Lecture Notes 2 – Knowledge Representation

Patrick Doherty

February 7, 1998

1 Outline

1. Introduction to Knowledge Representation.
   (a) Why and How?
2. Logic in AI.
   (a) Logic for Analysis and Specification.
   (b) Logic as a KR Language with inference engine.
   (c) Pros and Cons.
3. Epistemic and Heuristic Adequacy.
4. A Knowledge Representation framework.

Much of this material is taken directly from the articles of Israel, McCarthy, and the book by Davis.

2 Introduction to KR

Knowledge representation is the re-*presentation* of knowledge. Three immediate questions come to mind:

1. What is knowledge?
2. What type of representation should be used to represent knowledge?
3. Why do we want to represent knowledge?

2.1 Why do we want to represent knowledge?

An answer to the third question will simplify our answer to questions one and two, so we can begin with it. "The aim of AI research is the design and creation of systems capable of succeeding at tasks whose accomplishment requires intelligence."[1] Intelligent behavior is generally dependent on

- Massive amounts of facts about the environment in which intelligent behavior is envoked, along with facts about the agent itself and its relation to the environment. This would include, among other aspects, capabilities and awareness of what an agent knows or believes.

[1]Israel, p.15.

- Large amounts of commonsense knowledge which permit intelligent behavior in situations where we do not have all the facts. Intelligent behavior is characterized by its flexibility; the ability to adapt smoothly to unique situations, not previously experienced.
- Different modes of reasoning such as deduction, abduction and induction which use facts and beliefs to derive other facts and beliefs useful for solving problems.
- There are most certainly a number of biological constraints acquired through evolution that via the agents symbiosis with its environment, contribute to efficient methods for coping with the environment.

Based on the aim of AI research and the dependency of intelligent behavior on knowledge and reasoning using such knowledge, it is clear that artificial agents should have representations of knowledge about their environment, themselves, and strategies for using the knowledge they have in an efficient way. This will be used when reasoning towards solutions to problems which arise in attempts to acheive goals. The goals may be specified by an outside source or by the agent itself.

In summary, in order to explain the intelligent behavior of an agent given some goal, we

1. attribute to the agent knowledge of the conditions relevant to the success of some behavior, and
2. attribute to the agent the ability to reason from such knowledge.

The point to emphasize here is that we would like to **construct** artifacts which behave "intelligently". The emphasis on construction of artifacts implies a strong engineering flavor with the associated constraints. Just because the tasks our artifacts will face require what we as humans call "intelligence" does not imply that the artifacts we build will mimic human intelligent behavior, whatever that is. This is an important point that will help in avoiding many fruitless discussions with psychologists and philosphers about issues pertaining to intelligence and cognition. There are certainly various degrees of intelligence and different criteria for determining whether an artifact's intelligent behavior is of so and so a degree. In fact, the type of intelligence artifacts will exhibit might well be qualitatively different in development from that of humans.

The goal of constructing intelligent artifacts is an ambitious one and we have only just begun. Implementing elementary types of intelligent behavior already provide their share of engineering problems. Although AI research strives for generality, this can only be reached by attention to detail. Problems and phenomona unforeseen in cognitive psychology and philosophy have arisen when forced to actually implement intelligent behaviors. One such phenomena almost completely missed in philosophy is the nonmonotonic nature of reasoning involved in everyday decision making. In conclusion, although one should not underestimate the help we can get from philosophy and psychology, we should not overestimate it either. The relation between AI, psychology and philosophy deserves a whole course. Unfortunately, we do not have the time to consider such relationships.

What we will do in this course is study a fundamental problem in AI. Given a goal, how may an agent use facts about actions and their effects, represented in a knowledge base, to determine the appropriate actions to acheive that goal? Our first problem is to decide how to represent actions and their effects in a knowledge base. The second is how to reason with such facts. From a philosophical and psychological point of view, our representations and problem statements might appear trivial. On the other hand, solutions to the problems that arise in such scenarios can be used in certain cases to determine appropriate actions for achieving goals. Attempting to find solutions to simple types of intelligent behavior generates new problems and highlights phenomena which might previously have gone unnoticed if we had chosen to stay at much higher levels of abstraction. One such phenomenon that will be both studied and used is that of the nonmononotonic nature of reasoning about actions and change. So, although our long term aim is quite ambitious, our short term goal is to get off the ground. This course will provide a detailed examination of one of the fundamental problems in the logic-based approach that need to be solved if our artifacts are to have a chance of exhibiting intelligent behavior of a higher degree. Although we will study particular problems related to reasoning about action and change, keep in mind that similar problems also arise in other areas such as explanation, diagnosis, text understanding and text generation.

