

Tablet PCs

These are a new brand of PC which have the functionality of a desktop PC and are more portable than a laptop. The advantage of having no keyboard or mouse (although they can be connected) makes the tablet ideal for carrying around and storing in a bag. With this in mind, educational establishments have been inundated with literature from manufacturers who are keen to target students with their products.


[4]

The new tablets have very accurate handwriting recognition software making them ideal for students sitting in on a lecture and taking notes. The notes can be edited later and they have the advantage of already being in a typed format.

Applications

Handheld devices which use handwriting recognition include:

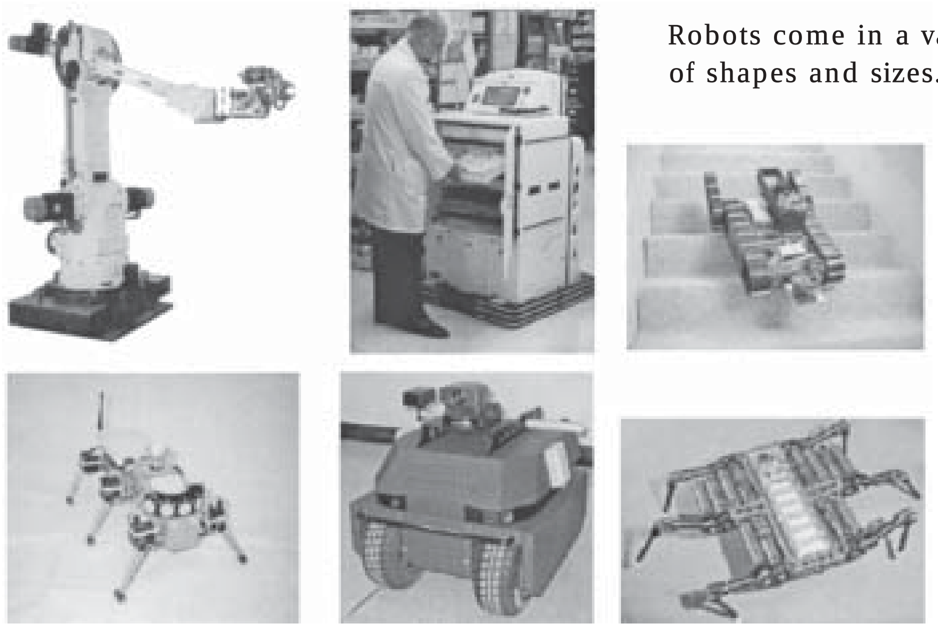
- PDA (Personal Digital Assistant)
- Tablet PC (mentioned above)
- Electricity meter reading (take the customer's signature as well)
- Post office form filling
- Supermarkets for stock control.

Complete Exercise 7 in Section 5

[4] Kind permission from RM (Research Machines)

Robots

Robots are automatic electro-mechanical machines. Some are not fully automatic but are remotely controlled by humans using TV and other feedback. Most are used to move parts into and out of fabricating machines such as machine tools. Others are used to manipulate fabricating devices such as spot welders, arc welders and paint sprayers. Others are used as remote manipulators in deep sea, radioactive laboratory, satellite, hot and dangerous factory, and other environments hostile to humans. Some are mobile, travelling on land, sea or air, propelled by wheels, water propellers, and air propellers (or jets). Guided missiles and drones are classes of robot.



Robots come in a variety of shapes and sizes.

The term ‘intelligent robots’ has been coined to differentiate robots equipped with vision systems, artificial neural control systems, the ability to learn, or some other aspect of artificial intelligence from the pure automatons. A ‘dumb’ robot, much like a ‘dumb terminal’, has no controlling processor and it merely repeats the same movements again and again regardless of the circumstances.

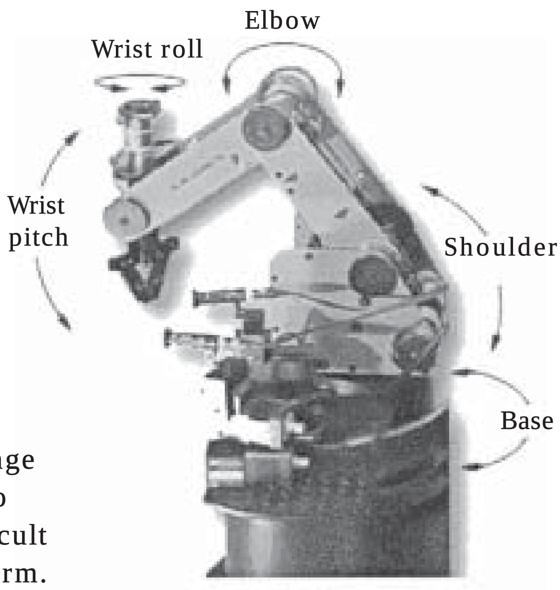
If the robot is spraying bicycles frames which are moving along a conveyor belt and one of the frames falls off the belt, a dumb robot will continue to spray into thin air. It has no knowledge of its surroundings and is an automaton in the purest sense.

Some robots have simple sensors which send feedback to a controller. These are better than dumb robots, but are still not really intelligent.

There are a great many different types of robots which perform a huge range of tasks. Despite their many differences there are some characteristics which all robots have in common. We will examine the common features of robots, look at the jobs they perform and consider the reasons for having robots in the first place.

Anatomy of a robot

Our first impression of a robot is usually of a machine which has arms and legs, is made in the image of a human and walks in a rather ungainly fashion. It seems natural to compare robots to humans and to some extent this comparison is valid as many robots are designed to replace jobs previously done by humans, so their body parts are similar. One of the most commonly used robots in industry is the robot arm.



The example on the right is a typical robot arm. It has three main joints: shoulder, elbow and wrist.

Each of the joints allows the robot to move on different planes and enables it to position its end effector to perform tasks.

The wrist is similar to a human wrist and can be designed with a wide range of motion. This enables the robot to reach into places that would be difficult or impossible to reach by a human arm.

This type of flexibility makes the robot's value much greater and its capabilities attractive. The grippers are at the end of the wrist. They are used to hold whatever the robot is to manipulate. Some robots have end-of-arm tools instead of grippers. A few examples of these tools are painters, arc welders and spot welders. All these put together make a manipulator. This enables a robot to pick up an object, manipulate it, and set it down where desired.

Sensors

Sensors are the first link between the typical automated system and the conventional process. A robot can use a wide variety of switches. First is a manual switch, which tells the robot yes or no (like the one in the bicycle example above). Virtually all manual switches are electric, and most often are used to turn the robot on or off, or to make adjustments to the automotive cycle. Limit switches are activated by levers, toggles, push buttons, plungers, rollers, whiskers, and just about anything else the developer can engineer to make the device automated.

Proximity switches do not require physical contact or light radiation to feel or sense an object. They are called proximity switches because they are able to sense a nearby object without actually touching it. This gives the robot the ability to do something that humans are not able to do, and is done by using electromagnetic waves or sonar.

Photoelectric sensors are sensitive to light radiation. Sometimes it is useful to detect electromagnetic radiation outside the visible range. This is where infrared is used. This would include the sensing of hot objects, as hot objects emit infrared rays. These are highly useful for machine malfunctions and are used as automatic emergency shutoffs.

The commonly used sensors include:

- bump (contact) sensors (switches see above)
- proximity switch
- light sensors/photoelectric (see above)
- temperature sensors
- pressure sensors.

Advantages of robots

- Increased productivity: Robots do not require breaks or even working hours; they can be put into production continuously day and night.
- Improved accuracy: The precisely controlled actions of a robot mean that in situations where detailed work is required they are more accurate than humans, e.g. checking circuit boards.
- Consistency: The first job of the day and the last will be done in exactly the same way with no loss of quality. A human can get tired and make mistakes.
- Reduced wage bill: With fewer staff employed and no wages for the robots this is an area of major cost savings.

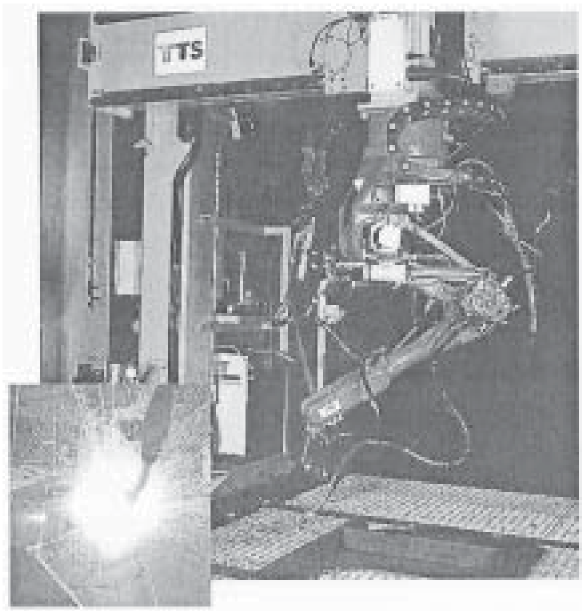
- Hostile environment: Here spray painting by a robot has many advantages. One of them is removing humans from the exposure to fumes. Many of the paints and coatings that are used contain harmful gases that are toxic to the human body.

  Off-shore oil production is conducted in an extremely hostile environment where robots are used to check rig platforms and weld joints.

Applications of robots

The vast majority of intelligent robots are used in manufacturing industries. In 1992 there were 800,000 industrial robots on the planet and this figure is estimate to rise to more than 1 million by 2004. Other uses are often related to working in remote or hostile environments.

Welding



In the mid-1990s welding was the number one application of industrial robots. The majority of these applications were in spot welding in a large number of automotive and truck industries, though arc-welding robots were steadily increasing. When spot welding and arc-welding are done manually, they present personal safety hazards. Also, when welding, one must wear protective gear that can be uncomfortable and sometimes very expensive. The main reason robots are used extensively in industrial welding is their productivity rate. On average robotic welders perform at the same rate as fifteen humans. This is due to the human need to adjust helmets, gas levels, and other safety equipment that robots do not need. Another benefit of using robots is that a skilled one can operate two to three robotic welders at one time. Besides the faster speed being achieved by robots the superior quality of the weld is easily visible.

Assembling

Robots are used to assemble watches, calculators, printers, circuit boards, electric motors, alternators and countless other products. They can be programmed to assemble anything that can be broken down into simple step-by-step instructions. The main problem with using a robot for assembly is in product design. Many older products were not designed to be assembled by a robot, so until they are redesigned, assembly will not be completely robotic. This is why an enormous amount of time is being spent in the research and development of everyday products that can be assembled by robots.

Military/police use

The Multipurpose Security and Surveillance Mission Platform (MSSMP) is designed to provide a rapidly deployable, extended-range surveillance capability for a range of operations and missions, including: fire control, force protection, tactical security, support for counterdrug and border patrol operations, signal/communications relays, detection and assessment of barriers (i.e. mine-fields, tank traps), remote assessment of suspected contaminated areas (chemical, biological and nuclear). The MSSMP system consists of three air-mobile remote sensing packages and a base station.

