Journal of Artificial Intelligence Research 59 (2017) 651-723 Submitted 09/16; published 08/17

Logical Formalizations of Commonsense Reasoning: A Survey

Ernest Davis DAVISE@CS.NYU.EDU
Department of Computer Science, New York University
251 Mercer St.
New York, NY 10012 USA

Abstract

Commonsense reasoning is in principle a central problem in artificial intelligence, but it is a very difficult one. One approach that has been pursued since the earliest days of the field has been to encode commonsense knowledge as statements in a logic-based representation language and to implement commonsense reasoning as some form of logical inference. This paper surveys the use of logic-based representations of commonsense knowledge in artificial intelligence research.

1. Introduction

The automation of commonsense reasoning has been recognized as a central problem in artificial intelligence since the inception of the field sixty years ago. One of the most studied approaches toward this goal has been to use formal mathematical logic. In this approach, commonsense knowledge is represented as a theory in a language based on some well-defined logic, and some substantial part of commonsense reasoning is implemented as valid (or approximately valid) inference with respect to the logic. This paper surveys this line of research, focusing on the representations that have been developed for various commonsense domains, rather than on inference techniques.

The paper is addressed to a reader with general knowledge of artificial intelligence, and some basic knowledge of formal logic, but readers who know less than that or more may still find it useful.

Consider what would be involved in creating an AI program that could understand texts such as the following two samples.

> **Sample 1:** On a mundane morning in late summer in Paris, the impossible happened. The Mona Lisa vanished. On Sunday evening, August 20, 1911, Leonardo da Vinci's best-known painting was hanging in her usual place on the wall of the Salon Carré between Correggio's Mystical Marriage and Titian's Allegory of Alfonso d'Avalos. On Tuesday morning, when the Louvre reopened to the public, she was gone. Within hours of the discovery of the empty frame, stashed behind a radiator, the story broke in an extra edition of Le Temps, the leading morning newspaper. Incredulous reporters from local papers and international news services converged on the museum. Georges Bénédite, the acting director, and his curators were speculating freely to the press.
> — "The story behind the theft of the Mona Lisa" (Scotti, 2009)

Figure 1: A lake divides into two lakes when the water level falls

> **Sample 2:** In allopatric speciation (from the Greek *allos*, other, and *patra*, homeland) gene flow is interrupted when a population is divided into geographically isolated subpopulations. For example, the water level in a lake may subside, resulting in two or more smaller lakes that are now home to separated populations (see figure [1]). Or a river may change course and divide a population of animals that cannot cross it.
> — *Campbell Biology* (Reece et al., 2011)

These texts draw on basic concepts and presume knowledge of basic facts from a variety of familiar domains:

- Time. The Mona Lisa was stolen *between* Sunday night and Tuesday morning. The species separates into two some generations *after* the the subpopulations are separated.
- Space. The Mona Lisa is taken *outside* the Louvre. The lake is originally a *connected* whole; when the water level *is lowered* it becomes two *separate* pieces.
- Physics. A painting cannot literally disappear; nor can it travel off a wall where it is hanging by itself. A river can change course, though this is rare.
- Biology. Two animals need to come close together in order to interbreed.
- Folk psychology. The museum curators do not *know* where the Mona Lisa is. They *hope* it will be recovered.
- Social institutions. A museum can own a painting.
- Facts about language. A word in one language can be formed out of two words in a different language.

It seems plausible, therefore, that if one wishes to build an AI program that can understand these kinds of texts — more concretely, that can answer simple comprehension questions such as "What happened to the Mona Lisa?" or "What would happen if the water level in the lake rose again, a month later?" — it is worthwhile trying to determine what are the concepts and facts in these commonsense domains.

To clarify: We do not, by any means, know how to represent all or most of the commonsense knowledge needed to understand these or comparable texts. In the domains of biology, social institution, and facts about language, little is known about the representation of commonsense knowledge or the automation of commonsense reasoning. In the domains of space, physics, and folk psychology, considerable work has been done, but much remains unknown; we do not how to represent the domain knowledge needed to understand these two particular passages. The representation of temporal information is very well understood, but that is the only domain in which representation is close to being a solved problem. This paper focuses primarily on the work that has been done and the theories that have been developed, but also discusses some of the important problems not yet solved.

Section 2 introduces the issue of commonsense knowledge and discusses its role in artificial intelligence. Section 3 discusses the various kinds of logic that have been studied as representations of commonsense knowledge. Section 4 discusses mereology, the analysis of the relation "X is part of Y" Section 5 discusses taxonomic knowledge. Sections 6-10 discuss specific domains: time, space, physics, knowledge and belief, folk psychology, and interactions between people. Section 11 briefly surveys some alternative approaches to collecting and representing commonsense knowledge. Section 12 discusses related undertakings in fields other than artificial intelligence. Some of the material is included here because it seems to me particularly central or important; some is included because it has been extensively studied; these are not identical.

2. Commonsense Knowledge

By "commonsense knowledge" we mean, roughly, what a typical seven year old[1] knows about the world, including fundamental categories like time and space, and specific domains such as physical objects and substances; plants, animals, and other natural entities; humans, their psychology, and their interactions; and society at large. We will not attempt to be precise about this, but let us indicate roughly which issues we are considering and which we are ignoring. Obviously, this body of knowledge in fact depends on place, time, culture, social standing, personal characteristics (e.g. unusual cognitive or physical abilities or disabilities), schooling, perhaps on language. We ignore all that; without embarrassment, we have in mind a 21st-century, first-world, urban, middle-class, normally-abled child, with an appropriate level of schooling, whose native language is English if that matters. We exclude book-learning, material explicitly taught at school. We largely exclude specialized knowledge, such as the fact that a king in a deck of cards counts for more than a jack; knowledge of conventions, such as the fact that the symbol $\triangleright$ indicates a "Play" button or that, in the US, one drives on the right side of the street; and knowledge of the specifics of language, such as the meanings of individual words. However, we include knowledge *about* these, such as the fact that people play cards for fun, drive cars to get places, and use language to communicate.

Commonsense knowledge and commonsense reasoning are relevant in many different AI tasks. A few examples:

- Natural language processing. Resolving ambiguities in natural language text and utterances is a matter of finding the most plausible interpretation; and determining the

1. A seven year old would find the jargon in the above quote about speciation daunting, but could certainly understand the idea if it were explained.

plausibility of an interpretation often requires commonsense reasoning. For instance, to understand the story about the Mona Lisa, a program must infer that the painting was stolen (never stated explicitly in the paragraph quoted.) Many additional, smaller-scale inferences are also involved; for instance the program must realize that that "The Mona Lisa vanished" is a figure of speech, because paintings do not literally vanish.

- Visual interpretation, particularly of video. Consider, for instance, what background knowledge would be required for an AI program that could watch the "horse's head" scene in *The Godfather* and understand that Hagen is threatening to kill Woltz unless he coöperates. Such a program must understand a great deal about wordless communication, threats, and the difficulties of decapitating someone else's horse and sneaking its head into their bedroom.

- Robotics. For instance, if a waiter robot goes to get a drink for a guest at a party, it should know not to use a glass that is broken, is dirty, has a cockroach in it, or has soap in it.

However, currently the problems of representing and using commonsense knowledge are in practice so daunting that most successful AI applications entirely avoid the use of any commonsense knowledge, and comparatively few practical applications go beyond a few elementary forms of commonsense reasoning. In practical applications such as web search, question answering, machine translation, computer vision, and robotics, the use of commonsense reasoning is minimal or non-existent. The work described in this paper is therefore, so far, mostly theoretical.

2.1 What Claims are Made for Logic-Based Analysis?

When a scientist proposes a logic-based theory for some form of reasoning, such as those discussed in this paper, either for use in an AI system or as a cognitive model, there are a number of different possible interpretations of what might be meant.[2]

The strongest claim is that analysis is valid at the *implementational* level; that some substantial part of carrying out of intelligent tasks can be implemented in building an AI system, or can be modeled in studying a cognitive system, as the application of a general-purpose theorem prover to a knowledge base that consists of sentences in the logical language. As we will discuss below, successful AI systems have been built in this way, though it is by no means the currently dominant approach to AI. As a cognitive theory, it would clearly require an explanation of how an effective symbolic theorem prover can be built using the wetware of neurons and the architecture of the brain.

A weaker claim is that the analysis has some validity at the *conceptual level*; the concepts that arise in the logical analysis correspond closely to the concepts explicitly involved in the intelligent system. For an AI system, this would presumably mean that one can connect the concept in the logical analysis to some symbol or data structure in the program, and

2. Marr's (1982) well-known distinction between theories at the computational level, the algorithmic level, and the implementational level is similar. However, we are limiting the discussion to logical theories, whereas Marr was considering theories in general; therefore, we are able to be more specific about the distinction between these levels.

one can show that there is some correspondence between the way the symbol is used in the program and the way that it is used in logical inferences in the theory involved in the intelligent system. As a cognitive claim, it would mean that the symbols in the theory correspond to concepts in the human or animal thinker; what exactly that amounts to is hard to say, since what it means for a person to have or use a concept is currently poorly understood (Murphy, 2002). For example, if an AI system uses the same symbols as a logical theory, but uses some non-logical form of reasoning, such as spreading activation, Bayesian networks, or reasoning by analogy (Falkenhainer, Forbus, & Gentner, 1989), then the logical theory would be valid at the conceptual level but not at the implementational level. The value of the theory in this case would be in revealing which domain concepts are important for reasoning.

The weakest claim is that the analysis is purely at the *knowledge level* (Newell, 1981). That is, the analysis characterizes abstractly what the reasoner knows about the problem and about the world and how that knowledge is involved in solving the task, but makes no claims about how the knowledge is represented or how the reasoning takes place. As a (perhaps misleading) analogy: Differential equations can be characterized logically by axiomatizing the theory of the real numbers and giving formal definitions of concepts such as the derivative. However, none of the axioms and few of the concepts are represented symbolically in a program that uses the Runge-Kutta method to numerically solves differential equations. The value of the formal theory is at the meta-level. The program is designed using an understanding of the mathematical theory, and the workings of the program can be justified in terms of the mathematical theory; indeed formal proofs of the correctness of the program can be constructed, However, the relation between the program and the theory is extremely indirect; nothing in the program "looks like" the theory.

McCarthy and Hayes (1969) also proposed a useful taxonomy of different senses in which a representation can be adequate to a domain. They write:

> A representation is called *metaphysically* adequate if the world could have that form without contradicting the facts of the aspects of reality that interests us. . . . [For example, the] representation of the world as a giant, quantum-mechanical wave function. . . .
>
> A representation is called *epistemically adequate* for a person or machine if it can be used practically to represent the facts one actually has about the world. . . .
>
> A representation is called *heuristically adequate* if the reasoning processes actually gone through in solving a problem are expressible in the language.

2.2 General Reading

From the earliest days of artificial intelligence, the representation of commonsense knowledge and the automation of commonsense reasoning has been identified as a key problem for AI in general (McCarthy, 1959, 1963, 1968) and for natural language processing in particular (Bar Hillel, 1960).

Representations of Commonsense Knowledge (Davis, 1990) is a textbook on logic-based representations of commonsense knowledge. *Commonsense Reasoning* (Mueller, 2006) is also a textbook on commonsense reasoning, chiefly focused on the use of the event calculus in understanding narrative. *Handbook of Knowledge Representation* (van Harmelen, Lifschitz, & Porter, 2008) is a collection of 25 survey articles covering the field of knowledge representation; this is certainly the most comprehensive presentation of the current state of the art. Davis and Marcus (2015) give a non-technical survey of the state of the art.

Knowledge Representation and Reasoning (Brachman & Levesque, 2004) is a textbook on knowledge representation with emphasis on logic-based representations. *Readings in Knowledge Representation* (Brachman & Levesque, 1985) and *Formal Theories of the Commonsense World* (Hobbs & Moore, 1985) are important collections that include many essays relevant to the material in this paper. *Formalizing Common Sense: Papers by John McCarthy* (Lifschitz, 1990) is a collection of the papers of John McCarthy on formalizing common sense.

The International Symposium on Logical Formalizations of Commonsense Reasoning meets every two years; its website is at http://commonsensereasoning.org.

The area of applied ontology addresses the development of ontology and knowledge representation for specific, often expert-level, applications. There is a substantial overlap between that area and the formalizations of commonsense reasoning discussed in this survey, not least because application-oriented reasoning often draws on commonsense reasoning as a foundation, and application-specific knowledge in a domain is an extension of commonsense knowledge of that domain. The International Conference on Principles of Knowledge Representation and Reasoning and the International Conference on Formal Ontology in Information Systems often have papers relevant both to logical representation of commonsense reasoning and to applied ontology.

3. Logic

A number of different strategies have been pursued in the encoding of commonsense knowledge. This paper focuses on developing representations of fundamental commonsense domain by hand by experts using mathematical logic as a framework. In section 11, we will briefly survey alternative approaches, such as web mining and crowd-sourcing.

A *logic* is a theoretical framework in which knowledge can be expressed symbolically and reasoning can be characterized. A knowledge engineer chooses a logic; uses the logic to define a formal *language*; and then expresses the knowledge he wishes to encode as *sentences* in the language. A *theory* is a set of sentences. Very roughly, a knowledge engineer is analogous to a programmer; a logic is analogous to a programming language; constructing a formal language is analogous to choosing a collection of identifiers and deciding on their meaning; and formulating a theory is analogous to writing a program.[3]

A logic consists of the following components:

- Syntax. A specification of the kinds of symbols that can be used and how they can be combined into sentences. Symbols are divided into *logical symbols*, (analogous to reserved words in a programming language) whose form and meaning is specified by the logic; and *non-logical symbols* (analogous to identifiers), whose form and meaning is specified by the knowledge engineer.

3. The aim of logic programming languages (Kowalski, 1979) is to turn this analogy into an identity.

- Semantics. The semantics of a logic is a mathematical characterization of the meaning of sentences in a way that supports a definition of entailment. In the literature discussed here, the most common approach to semantics is compositional, model-theoretic semantics. In this approach, the meaning of symbols and sentences are characterized in terms of a mathematical model,[4] and the logic supplies rules that define the meaning of a sentence in terms of its syntactic structure and the meaning of the symbols.

- Inference method: A method of symbolic computation that determines whether a sentence is a logical consequence of some set of axioms.

A logic-based representation is, necessarily, a declarative representation. That is, the knowledge of the real world and of the characteristics of the task to be executed and the specifications of the particular problem being addressed are encoded, largely or entirely, in a knowledge base expressed in a language consisting of sentences built out of symbols. Each symbol has a particular meaning or denotation, grounded in the real world; that is, a human reader of the knowledge base can say what the symbol means. Each sentence has a meaning grounded in the real world, which is determined by its constituent symbols and its structure; that is, a human reader who knows the meaning of the symbols can say what the sentence asserts about the real world, and what it would mean for the sentence to be true.

As contrasts to clarify the point, let us give a couple of examples of kinds of data structures that are not declarative representations. First, a neural network is not a declarative representation. The state of a neural network and the knowledge that it implicitly contains are encoded in numerical parameters. Only occasionally and fortuitously can an individual parameter or individual cell be associated with some specific intelligible feature of the world or of the task; and rarely if ever is it possible to associate any feature of the neural network with the knowledge of a general rule. Second, the data structures in a conventional computer program that carries out some AI task do not generally constitute declarative representation, even though the program is built compositionally out of symbols, because most of the symbols and the higher-level structures such as statements and procedures do not have a meaning in terms of the external world; they refer only to the internal state of the computation.

Not all declarative representations are logic-based. For example a system that represents the meaning of words in terms of a 1000 dimensional feature vector satisfies the above definition of a declarative representation but is hardly a logic-based representation. Logic-based systems are characterized by the following characteristics; these are matters of degree, rather than hard and fast rules, so a system can be logic-based to a greater or lesser extent.

1. The syntactic rules for combining symbols into sentences and the semantic rules for combining the meanings of symbols into the meanings of sentences are reasonably

4. In the knowledge representation literature, "model" is a treacherous word, used with multiple different meanings and shades of meaning. I have not attempted to impose a greater consistency in this paper. In the discussions of formal logic in this section, I use the word in the precise sense of mathematical logic: A model is a set-theoretic structure that grounds a symbolic theory. In later parts of this paper, I follow the KR literature in using the word more loosely to mean, at times, a theory, a category of theories, an approach to knowledge representations, perhaps other things. *Caveat lector.*

straightforward. This excludes representations like natural language with its intricate syntactic and semantic rules.

2. There is a well-defined model, and the denotation of the symbols is defined in terms of that model. Thus, for example, in a theory of space, if the symbols are given strict geometric definitions, then that conforms well to this requirement. If the symbols are simply English words, with no effort to disambiguate the different spatial relations they can entail — e.g. `In` is used as a predicate, with no attempt to distinguish the different meanings of the word as used in "the soup is in the bowl", "Boise is in Idaho", "the reflection in the mirror" "there is a crack in the bowl", and so on — then that fails this criterion. A representation that carefully discriminated the different meanings as $\mathtt{In}_1, \mathtt{In}_2$ and so on, but did not provide a geometric interpretation would be at an intermediate level.

3. The meanings of symbols and sentences are context-independent; again, in contrast to both natural language and programming languages.

4. The reasoning carried out by the program largely corresponds to inferences within the logic; and it is best described in terms of logical inference. As a negative example, consider a program that does reasoning in terms of distances between feature vectors. It is no doubt *possible* to formulate a logic-like structure that will justify operations of this kind, but that is hardly the most *natural* or *useful* way of describing them.

   This condition does not require that inference be carried out using general purpose deduction techniques, such as resolution (Genesereth & Nilsson, 1987). In particular it does not exclude the use of numerical calculations on representations that include numerical parameters. (Complex calculations with floating point numbers require care to ensure validity, as always.)

3.1 General Reading

There are many introductory textbooks for mathematical logic that cover propositional logic and first-order logic; for example, *Elementary Logic* (Mates, 1972) and *Introduction to Mathematical Logic* (Mendelson, 1979). *Logics for Artificial Intelligence* (Turner, 1980) and *Logical Foundations of Artificial Intelligence* (Genesereth & Nilsson, 1987) survey the use of logic in artificial intelligence.

The use of formal logic as a framework for representing commonsense knowledge in AI systems was first proposed by John McCarthy, (1959). Numerous papers have advocated a logic-based approach to representation and reasoning (Green, 1969; McCarthy & Hayes, 1969; Hayes, 1977; Hayes, 1978; McDermott, 1978; Moore, 1982; Thomason, 2003).

Lifschitz, Morgenstern, and Plaisted (2008) combine an introduction to the use of logic in knowledge representation with a survey of the state of the art in automated deduction.

Minsky (1975) and McDermott (1987) wrote important critiques of the use of logic in AI. However, the diminished importance of logic in AI technology has little to do with the issues raised in these critiques and everything to do with the success of corpus-based machine learning, which neither Minsky nor McDermott anticipated or welcomed. Charniak (1993) discusses his own conversion from knowledge-based methods to corpus-based statistics.

`On(BlockA,BlockB,0).  On(BlockB,Table,0).`
`On(BlockC,Table,0).  Clear(BlockA,0).`
`Clear(BlockC,0).`
	Starting state.

`On(BlockC,BlockB,3).`	*Goal state.*
$\neg$`[On(BlockA,BlockC,2)` $\wedge$ `On(BlockB,BlockC,2)].`	*Domain Constraint.*
`PutOn(BlockA,BlockC,0)` $\Rightarrow$ `On(BlockA,BlockC,1).`	*Causal Axiom.*
`PutOn(BlockA,BlockC,0)` $\Rightarrow$ `Clear(BlockC,0)` $\wedge$ `Clear(BlockA,0).`	*Precondition Axiom.*
`PutOn(BlockA,BlockB,2)` $\Rightarrow$ `[On(BlockC,Table,2)` $\Leftrightarrow$ `On(BlockC,Table,3)].`	*Frame Axiom.*

Table 1: Axioms from a propositional encoding of a blocks-world planning problem

3.2 Propositional Logic

In *propositional logic* a sentence consists of some number of atomic ground formulas connected by Boolean operators, e.g.

$\texttt{On}(\texttt{BlockA}, \texttt{Table})$.
$\neg\texttt{Red}(\texttt{BlockA})$.
$\texttt{On}(\texttt{BlockA}, \texttt{Table}) \vee \texttt{On}(\texttt{BlockA}, \texttt{BlockB})$

(Throughout this paper, we will use `typewriter font` to indicate formulas in a formal language.)

The logical symbols are the Boolean operators. As far as the logic is concerned, the non-logical symbols are atomic (i.e. do not have meaningful subparts); e.g. "`On(BlockA,Table)`" above is simply a string of 16 arbitrary characters; it might as well be `P1074`. A human knowledge engineer may choose to construct them in some kind of compositional form, as above.

General rules cannot be stated in propositional logic, so the logic is not in itself adequate for axiomatizing any domain. However, the propositional calculus does often suffice to solve specific problems. The problem specification, and all potentially relevant instances of the general axioms are asserted in the propositional calculus, and then the logic allows inference to be done over these. For instance, in a SAT-based planner (Kautz, McAllester, & Selman, 1996), given a problem specification, the planner generates an atom for every possible value of every fluent (time-varying term) and for every possible action at each of a collection of time points. The planner generates sentences that characterize the relations between fluent values and actions. Table 1 shows a few of the sentences that would be generated as a propositional encoding of the blocks world planning problem in figure 2. (This kind of axiomatization is discussed further in section 6.2).

In the worst case, the problem of finding a valuation satisfying a set of sentences in the calculus is NP-hard (intractable). However, in the last two decades, inference engines have been developed that, in practice, can often find satisfying solutions or carry out inferences for enormous sets of sentences, with thousands or millions of atoms. This technology has had extensive practical applications in software and hardware verification (Prasad, Biere, & Gupta, 2005; Gomes et al., 2008).