2.2 What is knowledge?

Based on the previous discussion, the extremely complex philosophical question about what knowledge is can be diffused somewhat. We will call the things believed, known, or assumed, *propositions*. An important property of propositions is that they are truth valuable. Propositions can be true, false, unknown, or perhaps both true and false. Whatever view of the world one might have it is characteristic of propositions that we can assign truth values to them. A second important property of propositions is that logical relations can hold between them, in particular that of logical consequence. For example, a set $P_1, \ldots, P_n$ may be a good reason for believing Q. In other words, we can infer Q on the basis of believing $P_1, \ldots, P_n$. For the time being, we will remain neutral as to the choice of inference or consequence we have in mind.

The next important step is to model an agent's beliefs. If we assume that at any one time an agent has a set of beliefs (propositions), then we can characterize the agents cognitive state (its set of beliefs) in terms of it accepting as true sentences that express the propositions. Shifting to a syntactic characterization of propositions has its advantages. We can use the rules of formal languages to combine propositions into complex forms and define arguments as sequences of sentences, each with a distinguished element called it's conclusion.

Clearly, our assumptions are quite strong and to philosophers and cognitive psychologists certainly coarse oversimplifications. One can spend much time discussing the existence or nonexistence of propositions with philosophers, or their relation to language. The standard notion of knowledge in philosophy is *justified true belief.* In AI the term knowledge is used rather as a blanket term for information in the knowledge base which as we pointed out previously could be interpreted as facts, beliefs, assumptions etc. We can also shed a great deal of ink over the definition of a cognitive state and its representation in the mind as being linguistic or non-linguistic. These are fascinating issues, but we leave them for other courses to give them the full and critical treatment they deserve. For our purposes, these assumptions work as a loose metaphor for what our intelligent architectures are supposed to mirror. In conclusion, knowledge is defined as propositions and propositions have syntactic representations in the form of well-formed formulas in a logical language.

2.3 What type of representation should be used to represent knowledge?

There are a number of proposals as to what type of representation language should be used to represent knowledge in a knowledge base, to what degree it should be formalized, and even to what degree it is possible to formalize knowledge and its use. Again, we will take the easy way out. This course will choose logic as the language of choice. We will discuss its advantages and disadvantages and the different purposes of its use in the next section. It should be pointed out from the start that the use of logic as the KR language of choice is highly controversial and the verdict is still out. It is too early to determine whether logic, another representation, or even a combination of representations are the right way to go. The latter appears to be the most feasible. Whatever representation is chosen though, there will continue to be a logical element involved. It is reasonable to assume that other representations will still be analyzed using logical tools. We will have more to say about the role of logic in the next section.

As far as alternatives are concerned there are, among others, production rules, semantic nets, frame-based languages and neural nets, although it is debatable whether the latter is a knowledge representation or a tool for implementing knowledge representations. Most of you have had contact with these modes of representation in other courses and so are familiar with their advantages and disadvantages. Keep in mind that even if one chooses to use such tools as representation languages that it is still vital that we understand their logical properties at a more abstract level. What can and can not be expressed? Is the inference relation used sound? Such questions are answered for free when we choose the logic-based approach.

3 Logic in AI

It is surprising that at this stage in the development of AI, one still has to point out the strong connection between logic and AI and logic's relevence to AI.[2] The design and creation of devices that envoke intelligent behavior require valid reasoning. Logic is the study of reasoning. Consequently, both logic and AI share the same subject matter, reasoning. Logic plays two broad roles in AI:

1. as a source of languages and logics for artificial reasoners.
2. as a source of analytical tools and techniques which provide the underlying mathematical and conceptual framework for AI research.