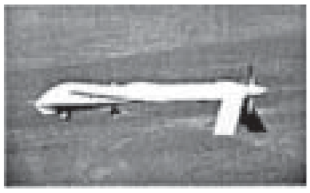
MSSMP system requirements include:

- high mobility
- remote operations over low-bandwidth tactical radio links
- long-endurance surveillance capabilities, and
- the ability for one operator to supervise several remote systems.

The flying robot has been used to fly down city streets, looking through upper- and lower-storey windows, providing lookout support ahead of advancing troops, and performing observations after landing on the roof of a two-storey building.

Acoustic and visual motion detection is used to detect, identify and locate targets of interest. Pre-programmed responses are activated upon detection and may include only an alert to the operator, or automatic transfer of a static image, laser range or an image stream.

Unmanned aircraft



During the second Gulf War in Iraq, in a widely reported news story, television viewers watched a video of Iraqi soldiers scouring the banks of Baghdad's Tigris River for what were reported to be downed American airmen and aircraft. In fact, the vehicles that had fallen to earth were unmanned drones, which the US Air Force had deliberately launched to gauge the strength of Iraqi air defences. After running out of power, the unmanned air vehicles had crashed, touching off the Iraqi militia's wild search for phantom pilots.

Now every branch of the US military is moving to produce or enhance unmanned air vehicles (UAVs) for reconnaissance, communications and combat. For fiscal year 2004 alone, the US Defense Department is requesting $1.39 billion to research or build novel air-borne drones. These robotic craft, it is anticipated, will be able to fly longer, fire more weapons, carry more sensors, and, increasingly, perform missions autonomously without human command or intervention.

Medicine

The goal is to avoid the tissue damage that comes with making a foot-long incision through the sternum and opening up the chest, which has been the typical procedure for open-heart surgery. With robot-assisted heart surgery, surgeons are able to operate through small incisions between the ribs. As medical science has gained an increasingly fine understanding of the human body, treatments have begun to emerge requiring surgery at the microscopic cellular level. Procedures conducted on the nerves, heart, brain and joints have now become commonplace through the application of telepresence robots in today's operating theatre. Robotics used for this purpose can guide a surgeon through tiny magnifiers to almost any area of the body while miniature grippers and manipulators cut away and suture affected tissue. The surgeon views and controls such operations through a computer monitor that in theory could be located many miles away.

Lawsuit against the surgical robot team

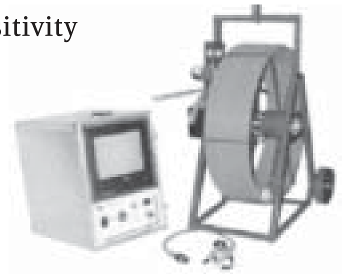
Following a number of successful operations the robot surgeons are being sued in court after a disastrous operation. In this case doctors used the da Vinci Surgical robot in an attempt to remove a cancerous kidney. Apparently the doctors were unable to complete the operation robotically and resorted to performing the surgery in the traditional fashion. Tragically, the patient died the next day. There were several mishaps during the surgery and the hours following it. Apparently the surgeon (using the robot) mistakenly nicked the patient's aorta and vena cava (two major blood vessels) and a vascular surgeon had to repair the vessels. Also, a post-operative x-ray showed that an absorbant pad had been left inside the patient thus requiring that the patient be re-opened to remove that pad (metallic tape is put on pads and sponges so that they will show in post-operative x-rays). A needle was also missing after the surgery and never accounted for, according to the lawsuit.

Pipe inspection

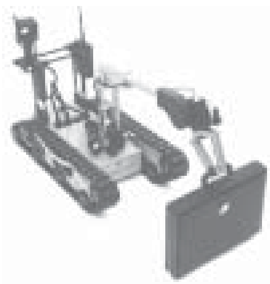
Robots are proving useful for routine pipe inspection in manufacturing and processing plants. When equipped with ultrasonic sensors or video transmitters, small robots can travel on or through pipes, ducts and conduits detecting fractures, corrosion and other problems. Their ability to reach inaccessible areas makes them useful for examination and cleaning operations or detecting the presence of dangerous gases and chemicals.

Specification for a pipe inspection system:

- small solid-state colour camera, low light sensitivity
- negotiates multiple bends
- compact and lightweight
- colour or black and white camera
- a 200-foot integrated pushrod or cable
- VCR (VHS) and video outputs.



Bomb disposal



Since the Second World War efforts have been made to invent devices to dispose of bombs safely. The advantages are obvious in terms of risk to life and the need for such automated machines has increased over the years with more dissident groups appearing across the globe.

Robots specifically designed for this purpose are available in many communities and are used routinely by law enforcement agencies. Their objective is to separate personnel from danger as they seek out and either detonate or relocate explosives to a more suitable location. These robots are generally telepresence machines controlled remotely by an operator stationed at a considerable distance.

Land-mine detection

Recent estimates suggest that over 110 million active land mines remain buried around the world, the majority of which end up injuring or killing civilians. About 50% of these victims are children. Clearing an area of land mines is dangerous, costly and very difficult. The development of robots capable of detecting land mines has become the object of considerable attention and research. The principal difficulties to overcome involve unpredictable terrain and different types of mines needing specific methods of detection. Olfactory sensors have been developed for this application allowing these robots to 'sniff out' the explosive agent.

Automated mail delivery

Large companies have to employ a team of staff to deliver mail each day throughout their building. Robot delivery systems are now being used to automate the process. They can follow a pre-determined path, stopping to deliver and pick up mail. The robots can also be programmed to control elevators and automatic doors.

Advantages

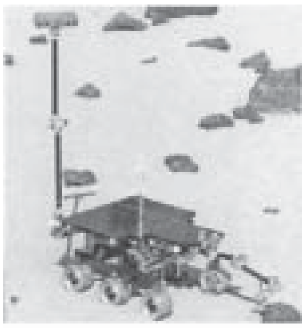
- Eliminates the need for mail centre staff to deliver mail manually, allowing them to concentrate on other areas
- Decreases the response time and travel time of internal mail
- Speeds up the delivery of internal and external mail through multiple delivery loops instead of once or twice per day.

Factory mobile robot

One particular use of robots on the factory floor is to pick up and deliver goods from one area of the plant to another. The robot carriers are equipped with proximity sensors and usually have both a flashing light and emit an annoying beeping sound. These are intended both to warn any humans of the presence of a robot and to avoid physical contact in case of injury.

The robots have a plan of the factory floor in their memory and are given instructions by a controlling computer. Some mobile robots have the ability to negotiate their way around an obstacle using sensors and artificial intelligence techniques.

Mars Explorer



Clearly, a robot which is being sent to another planet must have some form of independent decision-making capacity as it may encounter any number of unknown circumstances. In addition, the sheer distance from Earth would make a purely remote operation almost unworkable.

This robot is equipped with a number of artificial intelligence features.

A stereo vision system is used to supply images which are analysed and a map is created. This map is then passed to the obstacle-avoidance and path-planning modules. The obstacle-avoidance module sweeps arcs in front of the robot to analyse the terrain immediately in front of the robot while following the path created by the path-planning module. Data from both of these modules is passed to an ‘arbiter’ module which decides whether to continue on the current path or to change direction if there is a hazard ahead.

This is a state-of-the-art robot system which combines a vision/ perception system with artificial intelligence algorithms to navigate.

For further information and a look at the simulator try:

http://www.frc.ri.cmu.edu/projects/mars/

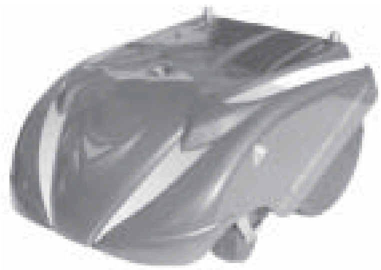
Vacuum robot

The image above is of a house-bound vacuum robot which can navigate a room avoiding obstacles while cleaning the carpet. The robot initially sweeps the room in a spiral pattern, and when an object is detected by the infrared sensors the guidance system plans a path to avoid collision. Once the job is finished, the robot will switch itself off.

Internal sensors can detect 'drop-offs' like the top of the stairs and the robot turns to avoid the hazard. When picked up, an internal sensor is activated and the robot switches itself off.

For more details try:
http://www.letsautomate.com/11020.cfm?

Lawn-mower robot (Mowbot)



This robot will mow the lawn. Mowbot is equipped with bump sensors which are activated when it encounters an object; it then moves in the opposite direction.

If the robot is accidentally turned over or lifted, the motor will shut down. There is also a rain sensor fitted.

The manufacturers claim that Mowbot is intelligent because it can detect when the power supply is low and navigate back to the recharging point where it will recharge itself.

Intelligent robots

All of the robots above have sensors and are involved in some sort of decision-making process. The fact that they all have sensors and a feedback mechanism is without doubt an improvement over the ‘dumb’ robots of the past, but are they ‘intelligent’?

Remember our three criteria of intelligence are the ability to:

1. communicate
2. retain knowledge
3. solve problems.

These robots satisfy some of the criteria to a limited extent, but few of them can truly be described as intelligent. Most of them are simply using an imperative program which responds to the input from the sensor and takes appropriate action.

An intelligent robot employs some aspect of artificial intelligence. For example, it could be equipped with:

- a vision system which can interpret images and then make a response;
- an artificial neural system which gives it the ability to learn;
- speech recognition which is able to respond to verbal commands.

Techniques like these are increasingly being applied to robots in all of the fields described in the previous section. This makes these robots more flexible and adaptable.

Complete Exercise 8 in Section 5

SECTION 3

Imagine a knowledge base for finding a fault in a car. A dialogue between the computer and the user might proceed as follows:

Computer	Does the engine turn when you engage the ignition?
User	*No.*
Computer	When you put the lights on do they go dim?
User	*Yes.*
Computer	When you turn the ignition is there a clicking sound?
User	*Yes.*
Computer	The starter motor in your car has jammed.

You can see that there is a decision-making process going on here and each question depends on the answer to the previous question. This type of process can be represented as a branching tree, more commonly called a **search tree**.

Many problems can be represented by a search tree. Various search techniques can be used to find the solution to the problem.

The search techniques apply to a search tree which contains all the different permutations or choices possible when starting from one state and trying to reach a 'goal' state. This might make a little more sense if we consider the following example.

We are faced with a number puzzle which contains the numbered tiles 1 to 5. In the initial state, the tiles are not in the correct order. Our objective or 'goal state' is to move the numbered tiles around the block so that the tiles read 1 to 5 in the correct order.