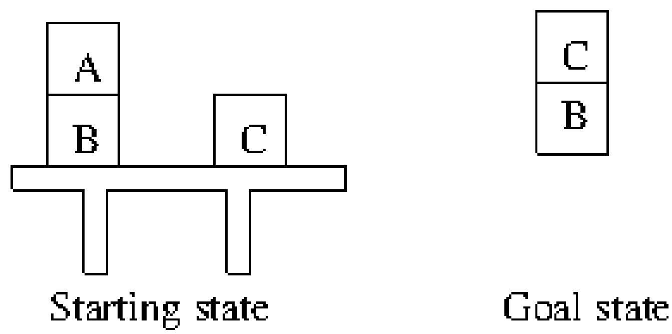


Figure 2: Blocks-world planning problem

Constraint networks are, from a logical perspective, propositional theories of a particular structure. In practice, the use of a constraint network can make the expressions of certain kinds of information more natural, and can accommodate particular forms of inference, such as various forms of Waltz propagation (Dechter, 2003; Rossi, van Beek, & Walsh, 2008).

The propositional calculus can be extended to include an additional specific theory $\mathcal{T}$ by allowing sentences that include ground formulas from $\mathcal{T}$. For example, $\mathtt{P} \vee \neg\mathtt{Q} \vee [\mathtt{3X} - \mathtt{2Y} \geq \mathtt{5}]$ is a sentence from the propositional calculus combined with the theory of linear inequalities. The technology for solving systems of such sentences is known as *satisfiability modulo theories*. This extension of SAT solving has likewise been applied both to planning (Wolfman & Weld, 1999) and to software verification (De Moura & Bjørner, 2011).

3.3 Predicate Calculus and Other Extensional Logics

The predicate calculus, also known as first-order logic, extends the propositional logic in two respects:

- There are three types of non-logical symbols: Constants, functions, and predicates. A constant symbol denotes an individual entity; a function denotes a mapping; and a predicate symbol denotes a relation.
- The logical symbols include the Boolean operators; the two quantifiers $\forall$ (for all) and $\exists$ (there exists); and variable symbols. Optionally, the equality sign $=$ may also be included.

The predicate calculus thus does allow the expression of generalizations, through the use of the quantifiers, and therefore is often adequate for the axiomatization of commonsense domains. For instance, table 2 shows a few of the general rules for the blocks world that underlie the propositional encoding in table 1.

Most of the domain theories we discuss in this paper will be predicate calculus theories. All or almost all mathematical theories can be expressed in the predicate calculus, though possibly with an infinite number of axioms characterized by a general rule (an "axiom schema") rather than individually enumerated. Almost all mathematical theorems can be construed as inferences in the predicate calculus.

$\forall_{\texttt{b1,b2,b3,t}}$ `b1` $\neq$ `b2` $\Rightarrow \neg$`[On(b1,b3,t)` $\wedge$ `On(b2,b3,t)].` $\forall_{\texttt{x,t}}$ `Clear(x,t)` $\Leftrightarrow$ `x=Table` $\vee \neg \exists_{\texttt{b}}$ `On(b,x,t).`	*State Constraints.*
$\forall_{\texttt{b1,b2,t}}$ `PutOn(b1,b2,t)` $\Rightarrow$ `On(b1,b2,Succ(t)).`	*Causal Axiom.*
$\forall_{\texttt{b1,b2,t}}$ `PutOn(b1,b2,t)` $\Rightarrow$ `Table` $\neq$ `b1` $\neq$ `b2` $\wedge$ `Clear(b1,t)` $\wedge$ `Clear(b2,t).`	*Precondition Axiom.*
$\forall_{\texttt{b1,b2,b3,x,t}}$ `PutOn(b1,b2,t)` $\wedge$ `b3` $\neq$ `b1` $\Rightarrow$ `[On(b3,x,Succ(t))` $\Leftrightarrow$ `On(b3,x,t)].`	*Frame Axiom.*

Table 2: First-order, linear-time encoding of some blocks-world axioms

The predicate calculus is Turing-complete; a general computer with unbounded memory can be characterized in the predicate calculus, and carrying out inference in the predicate calculus requires a general computer with unbounded memory or its equivalent. Therefore, the inference problem is semi-decidable. That is, there are algorithms for computing whether a conclusion ϕ is a consequence of axiom set Γ with the following properties:

- If ϕ is a consequence of Γ, then the algorithm will eventually return the answer "TRUE", though no computable upper bound can be given on how long the algorithm will take to terminate.
- If ϕ is not a consequence of Γ, then the algorithm will either return the answer "FALSE" or will not terminate.

However, no algorithm can be written that always terminates, always returns "TRUE" if ϕ is a consequence of Γ, and always returns "FALSE" if it is not.

Though the problem cannot be solved in general, there is a sophisticated technology of algorithms that in practice are often effective in doing first-order inference (Wos, Overbeek, Lusk, & Boyle, 1984; Lifschitz, Morgenstern, & Plaisted, 2008). Various properties of the theory involved can make the problem easier or harder. Some examples:

- If a theory has no function symbols, and the only quantifiers in any sentences are universal quantifiers whose scope is the entire sentence, then this is a "Datalog" theory. Datalog theories are fully decidable.
- If the sentences in the theory are all either atomic formulas or have the form

  $$\forall_{x_1 \ldots x_m}[\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_k] \Rightarrow \psi$$

  where all the ϕ_i and ψ are atomic formulas, then the theory is called a *Horn* theory. Horn theories are also semi-decidable, but in practice there are algorithms for Horn theories that often run much more efficiently than inference algorithms for non-Horn theories. The programming language Prolog is based on Horn theories.
- If the theory includes axioms that use both function symbols and equality — for example, the associativity axiom $\forall_{\mathtt{x,y,z}}\mathtt{F}(\mathtt{x},\mathtt{F}(\mathtt{y},\mathtt{z}))) = \mathtt{F}(\mathtt{F}(\mathtt{x},\mathtt{y}),\mathtt{z})$ — then in practice the problem of inference becomes very much more difficult; the search space explodes, and the heuristics for guiding search become harder to find.

`Yesterday(Rain).`
(Yesterday, it rained.)

$\neg\exists_{\mathtt{l}}$ `Know(Benedite,Location(MonaLisa,l)).`
(Bénédite does not know the location of the Mona Lisa.)

`Hopes(Benedite, Eventually(`$\exists_{\mathtt{x}}$ `Know(Benedite,Past(Steal(x,MonaLisa))))).`
(Bénédite hopes that he will eventually know who stole the Mona Lisa.)

Table 3: Sentences in Modal Logic

It is often useful, both for the design of knowledge bases and for moderate improvements in inference efficiency, to use a *sorted* logic (analogous to a typed programming language). In a sorted language, there is a hierarchy of sorts of entities and the sort of every constant symbol, function symbol, and variable and every argument to a function or a predicate is declared. For example, the sorts in a simple physical theory might be instants of time, regions of space, and physical objects. Formulas that violate the sort declarations e.g. `Parent(BFlatMajor, CivilWar)` are essentially category errors (Cohn, 1987).

3.4 Modal Logic

Modal logics extend the propositional or the predicate calculus by adding additional operators on sentences, known as *modal operators* (Hughes & Cresswell, 1968). In commonsense reasoning, the most important categories of modal operators are those that express temporal relations such as "ϕ is sometimes true;" or "ϕ will eventually be true" and those that express propositional attitudes such as "Person x hopes that ϕ is true" or "It is common knowledge that ϕ is true" (discussed in sections 9.1 and 10.1). Modal operators and quantifiers can be embedded inside one another. Table 3 illustrates a number of sentences in modal logics.

Often, though not necessarily always, the contents of a modal theory can be translated into a first-order theory. For instance, most sentences in a temporal modal logic can be expressed as a sentence in a non-modal first-order logic that includes instants of time as individuals.[5] Likewise, as we will discuss in section 9.1, some of the most important modal logics of knowledge can be translated into first-order logics in which "possible worlds" are individuals. However, even in such cases, there may well be significant advantages to using the modal language rather than the first-order language.

- Modal sentences are generally closer to a natural language expression of the fact.
- Modal sentences may be easier for a knowledge engineer to use.
- Propositional modal logics — that is, logics that have modal operators but no explicit quantifiers — are often both expressive enough for the purpose at hand and reasonably tractable, or at least decidable (Vardi, 1996).

5. In branching models of time, the translation of some modal operators into first-order logic may require quantifying over time lines as individuals.

3.5 Plausible Reasoning

The logics we have discussed so far have all been *deductive*; that is, the inference mechanism associated with the logic allows conclusions that are known with certainty to be drawn from information that is known with certainty using rules that are always valid. Unfortunately, in real world situations, reasoning of this kind rarely suffices. In most real situations, a reasoner's information is uncertain, and rules are suggestive but not conclusive, so the conclusions that a reasoner must derive are likewise uncertain, and may be overthrown by the acquisition of better information or other rules. This kind of reasoning is known as *plausible inference*.

There are many different forms of plausible inference that arise.

1. **Inference from frequencies.** For instance: Pierre lives in Paris. Most residents of Paris can speak French. Infer that probably Pierre can speak French.

2. **Inference from random processes.** For instance: I have drawn five cards from a well-shuffled deck. Infer that I am more likely to have a pair than a flush.

3. **Inference from typicality:** For instance: Tweety is a bird. Birds typically live in trees. Therefore Tweety presumably lives in a tree. (It is not clear that statistically *most* birds live in tree; e.g. chickens, who are numerous, do not.)

4. **Subjective likelihood.** For instance, if you (an adult) run into someone on the street who used to be a close friend but you haven't seen for ten years, they will probably recognize you and almost certainly remember you. (Note that there is no well-defined sample space or random process here.)

5. **Evidence combination.** For instance, if two friends tell you that a movie is worth seeing, then that is better evidence than if just one of them tells you. If you have disliked the director's earlier movies, that's evidence against it.

6. **Inference from vague terms.** For instance, if you are told that the CEO of International Widget is short, infer that he is less than 6 feet tall.

7. **Induction.** From many examples of P's that are Q, infer that the generalization that all P's are Q. For instance, if every mouse you have seen is small, infer that all mice are small.

8. **Scientific theorizing.** For instance, infer from a large quantity of indirect and partial information that the chemical properties of elements are a periodic property of the order of their atomic weights.

9. **Explanation (abduction).** For instance, if there are puddles all over the street, infer that it rained.

10. **Closed world assumption.** The objects that you know about are all the relevant objects. For instance, in figure 2, assume that blocks A, B, and C are all the blocks that are relevant, and that there is no block D which is invisible, or hidden behind C, or simply omitted by the artist as uninteresting.

11. **Predicate closure assumption.** Assume that, among a particular set of objects, you know every instance of some particular relation ϕ. For instance, among your personal friends, assume that you know all the immediate family relations. You do not expect to find out that two people whom you know separately are in fact mother and son, though it would not be surprising if they turned out to be acquainted.

12. **Unique names assumption.** Assume that two entities described in different ways are in fact not the same entity, unless you know otherwise. For instance, if you read about a new appointment to the District Court, assume that this is not the middle-aged guy with the mustache you see in the elevator every so often, unless you have some reason to think that it is the same person.

13. **Autoepistemic inference:** Assume, from what you know about how you learn things, that if some fact were true, you would have heard about it; since you haven't heard about it, it is presumably false (Moore, 1985b). For instance, if you have never heard that you have a younger brother, assume that you do not, since most people know about all their siblings.[6]

14. **Gricean maxims.** Assume that speakers and writers are obeying the Gricean (1957) maxims:

    - **Maxim of relevance:** Utterances are relevant.
    - **Maxim of quality:** Assertions are true, or at least believed by the speaker.
    - **Maxim of quantity:** Assertions contains all information that might be useful, and no information that is useless.

There is no established theory to characterize the ways in which these different forms of plausible inference relate to one another (e.g. perhaps some are special cases of others). Obviously several of these forms of inference can only be applied in restrictive circumstances; for example, if you do not happen to know that the proposition "Mark Twain had children," is true (which it is), you are not justified using negation as failure or the auto-epistemic inference to conclude that it is probably false, because a large fraction of people do have children, and one could easily not know it about some particular famous person. Specifying the circumstances under which these various forms of plausible inference are reasonable is a major part of the theory of plausible inference.

Some of these modes of inference, such as Gricean inference, are particularly tied to natural language. In other cases, such as the inference from vagueness, this is less clear cut; it is not clear whether vagueness is a property of *concepts* or just of *words* or if that is a useful distinction. If a mode of inference is particularly associated with language, then there is a case to be made that it should be incorporated into the process of language interpretation rather than into the (language-independent?) process of conceptual reasoning. However, again it is not clear whether this is a meaningful distinction.

There are three main categories of logics that support plausible inference: non-monotonic logics, discussed in section 3.5.1, probabilistic logics, discussed in section 3.5.2, and fuzzy logic, discussed in section 3.5.3.

6. Of course, there are many cases where this turns out to be false. But the point is that such cases are almost invariably very *surprising*; the person involved had previously assumed otherwise.

There is a large literature on using specific forms of non-monotonic logic in taxonomic reasoning, in temporal reasoning, and in the theory of knowledge. We will discuss these briefly in sections 5, 6, and 9.1. The applications of non-monotonic logic to other commonsense domains and the applications of probabilistic logic and fuzzy logic to commonsense generally have not been studied extensively.

Theories of plausible reasoning tend to take one of the above forms of plausible inference as central, and then try to model all the other forms as variants of the central form. We will see examples below.

3.5.1 Non-monotonic Logic

A logic is *monotonic* if it satisfies the following constraint: Suppose that Δ is a set of axioms and that ϕ is a conclusion of Δ. Let Γ be any other set of facts. Then ϕ is still a consequence of $\Delta \cup \Gamma$. That is, if you start believing Δ and infer ϕ, and then learn some more facts Γ, your inference of ϕ is still valid. A logic that is not monotonic is *non-monotonic* (Brewka, Niemelä, & Truszczyński, 2008).

All of the logics that we have discussed in sections 3.2 through 3.4 are monotonic. Indeed, it is easily seen that a logic must be monotonic if either of the following conditions are satisfied:

- (Semantic condition.) There is a concept of a *model* and of what it means for a sentence to be true in a model. We say that model $\mathcal{M}$ satisfies sentence ϕ if ϕ is true in $\mathcal{M}$. A model $\mathcal{M}$ satisfies theory Δ if $\mathcal{M}$ satisfies every sentence in Δ. Sentence ϕ is a consequence of theory Δ if every model that satisfies Δ also satisfies ϕ.

- (Syntactic condition.) There is a concept of a *proof* of a conclusion ϕ from axioms Δ. A proof is some kind of symbolic structure that cites ϕ as the conclusion and cites some sentences from Δ as axioms. There is a criterion that determines whether a given symbolic structure is indeed a proof. The criterion has the property that, as regards Δ, all that matters is that the sentences cited as axioms are in Δ; the validity of the proof does not depend on Δ in any other way. In particular, it does not depend in any way on sentences in Δ that are not cited in the proof.

Therefore, for a logic to be non-monotonic, both the semantics of the logic and the notion of proof in the logic must be either very non-standard, or non-existent.

Numerous non-monotonic logics have been proposed and studied. Space limitations here preclude giving technical details, but we will sketch a few of the best known.

Negation as failure. Negation as failure is a built-in feature of many logic programming languages such as Prolog. A Prolog program is a collection of Horn clauses; executing a program corresponds to proving an assertion. The language provides an operator `not(P)` which is considered to be proved true if the attempt to prove `P` fails.

Various forms of negation as failure are also a fundamental mode of inference in **answer set programming** (Gelfond, 2008) and in **default logic** (Reiter, 1980). Answer set programming is similar to logic programming, but the program returns the entire set of answers rather than a single answer. In default logic, plausible inferences are encoded as rules of the form, “If α is known to be true and β cannot be proved false, then infer γ.”

Circumscription. In circumscription the central form of inference is a variant of predicate closure (McCarthy, 1980). Non-monotonic inference takes place in two steps. In the first step, a non-monotonic rule takes a theory $\mathcal{T}$ and a formula $\phi(\mathtt{x})$, and generates a second-order axiom α that essentially states, "The only objects satisfying ϕ are those that $\mathcal{T}$ forces to satisfy ϕ." Once α has been derived, inference is just ordinary deduction from $\mathcal{T} \cup \{\alpha\}$.

Belief revision characterizes non-monotonic inference in terms of principles that govern how you should withdraw previous beliefs if you learn new facts which are inconsistent with what you knew previously (Peppas, 2008). The best known theory of belief revision is the AGM (Alchourrón, Gärdenfors, & Makinson, 1985) postulates.

3.5.2 Probabilistic Logic

The classical theory of probability is by far the oldest, best established, and most widely used theory of plausible reasoning. Probabilistic reasoning of various kinds has become very dominant in artificial intelligence generally (Pearl, 1987); in particular, it is one of the major paradigms in machine learning (Murphy, 2012). Though most of this AI work does not invoke a logical framework, probabilistic logics have been extensively developed for use in AI (Nilsson, 1986; Bacchus, 1988; Milch et al., 2007; Goodman et al., 2008). Little of this work has been directly connected to commonsense reasoning, however, and therefore we will not discuss it in the other sections of this paper.

The original impetus for the theory of probability was as a theory of inference for random processes (gambling). The theory of probability is indisputably valid in situations where uncertainty is clearly due to random processes, including random sampling, that generate outcomes with a well-established distribution. However, the plausible reasoning needed for AI applications generally and for commonsense reasoning specifically rarely falls within these clear-cut cases. For those, what is needed is some form of "subjective probability", in which the probability of a statement is taken to be its likelihood, given some other information. The theoretical underpinnings of subjective probability theory are more tenuous, though in practice it is very successful.

There are several different ways in which a "base logic", such as propositional logic, predicate logic, or modal logic, can be augmented with probabilities. The simplest method is to add probabilities purely at the meta-level. That is: The sentences of the language are just those of the base logic, with no mention of probabilities. However, rather than being characterized as true or false in the knowledge base, a sentence ϕ is characterized in terms of its likelihood $\mathtt{Prob}(\phi)$, which is a real number between 0 and 1.

One may wish to extend probabilistic logic to allow more complex uses of the probability operator. However such extensions become problematic, in terms of their intuitive meaning, their formal semantics, and their proof theory. Some possible extensions might include:

- Boolean combinations of probabilistic statements; e.g. $\mathtt{Prob}(\phi)\mathtt{=0.5} \vee \mathtt{Prob}(\psi)\mathtt{=0.8}$.
- Arithmetic constraints on probabilities; e.g. $\mathtt{Prob}(\phi) \geq \mathtt{0.5}$ or $\mathtt{Prob}(\phi) \leq \mathtt{Prob}(\psi)$.
- Probabilistic operators within the scope of quantifiers or modal operators; e.g.
$\forall_{\mathtt{c1,c2,f1,f2}}\ \mathtt{c1} \neq \mathtt{c2} \wedge \mathtt{Flip(f1,c1)} \wedge \mathtt{Flip(f2,c2)} \Rightarrow$
$\quad\quad \mathtt{Independent(Heads(f1),Heads(f2))}$

(Any two flips of two different coins are independent) or
`Believes(Ralph, Prob(Likes(Josephine, Ralph)) > 0.3)` .
(Ralph believes that there is better than a 0.3 chance that Josephine like him.)

- Embedded probability operators e.g, `Prob(Prob(On(BlockA,BlockB) > .5) < .3).`
(There is less than a 0.3 chance that the probability that block A is on B is more than .5.)
`Prob(Believes(Ralph, Prob(Likes(Josephine, Ralph)) > 0.9)) < 0.4)` .
(It is unlikely (less than 0.4) that Ralph is almost certain (greater than 0.9) that Josephine likes him.) (Gaifman, 1988)

Another major issue is the "direct inference" (Bacchus, Grove, Halpern, & Koller, 1992). In applying probabilities, it is necessary to go from assertions about the sizes of collections such as "There are 2,598,960 different 5-card poker hands, of which 5148 are flushes" to statements about likelihoods, like "Joe has drawn five cards from a shuffled deck. The likelihood that he has a flush is 0.00198." Under what circumstances are inference of this kind valid? Note that this inference in itself non-monotonic; if the first sentence is changed to "Joe has drawn five cards from a shuffled deck and gotten two pairs", then the conclusion is no longer valid.

3.5.3 Fuzzy Logic

Fuzzy logic (also called "possibilistic logic") deals with inference from vague information (Zadeh, 1987; Dubois, Lang, & Prade, 1994). Vagueness is ubiquitous in natural language, and seems to be different from uncertainty. In fuzzy logic, every sentence has a "degree of truth" which is a number between 0 and 1. For instance, the sentence, "Joe is tall" is true to degree 1 if Joe is 6'6"; to degree 0.6 if Joe is 5'11"; and to degree 0 if Joe is 5'3".

Fuzzy logic occupies a strange position in the knowledge representation research community. Almost without exception, people either love it or hate it;[7] they either think it captures a central aspect of reasoning, or they think it is useless. Everybody agrees that vagueness is pervasive in natural language; but it is not clear whether vagueness must or should be an aspect of the symbols in a knowledge base. On the other hand, the fact is that, in most knowledge bases of any size, the symbols *are* simply words, or at least very closely tied to words. No one building a large knowledge base excludes concepts corresponding to words that are vague; and no one has proposed any better theory for vague words. Nonetheless it is not at all clear what is gained by incorporating fuzzy logic into a knowledge base. People have used fuzzy logic as the basis for practical programs with no particular relation to natural language (e.g. elevator control). However, the opponents of fuzzy logic claim that whatever valid reasoning these programs are doing could be interpreted more perspicaciously as an approximate implementation of some other theory.