The latter role is uncontroversial. Logical techniques are quite useful in specifying and analyzing the behavior of classes of intelligent devices or agents. On the other hand, using logic as a knowledge representation language and a proof theory as the algorithm for reasoning is highly controversial. There are ungoing debates in the literature regarding the first role that are as yet inconclusive. Such discussons are most often characterized by a lack of understanding by participants of both sides as to the other sides definition of the problem discussed and the assumptions on which their arguments are based. Generic discussions are of the type philosopher–AI researcher or psychologist–AI researcher. The more practical types of discussions are of the type AIresearcher–AIresearcher. Here one might play a neural-net-type against a symbolic-type or a logic-symbolic-type against a procedural-symbolic-type. Before considering the first role, we will look at logic as a tool for designing and specifying a particular type of intelligent behavior, diagnosis.

3.1 Using logic for analysis and specification

Let's take a look at a task that requires intelligent behavior, diagnosis. We might want to diagnose an illness based on symptoms, or diagnose a problem with a car engine based on sensory information, or even diagnose faults in circuits. Most of you are already familiar with diagnosis via expert systems, or a visit to the doctor or gas station. Such familiarity suffices to convey the idea of diagnosis to other humans. But, how can we be more precise about what diagnosis is, so that we can provide a specification to be used in the design or construction of the diagnosis task? The goal would be to implement such a process that could be envoked by an agent exhibiting intelligent behavior in this respect that is on par with that of human diagnosticians.

Our question is "What is a diagnosis problem?" and secondly "How do we know that we have a solution to a diagnosis problem?". What we would like is a formal logical definition of a diagnosis problem and a precise logical specification of the notion of a solution to such a problem. Reiter [**?**] provides such a specification. We have a diagnosis problem when a device or system is observed to deviate from its desired specification. The first assumption then , is that there is a clear and precise specification of a system and its components.

[2] Parts of this section are based on [**?**].

Definition 3.1 *A* system *is a pair* $\langle SD, COMP\rangle$ *where* SD*, the* system specification*, is a satisfiable set of first-order sentences and* $COMP$ *is a set of constants, one for each distinguished component of the system.*

Definition 3.2 *An* observation OBS *is a finite set of first-order sentences.*

In addition, the signature of the logical language being used must contain a non-logical unary predicate constant Ab whose intended meaning is 'abnormal'. $COMP$ is also a subset of the signature. We will use $\{c_1, \ldots, c_n\}$ as component constants.

When the following set of sentences is satisfiable, this signifies the assumption that the system in question is functioning properly:

$$SD \cup \{\neg Ab(c_1), \ldots, \neg Ab(c_n)\}. \tag{1}$$

Let OBS be an observation of a set of faults. We have a diagnosis problem when the following set of sentences is unsatisfiable.

$$SD \cup \{\neg Ab(c_1), \ldots, \neg Ab(c_n)\} \cup OBS. \tag{2}$$

A diagnosis is going to be a *conjecture* (we can not know for sure) that certain of our components are faulty (Ab) and the rest are normal ($\neg Ab$). The problem is to specify which are the faulty components and which are not. Under the assumption that $SD \cup OBS$ is satisfiable and 2 isn't, retraction of enough components in $\{\neg Ab(c_1), \ldots, \neg Ab(c_n)\}$ should restore satisfiability. The components retracted would then represent our diagnosis of why the system is not functioning properly. Obviously, some constraints must be placed on what components we conjecture to be faulty, otherwise we could retract all $\neg Ab(c_i)$. So, a diagnosis will be a conjecture that a minimal set of components are faulty. Here is Reiters formal definition.

Definition 3.3 *A* diagnosis *for* $\langle\langle SD, COMP\rangle, OBS\rangle$ *is a minimal set* $\Delta \subseteq COMP$ *such that*

$$SD \cup OBS \cup \{Ab(c) \mid c \in \Delta\} \cup \{\neg Ab(c) \mid c \in COMP - \Delta\}$$

is consistent.

Based on the minimality of Δ , it follows that for each $c_i \in \Delta$,

$$SD \cup OBS \cup \{\neg Ab(c) \mid c \in COMP - \Delta\} \models Ab(c_i).$$

This states that the faulty components are logically dependent on the functioning components in $COMP - \Delta$.

