Knowledge Representation

Ernest Davis, New York University, New York, NY, USA

Abstract

In artificial intelligence, knowledge representation is the study of how the beliefs, intentions, and value judgments of an intelligent agent can be expressed in a transparent, symbolic notation suitable for automated reasoning. From a purely computational point of view, the major objectives to be achieved are breadth of scope, expressivity, precision, support of efficient inference, learnability, robustness, and ease of construction. Knowledge-based techniques have been applied successfully for many computational tasks including text interpretation and cognitive robotics. Many different general architectures have been used for knowledge representation, including first-order logic, other formal logics, semantic networks, and frame-based systems. The representation of temporal knowledge is both a problem of central importance in knowledge representation and an archetype of the kinds of issues that arise in developing representations for various domains. The use of machine learning techniques for the automatic construction of knowledge bases and knowledge representations is difficult, but has achieved some degree of success.

Knowledge Representation: Ideal and Reality

The knowledge-based approach to artificial intelligence (AI) conjectures that a large component of any AI system with broad intelligence will be a knowledge base: a transparent, symbolic representation of the agent's beliefs, intentions, and value judgments, together with reasoning processes that modify the knowledge base in rational ways. Perceptual and motor processes connect the knowledge base to the outside world. The field of knowledge representation (KR) is the study of how such a knowledge base can be constructed (van Harmelen et al., 2008).

The representation of knowledge in a knowledge base should be symbolic, transparent, and modular. That is, the atomic elements of a knowledge base are symbols, each of which has an associated meaning. It is possible to identify small substructures of the knowledge base that correspond to specific beliefs, judgments, and intentions. For example, the belief that Holbein painted a portrait of Thomas More in oil might be represented in a predicate calculus formula like

$$\exists_X \text{ Oilpainting}(x) \wedge \text{Painter}(\text{Holbein}, x) \wedge \text{Depicts}(x, \text{ ThomasMore})$$

or in a fragment of a semantic net (Figure 1).

Here the symbols are '∃' (meaning 'there exists'), 'x' (a variable), '∧' (meaning 'and'), 'OilPainting', 'Painter', 'Depicts', 'Holbein', and 'ThomasMore'. The parentheses and the comma indicate the syntactic structure.

Not every belief of the agent can be explicitly expressed in the knowledge base. For example, someone who knows that Holbein painted a portrait of Thomas More will also know that Holbein knew how to paint, that Thomas More was painted, that Thomas More was born before Holbein's death, and so on. These ancillary beliefs are not each separately represented in the knowledge base. Rather, they are derived as needed by combining the particular fact with general knowledge about painting.

The construction of a broadly intelligent artificial agent is the ultimate objective for AI, but realistic short-term projects focus on specific, narrowly defined tasks. For such a limited project, the problem of KR is one of finding data structures that express the knowledge needed for the particular application.

Thus broadly construed, the problem of KR would appear to be equivalent to computer programming generally. Any program, after all, uses some data structures and incorporates some kind of knowledge. The field of KR is distinguished in its focus on a broad range of fundamental knowledge, particularly commonsensical world knowledge; on representations that are highly expressive, particularly of various natural forms of partial knowledge; and on forms of reasoning that arise in intelligent tasks. As AI is the study of tasks that are easy for people to carry out but difficult to automate, so is KR the design of data structures that express knowledge that people deal with easily, but are hard to encode in a data structure. The design of a KR can be very roughly divided into five parts.

1. Architecture: The design of general-purpose schemas for KR and procedures for reasoning with these schemas.
2. Content: The characterization of the knowledge that is to be represented, particularly for fundamental categories such as time, space, and communication.

