12/1/88

Knowledge Representation and Inference Two

1.0 Introduction to Knowledge Representation and Inference

A "representation of knowledge" is a combination of data structures and interpretive procedures that, if used in the right way in computer program, will lead to "knowledgeable" behaviour. In this way we can "ascribe" knowledge to programs in the same manner as we ascribe it to each other - based upon observing certain behaviour; we say a program knows about objects in its domain, about events that have taken place, or about how to perform specific tasks. It is important to keep in mind that a data structure is not a knowledge representation, any more than an encyclopaedia is knowledge. We can say, metaphorically, that a book is a source of knowledge, but without a reader (an interpreter), the book is just ink on paper.

AI has therefore mostly been interested in understanding how to build programs which exhibit knowledgeable behaviour. One of the best summaries of what underlies all such attempts is the "knowledge representation hypothesis" of Brian Smith [see chapter 3 of the Big Red KR book]. This says:

The Knowledge Representation Hypothesis

> Any mechanically embodied intelligent process will be comprised of structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and b) independent of such external semantical attribution, play a formal but causal and essential role in engineering the behaviour that manifests that knowledge.

A further strong belief held by most AI researchers has been that in order to build programs that can behave knowledgeably programs will have to know something about what they know. In other words they will have to be reflective systems. Smith summarised this requirement in terms of what is called the reflection hypothesis. Which says:

The Reflection Hypothesis

> In as much as a computational process can be constructed to reason about an external world in virtue of comprising an ingredient process (interpreter) formally manipulating representations of that world, so too a computational process could be made to reason about itself in virtue of comprising an ingredient process (interpreter) formally manipulating representations of its own operations and structures.

These two hypothesis should not be taken as definitions of AI, or what makes a computer program and AI program. They are more concerned with the articulation of a framework in which to carry out AI research. As Smith himself wrote "... one of the most difficult questions is merely to ascertain what the [KR] hypothesis is actually saying - thus my interest in representation is more a concern to make it clear than to defend or deny it." The work of Smith and many other people in AI since it started about thirty years ago has resulted in a wide variety of "knowledge representation schemes". The most important consideration in examining and comparing all these schemes is the eventual "use" of the knowledge represented using them. Getting a feeling for what it means for one representation scheme to represent certain types of knowledge more easily than another - which involves the representations, the domain, and the reasoning strategies - is, at present, part of doing AI research. Put another way, there is NO "theory of knowledge representation". We don't yet know why some schemes are good for certain tasks and others not.

2.0 The Declarative versus Procedural Debate

Some years ago a battle raged in the AI community about whether knowledge should be represented "declaratively" or "procedurally". The declarative approach was typified by resolution-based theorem provers and the procedural approach was typified by systems like Winograd's SHRDLU system. It was a debate in some sense based upon the distinction between "knowing that" and "knowing how". The proceduralists asserted that our knowledge is primarily a "knowing how", a view which became associated with people like Minsky, Papert, Hewitt, and Winograd at MIT. The declarativists, on the other hand, believed that knowledge of a subject is intimately bound with the procedures for its use. This view was based upon a two part idea: a quite general set of procedures for manipulating facts of all sorts, and a set of specific facts describing a particular domain. One of the major advocates of this approach was McCarthy, and the approach became associated with Yale and Edinburgh.

The declarative-procedural battle was an important one in AI. Although it dissolved rather than got resolved, the knowledge representation techniques we have today are a direct result of much work done on the issues raised by the debate. Today there is still no "theory of knowledge representation". There are still many problems to be overcome. Perhaps the most significant difference is that now we seem to spend less time arguing and discussing them and more time trying to make them out to be something they are not. It is important that anybody attempting to use any of the various representation techniques available should develop a critical approach to their application. Failure to do this will almost certainly result in a failed project. However, if a critical and constructive approach is adopted towards the application AI knowledge representation techniques some very impressive results can be obtained. Be warned though, AI and knowledge representation in particular is not magic; to use it well demands a sound understanding of the particular problem being tackled.

2.0 Required Reading from Lecture 1

From: Readings in Knowledge Representation,

Chapter 1 - Some Problems and Non-Problems in Representation Theory, by Pat Hayes, page 3. 16 pages.

Chapter 3 - Prologue to "Reflection and Semantics in a Procedural Language, by Brian Smith, page 31. ✓

Chapter 5 - From Micro-Worlds to Knowledge Representation: AI at an Impasse, by Hubert Dreyfus, page 71. ✓

These three papers should give you an introduction to some of the issues, problems, and questions that KR+I is concerned with, together with a somewhat anti-AI view to give another perspective on things.

3.0 Small Group Tutorial Preparation

Using your notes of the KR+I-1 survey conducted during part two of lecture one organise the information collected about various KR techniques into a useful knowledge base which can support the answering of questions about what type of techniques there are, and what they can and cannot be used for. You may use a representation scheme of your own choosing, but must be prepared to defend your decision. You should also be prepared to comment upon the success of the knowledge elicitation exercise carried out to obtain the information, and upon any difficulties you experience in organising the information into a coherent knowledge base.

Tim Smithers
January 1988

Knowledge Representation and Inference Two

1.0 Introduction to the Course

The Knowledge Representation and Inference Two (KR+I-2) course is designed to follow on from the Knowledge Representation and Inference One course (Msc or AI/CS-3). The aim is to introduce a selection of more advanced techniques and ideas, together with some of the problems associated with them. The course will consist of nine Lectures, whose contents are described below, five Full Class Tutorials, and four Small Group Tutorials. Together with material presented in the lectures there will be required reading, tutorial preparations, and course work exercises to do.