Note the value of the logical specification above. Reiter's logical analysis of diagnostic problems and their solutions could be used as a description of a minimium criterion of success for any implemented diagnosis system. A diagnosis system would be deemed successful if for a particular diagnosis, it returned at least one of the minimal sets of components that the logical representation of the same problem would. Note also that one could implement the diagnosis system by constructing a theorem prover that generates the minimal sets of faulty components, but that the logical analysis is completely independent of questions regarding implementation. This clean separation of epistemological aspects of a problem from its heuristic aspects is worth striving after. We will have more to say about this in a later section.

Let's make the discussion more concrete by presenting an example due to Reiter [**?**] involving the diagnosis of a binary full adder. The components in the adder are $COMP = \{A_1, A_2, X_1, X_2, O_1\}$. The system specification consists of the following sentences.

$$\forall x. AndG(x) \wedge \neg Ab(x) \rightarrow out(x) = and(in1(x), in2(x)). \tag{3}$$

$$\forall x. XorG(x) \wedge \neg Ab(x) \rightarrow out(x) = xor(in1(x), in2(x)). \tag{4}$$

$$\forall x. OrG(x) \wedge \neg Ab(x) \rightarrow out(x) = or(in1(x), in2(x)). \tag{5}$$

$$\forall x. OrG(x) \wedge \neg Ab(x) \rightarrow out(x) = or(in1(x), in2(x)). \tag{6}$$

$$AndG(A_1) \wedge AndG(A_2). \tag{7}$$

$$XorG(X_1) \wedge XorG(X_2) \wedge OrG(O_1). \tag{8}$$

$$out(X_1) = in2(A_2). \tag{9}$$

$$out(X_1) = in1(X_2). \tag{10}$$

$$out(A_2) = in1(O_1). \tag{11}$$

$$in1(A_2) = in2(X_2). \tag{12}$$

$$in1(X_1) = in1(A_1). \tag{13}$$

$$in2(X_1) = in2(A_1). \tag{14}$$

$$out(A_1) = in2(O_1). \tag{15}$$

$$in1(X_1) = 0 \vee in1(X_1) = 1. \tag{16}$$

$$in2(X_1) = 0 \vee in2(X_1) = 1. \tag{17}$$

$$in1(A_1) = 0 \vee in1(A_1) = 1. \tag{18}$$

In addition there are some axioms for boolean algebra not specified here. Below, is a figure of the binary full adder.

Figure 1: Binary Full Adder

Suppose the adder is given the inputs 1, 0, and 1 and it outputs 1, 0 in response. This can be described as the following observation OBS:

$$in1(X_1) = 1 \wedge in2(X_1) = 0 \wedge in1(A_2) = 1. \tag{19}$$

$$out(X_2) = 1 \wedge out(O_1) = 0. \tag{20}$$

Note that the observations tell us that the adder is faulty. Both circuit outputs are wrong for the given inputs. Although we have not yet studied the proper notion of entailment for computing conjectures, we can list the minimal sets of components representing a diagnosis of the binary full adder:

$$\{X_1\} \text{ or } \{X_2, O_1\} \text{ or } \{X_2, A_2\}$$

3.2 Using logic as a knowledge representation

The logic-based approach to knowledge representation was proposed by John McCarthy in the late 1950's in the form of a research strategy aimed at the design, implementation and analysis of artificial reasoners. The basic idea was a program called the "Advice Taker" that would solve problems by manipulating sentences in formal languages. The advice taker would get its instructions by telling it, in the form of declarative sentences, about its environment and what is wanted from it. The declarative nature of the instructions would ensure the ability to incrementally add knowledge to the device with little or no knowledge of what it already knows. This is similar to the way humans instruct each other using natural language and not at all like the methods used to instruct a computer via a program. In solving a problem, the assumption is that the advice taker would have access to the immediate logical consequences of its knowledge base and of what it is told. In McCarthy's own words,

> The "Advice Taker" is a proposed program for solving problems by manipulating sentences in formal languages.
>
> The main advantages we expect the advice taker to have is that its behavior will be improvable merely by making statements to it, telling it about its symbolic environment and what is wanted from it. To make these statements will require little if any knowledge of the program or the previous knowledge of the advice taker. One will be able to assume that the advice taker will have available to it a fairly wide class of immediate logical consequences of anything it is told and its previous knowledge. This property is expected to have much in common with what makes us describe humans as having common sense. We shall therefore say that a program has common sense if it automatically deduces for itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows.