Initial state

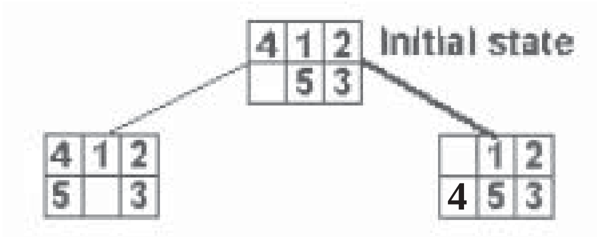
4	1	2
	5	3

Goal state

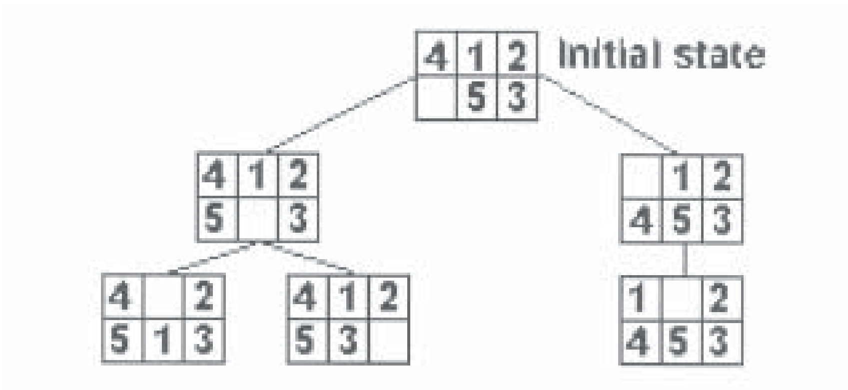
1	2	3
4	5	

It is possible to construct a search tree to address this problem.

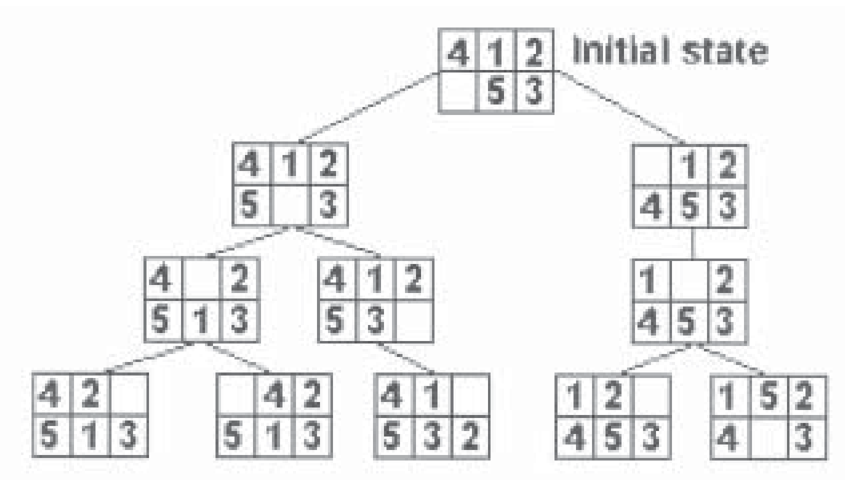
Top level Starting at the top of the tree we can make two moves; move the 5 into the blank or the 4 into the blank.



Level 2 From the left-hand branch we can make two new moves; move the 1 or the 3 into the blank. *(Moving the 5 into the blank has already been considered, it is the initial state.)* From the right-hand branch we have only one new move, i.e. move the 1 into the blank.



Level 3 From the leftmost branch: move the 2 into the blank or move the 4 into the blank.
From the inner-left branch: move the 2 into the blank.
Moving the 3 takes us back to Level 2 so can be ignored.

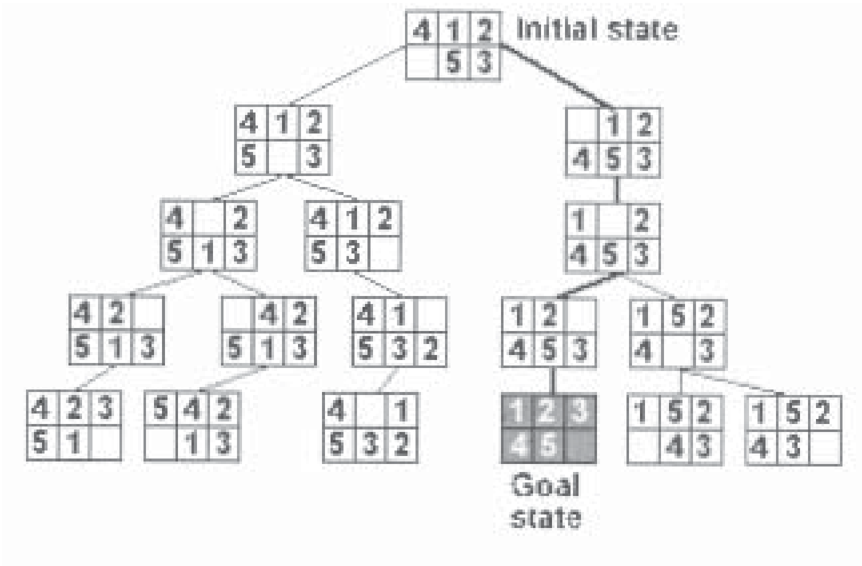


Level 4 The leftmost branch: move the 3 into the blank
The second left branch: move the 5 into the blank
The third left branch: move the 1 into the blank
The inner right branch: move the 3 into the blank (goal state)
The right branch: move the 4 into the blank or the 3 into the blank

As we have now reached the goal state, the search can stop.

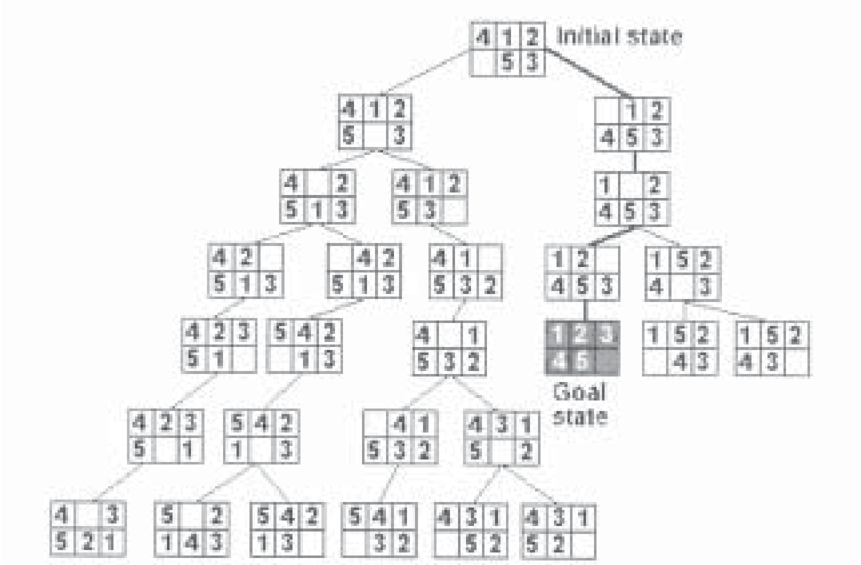
The full four levels can be seen on the next page.

Tile puzzle



The search tree above contains the moves from one node to the next until the ‘goal’ is reached. This is how many systems work to find a solution to a problem.

Below is an expansion of the tile problem search tree showing further but not all levels.



Most real life problems involve a much larger search tree and if every node was searched, it might take a great deal of time to reach the goal state, in fact a full search might be impractical. Consider chess where there are 10^{120} possible states; that’s more atoms than in the entire universe!

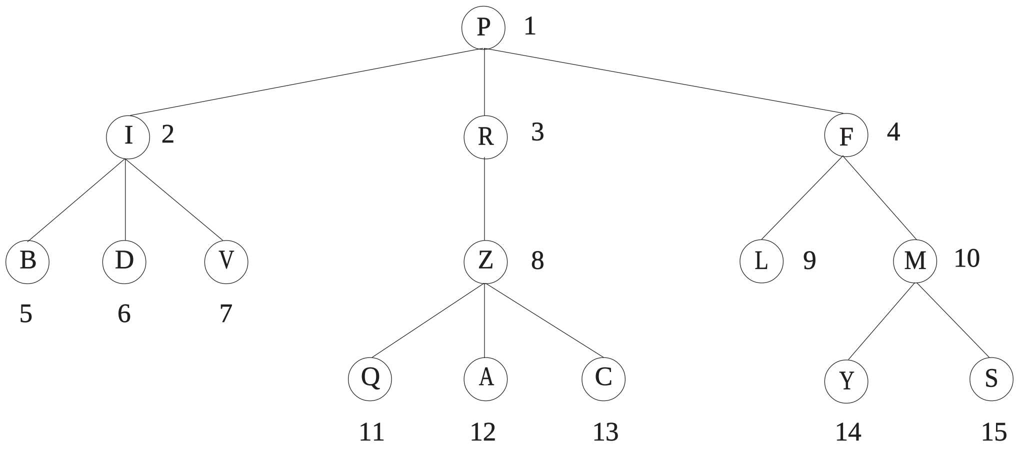
One of the areas of research in the field of artificial intelligence is aimed at finding ways of reducing the searching involved when the tree is just too large to find a solution. More of this topic is discussed in Higher Computing but we will restrict ourselves here to the two most common types of search.

Two search techniques are commonly used in these situations:

1. breadth first
2. depth first.

Breadth-first search

This is where the tree is searched from left to right working its way down layer by layer until the goal state is found.

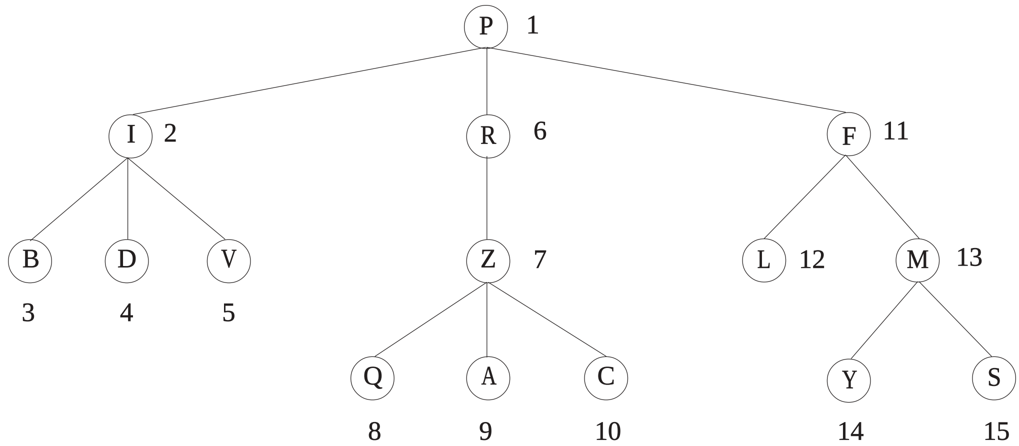


The numbers indicate the path of a breadth-first search. It starts at the top of the tree and traverses from left to right as it works its way down each level of the tree. When we examined the tile puzzle we conducted a breadth-first search as the path considered each permutation running from left to right and each level in turn.

The advantages of breadth-first searching are that the shallowest solution will be found first, i.e. the one nearest the top of the tree. If the goal state was at F then the path taken would be P→I→R→F. Because it is near the top it was found quickly.

Depth-first search

We consider the same tree as before but take a different route.



Again, the numbers indicate the route followed.
This time we have started at the top level and followed a path down the left side of the tree (P→I →B) until we reach the lowest point on the left, which is B. From node B we move back up the tree to the previous node, that is I, and then down to D.