3.5.4 Further Readings (Plausible Reasoning)

Theories of Probability (Fine, 1973) surveys the different foundations and interpretations associated with probability theory. *Readings in Uncertain Reasoning* (Shafer & Pearl, 1990)

7. For instance, in *Handbook of Knowledge Representation* (van Harmelen, Lifschitz, & Porter, 2008), it is mentioned once, in passing.

is a collection of seminal articles. *Reasoning About Unceretainty* (Halpern, 2003) is a monographic treatment of plausible reasoning from an abstract, axiomatic standpoint.

Uncertainty in Artificial Intelligence (UAI) is an annual conference (http://auai.org).

4. PartOf Relations

Our discussion now turn from logic as a general framework to the specific content of domain theories.

A fundamental relation between two entities of the same sort is the `PartOf` relation. This can relate two physical objects (e.g. John's head is part of John); two collections (e.g. John's nuclear family is part of his extended family); two time intervals (e.g. the month June 2014 was part of the year 2014); two events (e.g. the Battle of Gettysburg was part of the Civil War); two spatial regions (e.g. Alabama (as spatial region) is part of the United States (as spatial region); social organizations (e.g. the Senate is part of the Congress; Alabama, the state, is part of the United States, the country); and more abstract entities (e.g. quantum physics is part of physics).

Mereology is the study of `PartOf` relations. Whether a representation uses a single `PartOf` relation for all sorts or whether there is a separate relation for each sort is a purely a matter of minor notational convenience. (Either one has to have multiple `PartOf` relations for different sorts, or one has to have an axiom that asserts that `PartOf(x,y)` can only hold if `x` and `y` are the same sort.) By convention, we will take `PartOf(x,y)` to be the non-strict relation which includes the case where `x=y`; the strict relation is denoted `ProperPart(x,y)`.

$\forall_{x,y}$ `ProperPart(x,y)` $\Leftrightarrow$ `PartOf(x,y)` $\wedge$ `x` $\neq$ `y`.
$\forall_{x}$ `PartOf(x,x)`.

The part-of relation defines a partial ordering over a sort where it applies; that is, `ProperPart` satisfies the following two axioms:

$\forall_{x,y}$ `ProperPart(x,y)` $\Rightarrow$ $\neg$`ProperPart(y,x)`.
$\forall_{x,y,z}$ `ProperPart(x,y)` $\wedge$ `ProperPart(y,z)` $\Rightarrow$ `ProperPart(x,z)`.

The `PartOf` relation on different sorts often correspond closely to one another. For example, if object `x` is `PartOf` object `y`, then at any given time the region occupied by `x` is `PartOf` the region occupied by `y`. Natural measures on a space of entities generally conform to natural `PartOf` relations, in the sense that if `PartOf(x,y)` then `Measure(x)` $\leq$ `Measure(y)`. This holds, for instance, of `x` and `y` are physical objects and the measure is mass; or if `x` and `y` are spatial regions and the measure is volume; or if `x` and `y` are collections and the measure is cardinality.

Two further relations can be defined in terms of `PartOf`. Two entities are *disjoint* if they have no common part; they *overlap* if they have a common part but neither is part of the other.

$\forall_{x,y}$ `Disjoint(x,y)` $\Leftrightarrow$ $\neg\exists_{z}$ `PartOf(z,x)` $\wedge$ `PartOf(z,y)`.
`Overlap(x,y)` $\Leftrightarrow$ $\neg$`Disjoint(x,y)` $\wedge$ $\neg$`PartOf(x,y)` $\wedge$ $\neg$`PartOf(y,x)`.

The five relations `ProperPart(x,y)`, `ProperPart(y,x)`, `x=y`, `Disjoint(x,y)`, and `Overlap(x,y)` form a jointly exhaustive pairwise disjoint (JEPD) set of relations over a sort; that is, given any `x`, `y`, exactly one of the relations holds between them. When applied to spatial regions, this collection of five relations is known as RCC-5 (section 7.1).

5. Taxonomy

A taxonomy (also known as a "semantic net" or an "ontology", though both of these terms are broader and less precise) is a knowledge base that records categories and entities and the fundamental relations between them: An entity `e` may be an *instance* of a category `c`, denoted `Inst(e,c)` , and category `c` may be a *subcategory* of category `d`, denoted `IsA(c,d)`. The `Isa` relation can be defined in terms of `Inst` as follows:

$$\forall_{\mathtt{x,y}}\ \mathtt{IsA(x,y)} \Leftrightarrow \mathtt{x} \neq \mathtt{y} \wedge \forall_{\mathtt{e}}\ \mathtt{Inst(e,x)} \Rightarrow \mathtt{Inst(e,y)}.$$

`IsA` is essentially just the `ProperPart` relation as applied to categories, so all the discussion in section 4 carries over.

A *semantic network* is an implementation of a taxonomy as a labelled directed graph, where the vertices correspond to entities and categories, and labelled arcs indicate `IsA` and `Inst` relations. Figure 3 shows a simple semantic network. Here, `Lassie` and `Tweety` are entities; `Dog, Cat, Robin, Penguin, Bird, Mammal,` and `Animal` are categories. `Lassie` is directly marked as an element of `Dog`; since `Dog` is a subcategory of `Mammal` and `Mammal` is a subcategory of `Animal`, it follows that `Lassie` is also an element of `Mammal` and of `Animal`.

Taxonomies are ubiquitous in knowledge-based AI. They are important because many of the properties of an entity, and in particular many of its most essential properties, are determined by the categories it belongs to. Furthermore, these exist techniques for collecting large amounts of quite accurate taxonomic information using web mining (section 11).

The standard taxonomy of the animal kingdom is unusually clear-cut. The entities are well-defined; the categories and the membership and subcategory relations are mostly very well established scientifically; and the taxonomy is tree-structured so no two categories partially overlap.[8] Many taxonomies, such as the taxonomy of events, or ideas, or social groups, are much more nebulous, in all respects. Systematically constructing a taxonomy involves large numbers of decisions that can be difficult, subtle, or arbitrary.

- Deciding which entities to include.
- Individuation. What constitutes a single entity? When are two things just two manifestations of the same entity?
- Deciding on the categories.
- Deciding on the instance relations
- Deciding on the subcategory relation.

8. There are of course other categories of animals, like "egg-laying" or "pet", that do not conform to the tree structure.

In almost any large, realistic domain, doing this carefully and methodically is difficult and involves many borderline cases and arbitrary decisions. Consider, for example, the problem of individuating books for a library catalogue. At first blush, it seems easy: "Moby Dick" is a book, case closed. However: There are multiple copies of the identical book. There are multiple editions, sometimes almost identical, sometimes identical except for features like illustrations or forewords, sometimes enormously different, sometimes with different authors. There are translations. There are multi-volume books. There are bound volumes that contain multiple books. There are things that are borderline as to whether they are books at all — personal notebooks, for instance, or bound journals, or audiobooks. Librarians have elaborate systems of rules for all these; still there are many borderline cases.

Semantic nets can also associate properties (unary predicates) with entities and categories. For instance in figure 3, `Furry` is a property, and the arrow from `Mammal` to `Furry` indicates that all mammals are furry. A property associated with a category corresponds to the statement that every element of the category has the property; in this case, the axiom $\forall_{\mathtt{x}}\ \mathtt{Mammal(x)} \Rightarrow \mathtt{Furry(x)}$. From that, it directly follows that every element and every subcategory of `Mammal` has the property `Furry`; the technical term is that `Dog` and `Lassie` *inherit* the property `Furry` from `Mammal`.

It is also common for a semantic net to indicate default properties of categories. For instance in figure 3 the arrow from `Bird` to `CanFly` indicates that there is a default rule that a bird can fly. This default is inherited by the subcategory `Robin` and the element `Tweety`, but it is specifically marked as canceled for `Penguin`, and therefore it is not inherited by `Penguin` or by any subcategories or elements of `Penguin`. Default inheritance is inherently more problematic than monotonic inheritance; there is a large literature on the proper way of encoding default rules and characterizing default inheritance (Etherington & Reiter, 1983; Touretzky, 1984; Brachman, 1985).

Semantic nets are in many ways similar to the hierarchies of classes and objects used in object-oriented programming.[9] Entities are analogous to objects; categories are analogous to classes; and the subcategory relation is analogous to the subclass relation. Inheritance of properties is analogous to inheritance of data fields and methods, and canceling of properties is analogous to overriding of methods. It may be noted that in programming languages such as C++ where a class may have multiple superclasses, the rules on inheritance and overriding become quite complicated.

It is tempting to extend a semantic net architecture by allowing an arc between nodes to represent a relation between categories; e.g. the statement "Birds have wings" could be represented as an arc from `Wing` to `Bird` labelled `PartOf`. This can be done but considerable care and precision is needed to make sure that quantification relations and implicit temporal qualifications are represented unambiguously. Woods (1975) warns in detail of the errors that can be made; despite this classic paper, however, these kinds of errors persist in semantic nets to the present day.

Consider for instance the small semantic net in figure 4. The arc labelled `AtLocation` from `person` to `restaurant` presumably means either "In any restaurant during working hours there are some people inside" or "In any restaurant there are sometimes people inside", or "Many people are sometimes in a restaurant at some time" or possibly "At any

9. The use of semantic nets in AI knowledge representation certainly predates the use of classes and inheritance in programming language technology; to what extent it was an influence, I do not know.

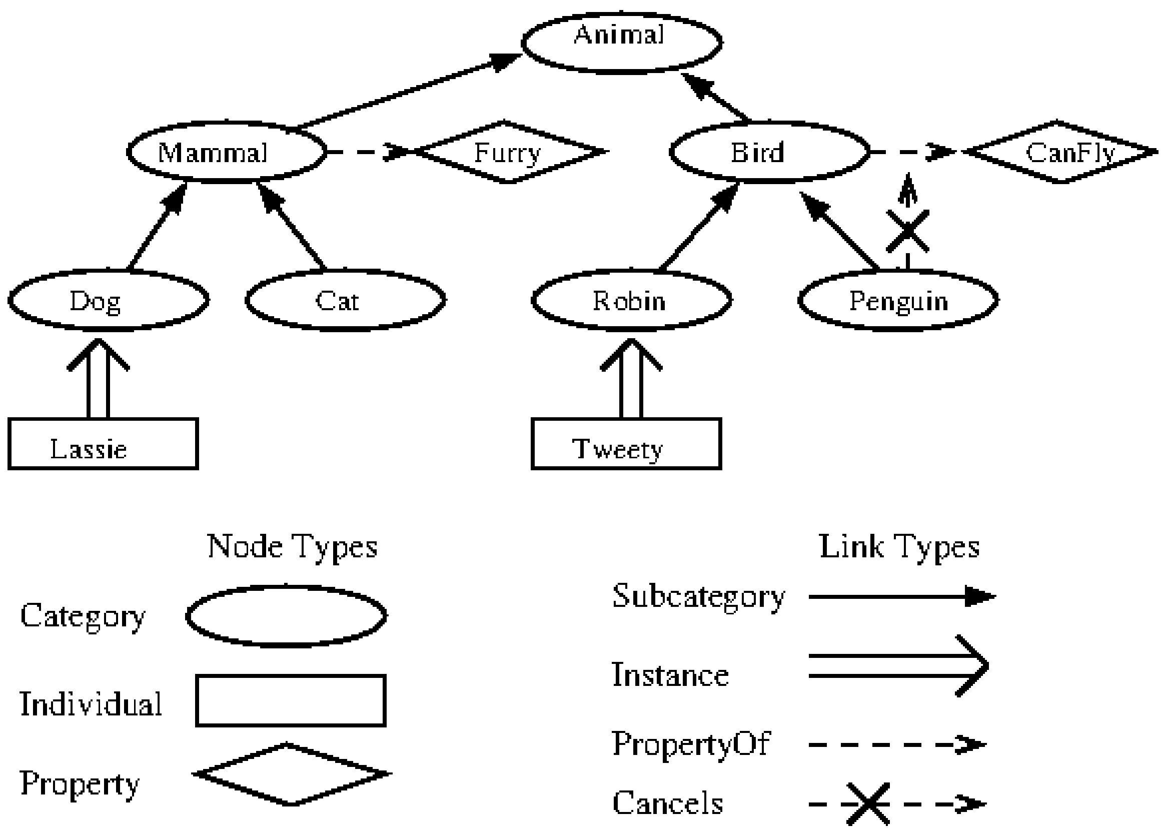


Figure 3: Simple taxonomy

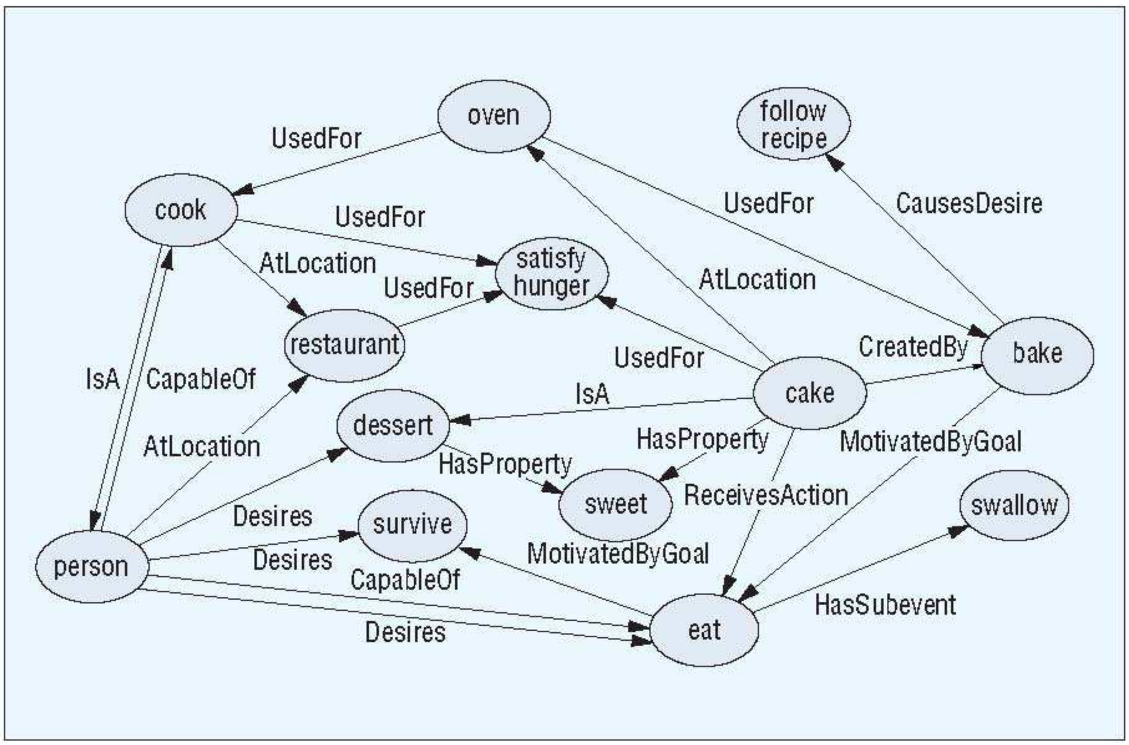


Figure 4: Small piece of the semantic network "ConceptNet" (Havasi, Pustejovsky, Speer, & Lieberman, 2009)

.

given time, there is someone inside some restaurant (somewhere in the world)". What kind of quantifiers ("All", "some", "many") should be attached to the head and tail of the arc, and to the implicit temporal argument, and with what scope, is unspecified. It is possible to extend the graphical representation to make all this unambiguous (Hendrix, 1979); but at that point the appealing simplicity of the semantic network is lost, and consequently the advantage over standard first-order notation becomes less clear.

A *description logic* is a representational framework in which taxonomic information can be expressed elegantly and naturally, and which supports efficient taxonomic inferences. Generally such logics are equivalent to a tractable subset of first-order logic. Description logics have been widely applied in knowledge-based applications, particularly the semantic web (Baader, Horrocks, & Satler, 2008).

5.1 General Reading

The literature on semantic networks and inheritance in AI is enormous (Findler, 1979; Brachman, 1979; Sowa, 1991; Sowa, 1992). The organization of entities in taxonomic hierarchies goes back to Porphyry (3rd century). The term "semantic net" was first used in a proposal by Richard Richens (1956; quoted in Sowa, undated, p. 3) who was working on machine translation.

> I refer now to the construction of an interlingua in which all the structural peculiarities of the base language are removed and we are left with what I shall call a "semantic net" of "naked ideas".

Quillian (1968) introduced "spreading activation" as a form of associative reasoning in semantic networks.

6. Time and Action

The logical representation of time has been studied very extensively in a number of different fields, including knowledge representation, programming language semantics, linguistics, and philosophy. It is certainly the best understood of commonsense domains.

The kinds of information that one might need to express in a temporal language include:

- Order information. The Mona Lisa was stolen *between* Sunday night and Tuesday morning.
- Metric information. The story broke in the newspapers *within hours* of the discovery of the empty frame.
- Hypotheticals. *If* I don't eat lunch, *then* I will be hungry by 4:00. *If* I eat lunch, *then* I will be OK until dinner time.

Models of the time structure differ along two features, each with two different options, so overall there are four different possible structures (figure 5).

- The time structure can be *linear* or *forward-branching*. In a linear model the instants of time are totally ordered. In a forward-branching model, the set of instants that

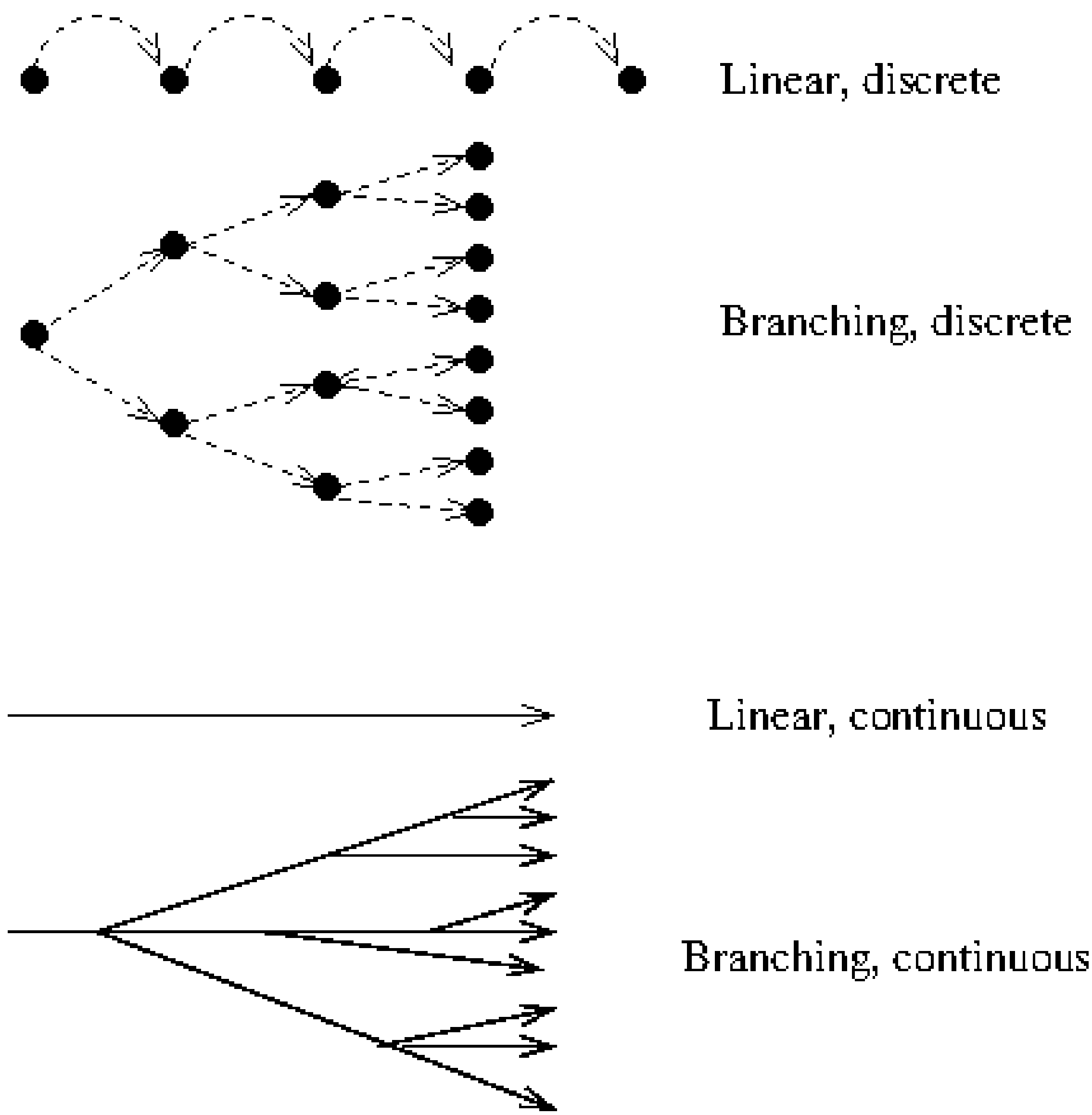


Figure 5: Time Structures

precede a given instant T are totally ordered, but there may be separate branches following T. Branching can correspond to choices made by an agent, or to the outcome of random processes, or both. (Other structures, e.g. structures with cycles, are logically possible, but rarely useful.)

- The time structure can be *discrete* or *continuous*. In a discrete time line, each time point has a unique immediate predecessor and one (in a linear model) or more (in a branching model) immediate successors. Thus, any totally ordered subset of the structure is isomorphic to a subset of the integers. In a continuous time line, each maximal totally ordered subset of the structure is isomorphic to the real line.

In a branching structure, a "time line" is one possible chronology, over all time; technically, a setwise maximal totally ordered subset of the time structure. In a linear structure, the time structure is the only time line.

Temporal languages vary along further features, beyond those that reflect the choice of time structure: A language may be first-order with instants of time or intervals represented explicitly; or it may be modal, with temporal relations expressed in modal operators and no explicit temporal entities.