McCarthy states that the "ultimate objective is to make programs that learn from their experience as effectively as humans do", but "in order for a program to be capable of learning somthing it must first be capable of being told it". Consequently, he concentrates on the following feature of a system that will evolve intelligence of human order:

1. Interesting changes in behavior must be expressible in a simple way.

The criterion for success will be to "achieve a system which can be told to make a specific improvement in its behavior with no more knowledge of its internal structure or previous knowledge than is required in order to instruct a human". Viewed from this context, logical languages seem to be the ideal choice as a knowledge representation language. Why is this?

McCarthy lists the advantages of the use of declarative sentences:

1. Advantage can be taken of previous knowledge.
2. Declarative sentences have logical consequences and it can be arranged that the machine will have available sufficiently simple logical consequences of what it is told and what it previously knew.
3. The meaning of declaratives is less dependent on the order in which they are presented than is the case with imperatives.
4. The effect of a declarative is less dependent on the previous state of the system so that less knowledge of this state is required on the part of the instructor.

In conclusion, the basic idea is that the knowledge required for intelligent behavior will be explicitly represented in a collection of manipulable data structures called a knowledge base, and that the rest of the knowledge required for solving problems will be inferred from it. Since the goal is to implement such a system, constraints must be placed on the language and inference methods, for example, recursive languages and effective decision procedures. There must also be a means for systematically determining the meaning of the knowledge in the knowledge base and the validity of the inferences based on the particular inference relation chosen. In other words, a formal mathematical semantics is needed. Taken together, formal logical systems are ideal as a medium for knowledge representation.

Although McCarthy presented these ideas in the late 1950's, many of his observations still hold today. In fact a modern variant of McCarthy's strategy is encapsulated quite nicely in Brian Smith's *knowledge representation hypothesis*:

> Any mechanically embodied intelligent process will comprise 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 engendering the behavior that manifests that knowledge.

3.3 Objections to the Logic-Based Approach

What, if any, are the objections to such a strategy and what might prevent its success? Israel lists three:

1. The alleged, or real, expressive inadequacies of logical languages.
2. The difficulty of theorem-proving, or finding derivations.
3. The nonmonotonicity or defeasibility of much of ordinary reasoning.

A reasonable reply to the first objection is the fact that although logical languages may be unnatural to humans, this is not necessarily the case for machines. Secondly, many critics have FOPC in mind when stating such an objection. There are obviously a great variety of logical languages and logics that are not as unnatural as FOPC and are obvious candidates for KR languages. For any domain, we simply have to find a suitable logic that naturally expresses facts about the domain. For example, frame-based languages are reasonable candidates for representing information about inheritance of properties and their underlying semantics is essentially a four-valued logic.

The second objection is not so easily dismissed even by proponents of the logical approach. There are two aspects to this objection, undecidability and efficiency. There is a tradeoff between the expressive power of our KR languages and the computational tractability of them. Increasing expressivity, decreases computational tractability and vice-versa. Much current research involves developing specialized languages and efficient inference procedures for reasoning about specific domains. Another line of research looks for fragments of FOPC that are tractable in general. Logic programming is an example of an efficient inference engine for the horn fragment of FOPC. Yet another area concentrates on tractable approximations of ideal but intractable notions of inference. Much of the research is guided by characteristics specific to particular domains. In addition, although the declarative nature of a KR language is generally considered a feature, there are some notions we want to model that are more easily represented as procedures. For example, representing addition as a relation is not very efficient. Methods such as semantic attachment have been developed to deal with such problems though. There is still much debate in the area regarding these points. Many researchers have abandoned McCarthy's original idea about one specific KR language and concentrate on what are currently known as hybrid systems. These consist of special purpose reasoning systems that operate over special purpose representations. The major problem is in integrating the various modules and showing that the operation of the system as a whole meets the requirements of the specification of the system.