Let's suppose the goal state is at node F as before. The path taken in a depth-first search is:

P→I →B→I→D→I→V→I→P→R→Z→Q→Z→A→Z→C→Z→R→P→F

This took a lot longer to find than the breadth-first search but that was simply because the goal state was near the top right of the tree. If the goal state is located far down the leftmost branch then depth-first search would reach the goal before breadth-first.

In the tile puzzle we could have searched down the leftmost branch first and it is possible that we would have reached the goal state, after all there is more than one path to solve the puzzle; the trick is to find the fastest path.

Consider how a chess player decides which move to make next. He does not calculate every path because he knows that some are clearly bad and will give him a poor or lost position. From past experience he can choose a path for the first several moves which he knows are good as he has played them before or some other top players have used them. Such strategies are referred to as ‘book moves’ in the world of chess as the paths have already been played at some point by grandmasters or other top players and are recorded in textbooks. What the chess player is really doing here is choosing a path down the search tree which he believes will lead him to his goal, to win.

Look back at the search tree for the tile puzzle. Which search techniques would arrive at the goal state in the shortest time?

Complete Exercise 9 in Section 5

SECTION 4

The programming languages which you have used will be based on a set of instructions that are executed in order by the computer. They include BASIC, COMAL, True BASIC, Pascal and many others. Collectively these are called **imperative** languages.

The language that we will examine in this section is called Prolog and it is very different from an imperative language.

Prolog (**Pro**gramming in **Log**ic) is a language used mainly by the artificial intelligence community, e.g. natural language processing, knowledge representation, expert systems, machine learning, and increasingly in software engineering for program specification. Prolog works in a two-valued (yes/no, true/false) logic where every query given to the program database either succeeds or fails.

Prolog is the main alternative to LISP as an AI programming language and exists in several dialects, the Edinburgh (or DEC-10) syntax being the most widespread. Programming in Prolog is very different to programming in imperative languages such as BASIC and C and emphasises problem description over the low-level mechanics of how to solve a particular problem. Prolog is described as a **declarative** language.

Introduction

Prolog programs are made up of clauses, which in plain terms mean a ‘fact’. This fact can look like any of the following:

Christine likes Michael	Mark is happy	Albert is married to Ann
Claire is a student	A car has wheels	Alison is Peter’s mother
The above clauses can be written in Prolog as:		
likes(christine, michael).	is_happy(mark).	is_married_to(albert, ann).
is_a_student(claire).	has_wheels(car).	mother(alison, peter).

It is important to understand the order in which the facts are read. From the above examples you can see that the 'verb' or link term is placed outside the brackets and the objects to which it applies are inside. Also the first object is applied by the link to the second (if there is one). In other words we would read the following fact…

is_good_at(mark, tennis) *mark is good at tennis*

There are a few syntax rules which you have to observe when writing facts in Prolog.

1. Spaces are not allowed, particularly in front of the brackets. Instead of a space the underscore is often used.
2. All facts are written in lower case. Upper case is reserved for variables.
3. You must finish each line with a full stop.

When you enter lines of Prolog into the computer most of the errors which you encounter are likely to be related to the three rules above.

We cannot assume facts which are not present even though they may seem obvious to us, for example:

is_married_to(albert, ann)

Clearly this means that Albert is married to Ann and common sense tells us that Ann is married to Albert but this is **not** true in Prolog. We would need another fact to tell us that Ann is married to Albert.

A free version of Prolog is available from:

http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/lang/Prolog/impl/Prolog/0.html

Example 1: Diary

A new student arrives at school and she does not know anyone. She has to learn the names of all of her teachers; the subjects they teach; the room where each lesson is taught; and, of course, details about any new classmates.

She makes notes in her diary during the first week.

My first class was English with Mr Brooks, we were studying Shakespeare's Macbeth and the homework is to read over the second act. The girl sitting next to me was called Gemma, she likes playing tennis and swimming and her favourite group is 'Busted'. Her mobile is 07887034. Have to leave room M1 and go to Science room 2 for my next lesson.

Mrs Crawford is the science teacher. She taught us about microbes and germs – yuk! I don't like this subject but I did meet a new friend called Alison. Her phone number is 07089087. She likes playing basketball, tennis and listens to hip-hop. I'm off to Mathematics next in room M1 with Mr Hopfield….not looking forward to this. Nearly forgot my homework! Page 11 of the text book.

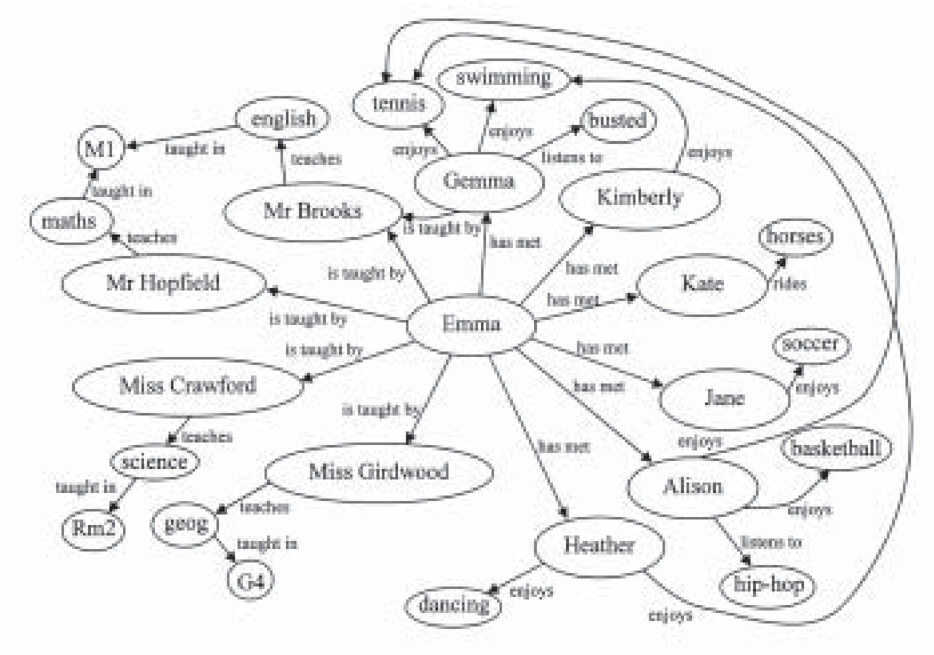
Well it turned out Maths wasn't that bad, we did quadratic equations (my favourite) and were given Ex 3 to finish for Homework. I met up with Gemma at the interval and she introduced me to some new friends; Jane who is the captain of the girls' soccer team, mobile number 07886534; Heather, she enjoys dancing and tennis; Kimberly, she likes swimming and her mobile is 07676544; Kate goes horse riding and her mobile is 01657778.

Just met my new Geography Teacher, Miss Girdwood, the lesson was about glaciers and I have to do page 34 of the text book for homework. We were in G4.

from Emma McColl's diary

Semantic nets

The extract from Emma's diary can be represented in a visual form called a semantic net. This is a convenient means of displaying the data and the relationships. The semantic net below has been developed from *some* of the data in Emma's diary.



You will notice that I have not included the mobile phone numbers, the class topic or the homework data.

In Prolog the above semantic net would be coded as the following **knowledge base**:

```
is_taught_by(emma,mr_brooks).
is_taught_by(emma,mr_hopfield).
is_taught_by(emma,miss_crawford).
is_taught_by(emma,miss_girdwood).
is_taught_by(gemma,mr_brooks).
teaches(mr_brooks,english).
teaches(mr_hopfield,maths).
teaches(miss_crawford,science).
teaches(miss_girdwood,geography).

has_met(emma,gemma).
has_met(emma,kimberly).
has_met(emma,kate).
has_met(emma,jane).
has_met(emma,alison).
has_met(emma,heather).

listens_to(alison,hip_hop).
listens_to(gemma,busted).
taught_in(science,rm2).
taught_in(geography,g4).
taught_in(maths,m1).
taught_in(english,m1).
enjoys(gemma,tennis).
enjoys(kimberly,swimming).
enjoys(jane,soccer).
enjoys(alison,basketball).
enjoys(heather,dancing).
enjoys(gemma,swimming).
enjoys(heather,tennis).
enjoys(alison,tennis).
rides_horses(kate).
```

Enter the knowledge base into Prolog, save it as 'EmmaDiary'. Compile it and if there is an error, check the syntax for mistakes.

Queries

A query is when the user asks questions of the knowledge base. The most basic query is to check a fact in the knowledge base which has a single argument, i.e. one with only a single term in the brackets. In the knowledge base you may have noticed that there is only one fact with a single argument, rides_horses(kate). I could also have written this as rides(kate,horses).

If we were to include the gender of each person in the knowledge base then those facts could all be written as:

```
female(emma).
female(gemma).
female(kate).
male(mr_hopfield).
```

We will test our knowledge base to see if Kate rides horses. With the knowledge base loaded into Prolog...

Enter this query into your computer.

```
? rides_horses(kate).
yes
```

In some versions of Prolog, you enter a query with a ? as shown, in others you enter it in a dialogue box.

The knowledge base has been searched and a match was found so the reply *true* or *yes* is sent back to the user.

Try a few more queries:

```
? has_met(emma,kate).
? is_taught_by(emma,miss_girdwood).
? enjoys(kimberly,swimming).
? listens_to(gemma,busted).
```

Let's try a few which should result in a negative response. That's *false* or *no* in Prolog.

? has_met(kate, emma).
? enjoys(kimberly,soccer).
? listens_to(alison,busted).

Once again, you should see that the fact that Emma has met Kate does not (in Prolog) mean that Kate has met Emma.

Variables

Asking the knowledge base simple queries like the examples above isn't really very productive. The information becomes a little more useful when we start to use variables.

Suppose we wish to find all of the people who like swimming. This time we enter the query:

? enjoys(X,swimming).

In this query we have used an 'X' instead of someone's name. This is a variable in Prolog and it either is a capital letter or starts with a capital letter. Examples of variables include:

X, Y, Z, Who, Person, Find

Back to the query...enter the query above. The reply should be

X= kimberly
X= gemma

Let's try a few more.

We want to find all of the teachers who teach Emma. This time the variable will be the other argument in the clause, i.e.

? is_taught_by(emma,X).

The answer should be

X= mr_brooks, X= mr_copfield, X= miss_crawford, X= miss_girdwood

The order the answers appear in is the same order as they were entered into the knowledge base. This is because Prolog searches the knowledge base from top to bottom.

Task 1: Entering queries

Write down your expected answers to the following queries:

1. ? enjoys(X, tennis).
2. ? enjoys(alison, X).
3. ? is_taught_by(X, mr_brooks).
4. ? taught_in(X, m1).
5. ? has_met(emma,X).
6. ? enjoys(heather,What).
7. ? enjoys(kimberly,What).
8. ? taught_in(Subject, g4).
9. ? listens_to(Who, hip_hop).
10. ? teaches(miss_crawford, maths).
11. ? taught_in(english, rm2).
12. ? taught_in(geography, Room).

Now enter each query into Prolog, to check your answers.