6.1 Intervals

In a linear model of time[10] an *interval* `i` is a set of time points satisfying the following conditions

- There are at least two points in `i`. (Single point intervals do not satisfy the theory described below.)
- If `x` and `y` are both in `i` and `x` $<$ `z` $<$ `y`, then `z` is in `i`.

Any two intervals `i` and `j` must satisfy exactly one of 13 relations (Allen, 1983): `Before(i,j)`, `Meets(i,j)`, `Overlaps(i,j)`, `Starts(i,j)`, `Equal(i,j)`, `StartsI(i,j)`, `During(i,j)`, `DuringI(i,j)`, `FinishesI(i,j)`, `Finishes(i,j)`, `OverlapsI(i,j)`, `MeetsI(i,j)`, and `BeforeI(i,j)` (figure 6). (The relations ending in `I` are the inverse of the relation of the same name without `I`; e.g. `BeforeI(i,j)` is equivalent to `Before(j,i)`.) Moreover, one can construct a 13×13 "transitivity table" which makes it possible to infer the possible relations between two intervals `i` and `k` given the relation between `i` and `j` and the relation between `j` and `k`. For example, if `Starts(i,j)` and `Meets(j,k)` then `Before(i,k)`. If `Starts(i,j)` and `StartsI(j,k)` then either `Starts(i,k)`, `Equal(i,k)`, or `StartI(i,k)` (figure 7). The JEPD collection of 13 relations and the associated transitivity table are known as the "Allen interval calculus."

If, in the time structure, all intervals have endpoints, then any interval relations between `i` and `j` can be expressed as order relations on their endpoints, and the transitivity relations between intervals can be computed simply by using the transitivity of order relations on point. Any relation involving an interval can be rewritten as a relation on the two endpoints. Hence, using bounded intervals as entities in addition to, or instead of, instants, does not

10. Intervals can be defined in the same way in a branching model; however, there is a larger set of possible relations.

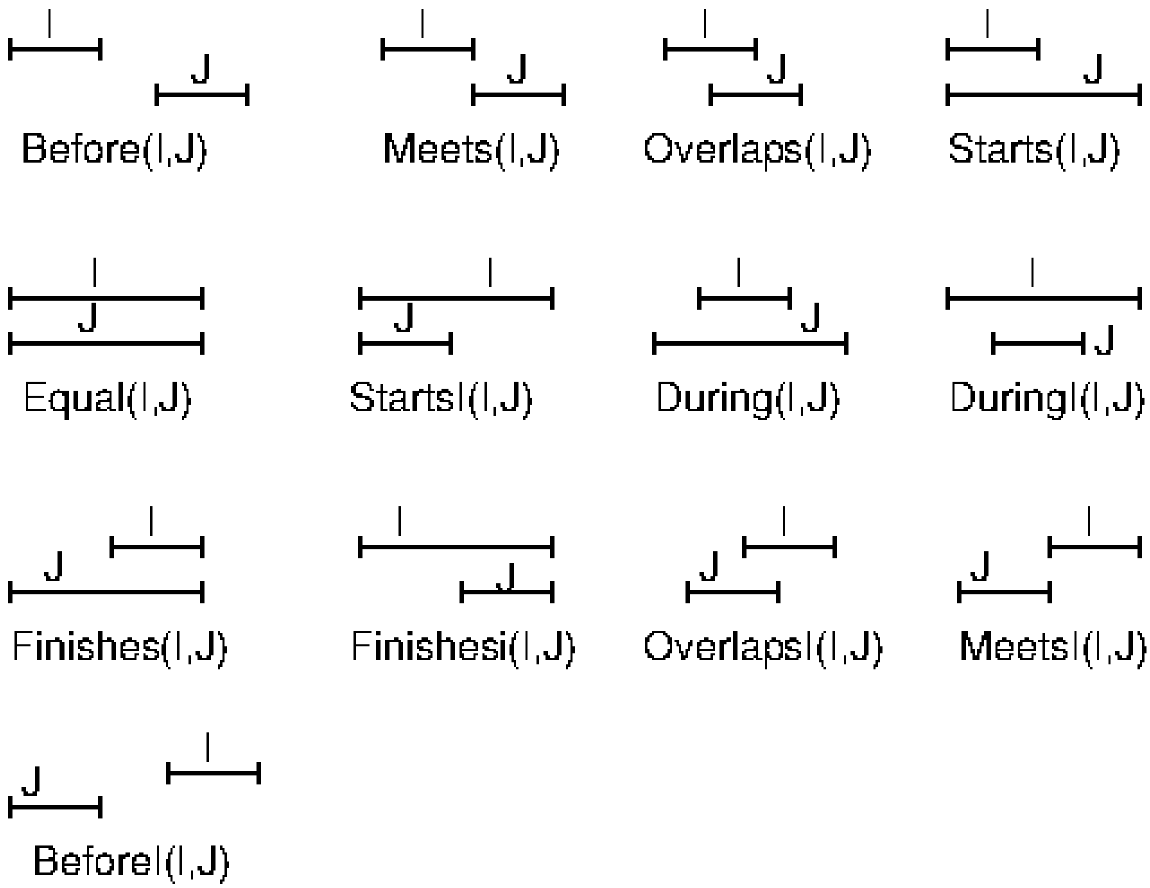
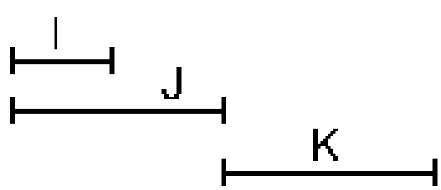
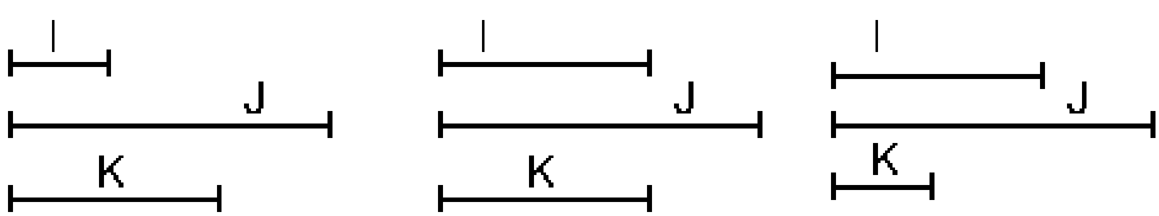


Figure 6: Interval relations



If `Starts(i,j)` and `Meets(j,k)` then `Before(i,k)`



If `Starts(i,j)` and `StartI(j,k)` then either `Start(i,k)`, `Equal(i,k)` or `StartsI(i,k)`

Figure 7: Transitivity rules for intervals

actually increase the expressivity of the language; it may, however, make the language more natural. It also allows the knowledge engineer to remain agnostic about what happens at the boundary instant between two states, which is sometimes advantageous (see section 6.7).

The computational characteristics of constraint systems over Allen's relations has been completely characterized (Nebel & Bürckert, 1995).

6.1.1 Modal Temporal Logics

Modal logics use modal operators to make assertions about the state of the world at times other than "now".

- Tense operators correspond roughly to tenses in languages;[11] for example `Future(`ϕ`)` (ϕ will be true at all future times) or `Past(`ϕ`)` (ϕ was true at all past times).
- Sequential logics, used for discrete models of time, have an operator `Next(`ϕ`)` (ϕ will be true at the next instant of time after "now"). Such logics sometimes also include `Prev(`ϕ`)`.
- Dynamic operators have forms such as $[\mathtt{a}]\phi$, meaning that if action `a` is executed now, then ϕ will be true when `a` completes. Here, each action `a` has a corresponding modal operator $[\mathtt{a}]$. If the language of actions here includes programming language operators, like `Sequence`, `Conditional`, and `Loop`, then this modal language can be used to define a programming language semantics.

In areas other than AI, such as software verification, mathematical logic, and philosophy, temporal logics have been the most widely used logical representation of time. However, in automated commonsense reasoning, they have been less studied than such representations as the situation calculus and the event calculus.

6.2 Basic Theories of Action

A major application of temporal representations has been reasoning about the effects of actions, for the purposes of planning. The axioms needed for such a theory can generally be divided into the following categories:

- **Domain constraints:** Constraints on what relations can hold at a single time. For instance, the simple blocks world[12] obeys the following domain constraints among others: Every block is either supported by a single different block or is on the table. A block is clear only if there is nothing on top of it.

11. The actual semantics of tenses in natural languages are complex, and involve many considerations other than just temporal sequence (Steedman, 2012).
12. The "blocks world" originally included a comparatively wide range of block-related phenomena and tasks: blocks of different sizes and shapes; complex structures such as arches (Winston 1975); events such as blocks falling (Funt, 1979); and a range of AI tasks including vision (Waltz, 1970) and natural language (Winograd, 1972). Soon, however (Sussman, 1975), the phrase came to denote a much more limited toy world: blocks are all cubical and all the same size and shape; blocks can only be stacked one on top of another; the only action is to move a single block from the top of one stack to the top of another; and the only tasks under consideration are projection and planning.

- **Precondition axiom:** The action "Put block A on block B" can only be carried out if both A and B are clear on top.
- **Positive causal axioms:** Executing an action causes some relation to hold. For instance: If you do the action "Put block A on block B", one result is that block A will be on block B.
- **Negative causal axioms:** Executing some action caused some relation not to hold. For instance: If block A is on block C and you do the action "Put block A on block B", one result is that A is no longer on C.
- **Frame axiom:** Executing a particular action leaves some state unchanged. For instance: Executing the action "Put block A on block B" leaves all "on" relations not involving block A unchanged.
- **Atemporal axioms.** Axioms that involve only static properties and relations. For instance: The table is not a block.

We have already seen the representation of some of these in tables 1 and 4; we will use them as examples for additional temporal representations below.

6.3 Situation Caculus

The situation calculus was introduced by McCarthy and Hayes (1969) and has been often used since, particularly in the context of automated planning (Reiter, 2001; Lin, 2008).

The situation calculus assumes a forward-branching model of time. Generally, the time structure is taken to be discrete, though continuous time can also to some extent be accommodated (Pinto, 1997). Branching generally corresponds to the actions of a single actor.

There are a few different notational schemes one can use for the situation calculus; the following one is typical.

There are three sorts of entities. A *situation* is an instant of time. A *fluent* is an entity that takes on different values in different situations; for example, `On(BlockA,BlockB)` is a Boolean fluent, which takes values `True` and `False`; `President(US)` is a person-valued fluent. A *action* creates a transition from one situation to a successor; for example `PutOn(BlockA,BlockB)`.

There are four general symbols in the theory. `Holds(s,f)` is a predicate which is true if Boolean fluent `f` is true in situation `s`. `Value(s,f)` is a function that denotes the value of non-Boolean fluent `f` in situation `s`. `Result(s,a)` is a function that denotes the situation that follows if action `a` is executed in situation `s`. `Poss(a,s)` is a predicate that holds if action `a` is feasible in `s` (i.e. the preconditions of `s` are satisfied.)

Table 4 shows some blocks world axioms expressed in the situation calculus. This is quite similar to the encoding in table 2. The important difference between the two is table 2 uses a linear time structure, in which a time instance `t` has a unique successor `Succ(t)` and only one action is executed in a given situation, whereas the situation calculus is a branching-time model, in which a situation `s` has a different successor `Result(s,a)` for every action `a` that is feasible in situation `s`. In the situation calculus, a goal G is achievable from a starting state $S0$ if there is a situation $SG > S0$ satisfying G; and the plan to achieve G is

$\forall_{\text{b1,b2,b3,s}}$ `b1` $\neq$ `b2` $\Rightarrow$ $\neg$`[Holds(s,On(b1,b3))` $\wedge$ `Holds(s,On(b2,b3))].`	*Domain Constraint*
$\forall_{\text{b1,b2,s}}$ `Poss(PutOn(b1,b2),s)` $\Rightarrow$ `Holds(Result(s,PutOn(b1,b2)),On(b1,b2)).`	*Causal Axiom*
$\forall_{\text{b1,b2,b3,x,s}}$ `b3` $\neq$ `b1` $\wedge$ `Poss(PutOn(b1,b2),s)` $\Rightarrow$ `[On(b3,x,Result(s,PutOn(b1,b2)))` $\Leftrightarrow$ `On(b3,x,s)].`	*Frame Axiom*
$\forall_{\text{b1,b2,s}}$ `Poss(PutOn(b1,b2),s)` $\Leftrightarrow$ `b1` $\neq$ `b2` $\wedge$ `Holds(s,Clear(b1))` $\wedge$ `Holds(s,Clear(b2)).`	*Precondition Axiom*

Table 4: Situation calculus encoding of some blocks-world axioms

the sequence of actions executed on the time line from $S0$ to SG. In the linear time model, G is achievable if it is consistent that G holds in some situation $SG > S0$.

6.4 The Frame Problem and the Yale Shooting Problem

The frame axioms, which characterize what fluents remain the same as a result of an action, are widely felt to be problematic. This problem, known as the "frame problem",[13] first identified by McCarthy and Hayes (1969), has a number of different aspects:

- Depending on how a dynamic theory is formulated, and how frame axioms are stated, there may have to be a lot of frame axioms, or they may have to be long, or they may depend on having a lot of ancillary axioms (e.g. unique names assumptions) (Davis, 1990, ch. 5; Schubert, 1990).

- Frame axioms tend to clog up the inference process. If you try to construct a proof of a temporal projection problem — that is, predicting the final state of a dynamic system given the starting state and the actions executed — you are very likely to find that an inordinate fraction of the proof consists in moving fluents forward, unchanged, from one situation to the next.

  In particular, frame axioms create misery for naïve backward-chaining solutions to planning problems, because one way that goal G can be true at time T is if G was true at time $T-1$, and the action at time $T-1$ didn't affect it. However, that fact in itself does not constitute progress toward making G true starting in a state where it is false.

- Frame axioms, and especially attempts to automate the generation of frame axioms, tend to interact strangely and badly with extensions of a temporal theory beyond simple causal axioms. Extensions that require care include state constraints (this interaction is known as the "ramification problem") (Thielscher, 1997, 2000; McIlraith, 2000), axioms that characterize behavior over extended time, continuous time, and theories of knowledge and belief (Scherl & Levesque, 2003; Morgenstern, 1991).

13. Cognitive psychologists and philosophers somehow got hold of this term and misunderstood it, so now in those disciplines the phrase "the frame problem" means the problem of determining which facts are relevant in drawing a desired conclusion or carrying out a desired task — also an important problem, but only distantly related (Pylyshyn, 1987).

- The "Yale Shooting Problem" proposed by Hanks and McDermott (1987), demonstrates that, even in very simple cases, simple methods of deriving frame axioms non-monotonically could lead to wrong results.

There is an enormous literature on the frame problem in general and the Yale Shooting Problem in particular (Brown, 1987). Within the context of highly structured dynamic theories, the problem can be solved (Stein & Morgenstern, 1994; Doherty & Łukaszewicz, 1994; Shanahan, 1997; Reiter, 2001). However, if the dynamic theory requires causal axioms that are not of a standard form, then frame axioms must be hand-constructed (Davis, Marcus, & Chen, 2013).

The frame problem also arises in program verification; a specification for a program or function must specify, explicitly or implicitly, everything that does not change as a result of executing it. This can be challenging if the programming language has features like aliasing (Kogtenkov, Meyer, & Velder, 2015).

Another problem that arises in constructing theories of actions is that it is often difficult to formulate sufficient conditions for an action to be feasible, because the number of things that can go wrong is almost unlimited. For instance, to start your car, you must be seated in the driver's seat, and you must able to reach the pedals, and you must have the car key and the ignition must not be blocked with chewing gum, and the car must have gas, and the battery must not be empty, and there must not be a potato in the tailpipe .... This is known as the "qualification problem"(McCarthy, 1977). It is difficult to formulate rules that cover all these cases, and once they are formulated, it is burdensome to include all these conditions in a problem statement. As with the frame problem, it would be desirable to use non-monotonic logic so that the more far-fetched of these considerations only arise when they are relevant; and as with the frame problem, care is required to make sure that the non-monotonic inference works out properly (Lin & Reiter, 1994).

6.5 McDermott's Temporal Logic

McDermott's (1982) temporal logic is similar to the situation calculus, but instead of using the `Result` function, it uses the predicate `Occur(t1,t2,e)` (event `e` occurred starting at time `t1` and ending at `t2`) to characterize the occurrence of an event. It thus drops the assumption that an event occurs from one situation to a "next" situation. As a result, the language lends itself more naturally to expressing situations in which events occur concurrently or with an indeterminate time relation. For instance, suppose that we want to represent a blocks world in which an agent has two hands and can use them asynchronously to build towers. In the situation calculus, this is difficult and awkward to represent; in McDermott's logic, it is comparatively straightforward, using axioms of the kind shown in table 5. (The axioms here are formulated so that they work correctly in either a discrete or a continuous model of time.)

However, apparently McDermott's logic is much harder to use in automated reasoner for tasks such as planning; certainly it has been much less studied than the situation calculus. Note that the number of different *kinds* of axioms in table 5 and their logical complexity is very much increased over table 4, making it more difficult to systematize the construction of theories and the organization of automated reasoners.

$\forall_{\texttt{h,b1,b2}}$ `Grasps(h,b1)` $\Rightarrow \neg$`On(b1,b2)` $\wedge \neg$`On(b2,b1)`.
(*State Constraint:*
While hand `h` is grasping block `b1`, `b1` is not on top of any other block, or vice versa.)

$\forall_{\texttt{h,b1,b2,ta,tx,tb}}$ `Occurs(ta,tb,PutOn(h,b1,b2))` $\wedge$ `ta` $<$ `tx` $<$ `tb` $\Rightarrow$
`Grasps(tx,b1)`.
(*State constraint during action:*
While hand `h` is carrying out the action `Puton(h,b1,b2)`, it is grasping `b1`.)

$\forall_{\texttt{h,b1,b2,ta,tx,tb}}$ `Occurs(ta,tb,PutOn(h,b1,b2))` $\Rightarrow$ `Holds(ta,Clear(b1))`.
(*Precondition at beginning of action:*
When the agent begins the action `Puton(h,b1,b2)`, `b1` must be clear.)

`Occurs(ta,tb,PutOn(h,b1,b2))` $\Rightarrow$
$\exists_{\texttt{tc}}$ `tc` $<$ `tb` $\wedge$
$\forall_{\texttt{td}}$ `tc` $\leq$ `td` $<$ `tb` $\Rightarrow$ `Holds(tb,Clear(b2))`.
(*Condition at ending of action:*
If the action `PutOn(h,b1,b2)` ends at time `tb`,
then `b2` is clear over some open interval `(tx,tb)`.)

Table 5: Partial axiomatization of an asynchronous two-handed blocks world in McDermott's temporal logic

6.6 Event Calculus

The *event calculus* was developed for representing narratives (Kowalski & Sergot, 1986). The primary goal was therefore developing a representation that elegantly expresses the events in narratives, rather than one that encodes a predictive theory. It uses a linear model of time, either discrete or continuous. As developed by Mueller (2006, 2008), it is a sorted logic with three sorts: events, Boolean fluents, and timepoints (the logic conflates time points and durations). Table 6 shows the central predicates of the theory.

Table 7 shows part of the axiomatization of a hot air balloon, controlled by a heater (Miller & Shanahan 1999). The balloon rises at velocity V if the heater is on and falls at velocity V if it is off.

6.7 Continuous Time

Continuous models of time are used in circumstances in which it is important to reason about continuous motion or continuous change of numerical measures. Continuous time structures can be either linear or branching. Either way, a time line in a continuous structure is generally taken to be isomorphic to the real line, or the positive real line.

A full axiomatization of continuity requires a second-order axiom or axiom schema equivalent to Dedekind's "greatest lower bound" axiom for the real numbers. This is rarely incorporated directly in commonsense reasoning systems. Rather, the knowledge engineer posits primitives whose intended interpretation is grounded in a continuous model, and

`Happens(e,t).` Event `e` happens at time `t`.
`HoldsAt(f,t).` Fluent `f` holds at time `t`.
`ReleasedAt(f,t).`
Fluent `f` may change its value at time `t`. More precisely, a fluent that is not released is subject to frame axioms and requires an event to occur to change it. No frame axioms apply to a fluent that is released.
`Initiates(e,f,t).` Event `e` causes fluent `f` to be true at time `t`.
`Terminates(e,f,t).` Event `e` causes `f` to be false at time `t`.
`Releases(e,f,t).` Event `e` causes `f` to be released at time `t`.
`Trajectory(f1,t1,f2,t2).`
If fluent `f1` is initiated at time `t1` then `f2` will become true at time `t1 + t2`.
`AntiTrajectory(f1,t1,f2,t2).`
If fluent `f1` is terminated at time `t1` then `f2` will become true at time `t1 + t2`.