1.1 Lectures

Lectures will take place on Tuesdays, 2-4pm, in Lecture Theatre 3 (LT3), Appleton Tower (AT).

Lecture 1: Introduction to course and review of KR+I-1 material.

Lecture 2: Non Classical Logics - many-valued logics, situational logic, and modal logics including the Lewis modal aixom systems S1 to S5. An attempt will also be made to illustrate the application of some of these logics.

Lecture 3: Non-monotonic logics and Truth Maintenance - Chronological Backtracking, Dependency Directed Backtracking, and the deKleer Assumption-based Truth Maintenance System (ATMS).

Lecture 4: Uncertainty and Default Reasoning - sources of uncertainty, probability theory, Baye's rule, Dempster/Schafer theory, and incidence calculus. And some proposed methods for introducing default reasoning.

Lecture 5: Blackboard Systems - the basic technique and a survey of some example systems.

Lecture 6: Planning - goal-directed planning, failure-directed planning, non-linear planning, with an example of planning for robotic assembly.

Lecture 7: Object oriented programming - what is it, why is it useful.

Lecture 8: More on Frame-based systems - lying about trees and what isa's aren't. Plus some examples of other problems and possible ways around them illustrated using the Knowledge Representation language being built for the Edinburgh Designer System (EDS).

~~Lecture 9: Intelligent Knowledge-based Systems Architectures - the EDS and AI Toolkits.~~

1.2 Tutorials

In order to join the tutorial schemes for the MSc and AI/CS-3 courses together a scheme of mixed Full Class and Small Group tutorials will operate. The Full Class tutorials will take place on alternate Thursdays from 4-5pm in LT2, AT, starting on Thursday 14 January. For MSc students the Small Group tutorials will take place on alternate Thursdays from 4-5pm at either South Bridge or Forrest Hill (depending on the group), starting on Thursday 21 January. For AI/CS-3 students Small Group tutorials will take place on alternate Fridays from 12-1pm at Forrest Hill, starting on Friday 22 January. The exact location and names of tutors will be announced during the first full class tutorial on

Thursday 14 January.

1.2.1 Full Class Tutorials

These are intended to broaden the scope of the course presented in the lecture series. The material presented will not be examinable, but attendance is compulsory for all MSc students taking the KR+I-2 course, and highly recommended for all AI/CS-3 students. The subjects of the Full Class tutorials is as follows:

Tutorial 1 (Thursday 14 January):
"What is AI: A personal view",
Tim Smithers, Department of Artificial Intelligence.

Tutorial 3 (Thursday 28 January):
"Connectionism",
Dave Willshaw,
Centre for Cognitive Science.

? Big in the states.

Tutorial 5 (Thursday 11 February):
"Cognitive Neuroscience",
Brendan McGonigle, Laboratory of Cognitive Neuroscience.

entertaining speaker.

Tutorial 7 (Thursday 25 February):
"Cognitive Science",
Mark Steedman, Centre for Cognitive Science.

Tutorial 9 (Thursday 10 March):
"The Speech to Text System", AlvI
Henry Thompson, Department of Artificial Intelligence and Centre for Cognitive Science.

follows lecture on system architecture

1.2.2 Small Group Tutorials

These are intended to provide an opportunity to discuss in small groups aspects of some of the material presented in the lectures. They are compulsory for all MSc and AI/CS-3 students. Each student is expected to prepare for these tutorials. The discussions will be started by the tutors selecting at random a member of the group to present what she or he has prepared. (Note the random selection algorithm means that having been selected once does NOT mean you will not be asked again!) The subjects of the Small Group Tutorials will be announced at the immediately preceding lecture, and will be drawn from either or both of the preceding lectures as follows:

Tutorial 2: Lecture 1

Tutorial 4: Lecture 2 and/or 3

Tutorial 6: Lecture 4 and/or 5

Tutorial 8: Lecture 6 and/or 7 and/or 8

The tutors for the groups will be:

MSc:	AI/CS-3:
Andy Bowles (SB)	Karl Millington (FH)
Rajiv Trehan (SB)	Tim Smithers (FH)
Wade Troxell (FH)	
Tim Smithers (FH)	

1.3 Course Work

Course work will consist of Required Reading and Exercises, both pencil and paper and computational. The Required Reading will be set at the end of each lecture and will mostly be taken from the "Big Red KR book" [Brachman and Levesque Ed --Readings in Knowledge Representation]. Further, recommended reading may also be given at the end of each lecture. This material is strongly recommended, but will not be directly examinable, although course work and Small Group Tutorial preparations may assume a knowledge of it.

Four course work exercises will be set in weeks 2, 4, 6, and 8, each for handing in two weeks later. Exercises 4 and 8 will each count 10% towards the final course mark in conjunction with exam mark. Course work exercises will be pencil and paper and some programming. Late submissions WILL NOT be marked.

1.4 Examination

The KR+I-2 MSc examination will consist of a one and a half hour paper. Answers to two questions out of three will be required. The first question on the paper will be a compulsory question. Any material presented in lectures, contained in required reading, prepared for tutorials, or set as course work exercises will be examinable. The material presented in Full Class Tutorials and recommended reading will NOT be examinable.

Tim Smithers
January 1988