Arithmetical operations

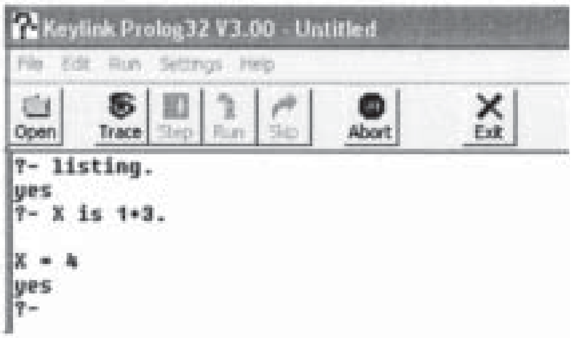
Prolog is a computer language developed for symbolic computation rather than basic arithmetic. However, it is possible to 'ask' Prolog some arithmetic questions directly and it will provide an answer.

The basic operators used in Prolog are restricted to the following:

+ Addition
– Subtraction
* Multiplication
/ Division

Start a new knowledge base. Enter **X is 1+3.**
(Remember the full stop.)

Prolog will evaluate the sum and give the answer as 4.



Enter the sums below into Prolog:

1. X is 5+7.
2. Y is 12–5.
3. Number is 3*6.
4. Z is 3*4+2*6 – 12/4.

It is possible to use variables such as X and Y to work out the answer to some more complex sums. Enter the line below.

X is 2, Y is 5, W is X+Y.

We can also use the operators > (greater than) and < (less than). These can be used to compare values to each other, e.g.

? 5 > 4.

Prolog checks to see if indeed 5 is greater than 4 and responds with 'yes'.

Use Prolog to work out the answers to the following sums.

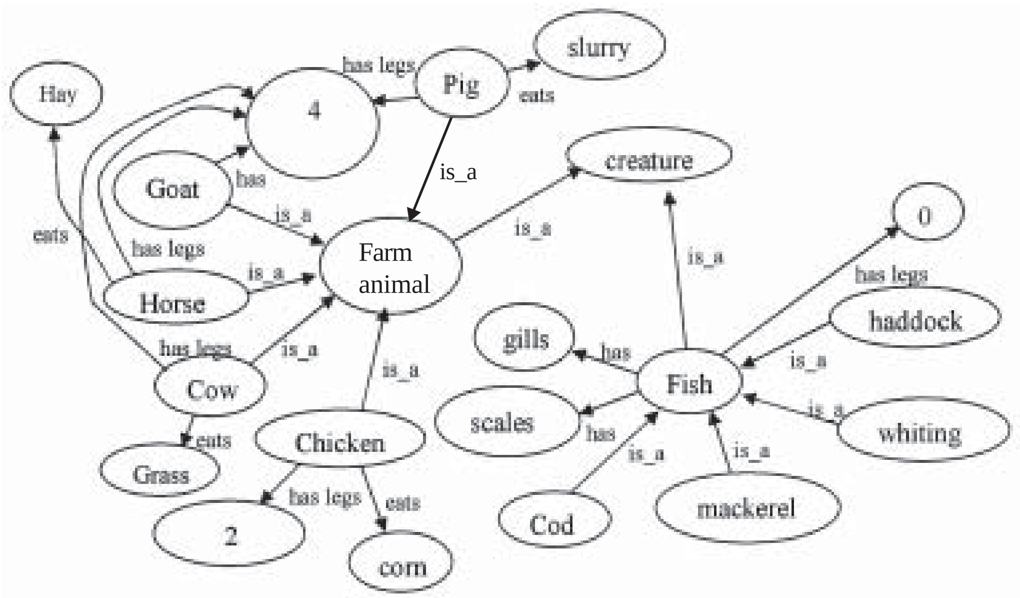
1. thirty seven plus twenty five
2. eighteen minus 6
3. two times 12 plus three times seven
4. four times six minus three times two
5. Is five times fifteen greater than 6 times sixteen?
6. Is six times eleven less than five times thirteen?
7. If X is fifteen and Y is fourteen and W is X times Y, what is W?
8. If X is twenty and Y is eleven and W is X times Y minus 20, what is W?

Example 2: Creatures

The following information is about creatures.

A pig has four legs, eats slurry, lives on a farm. Cows have four legs, live on a farm, eat grass. Chickens have two legs, live on a farm, eat corn. Horses have four legs, eat hay, live on a farm. Goats have four legs, live on farm. Cod, mackerel, whiting and haddock are fish, all fish have scales and gills.

Semantic net



The knowledge base would be:

```
is_a(fish, creature).
is_a(farm_animal, creature).
is_a(cod, fish).
is_a(haddock, fish).
is_a(whiting, fish).
is_a(mackerel, fish).
is_a(cow, farm_animal).
is_a(pig, farm_animal).
is_a(horse, farm_animal).
is_a(chicken, farm_animal).
is_a(goat, farm_animal).
has_legs(goat,4).
has_legs(pig,4).
eats(chicken, corn).
eats(cow,grass).
eats(horse,hay).
eats(pig,slurry).
has(fish,gills).
has(fish,scales).
has_legs(chicken,2).
has_legs(horse,4).
has_legs(cow,4).
```

Enter the facts on the left into a new knowledge base.

Try the following queries:

```
? has_legs(chicken,2).
? has_legs(X,4).
? is_a(Y,farm_animal).
? eats(horse, X).
```

Rules

Searching the knowledge base is fairly easy and there is a limit to the amount of useful information which can be extracted. Queries are fine for checking the knowledge base but you can't do much more. We need rules to create more relationships.

An example of a rule is

```
liveinsea(X) :-
is_a(X,fish).
```

This means X lives in the sea IF X is a fish.

> The :- part means **IF** in Prolog. Rules are made up of a **head** which is the conclusion and a **body** which is made up of one or more **sub-goals**. In order for the condition to be true all of the sub-goals must be true.

This is much more efficient than writing out a fact for each of the fish. Add the above rule to your knowledge base and let's test it. Now recompile the knowledge base. Try the following queries:

1. ? liveinsea(cod).
2. ? liveinsea(haddock).
3. ? liveinsea(X).

Now write a similar rule which states that a farm animal lives on a farm. (Seems obvious to us but remember Prolog has to be told.)

```
liveonfarm(X):-
is_a(X, farm_animal).
```

Add this to the knowledge base, recompile it and test it with the following queries:

1. ? liveonfarm(cow).
2. ? liveonfarm(horse).
3. ? liveonfarm(X).

Task 2: Simple rules

Now that you are familiar with writing rules, add rules for each of the following and check them with suitable queries.

- A fish has scales
- A fish has gills

We still have to create a rule to say that all farm animals are creatures. Try this yourself now. Test your new rule on each of the farm animals.

In Prolog we can have more than one rule with the same name but with different sub-goals. This may seem strange but our 'creatures' knowledge base needs just that.

A moment ago you were asked to create a rule to say all farm animals are creatures. You should have something like this:

```
is_a_creature(X) :-
liveonfarm(X).
```

This rule works fine for all of the farm animals but it's not much use to a fish! We need another rule for the creatures that live in the sea.

```
is_a_creature(X):-
liveinsea(X).
```

Add this new rule to the knowledge base, recompile, and test it with the following queries:

1. ? is_a_creature(cod).
2. ? is_a_creature(haddock).
3. ? is_a_creature(mackerel).

It's worth while taking a closer step-by-step look at how Prolog searches through the knowledge base to find a goal or a sub-goal. This is called a **trace**.

Trace

When doing a manual trace it helps to have the knowledge base nearby. The listing of the knowledge base is missing some lines for simplicity.

We will trace the query **? is_a_creature(cod).**
Prolog starts at the top of the KB and works through each line trying to find a match.

? is_a_creature(cod).	**Comments**
27 is_a_creature(X):-	X=cod
28 liveonfarm(X).	sub-goal for rule line 27
?liveonfarm(cod).	search for a matching sub-goal 28
21 liveonfarm(X):-	match on line 21
22 is_a(X,farm_animal).	sub-goal to rule 21 (X=cod)
?is_a(cod,farm_animal).	search for a matching sub-goal 22
FAIL	there is no match

The sub-goal 22 fails so the rule on line 21 fails.
So sub-goal 28 for rule 27 fails so rule 27 fails. We are back to the original query.

? is_a_creature(cod).	**X=cod**
29 is_a_creature(X):-	match on line 29
30 liveinsea(X).	sub-goal for rule line 30
?liveinsea(cod)	search KB for a matching sub-goal 30
25 liveinsea(X):-	match on line 25
26 is_a(X,fish).	sub-goal for rule line 26
?is_a(cod,fish).	search for matching sub-goal
2 is_a(cod,fish).	match on line 2
SUCCESS	on sub-goal line 26
so SUCCESS	on sub-goal line 30
so SUCCESS	on query
Yes	

The trace is useful because it provides an understanding of how rules work. **Each sub-goal of the rule must be successful for the rule to be successful.** The technique used above to trace the query should be practised. Once mastered it can be applied to any query of any knowledge base.

1. is_a(haddock, fish).
2. is_a(cod, fish).
3. is_a(whiting, fish).
4. is_a(mackerel, fish).
5. is_a(cow, farm_animal).
6. is_a(pig, farm_animal).
7. is_a(horse, farm_animal).
8. is_a(chicken, farm_animal).
9. is_a(goat, farm_animal).
10. has_legs(goat,4).
11. has_legs(pig,4).
12. eats(chicken, corn).
13. eats(cow,grass).
14. eats(horse,hay).
15. eats(pig,slurry).
16. has_legs(chicken,2).
17. has_legs(horse,4).
18. has_legs(cow,4).

19. has_scales(X):-
20. is_a(X,fish)

21. liveonfarm(X):-
22. is_a(X, farm_animal).

23. has_gills(X):-
24. is_a(X,fish).

25. liveinsea(X) :-
26. is_a(X,fish).

27. is_a_creature(X):-
28. liveonfarm(X).

29. is_a_creature(X):-
30. liveinsea(X).

Let's trace the query: **? has_scales(X)**

? has_scales(X)	**Comments**
19 has_scales(X):-	match on line 19
20 is_a(X,fish)	sub-goal for rule line 19
?is_a(X,fish)	search for a match
1 is_a(haddock, fish).	match on line 1 sub-goal succeeds
rule SUCCESS	line 19 succeeds
X=haddock	
?is_a(X,fish)	back to check for more
2 is_a(cod, fish).	match on line 2 sub-goal succeeds
rule SUCCESS	line 19 succeeds
X=cod	
?is_a(X,fish)	back to check for more
3 is_a(whiting, fish).	match on line 2 sub-goal succeeds
rule SUCCESS	line 19 succeeds
X=whiting	
?is_a(X,fish)	back to check for more
4 is_a(mackerel, fish).	match on line 2 sub-goal succeeds
rule SUCCESS	line 19 succeeds
X=mackerel	
?is_a(X,fish)	back to check for more
FAIL	no more matches
No	

```
1.  is_a(haddock, fish).
2.  is_a(cod, fish).
3.  is_a(whiting, fish).
4.  is_a(mackerel, fish).
5.  is_a(cow, farm_animal).
6.  is_a(pig, farm_animal).
7.  is_a(horse, farm_animal).
8.  is_a(chicken, farm_animal).
9.  is_a(goat, farm_animal).
10. has_legs(goat,4).
11. has_legs(pig,4).
12. eats(chicken, corn).
13. eats(cow,grass).
14. eats(horse,hay).
15. eats(pig,slurry).
16. has_legs(chicken,2).
17. has_legs(horse,4).
18. has_legs(cow,4).

19. has_scales(X):-
20. is_a(X,fish)

21. liveonfarm(X):-
22. is_a(X, farm_animal).

23. has_gills(X):-
24. is_a(X,fish).

25. liveinsea(X) :-
26. is_a(X,fish).

27. is_a_creature(X):-
28. liveonfarm(X).

29. is_a_creature(X):-
30. liveinsea(X).
```

Try writing a trace using the same method as above for the following queries:

1. ? eats(X, slurry).
2. ? eats(cow, X).
3. ? is_a_creature(pig).
4. ? has_gills(haddock).
5. ? is_a(X, farm_animal).
6. ? is_a(X, fish).
7. ? liveonfarm(X).