The third objection to logic was voiced quite early in the history of AI and was in fact noticed by proponents of the logic-based approach themselves in the form of the frame problem. The obvious reply to the third objection is not at all inconsistent with the logic-based approach, but considerably complicates the second objection. The idea is to simply develop logics whose consequence relations lack the property of nonmonotonicity. Such formalisms and their use are current areas of research in AI. The discovery of the ubiquity of nonmonotonic reasoning in everyday decision making and development of nonmonotonic logics should be considered one of the great success stories of research in AI. The conceptual success is somewhat blemished by the difficulty in applying the formalisms in an efficient manner. Althought we have the needed expressive power, in the worst case some of our nonmonotonic formalisms are not even semi-decidable. This is yet another area of current research.

In the beginnings of knowledge representation, many thought that FOPC and deductive reasoning would suffice as the basic backbone of any intelligent system. It is clear today that this is not the case and that much of our everyday commonsense reasoning is not deductive at all. As regards explanation, or a specific task involving explanation such as diagnosis, much of our reasoning is abductive. In predicting what may happen in the future, a capability needed in order to plan, we may have to jump to conclusions, conjecturing that somthing is the case in order to continue reasoning. Later on, we may find that our conjecture was not correct at all causing a modification of world description and thus our behavior towards that description. Such reasoning is non-deductive. In order to deal with reasoning tasks of this sort, new logics have been developed, or extensions of FOPC have been proposed where the notion of consequence lacks the property of monotonicity. We will have much to say about such nonmonotonic logics during the course, and we will in fact use them as tools in investigating phenomena related to action and change. Although the third objection can be dealt with, the use of nonmonotonic logics makes the second objective even more valid. This will also be discussed.

4 Epistemological and Heuristic Adequacy

At a very early stage McCarthy made a distinction between the epitomological and heuristic adequacy of knowledge representation languages.

- Epistemological Part of AI studies
  - What kinds of facts about the world are available to an observer with given opportunities to observe,
  - how these facts can be represented in the memory of a computer,
  - and what rules premit legitimate conclusions to be drawn from these facts.
- Heuristic Part of AI studies issues such as
  - how to search spaces of possibilities
  - how to pattern match, etc.

Advantages of the Separation

1. Same problem of what info is available and what conclusions can be drawn from it are similar for many reasoning tasks.
2. Single solution to epistemological problem can support a variety of heuristic approaches to the problem.
3. AI is a difficult scientific problem. Divide and conquer is a good methodology.
4. Difficult to formalize facts of common knowledge. Confining oneself only to special cases doesn't face difficulties involved in general intelligence.

4.1 Reichgalt's 4 levels of KB Systems

Reichgalt also makes an interesting distinction between four levels of a knowledge-based system that further clarify what is meant by epistemological adequacy. He distinguishes between the following levels:

4 levels and criteria for evaluating:

Implementation – Efficiency.

Logical – Clear syntax and semantics with sound inference rules.

Epistemological – Naturalness with which a representation can be constructed and understood and its modularity. Small additions imply small changes to the KB.

Conceptual – Direct correlation with phenomena modeled.

5 Knowledge Representation Framework

Now that we are familiar with the origins of KR in AI, different roles logic plays in AI, and the pros and cons of using logic for KR, let's present a methodology of sorts for developing declarative representations for the knowledge used for intelligent behavior by an artificial agent.[3]

We will define *commonsense reasoning* as the performance of *inference* on a body of *object-level* information, which describes the subject matter of the reasoning. In our case, the subject matter will be action and change. Object-level information is to be distinguished from *control-level* information, which describes the internal state of the reasoner and can be used to control the process of inference. Recall our previous discussion concerning epistemology and heuristics and McCarthy's original Advice Taker. Inference is the process of deriving new information form old. *Knowledge-Based Systems* will be defined as a program consisting of two parts:

- The *knowledge base* – A data structure encoding a body of object-level information.
- The *knowledge base manager* – A collection of procedures that perform inferences on the information in the knowledge base, including procedures that add and delete information.