Table 6: Predicates of the Event Calculus

```
Initiates(TurnOnHeater(a,b), HeaterOn(b), t).
Terminates(TurnOffHeater(a,b), HeaterOff(b), t).
HoldsAt(Height(b,h1),t) ∧ HoldsAt(Height(b,h2),t) ⇒ h1=h2.
HoldsAt(Height(b,h1),t1) ⇒
    Trajectory(HeaterOn(b),t1,Height(b,h+V·t2),t2).
HoldsAt(Height(b,h1),t1) ⇒
    AntiTrajectory(HeaterOn(b),t1,Height(b,h+V·t2),t2).
ReleasedAt(Height(b),t).
```

Table 7: Axomatization of a Hot Air Balloon

posits axioms whose justification is that they are true in a continuous model. For instance one might posit that if the distance from object A to B is less at time T1 than at time T2 and F is a value between those two distances, then the distance from A to B is F at some time between T1 and T2.

$$\forall_{\mathtt{a,b,t1,t2,f}}\ \mathtt{t1} < \mathtt{t2} \land \mathtt{Dist(Place(a,t1),Place(b,t1))} < \mathtt{f} < \mathtt{Dist(Place(a,t2),Place(b,t2))} \Rightarrow \exists_{\mathtt{tx}}\ \mathtt{t1} < \mathtt{tx} < \mathtt{t2} \land \mathtt{Dist(Place(a,tx),Place(b,tx))} = \mathtt{f}.$$

In using a continuous model of time for axiomatizing a particular domain, two "nuisance" issues unavoidably arise and must be addressed. The first is the problem of boundary points. If a Boolean fluent f is true over an interval (t1, t2) and then false over interval (t2, t3), what value does it have at the boundary point t2? In other words, is the fluent true over the closed interval [t1, t2]; the open interval (t1, t2) or one or the other of the half-open intervals (t1, t2] or [t1, t2). In some cases, this is a distinction without a difference and the question can safely be left unresolved. In other cases, there is only one possibility consistent with the topology of the situation: for example, if q is a continuous numeric fluent, then the Boolean fluent $q > 0$ must be true over an open interval and the fluent $q \geq 0$ must be true over a closed interval. A more complex example; suppose that objects A and B change shape continuously, and suppose that we are using the RCC system, described below in section 7.1, to characterize their spatial relations. If there is a transition from TPP(A,B) to EQ(A,B), then necessary EQ(A,B) holds at the boundary point; if there is a transition from TPP(A,B) to NTPP(A,B) then necessarily TPP(A,B) holds at the boundary point. However, in many cases quite careful thought is required to make sure that all the decisions made about boundary points are consistent one with another, and consistent with the other axioms being stated. Branching continuous time structures are even harder to deal with in this respect than linear continuous time structures, because of the complex topology at the branching points.

The second nuisance problem has to do with "Zenonian" behavior, in which a Boolean fluent changes its value infinitely often in a finite time interval. If a fluent f is true from t=0 to t=1/2, false from t=1/2 to t=3/4, true from t=3/4 to t=7/8, and so on, then that can be problematic. In each case, careful analysis is required to determine whether the issue can arise; if it can arise, whether it is problematic or can be tolerated; and, if it cannot be tolerated, how best to rule it out (Davis, 1992a).

6.8 Further Readings

Fisher (2008) surveys temporal representations and reasoning in AI. Van Benthem (1983) surveys logics of time from a philosophical perspective.

Action languages are formal, logic-based approaches to representing actions and their consequences (Gelfond & Lifschitz, 1998; Lee, Lifschitz & Yang, 2013).

Temporal Action Logic (TAL) (Sandewall, 1994; Doherty & Kvarnström, 2008) is a theory (more precisely, a family of theories) of time and action, expressed in first-order logic with a variant of circumscription. It has a rich and highly systematic structure, and has been implemented in automated planners (Doherty & Kvarnström, 2001).

7. Space

Commonsense spatial reasoning arises in many AI tasks, including computer vision, particularly interpreting video and schematic diagrams; robotic manipulation; robotic navigation; many forms of physical and biological reasoning; even many aspects of folk psychology and folk sociology (Cohn & Renz, 2008). For instance a reader of the Mona Lisa story must understand that the Mona Lisa was originally *inside* the Louvre; that the thief came *into* the Louvre, presumably *through* a door or a window, went *to* the Salon Carré, came onto spatial *contact* with the painting, and remained in contact with it as he took it *out of* the museum. The reader of the speciation text must understand that a region that is a single *open container* at one level may consist of multiple containers at a *lower* level; that creatures need to be *close* or *in contact* in order to breed, and so on (Davis, 2013a).

Considering that there is a large body of computer science dealing with space, subsuming computational geometry and large areas within computer graphics and scientific computing, and considering that there is an enormous body of mathematics dealing with space, going back long before Euclid, one might reasonably suppose that everything needed for commonsense spatial reasoning was accomplished long ago, and that AI researchers will be able to find everything they need in the published literature. This is true to the extent that, as far as I know, any spatial relation that is needed for commonsense reasoning can in principle be defined in the language of the geometry of $\mathbb{R}^3$ and any inference that is needed can be justified as a geometric theorem. But that is hardly satisfying, as a computational theory; it can hardly be supposed that AI programs represent spatial information in terms of arbitrary logical formulas in the language of geometry plus set theory; or that spatial reasoning involves arbitrary theorem proving over that horribly intractable general theory.

Certainly, the computer science subfields of computer graphics, solid modeling, and computational geometry do present many tractable classes of representations of spatial information together with powerful algorithms. However, these are not a good fit to the needs of commonsense reasoning. The representations used in computer graphics and solid modeling are generally precise and complete (up to some level of approximation), whereas commonsense reasoning often involves reasoning from information that is highly incomplete. Consider, for example, the quotations at the top of this article; we are not told the location of the Salon Carré in the Louvre or the shape of the lake, nor do we need that information. Computational geometry does sometimes deal with incomplete information, but the geometric properties and relations that interest computational geometers are not the same as the ones that arise in commonsense reasoning.

Unfortunately, the disconnect between theory and practice persists in the literature on automated commonsense spatial reasoning. In many or most cases, the problems that have been studied are mathematically natural, but not particularly relevant to any kind of AI task or commonsense reasoning. In consequence, the literature consists to an unfortunately large degree of results that are of purely mathematical interest (if that). Meanwhile, though we can *ad hoc* define the vocabulary that we would need for the spatial inferences in our two sample texts, and we can justify the inferences geometrically, we have no theory that supports an elegant representation or efficient inference.

Logic-based representations for commonsense spatial reasoning vary along a number of dimensions:

- The space under consideration. Some theories are specific to $\mathbb{R}^2$. A few are specific to $\mathbb{R}^3$. Some apply generally to $\mathbb{R}^k$ for $k \geq 2$. Some apply generally to an topological space. It is rare that a theory is proposed that is inconsistent with the geometry of $\mathbb{R}^2$ or $\mathbb{R}^3$. (The geometry proposed by Fleck (1987) is one such.)

- The spatial entities considered. Probably the majority of the work in the area takes the universe of entities to be a class of regions, where a region is defined as a subset of the overall space that is "well-behaved" in some sense. Well-behavedness conditions can include such properties as topological regularity,[14] connectedness, boundedness, being a polygon, and so on. However, many other kinds of spatial entities have been considered in the literature including points, curves, surfaces, vectors, directions, angles, rigid mappings, and sets of regions.

- Static or dynamic. There is a substantial body of work on reasoning about continuous spatial change (Muller, 1998; Galton, 2000; Davis, 2012; Hazarika, 2012).

- Relations and properties included. This category obviously depends on the class of entities; there is an enormous range of possibilities. A few specifics will be discussed below.

- Richness of the logical language. Problem specifications can be restricted to simple grounded constraints, or ground Boolean formulas, or can include the full first-order language.

A general difficulty with these spatial languages that, with rare exceptions, inference is intractable (anywhere from NP-hard to high levels in the hierarchy of undecidability) even for languages with a very limited set of relations and restricted logical forms.

7.1 Region Connection Calculus

Probably the most extensively studied theory of qualitative spatial reasoning is the Region Connection Calculus[15] known as RCC-8 (Randell & Cohn, 1989; Randell, Cui, & Cohn, 1992). The entities are topologically regular regions in an arbitrary topological space. The theory uses eight different base binary relations between regions, as illustrated in table 8 and figure 8; any pair of regions **A** and **B** satisfies exactly one of these.

Further relations can be defined as disjunctions of these. For instance we define `PP(A,B)` (`A` is a proper part of `B`) as holding if either `TPP(A,B)` or `NTPP(A,B)`. The relation `O(A,B)` holds if `A` and `B` overlap in any way; that is, if any RCC8 relation other than `EC` or `DC` holds between them.

Figure 9 shows a basic dynamic theory of RCC8; assuming that `A` and `B` are changing continuously, the relation between them must follow the arcs in the diagram. For instance, if at one time the relation `NTPP(A,B)` holds, then it can only transition to `TPP(A,B)` or `EQ(A,B)` and not to any of the other relations.

14. A region R is "topologically open regular" if R is equal to the interior of the closure of R; it is "topologically closed regular" if R is the closure of the interior of R.
15. RCC-8 is closely related, both to the RCC-5 set of mereology relations discussed at the end of section 4 and to the Allen interval calculus discussed in section 6.1.

`EQ(A,B)` (EQual) — `A` and `B` are equal.
`NTPP(A,B)` (Non-Tangential Proper Part). — `A` is a subset of the interior of `B`.
`TPP(A,B)` (Tangential Proper Part) — `A` is a subset of `B` and the boundary of `A` meets the boundary of `B`.
`NTPPI(A,B)` (Non-Tangential Proper Part Inverse) — `B` is a subset of the interior of `A`. `TPPI(A,B)` (Tangential Proper Part Inverse) — `B` is a subset of `A` and the boundary of `B` meets the boundary of `A`.
`OV(A,B)` (OVerlap) — `A` and `B` overlap, but neither is a subset of the other.
`EC(A,B)` (Externally Connected) — `A` and `B` meet only at the boundary.
`DC(A,B)` (DisConnected) — `A` and `B` have no points in common.

Table 8: RCC-8 relations

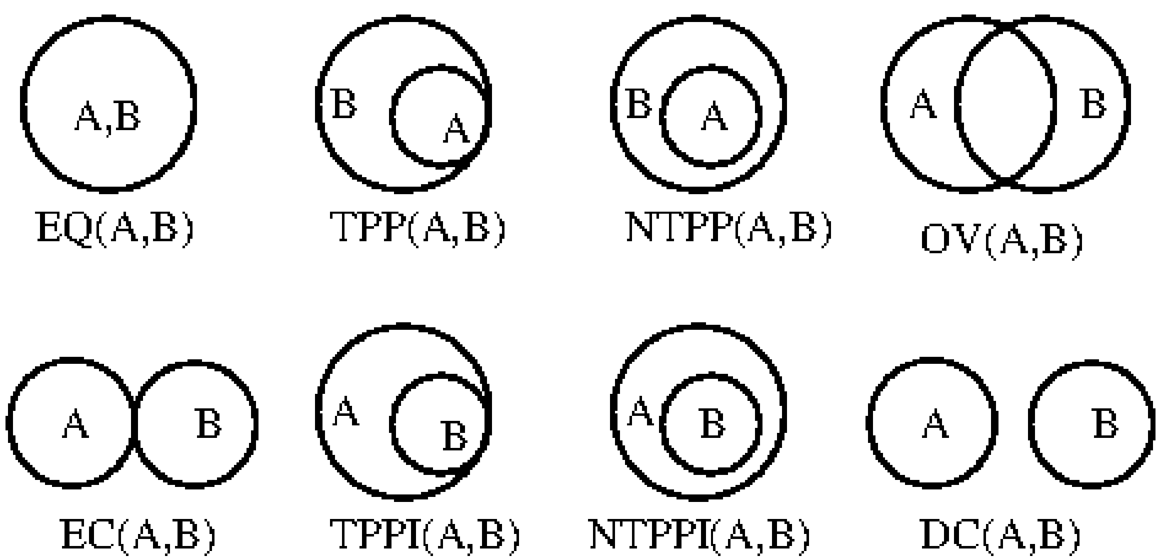


Figure 8: RCC8 relations

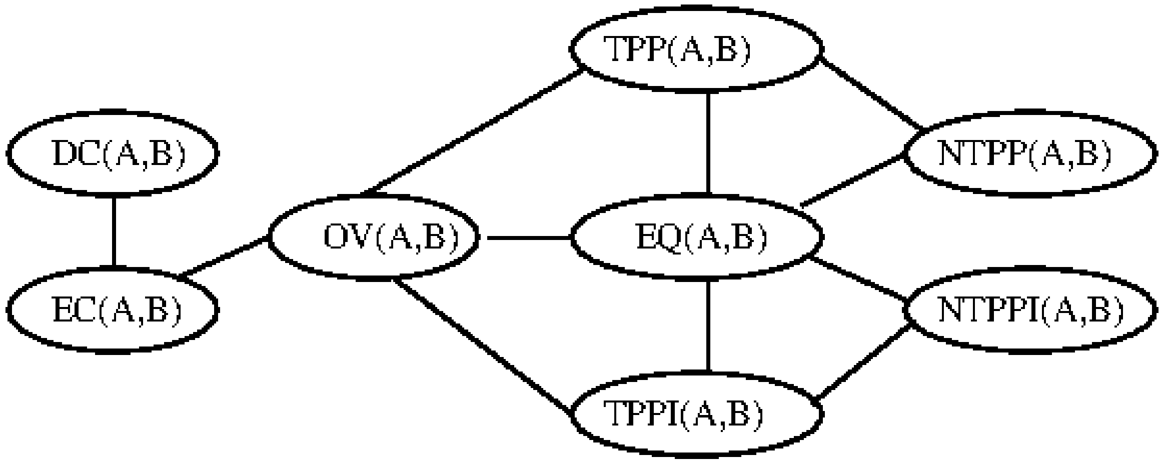


Figure 9: RCC8 transitions

As with the interval calculus (section 6.1), a transitivity table can be computed for the RCC-8 relations; the table can then be used to carry out some elementary spatial reasoning. For instance we can use the transitivity table to reason that, since on Tuesday morning the Mona Lisa is `DC` to the Louvre and the Salon Carré is `PP` to the Louvre, the Mona Lisa must be `DC` to Salon Carré.

As a more complex example of reasoning, Table 9 illustrates how the `RCC` theory of continuous motion can be used to infer that the fish cannot interbreed once the lakes have split. Supplemented with a sufficient spatio-temporal theory of topological relations and continuous change, the conclusion, "No two fish from the separate populations can come into contact." The representation may seem rather complex, but a simpler one would hardly be adequate. In fact, the actual inference involved in understanding the text is substantially *more* complex than table 9 indicates. This inference is stated in terms of the original fish in the two populations but the inference needed for the actual text requires understanding that the populations remain separated as one generation passes to the next. Moreover, many egg-laying fish, unlike mammals, do not have to come into physical contact to breed. Finally, constraint #7, that a fish must always be inside a lake, is something of a cheat; the real constraint is that a fish will soon die unless it is surrounded by water. All in all, a truly adequate formulation of the inference would probably be four or five times as long.

7.1.1 9-intersection Model

An alternative to RCC-8 for representing topological relations is the "9-intersection model" (Egenhofer, 1994; Egenhofer & Franzosa, 1991). This works as follows. For any region A, consider the three spatial sets A^o (the interior of A), A^- (the boundary of A) and A^c (the complement of A). Then given two regions A and B, their topological relations can be characterized by taking the nine pairwise intersections of the associated sets and asserting whether they are empty or non-empty. Table 10 shows the intersection table corresponding to the RCC-8 relation `TPP(A,B)` and one form of `EC(A,B)` (it excludes cases where A is an interior hole of B or vice versa).

Predicate symbols:
`Member(x,p)` — Individual `x` is a member of population `p`.
`PP(r,q)` — Region `r` is a partial part of region `q`.
`Inst(x,c)` — Entity `x` is an instance of category `c`.
`Extant(t,x)` — Object `x` is extant at time `t`.
`ContinuousMoving(x)` — Object `x` moves continuously.

Function symbol:
`Place(t,x)` — The region occupied by object `x` at time `t`.

Constant symbols:
`Pop1, Pop2` (two populations of fish).
`Lake1, Lake2` (lakes 1 and 2).
`Fish, Lake, Time` (the categories of fishes, lakes, and time instants).
`T0` (some time instant after the water level has fallen).

Given:

1. $\forall_{\mathtt{f}}$ `Member(f,Pop1)` $\Rightarrow$ `PP(Place(T0,f),Place(T0,Lake1))`.
(The fishes in population 1 are all in lake 1.)
2. $\forall_{\mathtt{f}}$ `Member(f,Pop2)` $\Rightarrow$ `PP(Place(T0,f),Place(T0,Lake2))`.
(The fishes in population 2 are all in lake 2.)
3. $\forall_{\mathtt{f}}$ `Member(f,Pop1)` $\vee$ `Member(f,Pop2)` $\Rightarrow$ `Inst(f,Fish)`.
(Each element of two populations is a fish.)
4. `Inst(Lake1,Lake)` $\wedge$ `Inst(Lake2,Lake)` $\wedge$ `Lake1` $\neq$ `Lake2`.
(Lakes 1 and 2 are lakes and are unequal.)
5. $\forall_{\mathtt{t}}$ `t` $>$ `T0` $\Rightarrow$ `Extant(Lake1,t)` $\wedge$ `Extant(Lake2,t)`.
(Both lakes are extant at all times after `T0`.)
6. $\forall_{\mathtt{x,y}}$ `Inst(x,Lake)` $\wedge$ `Inst(y,Lake)` $\wedge$ `x` $\neq$ `y` $\wedge$ `Inst(t,Time)` $\wedge$
`Extant(x,t)` $\wedge$ `Extant(y,t)` $\Rightarrow$
`DC(Place(t,x),Place(t,y))`.
(Two different lakes must be in disconnected regions.)
7. $\forall_{\mathtt{f,t}}$ `Inst(f,Fish)` $\wedge$ `Inst(t,Time)` $\Rightarrow$
$\exists_{\mathtt{l}}$ `Inst(l,Lake)` $\wedge$ `Extant(t,l)` $\wedge$
`PP(Place(t,f), Place(t,l))`.
(A fish must always be inside a lake.)
8. $\forall_{\mathtt{x}}$ `Inst(x,Lake)` $\vee$ `Inst(x,Fish)` $\Rightarrow$ `ContinuousMoving(x)`.
(Lakes and fish move continuously.)

Infer: $\forall_{\mathtt{f,g,t}}$ `t` $\geq$ `T0` $\wedge$ `Member(f,Pop1)` $\wedge$ `Member(g,Pop2)` $\Rightarrow$
`DC(Place(t,f),Place(t,g))`.

As discussed in the text, the purely spatio-temporal axioms relating `Place`, `ContinuousMoving`, and the RCC-8 relations are omitted.

Table 9: Inferring that the two populations of fish cannot come into contact

TPP(A,B)	A^o	A^-	A^c
B^o	Non-empty	Non-empty	Non-empty
B^-	$\emptyset$	Non-empty	Non-empty
B^c	$\emptyset$	$\emptyset$	Non-empty

EC(A,B)	A^o	A^-	A^c
B^o	$\emptyset$	$\emptyset$	Non-empty
B^-	$\emptyset$	Non-empty	Non-empty
B^c	Non-empty	Non-empty	Non-empty

Table 10: 9-intersection model of topological relations

Over the space of closed regular regions, the expressivity of the 9-intersection model is almost the same as RCC-8, and it is harder to use (e.g. to express composition tables). However, it lends itself more naturally to certain kinds of extension. For instance, it can be applied directly to non-regular regions; or the description of the intersection can be augmented with additional information, such as the number of connected components.

7.2 Exact Representation

As discussed above, most of the spatial computations done in computer science use geometric representations of shape that are exact up to some small tolerance. The algorithms that work on these can justified in terms of axiomatic Euclidean geometry, a theory that has been very well established for more than a century (and largely established for more than two millennia). Such representations are certainly compatible with a logical approach; they are easily given an exact semantics, and the calculations are, exactly or approximately, sound consequences of the representations. Why, then, have we neglected these?

One reason is that, though these algorithms *can be* formalized within a logical framework, in practice incorporating an explicit framework of symbolic logic rarely adds anything useful.[16] That is to say, the computer scientists who develop these data structure algorithms and the programmers who implement them as programs of course think about the geometry that underlies them, but rarely if ever think about the *axiomatics* of the geometry, or any other logical aspect of the geometry or the algorithms. The axiomatics would certainly arise if one is using proof verification technology to formally prove the correctness of the algorithms, and might come up in integrating these kinds of algorithms with other symbolic reasoning.

The deeper reason is that people often can, and often must, do spatial reasoning using purely qualitative information. Both of our sample texts, particularly the second, depend strongly for their understanding on various forms of spatial reasoning, and in neither are the exact shapes or dimensions given. It seems, therefore, that important kinds of commonsense spatial reasoning involves only qualitative reasoning; and conversely, qualitative spatial reasoning seems to be much more important in commonsense reasoning than in

16. This is equally true in mathematical practice; a mathematician working on a problem that is not specifically in the area of mathematical logic rarely thinks about the axiomatic foundations of the area, let alone the expression of those foundations in logical terms (Davis & Hersh, 1985).

many other areas of computer science. Therefore, research in AI commonsense reasoning research has tended to gravitate toward that problem. However, the roles of qualitative versus exact spatial representations in cognition and in artificial intelligence, and how they can be integrated, is not well understood and much debated.

7.3 Further Readings

Handbook of Spatial Logics (Aiello, Pratt-Hartmann, & van Benthem, 2007) is a collection of papers on logics of space.

Qualitative representations reasoning techniques have been developed for many other spatial properties and relations (Cohn & Renz, 2008) including distance (Clementini, Di Felice, & Hernández, 1997), size (Gerevini & Renz, 2002), direction (Ligozat, 1998; Renz & Mitra 2004), betweenness, convexity, (Pratt, 1999), and connectedness (Kontchakov, Pratt-Hartmann, Wolter, & Zakharyaschev, 2010). Much of this work has taken the form of defining a JEPD set of qualitative relations for some property; computing the transitivity table manually, automatically, or semi-automatically; implementing an inference engine based on the transitivity table; and then studying the theoretical properties (e.g. computational complexity) of the representation and the empirical properties of the inference engine.

The logical and computational properties of various qualitative languages have been extensively studied (Grzegorczyk, 1951; Bennett, 2001; Pratt-Hartmann, 2007; Davis, 2013b).

Hahmann and Grüninger (2011) take a very different approach to qualitative spatial reasoning, based on abstract algebra and category theory.

8. Physics

The computer science literature on physical reasoning is vast, and the AI literature on commonsense physical reasoning, though much smaller, is quite substantial (Davis, 2008a), but formal logic is significantly involved in only a tiny fraction of either of these. In view of this, we will begin by discussing why these non-logic based theories are insufficient for commonsense physical reasoning, and then we will survey the small literature on logic-based theories.