You should now be an expert at performing a trace!

Now let's try a few queries to test the knowledge base. Enter the following:

1. ? eats(pig,X).
2. ? eats(horse, What).
3. ? eats(Animal, grass).

Task 3: Writing queries with AND

Enter queries which will answer the following questions:

4. Which animals are farm animals?
5. List the fish in the knowledge base.
6. Which animal eats corn?
7. Which creature lives on a farm AND eats hay?

The last question requires you to combine two queries together using the operator AND. You should have written:

? is_a_creature(X) **,** eats(X,hay). (*The comma is used for* AND)

Prolog allows other 'operators' to be used in queries. You will be familiar with them from your mathematics class. > is for 'greater than' and < is for 'less than'. We can use them when numbers are involved. We will look further at operators in the next example but first let's practise using the AND operator.

Use AND to find answers to the following:

8. which creatures live on a farm AND have 2 legs?
9. which creature eats slurry AND has four legs?
10. what creature eats corn AND has two legs?
11. what creatures live in the sea AND have gills?

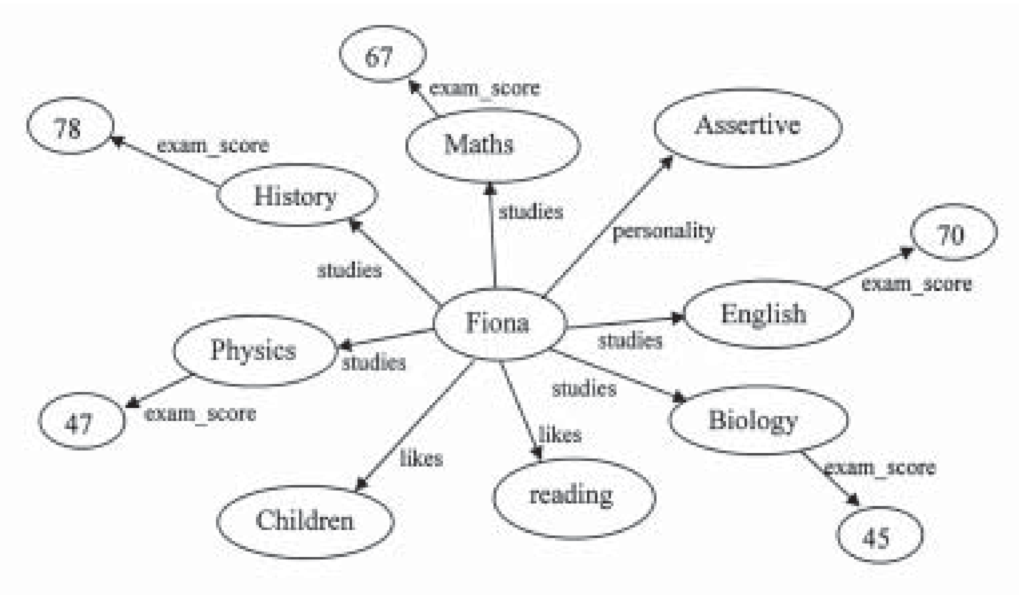
There is a limited number of meaningful queries we can do on this knowledge base. Time for a new example!

Example 3: Career choice

This exercise is based on the subjects and marks which a student has achieved in her exams. The purpose of the knowledge base is to provide her with career advice. Here is written transcript of her details:

Fiona studies English, Maths, History, Biology and Physics. Her scores in the exams for each subject are: English 70, Maths 67, History 78, Biology 45, Physics 47. Fiona likes reading and has quite an assertive personality. She also likes working with children.

Your first task in this exercise is to create a knowledge base from the semantic net shown below.



Task 4

Write out the facts and relationships on paper for the semantic net above. The best way to make sure you haven't missed anything is to tick each arrow on the semantic net as you write a clause to represent it in Prolog. When you have finished writing it out, show it to your tutor.

Once your tutor is satisfied with the knowledge base you will be told to enter it into Prolog. Make sure you correct any errors which may appear, e.g. forgetting to put the full stop at the end of the line.

Save the knowledge base and call it 'Careers'.

Rules

If you are thinking of a career in primary teaching then you have to have scored at least 68 in the English exam. We need to write a rule to check the English exam score.

```
primary_teacher(Fiona):-
exam_score(english, X),
X>68,
nl.          (the nl means take a new line)
```

This rule combines some of the ideas from the earlier examples. Reading from the top line the rule means Fiona could be a primary teacher if her English exam score is greater than 68. Remember that the commas represent AND.

Add this rule to your knowledge base and test that it works. Is Fiona suited to be a primary teacher?

Task 5

- Write a rule for going to college to study Social Science. To be eligible for this you need to have scored more than 70 in the History exam.

  Enter the rule and find out if Fiona is suited to this course. Show this to your tutor.

- To train as a nursery nurse the college is looking for people who like reading and children. Write a rule for them.

  Add the rule and test it to see if Fiona is a suitable candidate.

- To pass the English exam you must have over 50 marks. Write a rule which will check Fiona's English score and report if she has passed the exam.
  Call the rule: **passed_english(Fiona).**

  Add this to the knowledge base and show this to your tutor.

- Now write similar rules for each of the other subjects.

  Add the rules and test them to see which subjects Fiona passed.

- The police are looking for officers who are assertive and have passed in English. Use the previous rule which tests if Fiona passed English applying this new rule.

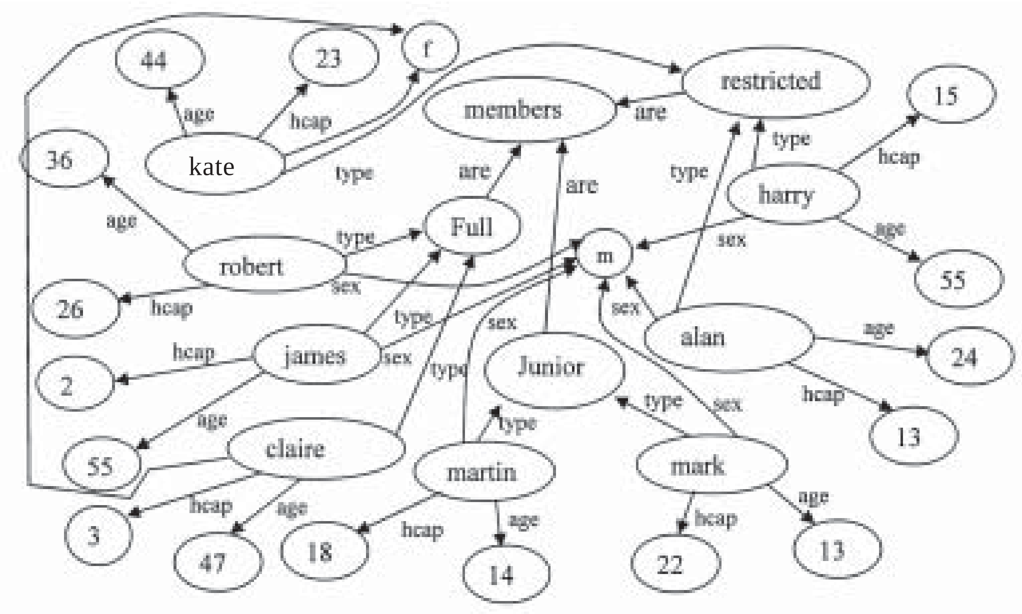
  Add this rule and find out if Fiona is a suitable candidate to join the police.

- To apply for a college course in Science Fiona has to have passed English and Biology. Will she get in?

Example 4: Golf club

The Mearnskirk Golf Club has different types of membership depending on age and access times. Here are some members' details.

Kate is 44 and has a handicap of 23; she has restricted membership. Martin is a junior member and is 14 with a handicap of 18. Alan has a handicap of 13 and has restricted membership; he is 24. James is 55 with full membership and a handicap of 2. Harry has restricted membership, is 55 and has a handicap of 15. Robert has a handicap of 26, is 36 years old and has full membership. Mark is 13, has junior membership with a handicap of 22.



Complete the knowledge base for the golf club. Get you tutor to check the knowledge base before going any further.

Now test the knowledge base with a few queries.

Task 6: Writing queries

1. write a query to check if James is 55.
2. write a query to check if Alan is a junior.
3. write a query to list all junior members.
4. write a query which will list all restricted members.
5. write a query to list all of the members with full membership.
6. write a query which will produce a list of all females.

Now for a few complex queries:

7. write a query which will find restricted members who are male.
8. write a query which will list all restricted members who are female.

Show your tutor each of the above queries.

More rules

Junior members have to become seniors when they reach the age of 16. We would like a rule which will find all members who are currently 15 so that a letter can be sent out this year to inform them that their membership type will change next year.

- Write a **rule** which will identify all members who are currently 15. Name this rule **last_year_junior(X)**. Add this to the knowledge base and test that it works. Show this to your tutor.
- In Prolog enter the query **? last_year_junior(X)**.

The ladies' medal is open to all ladies who hold full membership. To find all those entitled to enter the medal create the following rule:

```
ladies_medal(X):-
female(X),
type(X,full).
```

In Prolog enter the query **? ladies_medal(X)**.

The result should show that only Claire is eligible to enter (I think she might win!). Note how the comma was used in the rule as AND.

Task 7: Rules and Trace

- There is a competition for all the gentlemen who have restricted membership. Write a rule which will find the people who can enter the competition. Add this to the knowledge base and test that it works.

- Do a manual trace of the above rule and show it to your tutor.

Example 5: The band

The details given below relate to a school band, its members and where its concerts are performed.

The school band's conductor is Mr Anderson, who is the principal teacher of Music. The members of the band include Peter who plays the trumpet and is learning the piano; Mary who plays the clarinet and the flute; Michael plays the trombone; Jean plays the guitar and the violin; Ian plays the violin; John plays the flute. The band performs in the school, in the town hall and the local community centre.

Knowledge base:

```
in_the_band(peter).
in_the_band(mary).
in_the_band(michael).
in_the_band(ian).
in_the_band(john).
in_the_band(jean).

plays(peter,trumpet).
learning(peter,piano).
plays(mary,clarinet).
plays(mary,flute).
plays(michael,trombone).
plays(jean,guitar).
plays(jean,violin).
plays(ian violin).
plays(john,flute).

performs(band,school).
performs(band,town_hall).
performs(band,community_centre).

conductor(band,mr_anderson).
```

Task 8
Your task is to create a semantic net from the knowledge base above.