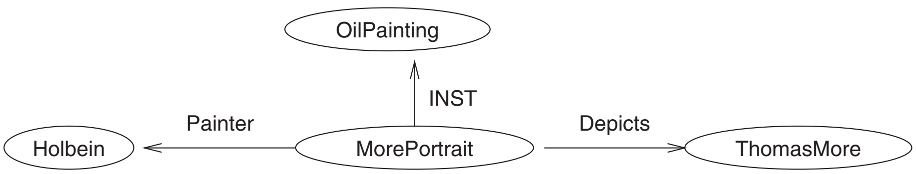

Figure 1 Fragment of a semantic net.

 http://dx.doi.org/10.1016/B978-0-08-097086-8.43048-5

3. Implementation: The specifics of instantiating the representation as a data structure and the reasoning process as a procedure.
4. Interface: The connection of the knowledge base to an external application.
5. Learning: Constructing the knowledge base, or parts of it, automatically by the analysis of a data corpus.

Desiderata

The design of a KR will generally aim toward the following objectives:

Scope: A KR for a limited application should cover all relevant aspects of the relevant domains. For a broadly intelligent aspect, an adequate KR theory must, first, describe the specifics of the representation of fundamental and universal human thought, such as time, space, simple physical categories, the human mind, and human relations (Davis, 1990; McCarthy, 1968), and, second, describe a general representational architecture that is applicable to any comprehensible object of thought.

Expressivity: The KR should be able to express the forms of partial and incomplete knowledge that naturally arise for the desired application.

For example, a standard form of representation is a collection of tables of values, known in computer science as a 'relational database'. Table 1 displays a table of paintings and a table of artists.

However, even within its narrow topical scope, such a representation may not achieve reasonable expressivity, because there is no way to express plausible partial states of knowledge such as

Delacroix was a generation older than Monet.
Rembrandt painted many oil paintings.
The Sistine Chapel and the sculpture of David were created by the same artist.

If such partial states of knowledge are important in the application, then a more expressive language must be found.

Here is another example: it is tempting to represent spatial information using a representation that corresponds to a single picture of the scene, sometimes called a diagrammatic representation (Glasgow et al., 1995). Such representations are easily understood and support efficient inference. However, they cannot be used to represent partial information. For instance, an agent might know that A is a mile from B and that A is a mile from C without knowing the distance from A to C. However, it is not possible for a single picture to display the distances from A to B and from A to C without committing itself to some particular distance between B and C. Therefore, this and other similar states of knowledge cannot be represented in a diagrammatic representation.

Precision: The meaning of a KR should be unambiguous, and, as far as the subject matter allows, well defined.

Support for reasoning: It must be possible to model the processes of reasoning involved in cognition in terms of well-defined operations over the KR. For example, as mentioned above, an agent who knows that Holbein painted a portrait of More should be able to conclude that Holbein knew how to paint, that More was born before Holbein died, and so on.

Efficiency of reasoning: The implementation of reasoning operations over the KR must also be efficient. Depending on the application, it may be necessary to draw conclusions from a large knowledge base in a fraction of a second.

Learnability: The KR should support modules that can learn from experience and can generalize individual cases to general rules.

Support for uncertainty: To be useful, a knowledge base cannot be limited to information known with certainty. Rather, the KR must be able to express relative degrees of certainty, and reasoning modules must use these (Pearl, 1988; Halpern, 2003).

Robustness: The information in a knowledge base will sometimes be wrong or corrupted. The effects of an error in the knowledge base should be limited in scope.

Ease of construction: Ideally, the information in a large knowledge base can be collected automatically or semi-automatically from existing data sets. If the knowledge base must be constructed manually, then a notation that can be used by users with little technical expertize is obviously preferable to one that requires a KR expert. The less the human labor, especially expert labor, required to build a knowledge base, the better.

Interface: A KR should support a suitable interface to the user or the application.

Expressivity and precision are features both of the architecture and of the content of a knowledge base. Scope is chiefly a feature of content, although expressivity and therefore architecture do bear on scope; some domains inherently require a very expressive language. Robustness, ease of construction, and support for reasoning, learning, and uncertainty are primarily features of the architecture. Efficiency of reasoning depends on the architecture, procedures, and class of problems being solved.