8.1 Mainstream Physical Reasoning in Computer Science

Reasoning about physical systems has been the subject of a substantial fraction of computer science since the advent of the electronic computer, and of a substantial fraction of applied mathematics since the seventeenth century. (Indeed, the electronic computer was originally developed, in large measure, in order to do calculations about physical systems.) At this point the software technology for physical computation, both for scientific computations, and for realistic world simulation for computer games, is extremely rich, powerful, and sophisticated. By contrast, the field of automated commonsense physical reasoning is much smaller, and the portion of that field that is logic-based is tiny.

Computational physical reasoning is almost entirely based on some form of *simulation;* that is, tracking the state of the system, completely specified, over a sequence of time points, using exact physical laws formulated as differential equations or difference equations. In computer games, such simulators are called "physics engines". The status of simulation vis-à-vis logic-based commonsense reasoning is very much the same as the status of exact spatial representations, discussed in section 7.2:

- There is a substantial body of work on axiomatizing the underlying physics, but, unlike geometry, there is nothing close to a definitive account. We will discuss this briefly in section 12.3.
- Adding a logical substrate to a physics engine, or recasting the algorithms used by a physics engine in logical terms, is possible in principle but essentially useless in practice. Logical analysis of the workings of a programs may be useful for clarifying the implicit assumptions that it makes.

Physics engines, though powerful, are in many ways poorly suited to the needs of commonsense reasoning. Davis and Marcus (2016) discuss a number of features of physical reasoning problems that are inherently difficult for simulation, including incomplete information, unknown physics, and irrelevant complexity. Two examples:

- **Incomplete information and irrelevant complexity**. Suppose that you have a closed can, half-way full of sand, and you shake it up and down a few times. You wish to infer that the sand stays in the can. In a logic-based approach, that inference is straightforward (Davis, Marcus, & Chen, 2013). In a pure simulation approach, it would be necessary to specify, as boundary conditions, the exact shape and initial position of each grain of sand and the exact trajectory of the shaking, and then it would be necessary to trace every collision of two grains of sand together.
- **Unknown physics.** In the speciation text, all we need to know about the creatures involved is that they are water creature, who cannot travel far by land or air. *All* other properties of the creatures involved — what they are, how large they are, how they move, how they live — are irrelevant. A reader who understands the text will immediately apprehend that this kind of speciation is plausible for fish, jellyfish, and octopuses, and implausible for otters, penguins, or frogs. Conversely, if one reads that some unknown creature has divided into two species in this one, one can infer that it is a water creature. In a simulation-based system, one would have to simulate some large sample of creatures, with different characteristics, different populations, and different motions to arrive at this conclusion.

8.2 Qualitative Reasoning

Most of the work in the AI literature on commonsense physical reasoning is on a collection of techniques known as "qualitative reasoning" (QR) (Bobrow, 1985; Forbus, 2008).[17] In QR, the state of a system at a given time is characterized in terms of a collection of one-dimensional parameters. The "qualitative state" of the system is a collection of physically significant inequalities or equalities between two parameters or between a parameter and an important physical constant. The qualitative state constraints the sign of the derivative

17. The terminology is somewhat unfortunate, since there are actually many other forms of physical reasoning using qualitative information.

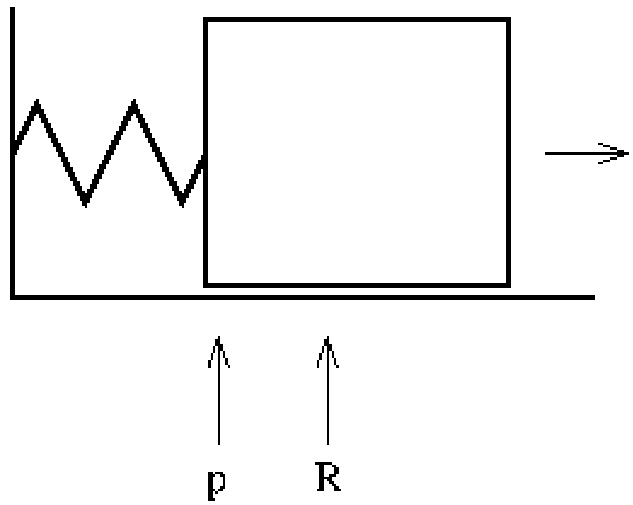


Figure 10: Mass on Spring

of the parameters. Using the signs of the derivatives, it can be determined what are the next possible qualitative states of the system. The program generates an "envisionment" of the system, which is a data structure expressing the possible sequences of qualitative states over time.

For example, in the simple "mass on spring" system shown in figure 10, the two parameters are the horizontal position (p) of the bob and its angular velocity (v). The position is characterized qualitatively in its relation to the rest length of the spring R; the velocity is characterized in terms of its sign. The system obeys the following rules.

$$\mathtt{sign}\left(\frac{\mathtt{dp}}{\mathtt{dt}}\right) = \mathtt{sign}(\mathtt{v}).$$

$$\mathtt{sign}\left(\frac{\mathtt{dv}}{\mathtt{dt}}\right) = \begin{cases} + & \text{if } \mathtt{p} < \mathtt{R} \\ 0 & \text{if } \mathtt{p} = \mathtt{R} \\ - & \text{if } \mathtt{p} < \mathtt{R} \end{cases}$$

Given this characterization of the dynamics of the system, the qualitative reasoner generates the envisionment shown in figure 11.

Qualitative reasoning was introduced in its general form in (Kuipers 1985; Forbus 1985; de Kleer & Brown, 1985). It has since been very much extended, and widely applied (Weld and de Kleer, 1990; Forbus, in preparation).

It is not difficult to give a logical account of qualitative reasoning; this is done for the de Kleer and Brown's (1985) component-based theory in and for Forbus' (1985) process-based theory by Davis (1990, 1992b). However, as with simulation, there is not much gained by doing so, except, arguably, to clarify the closed-world assumptions being made.

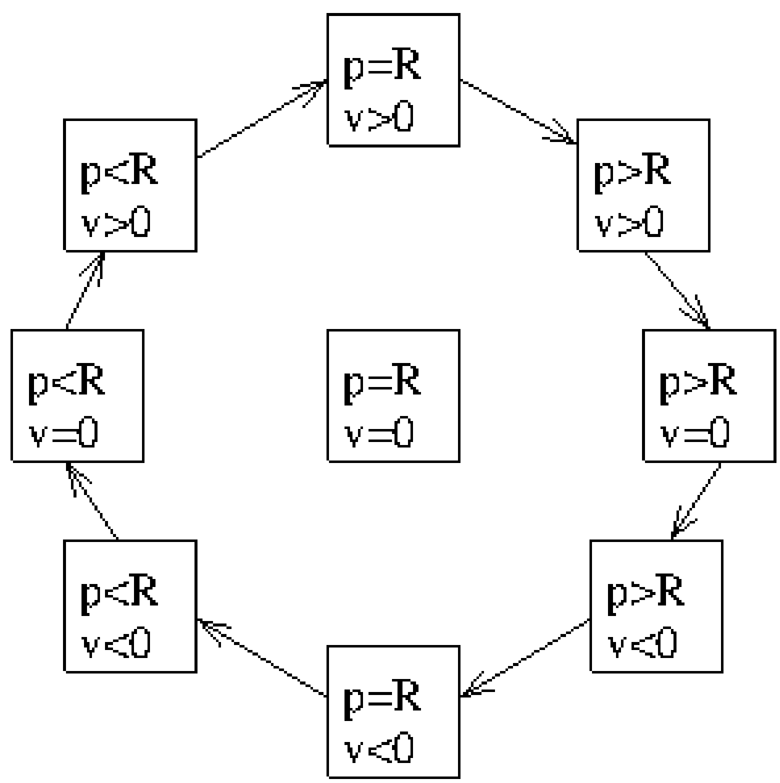


Figure 11: Envisionment

8.3 Logic-Based Analysis of Physical Reasoning

The literature of logic-based analysis of commonsense physical reasoning is small (a significant fraction is by the current author). It is somewhat scattershot; no systematic approach or framework for axiomatization has emerged. The physics addressed is mostly simple mechanical interactions of solid objects and of liquids with solid objects. The one near constant is that virtually all theories use a continuous model (linear or branching) or time and use Euclidean three-dimensional space as the basic spatial model.

A few examples of inferences that have been analyzed and justified in the literature:

- If water flows into a closed bathtub, then eventually the tub will overflow (Hayes, 1985).
- There is no way of carving a purely internal cavity in an object using a blade that is initially outside the object (Davis, 1993).
- If you have a cup partly full of water, and you lift it over an open pail and tilt it far enough, some of the water will pour from the cup into the pail (Davis, 2008b).
- If you have a small collection of small objects outside a large open box, and all of the objects and the box can be moved freely, then it is possible to move all the objects to a new location by loading them into the box, and then carrying the box, making sure to keep it upright (Davis, 2011).

Issues that have often arisen include the following:

Histories: Many useful inferences can be cast in terms of four-dimensional regions in space-time (equivalently, functions from an interval into regions of three-dimensional space) called "histories" (Hayes, 1979). It is therefore worthwhile including histories as entities in the ontology. The cost of doing so, however, is that axioms have to be stated that posit the existence of "all the histories that you will need".

Containment: A particularly important and common category of commonsense physical reasoning has to do with containers and their contents — how containers can be used to transport or protect their contents, and how to move stuff into and out of containers (Hayes, 1985; Davis, 2008b; Davis, 2011; Davis, Marcus, & Frazier-Logue, 2017).

Isolation: In physical reasoning, the most important form of the closed world assumption is an assumption of *isolation*. It is assumed that the problem statement includes (explicitly or implicitly) all the entities that could interact with the system under consideration. If you are going to make any useful inferences about what a tower of blocks is going to do, you have to be able to assume that no meteor, or SWAT team, or sudden plumbing disaster, is going to burst in and make havoc.

Inferring past transient states and constraining them. In everyday physical scenarios, it is often the case that, when a system changes from one equilibrium state, or quasi-equilibrium state, to another, it passes through a transient state that is extremely complicated, but whose details are entirely unimportant. For instance, if there is a tower of blocks, and a child knocks them over, the initial state is simple (a tower in stable configuration) and the final configuration is simple (the blocks are in some stable configuration on the floor) but in between there is a rapid sequence of changing states and events – blocks slide against one another and collide with one another, with contact patterns changing very rapidly. Even the fundamental physics of such transients is often not well understood; for instance, the thermodynamics of non-equilibrium systems is an important open problem (Lebon, Jou, & Casas-Vázquez, 2008). Both simulation and qualitative reasoning in principle have to track all of these changes of state. In a knowledge-based system, it is possible to skip much of that. However, the knowledge-based system must impose suitable constraints even in the transient condition, even if it cannot predict details. For instance, with the falling tower of blocks, it is important to infer that the blocks are reasonably near their starting positions and moving reasonably slowly. The axioms may not characterize the transient states exactly, but they should be strong enough to rule out the possibility that during the transient period the blocks turn into hummingbirds, make a quick trip to Hawaii, or attain Mach 3.

Order of magnitude reasoning. "Quick and dirty", "back of the envelope" reasoning about physical systems often involves reasoning about the general scale of the quantities involved, rather than their exact numeric value, and ignoring small quantities as negligible (Raiman, 1990; Weld, 1990).

Abstraction and multiple models: Certainly in formal scientific reasoning and probably in commonsense reasoning, a key first step in reasoning about a particular problem is choosing the right level of abstraction — deciding what aspects of the problem must be considered and which can be ignored. Moreover, it is often necessary to combine models; to carry out part of the reasoning using one model and then another part using a very different model. This problem has been often attempted, but with only limited success (Weld, 1992; Nayak, 1994).

9. Folk Psychology

A large part of human commonsense knowledge is our knowledge of people's minds: the properties and interactions of their knowledge, beliefs, perceptions, plans, goals, and emotions, and the relation of all these to their actions. People's commonsensical understanding of people's minds is known as "folk psychology" in both cognitive psychology and AI.

9.1 Knowledge and Belief

The most studied and best understood aspect of folk psychology is the theory of knowledge and belief. For example, in representing our sample text about the Mona Lisa, one needs to express facts like, "Before it happened, no one believed that the Mona Lisa could be stolen," "No one knew who had stolen the Mona Lisa, except the thief himself," "By midday Tuesday, many people knew that the Mona Lisa had been stolen," and so on.

One very common approach to the logical representation for knowledge and belief is a propositional modal logic, where there is a separate modal operator for each agent. That is, for each agent $\mathtt{A}$, we define modal operators $\mathtt{Know}_{\mathtt{A}}(\phi)$ ($\mathtt{A}$ knows proposition ϕ) and $\mathtt{Believe}_{\mathtt{A}}(\phi)$ ($\mathtt{A}$ believes proposition ϕ). These can be imbedded and combined with Boolean operators. Table 11 shows some examples which might arise in a card game.

For many purposes, it is desirable to extend this. The propositional logic can be extended to a first-order logic. The agent knowing can be elevated from an index to an

$\texttt{Know}_{\texttt{John}}(\texttt{InHand}(\texttt{Sara}, \texttt{KingOfSpades}))$
(John knows that Sara holds the king of spades.)

$\texttt{Know}_{\texttt{John}}(\texttt{InHand}(\texttt{Sara}, \texttt{KingOfSpades}) \vee \texttt{InHand}(\texttt{Tom}, \texttt{KingOfSpades}))$
(John knows that either Sara or Tom holds the king of spades.)

$\texttt{Know}_{\texttt{Sara}}(\texttt{InHand}(\texttt{Sara}, \texttt{KingOfSpades})) \vee \texttt{Know}_{\texttt{Sara}}(\neg\texttt{InHand}(\texttt{Sara}, \texttt{KingOfSpades}))$
(Either Sara knows that she holds the king of spades or she knows that she does not. That is, Sara knows whether or not she holds the king of spades.)

$\texttt{Know}_{\texttt{John}}(\texttt{Know}_{\texttt{Sara}}(\texttt{InHand(Sara,KingOfSpades)}) \vee$
$\quad \texttt{Know}_{\texttt{Sara}}(\neg\ \texttt{InHand(Sara,KingOfSpades)})).$
(John knows that Sara knows whether or not she holds the king of spades.)

Table 11: Examples of the propositional modal logic for knowledge and belief

$\forall_{\mathsf{a,c}}\ \texttt{Agent(a)} \wedge \texttt{Card(c)} \Rightarrow$
$\quad \texttt{Know(a,InHand(a,c))} \vee \texttt{Know(a,}\ \neg\texttt{InHand(a,c))}.$
(Everyone knows what cards they themselves hold.)

$\forall_{\mathsf{a1,a2,c}}\ \texttt{Past(Occur(Play(a1,c)))} \Rightarrow \texttt{Know(a2,Past(Occur(Play(a1,c))))}.$
(Everyone knows all the cards that have been played.)

$\texttt{Believe(Mary,}\ \neg\exists_{\mathsf{a,x}}\ \texttt{a} \neq \texttt{Mary} \wedge \texttt{Know(a,SSN(x,Mary))}.$
(Mary believes that no one besides her knows her social security number.)

$\texttt{Know(Frankie, Past(Believe(Frankie,}\ \forall_{\mathsf{x}}\ \texttt{Love(Johnnie,x)} \Rightarrow \texttt{x = Frankie)))}.$
(Frankie knows that she used to be believe that Johnnie loved only her.)

Table 12: Examples of an extended modal representation for knowledge and belief

actual argument, allowing quantification over agents.[18] A temporal representation can be integrated. Table 12 shows some examples of a language of this kind.

9.1.1 Characterizing Reasoning and Introspection

A major and largely unsolved problem in commonsense reasoning about knowledge and belief has to do with characterizing the agent's reasoning abilities. Can we admit a theory of belief in which an agent's beliefs can be inconsistent?[19] Can we admit a theory in which an agent fails to realize extremely obvious consequences of his beliefs or knowledge?

There are three types of answers here, none of them satisfactory:

18. If the agent is taken to be an index, as in $\texttt{Know}_{\texttt{A}}(\phi)$, then strictly speaking there is a separate modal operator for each agent. If the agent is taken to be an argument, as in $\texttt{Know(A,}\phi\texttt{)}$, then there is a single modal operator which takes two arguments.
19. This problem does not arise for knowledge, because, in the usual interpretation of "know", a fact that is known has to be true; and the set of true facts is necessarily consistent.

1. We assume nothing about the agent's reasoning capacity. We are perfectly happy imagining that the agent might believe that both that Tweety is a bird and that Tweety is not a bird; or might know that [Paris is in France and London is in England] but not know that [London is in England and Paris is in France]. However, this approach renders the theories of knowledge and belief almost useless; we cannot usefully interact with an agent whose knowledge is a completely random collection of facts depending on the exact phrasing.

2. We assume that the agent is a perfect reasoner, and instantaneously realizes all consequence of his knowledge and beliefs. In the technical terminology, knowledge and belief have the "consequential closure" property; this is known as the "assumption of logical omniscience" (Parikh, 1994). The problem is that this is not only false, but actually impossible, given what we know about the computational requirements of computing logical consequence.

3. We assume that the agent is an imperfect reasoner; he realizes some but not all of the consequences of his beliefs and knowledge. The problem is that logical consequences comes down to a series of small steps. Therefore, if an agent does not satisfy the assumption of logical omniscience, then at some point he is missing some glaringly obvious conclusion from his own beliefs. There does not seem to be any useful way of drawing a line in the sand, at least in logical terms. For instance, one might posit that a person knows facts that have short proofs from his set of axioms and does not know facts that require long proofs; but that fails in both directions. On the one hand, conversing in language with someone involves the assumption that they can understand the language and can do the necessary commonsense reasoning; and as far as we know, that can involve quite long chains of reasoning. On the other hand, if $\mathcal{L}$ is an NP-complete language, then by definition every short true statement $x \in \mathcal{L}$ has a short proof;[20] but it is not plausible that everyone knows the solution to every NP-complete problem.

The temptation is to write the problem off as unimportant; but that will hardly do. All kinds of interactions with other people involve assuming, both that they know a lot, and that you can rely on them to do various kinds of reasoning about a new situation. For instance when you communicate in speech or writing, you make various assumptions about the cognitive capacities and reasoning abilities of your audience, depending on the audience. A children's story makes one kind of assumption, a mystery novel makes a second kind, and a mathematical article makes a third kind; but they all make assumptions about what the audience knows and what it will understand; and since communication is largely successful, these assumptions must be reasonably realistic. Presumably what is actually needed is a detailed, domain-specific and even sometimes fact-specific theory about what various kinds of people do and do not know, and what kinds of reasoning they can and cannot do, but nothing close to such a theory has been developed. The gap between what we need for this kind of reasoning and what existing theories provide is enormous.

20. For example, if G is a small graph with a Hamiltonian path, then there is a short proof that G has a Hamiltonian path.

9.1.2 Theories with Consequential Closure

Let us return to the actual state of the art. The majority of work on logics of knowledge and belief assumes the consequential closure assumption: It is assumed that all of the "basic axioms" — e.g. logical axioms, properties of time, the axioms of knowledge and belief themselves — are known and that both knowledge and belief are closed under logical inference. Knowledge, though not belief, is *veridical*; the statement "A knows ϕ" entails that ϕ is true. Furthermore, two properties of introspection are often assumed:

- *Positive introspection:* If agent `A` knows ϕ, then he knows that he knows ϕ.
- *Negative introspection:* If agent `A` does not know ϕ, then he knows that he does not know ϕ.

A theory of a modal operator that posits all the above conditions (veridicality, positive and negative introspection) is called an "S5" theory, for historical reasons. A theory that posits positive and negative introspection but not veridicality is a "weak S5" theory. Most studies of the logic of knowledge and belief assume that knowledge conforms to an S5 theory and belief conforms to a weak S5 theory.

S5 models or other models with consequential closure are reasonable for applications where the limitations of other agents' or ones own reasoning powers are not important. For instance, in game playing, it is usual to assume that one's opponents play optimally, relative to their state of knowledge; therefore, an S5 logic is fine for an imperfect knowledge game, such as a card game. In applications where the limitations of reasoning are central, such as teaching, models with consequential closure cannot be used.

9.1.3 Possible Worlds Semantics

Modal theories of any kind that satisfy consequential closure can be characterized in terms of a *possible worlds* semantics, also known as a *Kripke model*. A possible world is one way that the world as a whole (in practice, the parts of the world under discussion) could be. Thus, a proposition ϕ is true in one world $w0$ and false in another world $w1$. A modal operator is expressed in terms of an *accessibility relation* between worlds; essentially, a set of labelled directed edges from one world to another. Thus, for example, in the theory of knowledge, there is a "knowledge accessibility relation" for each agent. World v is accessible from world u for agent a — that is, there is an arc labelled a from u to v — if world v is consistent with everything that a knows in world u. Therefore, if v is accessible from u and proposition ϕ is false in v, then $\neg\phi$ is consistent with what a knows in u; thus in u, it is not the case that a knows ϕ. Therefore, the statement that a knows ϕ in u. corresponds to the condition that ϕ is true in every world accessible form u.