Show this to your tutor.

Task 9
There is a place for a violin player in the county quartet. Write a rule called:

attend_audition(X)

The conditions of this rule are that the person must play in the band and play the violin. Test the rule to see if it works.

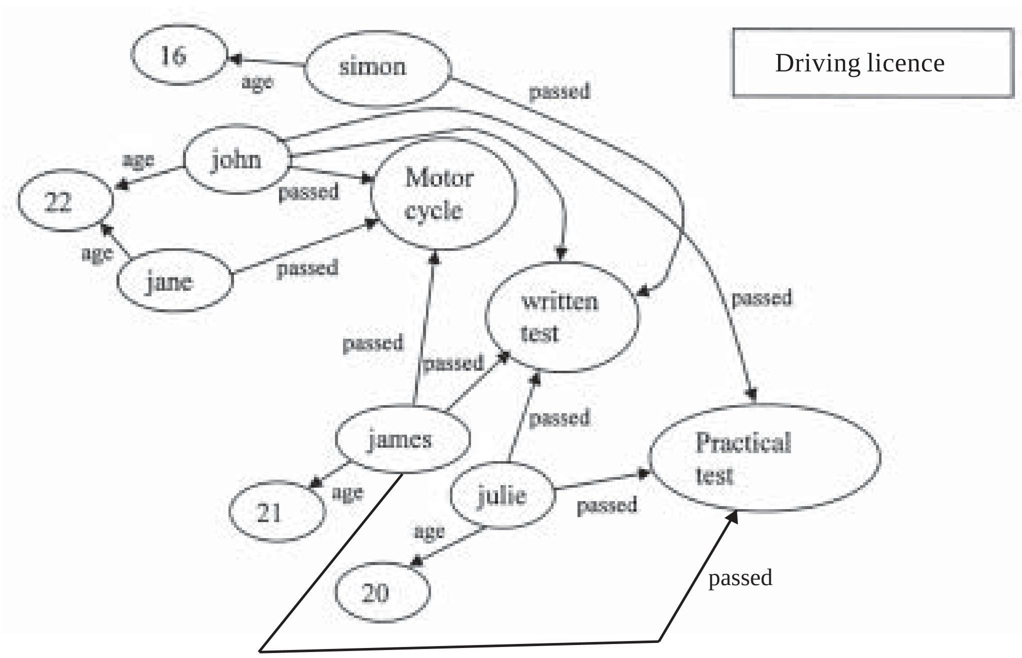
Practical exercise : Vehicle licence

Note for candidates
This exercise may be used to test Outcome 2 of the AI unit. The work submitted should be entirely your own. You may use any notes, books or online help while working on this task but you must not elicit help from others. Your tutor will observe you working on the tasks to ensure that you demonstrate the appropriate skills required to pass the outcome.

This knowledge base is about those people who have passed part or all of the requirement needed to get a driving licence.

Simon is 16 and has passed the written test but not the practical. John has passed the motor-cycle test and has passed both written and practical tests for a car; he is 22. Jane is also 22 years of age and she has only passed the motor-cycle test. James has passed the written and practical for a car and has also passed the motor-cycle test; he is 21. Julie is 20 and has passed both written and practical tests for a car.

Semantic net



Construction of a knowledge base

Write out the facts using the semantic net as a guide. Do not do this on the computer, but write it down on paper. Make sure you have a Prolog clause for every arrow on the net.

Show this to your tutor.

Once your tutor has approved your knowledge base enter it into Prolog and save it as 'Licence'.

Creating a rule

If someone has passed the written and practical tests for a car then they can drive a car. Write down a rule called:

can_drive_car(X):-

which has two sub-goals that can be used to find those people who can drive a car. Add this rule to your knowledge base.

Creation of queries and testing

Now write down on paper suitable queries which will test that the knowledge base is working. Write them from the following statements:

1. Is Julie 20 years of age?
2. Has James passed the written test?
3. Find all people who can drive a motorcycle.
4. List all people who have passed their written test for a car.
5. List all people who are 22 years of age.
6. List all those people who can drive a car.
7. List all those people who can drive a car and a motorcycle.

Enter each of your queries and test that the knowledge base gives the correct responses.

Obtain a **printed listing** of your knowledge base and give it to your tutor along with the **written queries**.

SECTION 5

Exercise 1: Intelligence

1. Why is it difficult to define intelligence?

2. What is an intelligence quotient (IQ)?

3. What criticisms have been expressed about IQ tests?

4. What is the name of the approach to intelligence which involves examining how people act?

5. Name three accepted indications of intelligence.

6. When defining artificial intelligence what is it often compared to?

7. Describe the Turing Test.

8. Why do you think the Turing Test has never been passed?

9. Why did early attempts at artificial intelligence involve game playing?

10. Why are computers good at playing simple games such as OXO?

11. When was the first time that a computer beat a world champion and at what game?

12. What game-playing event raised the profile of artificial intelligence?

13. What three characteristics do all games-playing programs have?

Exercise 2: Chatterbots

1. How does Eliza ‘appear’ to hold a conversation with the user?

2. What does Eliza do when it cannot find a suitable response?

   (i) ____

   (ii) ____

   (iii) ____

3. Why does Eliza not display intelligence?

4. What does a chatterbot attempt to do?

5. Where would you find chatterbots?

6. Name three commercial applications of chatterbots

   (i) ____

   (ii) ____

   (iii) ____

7. What feature of Alice makes it better at holding a conversation than Eliza?

8. Describe how an Internet search bot works.

9. Describe how a shopping bot operates.

10. What is the name of the annual prize awarded to the program which comes nearest to passing the Turing Test?

Exercise 3: Expert systems

1. Why did researchers have problems using conventional methods when trying to develop AI programs?

2. Name two hardware advances that aided the developments of AI.

   (i) ____

   (ii) ____

3. Explain what expert systems do.

4. List three advantages of using expert systems.

   (i) ____

   (ii) ____

   (iii) ____

5. Why are expert systems more reliable than human experts?

6. How have expert systems improved the safety functions of power stations?

7. Why do some people feel too much money is being spent on AI research?

8. Describe a medical expert system.

9. Describe two potential drawbacks of expert systems.

   (i) ____

   ____

   (ii) ____

   ____

Exercise 4: Artificial neural systems

1. List three characteristics of ANSs.

   (i) ______________________________

   (ii) ______________________________

   (iii) ______________________________

2. How do artificial neural systems learn?

3. Name three commercial uses of ANSs.

   (i) ______________________________

   (ii) ______________________________

   (iii) ______________________________

4. Name two specific tasks where ANSs are used.

   (i) ______________________________

   (ii) ______________________________

5. What problems occur with debt risk assessment if an ANS is used?

6. Why are credit companies now using ANSs to decide whether to offer their credit card to people?

7. Why have ANSs proved to be popular with stockbrokers?

8. Why is the postal service keen to have ANSs which are designed to recognise poor handwriting.

Exercise 5: Vision systems

1. How is an image captured in vision systems?

2. What feature of vision systems enables them to recognise images?

3. How are artificial neural systems used with vision systems?

4. What advantages do vision systems bring to the production line?

5. Describe a military use of vision systems.

6. How are vision systems used with expert systems in the analysis of satellite images?

7. How can vision systems combine with expert systems to help in the field of medicine?

Exercise 6: Natural language processing/speech recognition

1. What are the two main areas of NLP?

   (i) ______________________________

   (ii) ______________________________

2. Explain the main goal of NLP.

3. What must you do before using speech recognition and roughly how long will it take?

4. Why do we have to go through the process in question 3?

5. Name two important factors affecting accuracy when using speech recognition.

   (i) ______________________________

   (ii) ______________________________

6. List four recommendations when speaking into the microphone.

   (i) ______________________________

   (ii) ______________________________

   (iii) ______________________________

   (iv) ______________________________

7. Explain the difference between *continuous* and *discrete* SR.

8. What piece of advice would you give to someone using SR to ensure good results?

9. How does the SR program let the user know about words which are suspect?

10. Explain the difference between using SR to create a text document and using SR for command and control purposes.

11. Give an example of a 'command and control' usage.

Exercise 7: Applications of speech and handwriting recognition

1. Explain how SR can be used to help the disabled.

2. List three military uses of speech recognition.

   (i) ____

   (ii) ____

   (iii) ____

3. Explain how SR technology can be used to allow people who speak different languages to communicate.

4. Name one advantage of using handwriting technology rather than a conventional input method.

5. List two tips which will improve accuracy when using handwriting recognition.

   (i) ____

   (ii) ____

6. Name three commercial applications of handwriting recognition.

   (i) ____

   (ii) ____

   (iii) ____

7. Why are companies interested in developing natural language processing programs?

8. Describe two writing techniques which can improve the accuracy of handwriting recognition systems.

   (i) ____

   (ii) ____

9. Suggest a possible use for tablet PCs.

Exercise 8: Robots

1. What makes an intelligent robot 'intelligent'?

2. Name three commercial uses of robots.

   (i) ______________________

   (ii) ______________________

   (iii) ______________________

3. Name three end effectors used on a robot arm.

   (i) ______________________

   (ii) ______________________

   (iii) ______________________

4. Name three sensors used on robots.

   (i) ______________________

   (ii) ______________________

   (iii) ______________________

5. How does the Mars Explorer robot detect a hazard immediately ahead?

6. How does the Mars Explorer robot decide which path to take?

7. What does 'telepresence' mean?

8. How do mobile robots navigate around a factory?

Exercise 9: Search techniques

1. Describe how a depth-first search operates.

2. If the goal state is near the top of the search tree on the right, which search method would be best and why?

3. If the goal state is far down the leftmost branch of the search tree, which search technique would be best? Explain your answer.

4. In a search tree what is the name of the ‘state’ that is being sought?

5. Describe how a breadth-first search operates.

6. Why are search techniques necessary when playing chess?

SECTION 6

Answers to Section 4 tasks

Page 77–8, Queries

? has_met(emma,kate).	yes
? is_taught_by(emma,miss_girdwood).	yes
? enjoys(kimberly,swimming).	yes
? listens_to(gemma,busted).	yes
? has_met(kate, emma).	no
? enjoys(kimberly,soccer).	no
? listens_to(alison,busted).	no

Page 79 – Task 1, Entering queries

? enjoys(X, tennis).	X= gemma, X= alison, X= heather
? enjoys(alison, X).	X= basketball, X= tennis
? is_taught_by(X, mr_brooks).	X= emma, X= gemma
? taught_in(X, m1).	X= maths, X= english
? has_met(emma,X).	X= gemma, X= kimberly, X= kate, X= jane, X= alison, X= heather
? enjoys(heather,What).	What= tennis, What= dancing
? enjoys(kimberly,What).	What= swimming
? taught_in(Subject, g4).	Subject= geography
? listens_to(Who, hip_hop).	Who= alison
? teaches(miss_crawford, maths).	no
? taught_in(english, rm2).	no
? taught_in(geography, Room).	Room= g4
? teaches(Who, science).	Who= miss_crawford

Page 83

? is_a(cod,fish).	yes
? is_a(haddock,fish).	yes
? is_a(X, fish)	X= cod, X= haddock, X= whiting, X= mackerel

Page 83

has_scales(X):-
is_a(X,fish).

has_gills(X):-
is_a(X,fish).