Table 1 Relational database of paintings and artists

Work	*Medium*	*Genre*	*Artist*
Last Supper	Fresco	Religious	da Vinci
Melancholia	Drawing	Allegory	Durer
Thomas More	Oil	Portrait	Holbein
Mona Lisa	Oil	Portrait	da Vinci

Artist	*Country*	*Birth*	*Death*
Durer	Germany	1471	1528
Holbein	Germany	1497	1543
da Vinci	Italy	1452	1519
O'Keefe	USA	1887	1986

Applications

KR and automated reasoning have been applied, with greater and lesser degrees of success, to a very wide range of intelligent tasks. Here we will discuss briefly the representational issues involved in two particular important categories of applications: the interpretation of natural language text and action planning for robotic systems.

Text Interpretation and Information Retrieval

Many useful computational tasks involving text, such as document retrieval, can be effectively carried out based purely on superficial features of the text, such as word counts, with no representation of the meaning of the text or knowledge of the domain it refers to. However, tasks such as answering questions necessarily engage with the meaning of the text; therefore, performing these well requires a representation of the text and of the underlying domain. Other tasks such as machine translation, summarization, and information extraction can be carried out without using KR to an accuracy that may be useful but cannot be carried out with very high quality. Basic components of text interpretation, especially disambiguation, often require the application of world knowledge. For example, in interpreting the sentence "Jane knocked on Susan's door, but she did not answer", disambiguating 'she' to mean 'Susan' requires understanding the conventions of answering knocks. (By contrast, if the sentence ended, 'but she did not get an answer', then 'she' would refer to Jane.)

Attempts to use knowledge-based techniques in text understanding vary widely in terms of the depth of the knowledge represented, the breadth of the subject matter, and the degree to which the knowledge base is created manually as opposed to assembled automatically. At one end of the spectrum are systems that form a complete logical representation of the explicit and implicit content of the text, and that draw on a detailed knowledge base in carrying out inferences from the text (Balduccini et al., 2008); these necessarily are highly limited in scope, and require massive human labor. At the other end are systems whose knowledge base is derived purely from statistical analysis of a corpus of texts; the knowledge base is essentially a table of correlations between words (Dumais, 2005).

As of the time of writing, perhaps the greatest successes have been achieved by information extraction systems that lie somewhere between these two extremes. Information extraction systems are designed to extract from text information of a restricted, often predetermined form, such as the employee, company, and position in a business page story about hiring a new executive. The systems use a combination of general linguistic analysis, specific linguistic patterns, pre-existing online language tools such as WordNet, and basic background knowledge of the domain such as a hierarchy of semantic categories and semantic constraints (e.g., an employee must be a person). Some parts of these are constructed manually for the particular task; some are drawn from existing manually created tools; some are developed automatically, using machine learning techniques, both supervised and unsupervised.

Robotics and Planning

The application of knowledge-based techniques to robot control, sometimes called 'cognitive robotics,' has focused primarily on high-level planning, although researchers have also looked at integrating automated reasoning with lower level manipulation (Reiter, 2001; Ghallab et al., 2004; LaValle, 2006). Central issues in cognitive robotics include the following:

- *Causal modeling*: Developing a theory that characterizes how the world changes, both in response to the robot's actions, and as a result of processes independent of the robot.
- *Modeling perception*: A high-level model of what the robot can expect to perceive while executing a plan, and how it can use the perceptions to guide its actions.
- *Developing a high-level language of plans*: In the simplest case, extensively studied, a plan is simply a sequence of actions to be executed, but in richer domains, a plan may involve conditionals, loops, interruptions, concurrency, and so on.
- *Projection*: Predicting what will happen if the robot attempts to execute a specified plan.
- *Planning*: Constructing a plan that achieves a specified goal.
- *Multiagent planning*: Constructing a plan for an environment with multiple agents. Agents may be cooperative, independent, or antagonistic; if cooperative, there may or may not exist a centralized control (the boss).

A number of autonomous robotic systems, including the Mars rover, have made some use of knowledge-based reasoning.

Architectures

A number of KR architectures are briefly surveyed below. The particular architectures included here have been included, in part because of their importance in KR research and applications, and in part to illustrate the range of possible approaches to KR.

Logic-Based Systems

A logic-based system represents knowledge as a set of sentences in a logical language. Reasoning is implemented as inference procedures that carry out deductive proofs authorized within the logical theory (Geneserth and Nilsson, 1987).