What information is encoded in the inference procedures in distinction to what information is encoded declaratively in the knowledge base is a call made by the implementers of the system. Generally, we would like as much of the information as possible to be encoded declaratively as formulas in logic. Why? We have already discussed this previously, but briefly:

1. It is very expressive. Even incomplete information can be represented using disjunction and the existential quantifier.
2. There is a clear meaning for each piece of information. In addition, one can distinguish between somthing being false and not known.
3. It is easier to change or modify the information.
4. The same information can be used for many different tasks.

What modes of inference should be used by a knowledge based system? There are a number of possibilites:

1. *Deduction* – Deductive inferences are logically sound inferences. Inferred information is necessarily true if the premises are true.

[3]Parts of the following are taken from [**?**].

2. *Abduction* – Abductive inferences offer plausible explanations for some particular fact. Abduction is useful for explanation, planning and diagnosis, and is non-deductive.

3. *Induction* – Inductive inference infers general rules from specific examples. Induction is very useful for learning. It again is non-deductive.

Ideally, we would like to use deduction as much as possible because we know so much about it, and secondly, because it preserves truth, it simplifies implementation of the inference procedures in our KB-system. In addition, we already have a number of efficient procedures for fragments of FOPC. The problem is, as we shall see, that most of the interesting commonsense reasoning we will require, is non-deductive in nature.

The predominant methodology currently used for doing reserach in commonsense reasoning and knowledge representation is the following

> ...is to take a number of examples of commonsense inference in a commonsense domain, generally deductive inference; identify the general domain knowledge and the particular problem specification used in the inference; develop a formal language in which this knowledge can be expressed; and define primitives of the language as carefully and precisely as possible.

Not that this methodology completely excludes any reference to implementation. It is in the spirit of the logical analysis previously described for diagnosis. Davis lists three methodological assumptions on which the approach rests:

1. In designing a reasoning system, it is generally worthwhile to characterize the object level information that the system will use and the inferences that it will perform, independently of determining how the system should perform these inferences.

2. A number of fundamental commonsense domains arise so frequently in designing reasoning systems that it is worthwhile for the research community to study the basic structure of commonsense inference in these domains independently of any particular applications.

3. Certain characteristics of knowledge used in commonsense inference make it particularly important that the language used to express this knowledge be very carefully and tightly defined. In particular:

   (a) Commonsense inferences tend to use a great variety of partial and incomplete knowledge. Representations left open for interpretation may be ambiguous or unclear to the programmer.

   (b) The very familiarity of commonsense concepts and their closeness to natural language forms renders them liable to ambiguity.

Davis goes on to propose that "a commonsense reasoning system be designed as a knowledge base, recording information and performing inference on object-level information about some *microworld*", where a microworld is defined as "a restricted, idealized model of the world containing only those relations and entities of interest in the particular reasoning system being designed." In addition, the microworld is analyzed on three levels:

1. The definition of the microworld: its entities, its relations, and the rules that govern them. This definition is called the *ontology* of the domain.

2. The analysis of the type of information that the system must record and the types of inferences it must perform. The result of this analysis is generally a *formal language*, in which facts of the microworld can be expressed, and *axioms* or *rules of inference*, which characterize inferences that are reasonable within the microworld.

3. An *implementation* of the information in mutable data structures and of the inferences in procedures and predefined data structures.

Figure 2: Theory Strucure

Below we show a picture of the relationships and dependencies between the various parts of the system.

During the first part of the course, we will essentially follow this particular proposal, concentrating on ontology, axioms and particular forms of entailment specialized for the microworld domain of time, action and change. We will evaluate the feasibility of different ontologies and entailment relations in terms of temporal scenarios or narratives, problem-specific instances of the microworld. Afterwords, we will question the feasibility of the microworld methodology and propose a different methodology for reasoning about action and change that emphasizes the features associated with the interaction our physical agents must have with the real non-idealized world in which they are meant to function. Since this methodology is relatively new and it is not known how much more fruitful it might be than the traditional methodology being used, it will be left to the reader to evaluate its potential.

6 Discussion

The intention of this lecture has been to introduce you to the logic-based approach to knowledge representation, its origins, the potential problems that have arisen and the beauty and elegance of the idea. Again, the verdict is still out as to the feasibility of this approach. In any case, during the course, we will consider both roles of logic in the context of reasoning about action and change.