Page 86

Q1 ?eats(X,slurry).

15 eats(pig, slurry).	X= pig

Q2 ?eats(cow,X).

13 eats(cow, grass).	X= grass

Q3 ?is_a_creature(pig)

27 is_a_creature(X):-	X= pig
28 liveonfarm(X)	
?liveonfarm(X)	search for sub-goal to line 27
21 liveonfarm(X):-	found liveonfarm(X) at line 21
22 is_a(X, farm_animal).	
?is_a(pig, farm_animal).	search for sub-goal to line 22
6 is_a(pig, farm_animal)	match for sub-goal
SUCCESS	on rule line 21
SUCCESS	on rule line 22
yes	

Q4 ?has_gills(haddock).

23 has_gills(X):-	
24 is_a(X,fish)	X= haddock
?is_a(haddock, fish).	search for sub-goal to rule line 23
1 is_a(haddock, fish)	match on line 1
SUCCESS	on sub-goal line 24
SUCCESS	on rule line 23
Yes	

Q5 ?is_a(X, farm_animal)

5 is_a(cow, farm_animal)	match on line 5
SUCCESS	
X= cow	
?is_a(X, farm_animal)	search for any more solutions
6 is_a(pig, farm_animal)	match on line 6
SUCCESS	
X= pig	
?is_a(X, farm_animal)	search for any more solutions
7 is_a(horse, farm_animal)	match on line 7
SUCCESS	
X= horse	

?is_a(X, farm_animal)	search for any more solutions
8 is_a(chicken, farm_animal)	match on line 8
SUCCESS	
X= goat	
?is_a(X, farm_animal)	search for any more solutions
9 is_a(goat, farm_animal)	match on line 9
SUCCESS	
X= goat	
Q6 ?is_a(X,fish)	search for a match
1 is_a(haddock, fish).	match on line 1 sub-goal succeeds, rule
SUCCESS	
X=haddock	
?is_a(X,fish)	back to check for more
2 is_a(cod, fish).	match on line 2 sub-goal succeeds, rule
SUCCESS	
X=cod	
?is_a(X,fish)	back to check for more
3 is_a(whiting, fish).	match on line 2 sub-goal succeeds, rule
SUCCESS	
X=whiting	
?is_a(X,fish)	back to check for more
4 is_a(mackerel, fish).	match on line 2 sub-goal succeeds, rule
SUCCESS	
X=mackerel	
?is_a(X,fish)	back to check for more solutions
FAIL	no more matches
No	
Q7 ?liveonfarm(X)	
21 liveonfarm(X):-	match on line 21
22 is_a(X, farm_animal)	sub-goal for rule line 21
?is_a(X, farm_animal)	
5 is_a(cow, farm_animal)	match for sub-goal
SUCCESS	
X= cow	

```
?is_a(X, farm_animal)
6    is_a(pig, farm_animal)             match for sub-goal
SUCCESS
X= pig

?is_a(X, farm_animal)
7    is_a(horse, farm_animal)           match for sub-goal
SUCCESS
X= horse

?is_a(X, farm_animal)
8    is_a(chicken, farm_animal)         match for sub-goal
SUCCESS
X= chicken

?is_a(X, farm_animal)
9    is_a(goat, farm_animal)            match for sub-goal
SUCCESS
X= goat
```

Page 89 – Task 5, Careers

```
social_science(fiona):-
history(X),
X>70.

nursery_nurse(X):-
likes(reading),
likes(children).

passed_english(fiona):-
exam_score(english,X),
X>50.
```

(Similar for other exams)

```
career_police(X):-
passed_english(X),
personality(assertive).

science_course(X):-
passed_english(X),
passed_biology(X).
```

Page 91 – Task 6, Knowledge Base: Golf Club

```
age(martin, 12).
age(mark,33).
age(james,35).
age(kate,44).
age(claire,47).
age(robert,36).
age(harry,55).
age(alan,24).

sex(martin,m).
sex(mark,m).
sex(james,m).
sex(kate,f).
sex(clair,f).
sex(robert,m).
sex(harry,m).
sex(alan,m).

hcap(martin,18).
hcap(mark,22).
hcap(james,2).
hcap(kate,23).
hcap(claire,3).
hcap(robert,26).
hcap(harry,15).
hcap(alan,13).

type(alan,restricted).
type(harry,restricted).
type(mark,junior).
type(martin,junior).

type(claire,full).
type(kate,restricted).
type(robert,full).
type(james,full).
```

Page 91 – Task 6

```
Q1 ?age(james,55).
Q2 ?type(alan,junior).
Q3 ?type(X,junior).
Q4 ?type(X,restricted).
Q5 ?type(X,full).
Q6 ?sex(X,f).
Q7 ?sex(X,m),type(X,restricted).
Q8 ?sex(X,f),type(X,restricted).
```

Page 92 – Task 7, Rules and trace

```
gent_comp(X):-
type(X, restricted),
sex(X,m).
```

?gent_comp(X)	
41 gent_comp(X):-	match line 41
42 type(X,restricted),	sub-goal 1
43 sex(X,m).	sub-goal 2
?type(X,restricted)	search sub-goal 1
25 type(alan,restricted)	X= alan
?sex(alan, m).	search goal 2
15 sex(alan,m)	(match) goal 2
SUCCESS	on both sub-goals
SUCCESS	on rule
X=alan	
?type(X,restricted)	search for more
26 type(harry,restricted)	(match) X= harry
?sex(harry,m).	search sub-goal 2
14 sex(harry,m)	match goal 2
SUCCESS	on both sub-goals
SUCCESS	on rule
X=harry	
?type(X,restricted)	search for more
31 type(kate,restricted)	(match) X= kate
?sex(kate,m).	search sub-goal 2
FAIL	on goal 2
FAIL	on rule
no	

```
1.  age(martin, 15).
2.  age(mark,33).
3.  age(james,55).
4.  age(kate,44).
5.  age(claire,47).
6.  age(harry,55).
7.  age(alan,24).
8.  age(robert,36).

9.  sex(martin,m).
10. sex(mark,m).
11. sex(james,m).
12. sex(kate,f).
13. sex(claire,f).
14. sex(harry,m).
15. sex(alan,m).
16. sex(robert,m).

17. hcap(martin,18).
18. hcap(mark,22).
19. hcap(james,2).
20. hcap(kate,23).
21. hcap(claire,3).
22. hcap(harry,15).
23. hcap(alan,13).
24. hcap(robert,26).

25. type(alan,restricted).
26. type(harry,restricted).
27. type(mark,junior).
28. type(martin,junior).
29. type(claire,full).
30. type(james,full).
31. type(kate,restricted).
32. type(robert,full).

33. last_year_junior(X):-
34. age(X,15).

35. ladies_medal(X):-
36. type(X,full),
37. female(X).

38. restricted_medal(X):-
39. type(X,restricted),
40. male(X).

41. gent_comp(X):-
42. type(X, restricted),
43. sex(X,m).
```

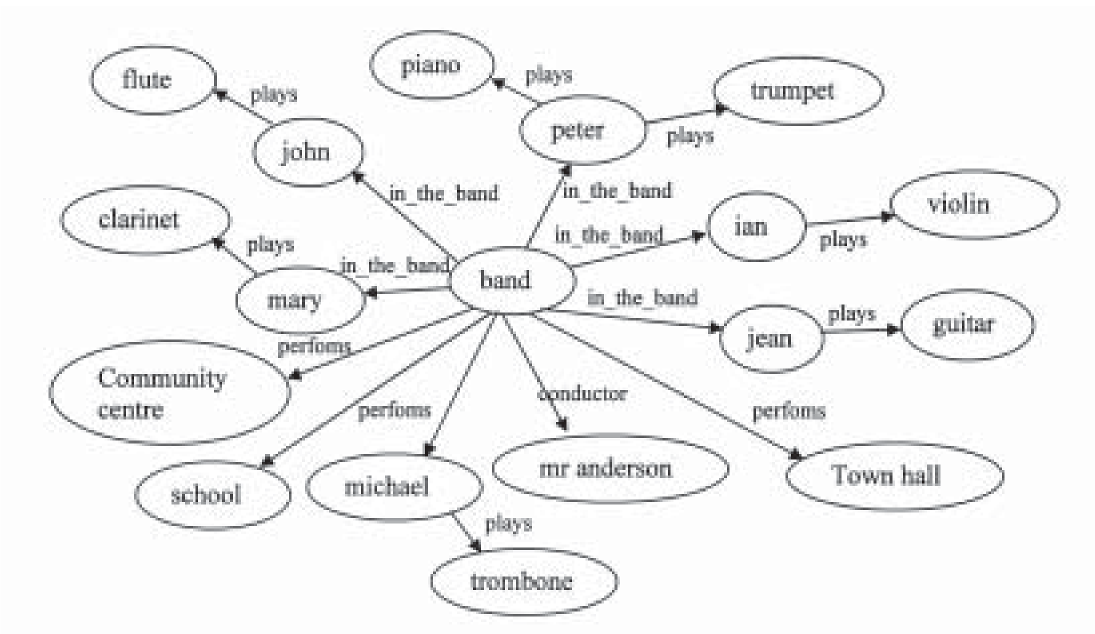
Page 93 – Task 8

```
in_the_band(peter).
in_the_band(mary).
in_the_band(michael).
in_the_band(ian).
in_the_band(john).
in_the_band(jean).

plays(peter,trumpet).
learning(peter,piano).
plays(mary,clarinet).
plays(mary,flute).
plays(michael,trombone).
plays(jean,guitar).
plays(jean,violin).
plays(ian violin).
plays(john,flute).

performs(band,school).
performs(band,town_hall).
performs(band,community_centre).

conductor(band,mr_anderson).
```



Page 93 – Task 9

```
attend_audition(X):-
in_the_band(X),
plays(X, violin).
```

Page 94/5 – Practical task solution

```
age(julie,20).
age(james,21).
age(jane,22).
age(simon,16).
age(john,22).

passed_written(julie).
passed_written(james).
passed_written(john).
passed_written(simon).

passed_practical(julie).
passed_practical(john).
passed_practical(james)

passed_motorcycle(john).
passed_motorcycle(jane).
passed_motorcycle(james).

can_drive_car(X):-
passed_written(X),
passed_practical(X).
```

Queries	Test results
1. ?age(julie,20).	1. yes
2. ?passed_written(james).	2. yes
3. ?passed_motorcycle(X).	3. X=john, X=jane, X=james
4. ?passed_written(X).	4. X=julie, X=james, X=john, X=simon
5. ?age(X,22)	5. X=jane, X=john
6. ?can_drive_car(X).	6. X=julie, X=john, X=james
7. ?can_drive_car(X), passed_motorcycle(X).	7. X=james, X=john