A logical theory characteristically consists of three parts:

- A syntax, which specifies what kinds of symbols are used, and how they are combined to form sentences.
- A semantics, which specifies how the meaning of a sentence relates to the meanings of the symbols it contains.
- A proof theory, which specifies what combinations of sentences constitute deductive proofs.

For example, first-order logic (also known as the predicate calculus) is the most widely used logic in KR. In first-order logic, there are seven kinds of symbols:

- Constant symbols represent individual entities. For example, the symbol Ad1517 may represent the year 1517.
- Predicate symbols represent properties or relations. For example, Painter may represent the two-place relation of a person painting a painting. Precedes may represent the relation of one event preceding another.
- Function symbols represent mappings. For example, BirthOf may represent the one-place mapping of a person to the event of his/her birth. CreationOf may represent the mapping of an artifact to the event of its creation.
- The Boolean operators are '$\wedge$' meaning 'and'; '$\vee$' meaning 'or'; '$\sim$' meaning 'not'; and '$\Rightarrow$' meaning 'implies'.
- Variables such as x, y, z represent indeterminate entities.
- The quantifiers are '$\forall$' meaning 'for all' and '$\exists$' meaning 'there exists'.

- Finally, the comma and the open and close parentheses indicate the structure of the sentence.

Constant, predicate, and function symbols are known as nonlogical symbols; they may be chosen, and their meanings assigned, by the designer of the knowledge base. Boolean operators, quantifiers, variables, and grouping symbols are known as logical symbols; their meanings are fixed by the logic. The distinction is roughly analogous to the distinction between open and closed categories in linguistics, or to the distinction between identifiers and reserved words in programming languages.

The syntactic rules for first-order logic establish that these symbols can be put together to form sentences such as the following:

$$\forall x,y \ \mathrm{Painter}(x,y) \Rightarrow \mathrm{Precedes}(\mathrm{BirthOf}(x),\mathrm{CreationOf}(y)) \quad (1)$$

The semantic rules establish that, given the above assignments of meanings to the symbols Painter, Precedes, BirthOf, and CreationOf, the sentence is true just if the birth of every painter precedes the creation of all his/her works. The proof theory establishes that, given sentence (1) above and the sentence Painter(Holbein,MorePortrait), it is possible to infer the following sentence:

Precedes(BirthOf(Holbein),CreationOf(MorePortrait))

This inference is valid regardless of the meaning of the nonlogical symbols (Mates, 1972).

Many other logics have also been used for KR. Some of these extend first-order logic to achieve greater expressivity. Modal logics define a variety of operators on sentences. For instance, given a base sentence, such as 'Bill Clinton is President,' sentences such as 'Bill Clinton was President,' 'Al Gore knows that Bill Clinton is President,' or 'Bill Clinton ought to be President' can be generated by applying temporal, epistemic, or deontic operators (Turner, 1984). Fuzzy logic allows the use of vague terms that hold to greater and lesser degrees; for example, 'George Washington was tall' is reasonably true, while 'Wilt Chamberlain was a tall person' is absolutely true (Zadeh, 1987). Probabilistic logics allow sentences to be tagged with a measure of certainty (Halpern, 2003).

Other KRs use logics less expressive than the predicate calculus in order to carry out inference more efficiently. For example, in the propositional calculus, the only logical symbols are Boolean operators, and a separate symbol must be used for each atomic proposition. Thus there is one symbol for the proposition, 'Leonardo painted the Last Supper,' another for the (false) proposition, 'Rodin painted the Last Supper,' and so on. General rules like formula (1) above can then only be stated by stating every instance for every pair of a painter and a painting. Despite this proliferation of symbols and formulas, however, as long as the underlying domain is not too large, it is often more effective to use the propositional calculus than the predicate calculus, because the best methods of inference known are much faster for the propositional calculus than the predicate calculus (Gomes et al., 2008).