For instance, consider the possible worlds structure in figure 12. The following statements hold:

> In world $W1$, A knows Q. The two worlds accessible for A from $W1$ are $W1$ itself and $W2$, and Q is true in both. Likewise, in $W2$ A knows Q and in $W3$, A knows $\neg Q$.
>
> In world $W1$, A does not know whether P, because P is true in $W1$ and false in $W2$, and both of these are accessible for A from $W1$

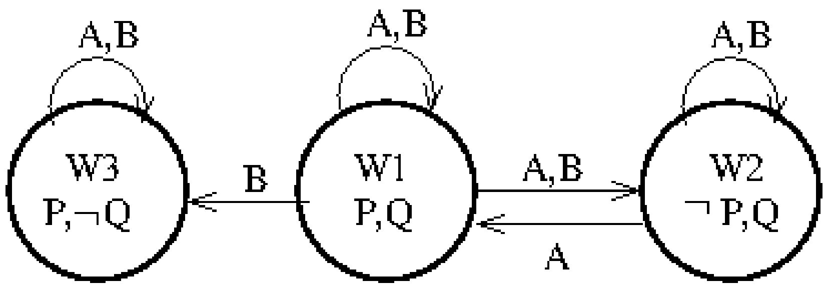


Figure 12: Possible Worlds

> In world $W1$, B does not know whether P, because P is true in $W1$ and $W3$ and false in $W2$ and all these are accessible for B from $W1$. Likewise B does not know in $W1$ whether Q, because Q is true in $W1$ and $W2$ and false in $W3$, and all these are accessible for B from $W1$. However B does know in $W1$ that $P \vee Q$, because that is true in all the accessible worlds.
>
> In world $W1$, B knows that A knows whether Q, because A knows whether Q in all accessible worlds.

In an S5 logic, the accessibility relation for any particular agent is an equivalence relation over possible worlds.

When the language of knowledge and belief is expressed in terms of possible worlds, it becomes a standard first-order theory, rather than a modal theory. From the standpoint of automated reasoning, this has the advantage that reasoning can be carried out using an inference engine for first-order logic rather than requiring specialized techniques to deal with the modal operators.

A *dynamic epistemic logic* is a logic that describes how knowledge can change over time (van Ditmarsch, van der Hoek, & Kooi, undated). One elegant logic of this kind can be constructed by combining a situation-based model of time with a possible-worlds model of epistemic relation by identifying situations with possible worlds. Thus a possible world/situation is a way that the world could possibly be at an instant. Statements about time then are expressed in terms of temporal relations between situations; statements about knowledge (or other propositional attitudes) are expressed as statements about all knowledge-accessible worlds. Table 13 shows some examples; here the three place predicate `KnowAcc(a,u,v)` holds if possible world `v` is knowledge-accessible from world `u` relative to agent `a`'s knowledge in `u`.

9.1.4 General Reading

The modal logic of knowledge and its possible worlds semantics was introduced by Hintikka (1962). Halpern and Moses (1985) give an overview of the modal logic of knowledge and its possible worlds semantics. The integration of the possible worlds semantics for knowledge with the situation based representation of time was first studied by Moore (1985a); that approach was pursued further in the KARO theory (van Linder, van der Hoek, & Meyer, 1998). *Reasoning about Knowledge* (Fagin, Halpern, Moses, & Vardi, 1995) presents logical

$\forall_{\texttt{t0a}}$ `KnowAcc(John,T0,t0a)` $\Rightarrow$
  $\exists_{\texttt{t1a}}$ `Occurs(t1a,t0a,Play(Mary,Club2)).`
(John knows in situation T0 that Mary has just played the two of clubs.)

$\forall_{\texttt{a1,a2,c,t1,t2,t3,t3x}}$ `Occur(t1,t2,Play(a1,c))` $\wedge$ `t2` $<$ `t3` $\wedge$ `KnowAcc(a2,t3,t3x)` $\Rightarrow$
  $\exists_{\texttt{t1x,t2x}}$ `t2x` $<$ `t3x` $\wedge$ $\wedge$ `Occur(t1x,t2x,Play(a1,c)).`
(Everyone knows all the cards that have been played.)

$\forall_{\texttt{tnx}}$ `KnowAcc(Frankie,Now,tnx)` $\Rightarrow$
  $\exists_{\texttt{t1x}}$ `t1x` $<$ `tnx` $\wedge$
    $\forall_{\texttt{t1y}}$ `BelieveAcc(Frankie, t1x,t1y)` $\Rightarrow$
      $\forall_{\texttt{a}}$ `Holds(t1y,Loves(Johnny,a))` $\Leftrightarrow$ `a = Frankie.`
(Frankie knows that she used to be believe that Johnnie loved only her.)

Table 13: Combining temporal and epistemic knowledge in a possible worlds representation

theories of knowledge and communication (discussed in section 10.2). Their theories use modal propositional logic and a linear model of time.

An alternative approach is to take the propositional argument of `Know` to be the string of characters expressing the proposition (Haas, 1986; Morgenstern, 1988). In principle, this gives a much more flexible theory than the modal theory or the possible worlds theory; in practice, it is hard to take advantage of that flexibility in any useful way. Also, care must be taken to avoid paradoxes of self-reference.

Dynamic Epistemic Logic (van Ditmarsch, van der Hoek, & Kooi, 2007) is a textbook and monograph surveying logics that combine knowledge and time.

Theoretical Aspects of Reasoning about Knowledge is a bi-annual conference on the logic of knowledge.

9.2 Plans and Goals

A second critical aspect of an agent's mental state is his set of plans and goals. For example, in the Mona Lisa story we want to infer (a) that the thief had the goal of possessing the Mona Lisa while remaining out of prison; (b) that the museum administration initially had the goal of preventing its artwork from being stolen; and (c) that once goal (b) failed, the museum administration adopted the goal of recovering the Mona Lisa.

Within the AI literature, the representation of plans and goals has been most extensively studied in the context of automated planning. The issues that arise in representing and reasoning about plans and goals for commonsense reasoning generally are somewhat different, for a number of interacting reasons:

- Automated planning is generally concerned with first-person plans; the reasoner is constructing plans for himself. Commonsense reasoning is often concerned with reasoning about the plans and goals of someone else.

- In automated planning, the form of inference under consideration is almost always, "Find a plan (or a good plan, under some measure of goodness) to achieve a given goal." In commonsense reasoning, other forms of reasoning are more prominent. In particular, the problem of *plan recognition* is often critical in understanding narrative (either text or video). Other kinds of inference include determining that a goal cannot be achieved, or finding constraints that apply to all (reasonable) plans to achieve a given goal.

- In automated planning, goals are almost always exogenous; the goal is stated as part of the problem statement. In commonsense reasoning, it is often necessary to infer the goals of another actor; therefore there has to be some underlying theory of what goals are plausible.
- The objective of automated planning is a plan that is executable; hence the emphasis tends to be on fully specified plans. Partially specified plans are constructed primarily as initial steps toward a fully specified plan. By contrast, in reasoning about a different agent, it is often impossible to determine all the details of his plan; one can only infer some of the characteristics of the plan.
- In automated planning, a reasoner is only interested in plans that he knows or believes to be feasible. However, a reasoner may see that the plan that a second agent is working on is in fact infeasible, either because the agent is more ignorant than the reasoner, or because the agent is an imperfect planner.
- In automated planning, a goal is usually a state to be achieved, once. In a broader setting, a goal may be to maintain an existing state (e.g. "Keep the museum's artwork in the museum") or to achieve a state repeatedly (e.g. "Attract at least 1000 visitors per day to the museum").

9.2.1 General Reading

Automated Planning: Theory and Practice (Ghallab, Nau, & Traverso, 2004) is a textbook covering high-level planning, with extensive discussion of logical analysis. *Planning Algorithms* (LaValle, 2006) is a textbook covering lower-level planning and its relation to robotics, with extensive discussion of mathematical model, but not much logic as such. (The intersection between the two books is very small.)

9.2.2 Plans and Goals in the Situation Calculus

In the most basic form of planning in situation calculus, a goal is a fluent to be achieved, and a plan is a sequence of primitive actions. This can be represented by extending the scope of the sort "action" in the situation calculus to include a sequence (possibly null) of actions and adding the primitives and axioms in table 14.

The language of plans can then be extended to include more powerful operators such as conditionals, loops, subroutines, and exception handling. These are useful if either

- the reasoner is developing a general plan schema applicable to many situations (this is generally the case in programming);
- the world is incompletely known, and the planner will gain knowledge during the course of execution (this is the case in event-driven programming);

Symbols:
`Null` – Constant. The null sequence of actions.
`Seq(p1,p2)` – Function. Execute plans `p1` and `p2` in sequence.
`Cond(q,p1,p2)` – Function. If fluent `q` do plan `p1` else `p2`
`While(q,p)` – Function. Repeat plan `p` while `q` is true.

Axioms:
$\forall_{s}$ `Result(s,Null) = s.`
$\forall_{s,p1,p2}$ `Result(s,Seq(p1,p2)) = Result(Result(s,p1),p2).`
$\forall_{s,q,p1,p2}$ `[Holds(s,q)` $\Rightarrow$ `Result(s,Cond(q,p1,p2)) = Result(s,p1)]` $\wedge$
`[`$\neg$`Holds(s,q)` $\Rightarrow$ `Result(s,Cond(q,p1,p2)) = Result(s,p2)].`
$\forall_{s,q,p}$ `Result(s,While(q,p)) = Result(s,Cond(q,Seq(p,While(q,p)),Null)).`

Table 14: Plan operators in the situation calculus

- the sequence of actions is long and structured, and can be more elegantly or compactly represented as a loop (compare the use of "repeat" symbols in musical scores).

In a hierarchical transition network (HTN) planner, plan actions are organized in a hierarchy (Nau et al., 2001). Actions have subactions that are constrained by preconditions and temporal constraints, and by *protections,* that ensure that a precondition of an action is not undone before it is needed. For instance, in the blocks world, there might be a rule that the action `Move(x,y,z)` (Move block `x` from support `y` to support `z`) can be achieved through the three actions `MakeClear(x)`, `MakeClear(z)`, `PutOn(x,y,z)` with preconditions `Block(x)` and `On(x,y)`; the constraint that the two `MakeClear` actions precede the `PutOn` action; and the protection that the *Clear* fluent achieved by the `MakeClear` steps persists until the beginning of the `PutOn` actions. This can be represented in McDermott's (1982) temporal logic using the axioms shown in table 15.

9.3 Intention Logic and BDI

Intention logic, developed by Cohen and Levesque (1990), is a theory relating actions, beliefs, intentions, and goals. There are four basic modal operators and two operators on actions, as shown in table 16.

More complex relations are constructed by combining these. For instance, having defined the operators $\Box\phi$ (ϕ will always be true), `Later(`ϕ`)` (ϕ is not now true, but will be later), and `Before(`ϕ,ψ`)` (ϕ will become true before ψ does)), one can define the concept of a persistent goal.

```
P-Goal(a,p) ≡
Goal(a,Later(p)) ∧ Believe(a,¬p) ∧
     Before(Believe(a,p) ∨ Believe(a, □ ¬p),
           ¬Goal(a,Later(p)).
```

New Symbols:
`Move(x,y,z)` — The action of moving block `x` from `y` to `z`.
`MakeClear(x)` — The action of making `x` clear.
`Protected(ta,tb,f)` — Fluent `f` remains true from time `ta` to time `tb`.

Axioms:

$\forall_{x,y,z,ta,tb}$ `Occurs(ta,tb,MakeClear(x))` $\Leftrightarrow$
    `[[ta=tb` $\wedge$ `Holds(ta,Clear(x))]` $\vee$ `%` Trivial case
    $\exists_{y,z}$ `Holds(ta,On(y,x))` $\wedge$ `Move(ta,tb,Move(y,x,z))].`

$\forall_{x,y,z,ta,tb}$ `Occurs(ta,tb,Move(x,y,z))` $\Leftrightarrow$
    `[x` $\neq$ `y` $\neq$ `z` $\neq$ `x` $\wedge$ `Block(x)` $\wedge$ `Holds(ta,On(x,y))` $\wedge$
    $\exists_{tc,td,te,tf,tg}$ `ta` $\leq$ `tc` $\leq$ `td` $\leq$ `tg` $\leq$ `tb` $\wedge$ `ta` $\leq$ `te` $\leq$ `tf` $\leq$ `tg` $\wedge$
        `Occurs(tc,td,MakeClear(x))` $\wedge$ `Occurs(te,tf,MakeClear(z))` $\wedge$
        `Occurs(tg,tb,PutOn(x,y,z)` $\wedge$
        `Protected(td,tg,Clear(x))` $\wedge$ `Protected(tf,tg,Clear(z))].`

Table 15: Axioms for HTN planning

Modal Operators
`Believe(a,`ϕ`)` – Agent `a` believes proposition ϕ.
`Goal(a,`ϕ`)` – Agent `a` has the goal of achieving ϕ.
`Happens(`α`)` – Action α will happen next.
`Done(`α`)` – Action α has just happened.

Action Operators
$\alpha : \beta$ – Sequence of action α then action β.
$\phi?$ – Action of testing the truth of proposition ϕ.

Table 16: Intention Logic

Proposition `p` is a persistent goal of agent `a` if `a` has the goal that `p` should be true later, `a` believes that `p` is not now true, and `a` will not abandon his goal of `Later(p)` until either he believes that `p` has been achieved or he believes that it is unachievable.

A similar model, known as BDI (beliefs, desires, intentions) model was developed by Rao and Georgeff (1991), based on Bratman's (1987) theory of rational action. A number of implementations have been developed, most notably the Procedural Reasoning System (Georgeff & Lansky, 1986), and these have been used in some real-world applications.

The logical theory of BDI is an extension of the branching time modal logic CTL*. The temporal structure is a discrete branching model of time. The semantics of the cognitive modal operators uses a possible worlds structure with accessibility relations. However, rather than identify a possible world with a single time point, as discussed in section 9.1.3, a world is instead considered to be a branching time structure over a set of time points. This enables a rich collection of semantic relations between modal operators.

There are two kinds of formulas in BDI (Rao & Georgeff, 1991): *state formulas*, which characterize a time point and *path formulas*, which characterize a time line. Any state formula is also a path formula, but not vice versa. The language uses the following symbols:

- The standard Boolean operators and quantifiers.
- The predicates `Succeeded(e), Failed(e), Done(e), Succeeds(e), Fails(e), Does(e)`, where `e` is a term denoting an event. These formulas characterize a particular time point. The past-tense predicates refer to the primitive action that ends at the reference time point. The present-tense predicates `Does(e), Succeeds(e), Fails(e)` apply to action `e` at time point v if the agent has resolved in v to execute `e`; so, given that resolution, that is the only possible continuation of the time line. For example, the predicate `Succeeded(PutOn(A,B))` is true at time point v if v is the end state after a successful execution of `PutOn(A,B)`; the predicate `Succeeds(PutOn(A,B))` is true at time point v if the agent intends in v to put `A` on `B` and succeeds in doing so.
- The modal operators `Believe(`ϕ`)`, `Goal(`ϕ`)`, and `Intend(`ϕ`)`, where ϕ is a state formula. These are state formulas.
- The modal operator `Optional(`ψ`)` where ψ is a path formula. This is a state formula. The formula `Optional(`ψ`)` holds at time point v if some time line p starting at v satisfies ψ.
- Three modal operators over path formulas $\psi_1 \cup \psi_2$, $\bigcirc\psi$, and $\Diamond\psi$. These apply to a reference time line p. The formula $\psi_1 \cup \psi_2$ states that, along p, ψ_1 holds until ψ_2 becomes true. The formula $\bigcirc\psi$ states that ψ holds over the path that starts at the second time point on p. The formula $\Diamond\psi$ states that ψ holds over some suffix of p.

9.4 Further Reading

Perception: Reasoning about agents involves reasoning about what they will or will not perceive. What an agent can or cannot perceive depends on the physical environment, and affects the agent's knowledge state. Perception can either be taken as an action, analogous to testing a Boolean condition in a program, or as a passive consequence of being in an environment (Levesque, 1996; de Giacomo & Levesque, 1999; Son & Baral, 2001).

Cognitive robotics (Levesque & Reiter, 1998; Thielscher, 2005; Levesque & Lakemeyer, 2008) is a research programme aimed developing knowledge representation and automated reasoning techniques specifically for the high-level control of robots. For the most part, the logical aspects of this work are combinations and extensions of the theories of time, planning, knowledge, and perception discussed above.

Emotion: An understanding of human emotions is of course important to narrative understanding or interacting with people. Marsella, Gratch, and Petta (2010) survey and analyze the large literature on computational theories of emotion. *The Cognitive Structure of Emotions* (Ortony, Clore & Collins, 1990) is a classic cognitive science analysis characterizing various emotions and words for emotions that is often drawn on in the AI literature (Steunebrink, Dastani, & Meyer, 2007).

Gordon and Hobbs (in preparation) develop a rich causal theory of emotions, with defeasible causal rules characterizing both the circumstances that give rise to emotions and the effects of emotions on actions. (The logic is a first-order logic with non-monotonic inference.)

10. Multiple Agents

From the standpoint of AI, probably the most complex situations faced by humans on a regular basis are those that involve interacting with other agents (van der Hoek & Wooldridge, 2008; Wooldridge, 2009).

Other agents can be dealt with one-on-one, in small groups, in large groups, or in enormous groups (e.g. a political election) They can be well-known or unknown. Communication can be perfect, imperfect, indirect, or non-existent. Agents can be cooperative, adversarial, or simply pursuing their own interests.

Game theory is the mathematical analysis of the interaction of agents; and there is a large literature on the logical formulation of game theoretic domains (Alur, Henziger, & Kupferman, 2002; Pauly, 2002; Pauly & Parikh, 2003). However, this kind of analysis is often quite far from commonsense reasoning, so we omit this in this paper.

10.1 Common Knowledge

Two agents, Ann and Bob, have common knowledge of proposition ϕ if Ann knows ϕ and Bob knows ϕ and Ann knows that Bob knows ϕ and Bob knows that Ann knows ϕ and Ann knows that Bob knows that Ann knows ϕ and so on. More generally, a set of agents S has common knowledge of a proposition ϕ if

- every agent in S knows ϕ; and
- (recursively) every agent in S knows that S has common knowledge of ϕ.

As we shall see below, common knowledge is important for idealized theories of communication.

Strictly speaking, common knowledge is a property of a specific characterization of a set of agents rather than of the actual set. For instance, suppose that Ann and Bob are playing a multiplayer online video game under the *noms de guerre* Attila and Babur, and suppose that Ann and Bob are classmates in "real life", but are not aware of the identities of each others' characters. Then the set { `Attila, Babur` } has common knowledge of the proposition, "The big monster is attacking with laser beams.", but { `Ann, Bob` } does not, and { `Ann, Bob` } has common knowledge that "The class has an assignment due next week", but { `Attila, Babur`} does not.

Common knowledge can thus be represented as a modal operator, $\texttt{ComKnow(S,}\phi\texttt{)}$ satisfying consequential closure and the following two axiom schemas:

$$\forall_s \ \texttt{ComKnow(s,}\phi\texttt{)} \Rightarrow \forall_{\texttt{a}} \texttt{a} \in \texttt{s} \Rightarrow \texttt{Know(a,}\phi\texttt{))}.$$
$$\forall_s \ \texttt{ComKnow(s,}\phi\texttt{)} \Rightarrow \texttt{ComKnow(s,ComKnow(s,}\phi\texttt{))}.$$

Parikh (2005) discusses the relation of logical omniscience and common knowledge.

10.2 Communication

Communication is an action that one actor performs with the intent that a second actor will perceive it and in some respect alter their actions or mental state.

The contents of an utterance, the characteristics of the code and of the medium, the constraints on the speaker, and the effect of the utterance on the hearer, can all vary with circumstances. One idealized model, often useful, is the following:

- The content of the utterance is a proposition ϕ.
- The speaker knows that ϕ is true.
- The speaker knows the identity of the recipient.
- When the utterance is finished, the recipient will know the identity of the speaker and the content of the communication.
- The speaker and the hearer have common knowledge of all the above.

It is easily shown that, if these conditions hold, then when the utterance is finished, the speaker and hearer have common knowledge of ϕ. Many weaker systems of conditions have been studied.

In reasoning about media of communication other than speech, features of the medium and time-delays between sending and receiving the communication may become important.

10.2.1 General Reading

Logic-based theories of propositional communication (i.e. the content is an atomic proposition) were considered in great depth by Fagin, Halpern, Moses and Vardi (1995). A logic of first-order communications was developed by Davis (2005).

A number of "agent communication languages" have been developed to facilitate communication between robots or softbots, many with a grounding in a formal logical theory (Labrou and Finin, 1997; Labrou, Finin, & Peng, 1999). Chopra et al., (2013) record a discussion of the major issues involved in this enterprise among six scientists of varying opinions. The FIPA ACL standard (Foundation of Intelligent Physical Agents / Agent Communication Language) (Poslad 2007; FIPA 2002) has been adopted by many agent programming languages and frameworks.

11. Other Approaches to Automating Common Sense

Broadly speaking, there are four alternative approaches in AI that have been applied to automating commonsense reasoning (Davis & Marcus, 2015). All of these create a symbolic knowledge base.

11.1 Informal Knowledge-Based Analysis

A knowledge base is created by hand by experts but the overall language and the inference rules tend to be left vague, and there are rarely any semantic models. This approach was popular in the 1970's and early 1980's, particularly in the research groups of Marvin Minsky (Charniak, 1972; Minsky, 1975) and Roger Schank (Schank & Abelson, 1977; Dyer, 1983).

11.2 Large-Scale Systems

Some large knowledge bases of commonsense knowledge have been constructed. The best known is CYC (Lenat, Prakash, & Shepherd, 1986). CYC uses a well-defined, rich representation language, which extends first-order logic, and a powerful inference system. Research-CYC, which is available to be licensed for research purposes, contains 500,000 concepts and 5,000,000 facts. However, there is no published evaluation or account that describes what kind of commonsense knowledge is included or what kinds of commonsense inference the system can perform.

11.3 Web Mining

There is a large body of work on systems that extract information from web documents, and some of this information relates to commonsense knowledge. For the most part, these systems are quite successful at extracting taxonomic knowledge and somewhat successful at extracting relations between individuals (e.g. Babe Ruth played for the New York Yankees). For example, the ProBase system (Wu, Li, Wang & Zhu, 2012) has collected 2.6 million categories with 20.7 IsA relations with 92% accuracy. Other notable projects of this kind include KnowItAll (Etzioni et al., 2004), NELL (Mitchell et al., 2015) and numerous others.