The chief virtues of logic-based representation are precision and modularity. The meaning of a formula in a logic-based representation is completely and exactly fixed by the logic and the meaning of the symbols included. It is independent of any other formula in the knowledge base, and any other aspect of the knowledge base. Moreover, in a monotonic logic, the validity of any proof depends only on the formulas used in the proof and not on any other aspect of the knowledge base. Another important virtue is that procedures for deductive inference over logic-based knowledge bases are known. These procedures characteristically generate only valid inferences and all valid inferences (in the technical jargon, they are sound and complete), and, are in practice, they can often be implemented so as to find useful answers quickly.

Their chief drawback is difficulty of use. Building a knowledge base in a logic-based representation inevitably requires a large investment of expert labor. Moreover, even experts in KR find the encoding of complex domains in logical languages surprisingly difficult and error-prone. Logic-based languages also tend to be fragile. In principle, any inconsistency in any part of a logical theory invalidates the entire theory, and in practice small and subtle inconsistencies in foundational parts of a logic-based knowledge base can lead to bizarre and unpredictable results.

Semantic Networks

A semantic network consists of labeled nodes connected by labeled arcs (Figure 2). A node may represent an individual

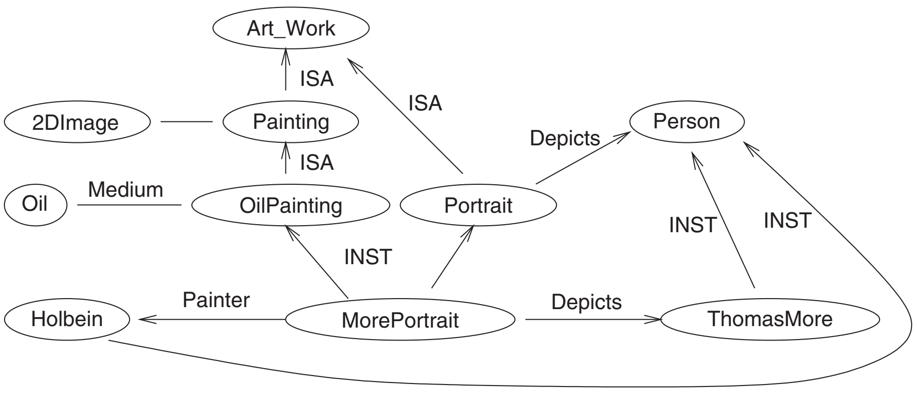

Figure 2 Semantic network consisting of labeled nodes connected by labeled arcs.

such as Holbein or a category such as oil paintings. An arc represents a two-place relation between nodes, such as the relation 'painter' between a work of art and a person.

The most important arcs in any semantic network are those labeled 'INST' and 'ISA.' An INST link goes from an individual *I* to a category *C* and asserts that *I* is an instance of *C*. An ISA link goes from category *C* to category *D* and asserts that *C* is a subset of *D*. Implicitly, therefore, if there is a chain of ISA links from *C* to *D*, then it can be inferred that *C* is a subset of *D*. If there is an INST link from *I* to *C* and then a chain of ISA links from *C* to *D*, then it can be inferred that *I* is an instance of *D*. For instance, in Figure 2, it can be inferred that oil paintings are works of art, and that the portrait of Thomas More is a work of art.

The central reasoning mechanism in a semantic network is inheritance: a category and its instances inherit the properties of the categories that contain them. For instance, in Figure 2, the node MorePortrait inherits the Medium arc to Oil from OilPainting, the Form arc to 2D-Image from Painting, and the Depicts arc to Person from Portrait. That is, the system can infer that the portrait of More is a two-dimensional image from the facts that it is a painting, and that all paintings are two-dimensional images.

The advantages of the semantic network architecture are that it is easy and natural to use, its meaning can be defined precisely, and the inheritance mechanism is easy to implement and efficient. The chief drawback is that it is not very expressive. In addition, the apparent simplicity of the representation can end up working against precision; an inexpert user can easily construct a semantic network that looks plausible, but that uses the same notation with two different meanings (Woods, 1975).

The minimal architecture described above is often supplemented with additional features, such as 'meta-arcs' that connect two arcs and describe the logical relations between them (Sowa, 2000). However, the more such features are added, the less simple and natural the representation becomes.

Frame-Based Systems

A frame represents either an individual or a category, and associates with its subject a collection of pairs of a slot with a corresponding filler. A slot is the name of a feature of an individual, and a filler for a slot of an individual is the value of that feature; for example, the Color slot for StatueOf Liberty has filler Green. The filler of a slot for a category contains information about the possible fillers of that slot for individual instances of the category. This can take many different forms; it can be a particular value, a range of values, a constraint on values, or a function that computes the value.

For instance, in frame OilPainting in Figure 3, the value of the Date slot is a constraint asserting that the date of a painting must take place within the painter's lifetime. (The specific syntax used here is invented, but the features are typical of frame-based systems.) As in semantic nets, frames inherit fillers through INST links, which connect an individual to a category, and through ISA links, which connect a subcategory to a supercategory. Other forms of inference in frame-based systems include computing or verifying the value of a slot using inherited procedures or constraints and categorizing a new

```
Frame MorePortrait
  Inst:OilPainting;
  Genre:Portrait;
  Depicts:ThomasMore;
  Shape:Rectangle (29.5 inch, 23.7 inch);
  Surface:Wood;
  Date:1527;
  Painter:Holbein;
  Owner:Frick;
  Image:more.gif {file with displayable image};
EndFrame

Frame OilPainting
  Isa:Painting;
  Medium:Oil;
  Surface: <OR> [Canvas, Wood];
  Date: <CONSTRAINT> [@.Painter.Birth <@.Date ≤ @.Painter.Death];
  Painter: <OR> [ <INSTANCE> Person, <INSTANCE> Workshop];
  Owner: <OR> [Lost, Destroyed, <INSTANCE> Person, <INSTANCE> Institution];
EndFrame
```

Figure 3 Frame-based systems.

individual as an instance of a category based on its features (Minsky, 1975) (Borgida et al., 1989).

Architectures for Large-Scale Systems

Several representational architectures are available for use in large-scale KR. The best known large-scale representational project is CYC, which has been under development since 1984 by a team led by Douglas Lenat (1995). The representation language is called CYCL. CYC is a commercial product, and the system as a whole is proprietary. Although substantial parts, including the structure of CYCL, the upper ontology, and a fraction of the content are openly available, very little has been published about CYC, and scientists outside the project have consistently found it difficult to determine how it can be effectively used, what content it contains, and how it has been evaluated.

Other representational architectures for large-scale systems include KIF (Knowledge Interchange Format) developed at Stanford and KQML (Knowledge Query and Manipulation Language) developed at the University of Maryland. All these contain a central logical component with many extensions of various kinds.

Fundamental Domains

The most difficult problems facing the designer of a knowledge base are those of content: what knowledge is needed for a given application, what concepts are needed, and how can the knowledge be effectively encoded in a given architecture?

KRs for specialized applications generally involve a protracted collaboration between the designers of the knowledge base and the domain experts. The interaction brings to light central concepts, distinctions, and forms of inference in the applications that are not evident to either group at the start (Friedland et al., 2004).

In encoding nonspecialized, commonsense knowledge, a popular methodology is to begin by defining a microworld, a small, coherent domain of study. Aspects of the real world that lie outside the microworld are ignored or dealt with ad hoc. A collection of commonsensically obvious inferences in the microworld is assembled. The researcher constructs a formal model that suffices to justify these inferences, defines symbols to express the concepts used in the inferences, and expresses the knowledge involved in the inferences in terms of these symbols in the selected representational format (Davis, 1990, 1998).

Time

The issues involved in content may be illustrated in a discussion of representations of time, certainly the most ubiquitous domain in automated reasoning.

Ontologically, time may be taken to be discrete or continuous, to be linear or branching, and to consist of points or of intervals. Which choice is made depends on the applications being considered. For instance, in a chess-playing program, it is natural to take time to be discrete, as nothing significant happens between successive states of the board, and to be branching, to express the options available to each player at each stage. In reasoning about physical processes, it is usually desirable to take time to be continuous, in order to describe continuous change, and linear, since no choices are being made. (Uncertainty about the future, as opposed to choices about the future, is easily accommodated in a linear theory of time.)

Many different representations have been devised for time. Some of the most popular include the following:

- *Time stamps*: If a linear model of time is used, and clock times are known precisely and are important, then each event can be marked by date and clock time.
- *Numerical constraints on time*: If a linear model of time is used, and clock times are important but are not fully known, then a reasonable representation can use a collection of symbolic constraints between the times of events, such as 'The date of the painting of More's portrait comes after Holbein's birth and before his death.'
- *The situation calculus*: This theory uses a branching, discrete model of time. There are three primary types of entities: situations, which are essentially instants of time; fluents, which are propositions whose value changes over time, such as 'Obama is President'; and actions, which cause one situation to change to a resultant situation. The two principal symbols are Result(s,a), which maps a situation s and an action a to the situation that results if a is performed in s, and Holds(s,f), the assertion that fluent f holds in situation s (McCarthy and Hayes, 1969).
- *The event/interval calculus*: Another representation for time popular in KR research is based around intervals and events. There are 13 possible temporal relations between intervals; interval i may entirely precede interval j or the end of i may coincide with the beginning of j and so on. The interval language contains primitives to describe these 13 relations, plus the relation Occurs(i,e), meaning event e occurs in interval i (Allen, 1983).
- *Tense logic*: This was developed in the philosophical literature to formalize the tenses of natural language. It is a modal logic, containing operators like 'φ will always be true,' 'φ has been true at some time in the past,' and so on (van Benthem, 1983).
- *Dynamic logic*: This representation is popular in formal theories of programming languages. It uses a discrete, branching model of time. Statements have the form 'after p has been executed, φ will be true', where p is an action or a computer program.

Other Domains

Other commonsense domains that have been extensively studied in the KR literature include the following:

- Spatial reasoning, in particular the development of reasoning systems that can effectively use qualitative information about shape and other spatial relations, rather than requiring exact geometric information.
- Spatiotemporal reasoning, in particular reasoning based on the fact that spatial relations change continuously over time.
- *Physical reasoning*: Here the challenge is to characterize the physical world in the terms in which the human agent encounters it, sometimes called the mesoscopic scale.
- *Mental states of agent*: Knowing, believing, hoping, fearing, and so on. The chief challenge here is to develop a theory

that is precise enough to support useful reasoning but not so idealized as to be entirely unrealistic.
- Communication between agents.

Learning and Knowledge Representation

In view of the immensity of the task of hand encoding a knowledge base of all commonsense knowledge, or even of developing a representation language adequate for such a knowledge base, and in view of the great success that machine learning has had in the past 20 years in many different tasks, the proposal has naturally been made that machine learning techniques be employed to develop both the representation language and the content for a knowledge (Etzioni et al., 2008).

Almost all such recent projects work from texts gathered from the World Wide Web. Projects differ widely in terms of the kinds of information that is gathered:

- The program may have a fixed set of categories and relations and may gather specific instances of these. For instance, the program may know of the category Person and the relation Father, and learn from text that Odysseus was a person and that Odysseus was the father of Telemachus.
- The program may learn both new relations and new instances; for instance, it may learn that Athlete and Team are categories, that Babe Ruth is an athlete, and that PlaysFor is a relation.
- The program may additionally learn semantic relations between categories and relations. For instance, it may learn that Athlete is a subcategory of Person and that PlaysFor is a relation between Athlete and Team.
- The program may learn more complex logical rules or probabilistic correlations, such as, 'If food F is made from ingredient G and G contains chemical C, then F contains chemical C.'

In general, such projects work by building up, side-by-side with the conceptual knowledge base, a collection of linguistic patterns that indicate the target relation and that can be matched against the text. For instance, if a phrase in the text matches the pattern '<*Class*> such as <*Instances*>,' it suggests that the enumerated instances in the text may be examples of the specified category. The accuracy and the coverage of any individual pattern may be low, but if many such patterns are used and are applied over a large corpus of text, the combined evidence may be substantial.

See also: Axiomatic Theories; Concept Learning and Representation: Models; Deductive Reasoning Systems; Logics for Knowledge Representation; Nonstandard Reasoning.

Bibliography

Allen, J., 1983. Maintaining knowledge about temporal intervals. Communications of the ACM 26 (11), 832–843.

Balduccini, M., Baral, C., Lierler, Y., 2008. Knowledge representation and question answering. In: van Harmelen, F., Lifschitz, V., Porter, B. (Eds.), Handbook of Knowledge Representation, Elsevier, Amsterdam, pp. 779–820.

Borgida, A., Brachman, R., McGuiness, D., Resnick, L., 1989. CLASSIC: a structural data model for objects. In: Proc. SIGMOD '89. Oregon, Portland, pp. 59–67.

Davis, E., 1990. Representations of Commonsense Knowledge. Morgan Kaufmann, San Mateo, California.

Davis, E., 1998. The naive physics perplex. Artificial Intelligence Magazine 19, 51–79.

Dumais, S., 2005. Latent semantic analysis. Annual Review of Information Science and Technology 38, 188–230.

Etzioni, O., Banko, M., Soderland, S., Weld, D., 2008. Open information extraction from the web. Communications of the ACM 51, 68–74.

Friedland, N., et al., 2004. Project Halo: toward a digital Aristotle. Artificial Intelligence Magazine 25, 29–47.

Genesereth, M., Nilsson, N., 1987. Logical Foundations of Artificial Intelligence. Morgan Kaufmann, San Mateo, California.

Ghallab, M., Nau, D., Traverso, P., 2004. Automated Planning: Theory and Practice. Morgan Kaufmann, San Mateo California.

Glasgow, J., Narayanan, N., Chandrasekaran, B., 1995. Diagrammatic Reasoning. MIT Press, Cambridge, Massachusetts.

Gomes, C., Kautz, H., Sabharawal, A., Selman, B., 2008. Satisfiability solvers. In: van Harmelen, F., Lifschitz, V., Porter, B. (Eds.), Handbook of Knowledge Representation, Elsevier, Amsterdam, pp. 89–134.

Halpern, J., 2003. Reasoning About Uncertainty. MIT Press, Cambridge.

LaValle, S., 2006. Planning Algorithms. Cambridge University Press, Cambridge.

Lenat, D., 1995. CYC: a large-scale investment in knowledge infrastructure. Communications of the ACM 38 (11), 33–38.

Mates, B., 1972. Elementary Logic. Oxford University Press, Oxford.

McCarthy, J., 1968. Programs with common sense. In: Minsky, M. (Ed.), Semantic Information Processing. MIT Press, Cambridge, Massachusetts, pp. 403–418.

McCarthy, J., Hayes, P., 1969. Some philosophical problems from the standpoint of artificial intelligence. In: Metzler, B., Michie, D. (Eds.), Machine Learning 4. Edinburgh U. Press, Edinburgh.

Minsky, M., 1975. A framework for representing knowledge. In: Winston, P. (Ed.), The Psychology of Computer Vision. McGraw-Hill, New York, pp. 211–280.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Mateo, California.

Reiter, R., 2001. Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press, Cambridge, MA.

Sowa, J., 2000. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole, Pacific Grove, California.

Turner, R., 1984. Logics for Artificial Intelligence. Wiley, New York.

van Benthem, J., 1983. The Logic of Time. Reidel Pubs, Dordrecht.

van Harmelen, F., Lifschitz, V., Porter, B., 2008. Handbook of Knowledge Representation. Elsevier, Amsterdam.

Woods, W., 1975. What's in a link: foundations for semantic networks. In: Bobrow, D., Collins, A. (Eds.), Representation and Understanding: Studies in Cognitive Science. Academic Press, New York, pp. 35–82.

Zadeh, L., 1987. Commonsense and fuzzy logic. In: Cercone, N., McCalla, G. (Eds.), The Knowledge Frontier: Essays in the Representation of Knowledge. Springer-Verlag, New York, pp. 103–136.