11.4 Crowd Sourcing

Some attempts have been made to use crowd-sourcing techniques to compile knowledge bases of commonsense knowledge (Havasi, Pustejovsky, Speer, & Lieberman, 2009). Many interesting facts can be collected this way, with much higher accuracy than is achievable by web mining. However, naïve users find it difficult and irksome to make the careful distinctions in quantifier structure, in distinguishing similar meanings of words, in notating temporal relations, and so on, that are needed to support reliable commonsense inference. The result is illustrated in figure 4, which was generated by the ConceptNet crowdsourcing project; the problems with this representation have been discussed in section 5.

12. Related Areas of Research

The domains of commonsense knowledge are central to many fields, and the techniques of logical representation and logical inference resemble, to greater and lesser degree, essentially *any* systematic representation of knowledge or method of reasoning. Therefore, there is an almost unlimited collection of areas of study in every intellectual area that to some extent relates to or parallels the logical representation of commonsense knowledge. For example, dancing as an activity seems very far from the design of AI systems; but the long-standing problem of developing a notation for recording choreography is in many ways similar to the problem of developing a representation for commonsense reasoning about human manipulation (Laumand & Abe, 2015).

This list of related areas of research is thus certainly incomplete and somewhat arbitrary.

12.1 Related Areas in AI

12.1.1 Natural Language Processing

The idea that extracting the meaning of texts is an important aspect of natural language processing technology has been steadily diminishing over time, but it has not yet been entirely eliminated, and perhaps very recently is enjoying a revival (Sachan & Xing, 2016; Liang, 2016). Any attempt to systematize the meaning of texts, or aspects of their meaning, is apt to run into many of the issues that arise in representing commonsense knowledge. In particular, various natural language *resources*, such as WordNet (Miller, 1995) and FrameNet (Baker, Fillmore, & Lowe, 1998) are to a substantial degree representations of commonsense knowledge. Likewise, there are *annotated corpora*, such as TimeML (Pustejovsky et al., 2003), PropBank (Kingsbury & Palmer, 2002), and AMR (Baranescu et al., 2013) in which texts are annotated with aspects of their semantics. Developing a schema for this kind of annotation is much the same enterprise as developing a logical representation for the domains of commonsense knowledge involved.

12.1.2 Planning

The issues in representing plans and goals for automated planning are obviously very closely related to the corresponding issues in automated reasoning, though there are significant differences, as we have argued in section 9.2.2 (Ghallab, Nau, & Traverso, 2004)

12.1.3 Robotics

The representations of manipulation and perception used in robotics tend to be much more low-level than those considered in commonsense reasoning and automated planning, including cognitive robotics. However, ultimately the subject matter is the same, and the two areas, with their very different emphases, will have to meet up in order to achieve robots that can act in a broadly intelligent way.

12.2 Related Work in Other Areas of Computer Science

12.2.1 Semantic Web

Representing the content of a web page or resource or a web-based transaction requires a representation of the domain concepts, and thus is closely related to the AI knowledge representation problem. In practice, KR and semantic web theories have interacted closely, and many of the leaders of the semantic web project have come out of the AI community (Berners-Lee, Hendler, & Lassila, 2001; Hendler & van Harmelen, 2008).

12.2.2 Databases

The development of useful database schemas for specific domains involves identifying the key features of the domain and often key constraints; this undertaking is quite similar to constructing a set of logical primitives for that domain. In particular, the problem of database merging — that is, combining two databases that have similar information, but with somewhat different systems of attributes — can require a rich representation of the content of the domain.

12.2.3 Automated Verification

The most successful application in computer science of automated logical inference has been to hardware and software verification. The logics used have generally been propositional logic, domain-specific extensions of propositional logic, first-order logic, and temporal modal logic.

Reasoning about software and commonsense reasoning have a number of common features. First and foremost is the temporal aspect; the state of a computation, like the state of the world, changes over time. A program is in many ways like a plan, and programs and plans share key operators such as sequencing and loops. Distributed systems can be viewed in terms of theories of knowledge and communication; each node of a system knows something and communicates what it knows to other nodes. Verifying software that is "about" some external domain, such as geometric software, requires combining reasoning about the software with reasoning about that domain. In all of these cases, the logical theories for verification and those for commonsense reasoning may have important elements in common.

12.3 Related Work in Other Disciplines

12.3.1 Linguistics

The area of semantics in linguistics involves the construction of a formal representation for the meaning of text. This is in principle very much the same as the KR problem; the major difference is that research in linguistics focuses on connecting the semantic representation to the linguistic form, whereas research in commonsense KR focuses on supporting commonsense reasoning.

The area of pragmatics in linguistics deals primarily with how the use of language in communication bears on its meaning; for example, the categorization of speech acts or the use of Gricean rules of inference. The commonsense theory of communication discussed in section 10.2 connects closely to this area of research.

Linguists also often include under the general category of "pragmatics" any use of world knowledge in interpreting text. More or less all of commonsense reasoning is relevant to pragmatics in this sense of the word. However, this area tends to be very much understudied in the linguistic literature, because it is hard to study in a controlled and circumscribed way.

12.3.2 Philosophy

The axiomatization of various fundamental domains has been a important thread in analytical philosophy for more than a century. Indeed many of the most important theories we have described above were originally developed by philosophers, in whole or in part, including logic, first-order logic, modal logic, temporal logic, epistemic logic, possible worlds semantics for modal logic generally, and the foundations of BDI theory. In particular, Carnap's (1958) development of logical theories of physics and of biology are quite similar in flavor to the domain theories we discuss here.[21]

12.3.3 Logical Foundations of Mathematics

One of the major accomplishments of nineteenth and twentieth century mathematics was the determination that essentially all rigorous proofs can be fully formalized as logical inference. (Whether this constitutes all of what is meant by "mathematics" or even all of what is meant by "proof" is a separate question.) The technology for constructing computer-verifiable proofs is now highly developed, though not very user-friendly, and very deep and complex theorems have been verified, including the prime number theorem (Avigad, Donnelly, Gray, & Raff, 2007; Harrison, 2009) and the Kepler conjecture (Hales et al., 2015). It was by no means obvious in, say, 1800 that this could be done even in principle, let alone in practice. Human-readable proofs involve appeals to intuition and understanding, and it was not obvious in advance that all these could be eliminated.

12.3.4 Axiomatization of Physics

The goal of formulating axiomatic foundations for physics in the sense that mathematical theories have been axiomatized was put forth by David Hilbert as the sixth of his famous collection of twenty-three unsolved mathematical problems (Corry, 2004). Mathematicians and physicists have mostly been unenthusiastic about this; still, there is a small literature on the subject. However, it generally focuses on formulation on the fundamental laws of physics, rather than on characterizing physical knowledge at the commonsense level.

13. Conclusion

The project of using logical languages to represent commonsense knowledge for AI systems is now almost sixty years old. Though the literature is large, the fraction of commonsense knowledge covered is presumably extremely small, and the impact of this approach on the practical AI technology is small and grows steadily smaller, as AI research becomes ever more dominated by corpus-based learning techniques that generate entirely opaque "representations".

The fact remains, however, that there are many forms of commonsense knowledge for which the *only* epistemically adequate representation known is some form of logical language, and many instances of commonsensical reasoning that can be reasonably characterized or approximated as logical inference, and cannot be carried out in any other known formalism. It therefore still seems likely that the study of the logical representation of com-

21. Thanks to Pat Hayes for bringing this to my attention.

monsense knowledge will ultimately play an important role in the development of human-level AI.

Acknowledgements

Many thanks to Hector Levesque for his feedback and suggestions. Thanks also to Andrew Gordon, Pat Hayes, Jerry Hobbs, Yves Lespérance, Vladimir Lifschitz, Gary Marcus, Leora Morgenstern, Sebastian Sardina, and Michael Wooldridge for helpful information and suggestions.

References

Aiello, M., Pratt-Hartmann, I., & van Benthem, J. (Eds.) (2007) *Handbook of Spatial Logics,* Springer-Verlag.

Alchourrón, C. E., Gärdenfors, P., & Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic, 50*(02), 510-530.

Allen, J. (1983). Maintaining knowledge about temporal intervals. *Comm. ACM, 23*, 123-154.

Alur, R., Henziger, T.A., & Kupferman, O. (2002). Alternating-time temporal logic. *Journal of the ACM, 49:*(5) 672-713.

Avigad, J., Donnelly, K., Gray, D., & Raff, P. (2007). A formally verified proof of the prime number theorem. *ACM Transactions on Computational Logic (TOCL), 9:*(1).

Baader, F., Horrocks, I, & Sattler, U. (2008). Description logics. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 135-180.

Bacchus, F. (1988). *Representing and Reasoning with Probabilistic Knowledge.* University of Alberta.

Bacchus, F., Grove, A., Halpern, J.Y., & Koller, D. (1992) From statistics to beliefs. *AAAI-92.*

Baker, C.F, Fillmore, C.J. & Lowe, J.B. (1998). The Berkeley FrameNet project. *ACL-1998.*

Baranescu, L. et al. (2013). Abstract Meaning Representation for Sembanking. *Proc. Linguistic Annotation Workshop.*

Bar-Hillel, Y. (1960). The present status of automatic translation of languages. In Alt, F. *Advances in Computers* Vol. 1. Academic Press, pp. 91-163.

Bennett, B. (2001) A categorical axiomatization of region-based geometry. *Fundamenta Informaticae, 46:* 145-158.

Berners-Lee, T., Hendler, J. & Lassila, O. (2001). The semantic web. *Scientific American, 284*(5) 34-43.

Bobrow, D. (ed.) (1985) *Qualitative Reasoning about Physical Systems.* MIT Press.

Brachman, R.J. (1979). On the epistemological status of semantic networks. In Findler, N.V. (Ed.) (1979) *Associative Networks: Representation and Use of Knowledge by Computers.* Academic Press, 3-50.

Brachman, R. J. (1985). I lied about the trees, or, defaults and definitions in knowledge representation. *AI magazine*, 6(3), 80.

Brachman, R.J. & Levesque, H. (1985). *Readings in Knowledge Representation.* Morgan Kaufmann.

Brachman, R.J. & Levesque, H. (2004). *Knowledge Representation and Reasoning.* Morgan Kaufmann.

Bratman, M.E. (1987). *Intentions, Plans, and Goals.* Harvard University Press.

Brewka, G., Niemelä, I., & Truszczyński, M. (2008). Nonmonotonic reasoning. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 239-284.

Brown, F. (Ed.) (1987). *The Frame Problem in Artificial Intelligence: Proc. of the 1987 Workshop.* Morgan Kaufmann.

Carnap, R. (1958). *Introduction to Symbolic Logic and its Applications.* Dover Pubs.

Charniak, E. (1972). *Toward a Model of Children's Story Comprehension.* Ph.D. thesis, MIT.

Charniak, E. (1993) *Statistical Language Learning.* MIT Press.

Chopra, A. et al. (2013). Research directions in agent communication. *ACM Trans. on Intelligent Systems and Technology, 4*(2).

Clementini, E., Di Felice, P., & Hernández, D. (1997). Qualitative representation of positional information. *Artificial Intelligence, 95:*(2) 317-356.

Cohen, P. & Levesque, H. (1990) Intention is choice with commitment. *Artificial Intelligence, 42* 213-261.

Cohn, A.G. (1987). A more expressive formulation of many sorted logic. *Journal of Automated Reasoning, 3:*(2), 113-200.

Cohn, A.G. & Renz, J. (2008). Qualitative spatial representation and reasoning. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 513-596

Corry, L. (2004). *David Hilbert and the Axiomatization of Physics (1898-1918): from* Grundlagen der Geometrie *to* Grundlagen der Physik. Kluwer.

Davis, E. (1990). *Representations of Commonsense Knowledge.* Morgan Kaufmann.

Davis, E. (1992a). Infinite loops in finite time: Some observations. *KR-92*

Davis, E. (1992b). Axiomatizing qualitative process theory. *KR-92*

Davis, E. (1993). The kinematics of cutting solid objects. *Annals of Mathematics and Artificial Intelligence, 9:*(3,4) 253-305.

Davis, E. (2005). Knowledge and communication: A first-order theory. *Artificial Intelligence, 166:* 81-140.

Davis, E. (2008a). Physical reasoning. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 597-620.

Davis, E. (2008b). Pouring liquids: A study in commonsense physical reasoning. *Artificial Intelligence, 172:* 1540-1578.

Davis, E. (2011). How does a box work? A study in the qualitative dynamics of solid objects. *Artificial Intelligence, 175*, 299-345.

Davis, E. (2012). Qualitative reasoning and spatio-temporal continuity. In Hazarika, S. (Ed.), *Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions.* IGI Global.

Davis, E. (2013a). Qualitative spatial reasoning in interpreting text and narrative. *Spatial Cognition and Computation, 13:*4, 264-294.

Davis, E. (2013b) The expressive power of first-order topological languages. *Journal of Logic and Computation, 23:*(5) 1107-1141

Davis, E. & Marcus, G. (2015). Commonsense reasoning and commonsense knowledge in artificial intelligence. *Comm. ACM, 58*(9) 92-105

Davis, E. & Marcus, G. (2016). The scope and limits of simulation in automated reasoning. *Artificial Intelligence, 233:* 60-72.

Davis, E., Marcus, G. & Frazier-Logue, N. (2017). Commonsense reasoning about containers using radically incomplete information. *Artificial Intelligence, 248* 46-84.

Davis, P. & Hersh, R. (1985). *The Mathematical Experience.* Birkhäuser.

de Giacomo, G. & Levesque, H. (1999). Projection using regression and sensors. *IJCAI-99.*

de Kleer, J. & Brown, J.S. A qualitative physics based on confluences. In Bobrow, D. (ed.) (1985) *Qualitative Reasoning about Physical Systems.* MIT Press.

De Moura, L., & Bjørner, N. (2011). Satisfiability modulo theories: introduction and applications. *Comm. ACM*, 54(9), 69-77.

Dechter, R. (2003). *Constraint Processing.* Morgan Kaufmann.

Doherty, P. & Kvanrnstöm, J. (2001). TALplanner: A temporal logic-based planner. *AI Magazine, 22*(3) 95-102.

Doherty, P. & Kvarnström, J. (2008). Temporal action logics. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 709-550.

Doherty, P., & Łukaszewicz, W. (1994). Circumscribing features and fluents. *Temporal Logic: Proceedings* 82-100.

Dubois, D., Lang, J. & Prade, H. (1994). Possibilistic logic. In Gabbay, D., Hogger, C.J., & Robinson, J.A. (Eds.) *Nonmonotonic Reasoning and Uncertain Reasoning. Handbook of Logic in Artificial Intelligence and Logic Programming,* vol. 3. Oxford University Press. 439-513.

Dyer, M. (1983). *In-depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension.* MIT Press.

Egenhofer, M. (1994). Topological similarity. *FISI Workshop on the Topological Foundations of Cognitive Science.*

Egenhofer, M. & Franzosa, R. (1991). Point-set topological spatial relations. *International Journal of Geographical Information Systems. 5:*(2) 161-174.

Etherington, D. W., & Reiter, R. (1983). On inheritance hierarchies with exceptions. *AAAI-83.*

Etzioni, O. et al. (2004) Web-scale extraction in KnowItAll (preliminary results). *WWW-04.*

Fagin, R., Halpern, J., Moses, Y. & Vardi, M. (1995). *Reasoning about Knowledge,* MIT Press.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence, 41*(1), 1-63.

Findler, N. V. (Ed.) (1979) *Associative Networks: Representation and Use of Knowledge by Computers.* Academic Press.

Fine, T. (1973). *Theories of Probability.* Academic Press.

FIPA (Foundations of Intelligent Physical Agents) (2002). FIPA Communicative Act Library Specifications. http://www.fipa.org/specs/fipa00037/index.html

Fisher, M. (2008). Temporal Representation and Reasoning. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 513-550.

Fleck, M. (1987). Representing space for practical reasoning. *IJCAI-87*, 728-730.

Forbus, K. (1985). Qualitative process theory. In Bobrow, D. (ed.) (1985) *Qualitative Reasoning about Physical Systems.* MIT Press.

Forbus, K. (2008). Qualitative modeling. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 361-394.

Forbus, K. (in preparation). *Qualitative Representations: How People Reason and Learn about the Continuous World.*

Funt, B. (1979). Problem solving with diagrammatic representations. *Artificial Intelligence, 13:*201-230

Gaifman, H. (1988). A theory of higher order probabilities. In *Causation, Chance and Credence: Proceedings* 191-219.

Gelfond, M. (2008). Answer sets. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 361-394.

Gelfond, M. & Lifschitz, V. (1998). Action languages. *Linköping Electronic Articles in Computer and Information Science, 3.*

Genesereth, M. & Nilsson, N. (1987) *Logical Foundations of Artificial Intelligence.* Morgan Kaufmann.

Georgeff, M.P. & Lansky, A. (1986) Procedural knowledge. In *Prof. IEEE Special Issue on Knowledge Representation, 74:* 1383-1398.

Gerevini, A. & Renz, J. (2002). Combining topological and size information for spatial reasoning. *Artificial Intelligence, 137:*(1-2) 1-42.

Ghallab, M., Nau, D. & Traverso, P. (2004). *Automated Planning: Theory and Practice.* Morgan Kaufmann.

Gomes, C. P., Kautz, H., Sabharwal, A., & Selman, B. (2008). Satisfiability solvers. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 89-134.

Goodman, N., Mansinghka, V.K., Roy, D.M., Bonawitz, K. & Tenenbaum, J.B. (2008). Church: A language for generative models. *Uncertainty in Artificial Intelligence.*

Gordon, A. & Hobbs, J. (in preparation). *How People Think People Think: A Formal Theory of Commonsense Psychology.* Cambridge University Press.

Green, C. (1969). Application of theorem proving to problem solving. *IJCAI-69.*

Grice, H.P. (1951). Meaning. *Philosophical Review, 66:* 377-388.

Grzegorczyk, A. (1951) Undecidability of some topological theories. *Fundamenta Mathematicae, 38:* 137-152.

Haas, A. (1986). A syntactic theory of belief and action. *Artificial Intelligence, 28:*3 245-292.

Hahmann, T. & Grüninger, M. (2011). A naïve theory for qualitative spatial relations. *Commonsense-2011.*

Hales, T. et al. (2015). A formal proof of the Kepler conjecture.
http://arxiv.org/abs/1501.02155

Halpern, J. (2003). *Reasoning about Uncertainty.* MIT Press.

Halpern, J. & Moses, Y. (1985). A guide to the modal logics of knowledge and belief. *IJCAI-85.*

Hanks, S., & McDermott, D. (1987). Nonmonotonic logic and temporal projection. *Artificial Intelligence, 33:*(3), 379-412.

Harrison, J. (2009). Formalizing an analytic proof of the prime number theorem. *Journal of Automated Reasoning, 43:*(3), 243-261.

Havasi, C., Pustejovsky, J., Speer, R., & Lieberman, H. (2009). Digital intuition: Applying common sense using dimensionality reduction. *IEEE Intelligent Systems, 24*(4), 24-35.

Hayes, P. (1977). In defense of logic. *IJCAI-77.*

Hayes, P. (1978). The naïve physics manifesto. In *Expert Systems in the Micro-electronic Age,* In Michie, D. (Ed.) Edinburgh University Press.

Hayes. P. (1985). Ontology for liquids. In Hobbs, J. & Moore, R. (Eds.) (1985). *Formal Theories of the Commonsense World.* ABLEX Publishing

Hazarika, S. (Ed.) (2012). *Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions.* IGI Global.

Hendler, J. & van Harmelen, F. (2008). The semantic web: Webizing knowledge representation. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 821-840.

Hendrix, G. G. (1979) Encoding knowledge in partitioned networks. In Findler, N. V. (Ed.) (1979) *Associative Networks: Representation and Use of Knowledge by Computers.* Academic Press. 51-92

Hintikka, J. (1962). *Knowledge and Belief.* Cornell University Press.

Hobbs, J. & Moore, R. (Eds.) (1985). *Formal Theories of the Commonsense World.* ABLEX Publishing

Hughes, G.E. & Cresswell, M.J. (1968). *An Introduction to Modal Logic.* Methuen & Co.

Kautz, H., McAllester, D. & Selman, B. (1996). Encoding plans in propositional logic. *KR-96*, 374-384.

Kingsbury, P., & Palmer, M. (2002). From TreeBank to PropBank. *LREC-02.*

Kogtenkov, A., Meyer, B. & Velder, S. (2015). Alias calculus, change calculus, and frame inference. *Science of Computer Programming, 97.* 163-172.

Konolige, K. (1985). Belief and incompleteness. In Hobbs, J. & Moore, R. (Eds.) (1985). *Formal Theories of the Commonsense World.* ABLEX Publishing

Konchakov, R., Pratt-Hartmann, I., Wolter, F. & Zakharyaschev, M. (2010). Spatial logics with connectedness predicates. *Logical Methods in Computer Science.*

Kowalski, R. (1979). Algorithm = logic+ control. *Comm. ACM* 22(7), 424-436.

Kowalski, R. & Sergot, M. (1986). A logic-based calculus of events. *New Generation Computing, 4*, 67.

Kripke, S. (1963). Semantical considerations on modal logic. *Acta Philosophia Fennica, Modal and Many-Valued Logics, 9:* 63-72.

Kuipers, B. (1986). Qualitative simulation. *Artificial Intelligence, 29:* 289-338.

Labrou, Y, & Finin, T. (1997). Semantics and conversations for an agent communication language. In Huhns, M. & Tambe, M. (eds.) *Readings in Agents,* Morgan Kaufmann.

Labrou, Y., Finin, T. & Peng, Y. (1999). Agent communication languages: The current landscape. *IEEE Intelligent systems 14*(2) 45-52.

Laumand, J. & Abe, N. (2015) *Dance Notations and Robot Motions.* Springer.

Lavalle, S.M. (2006). *Planning Algorithms.* Cambridge University Press.

Lebon, G., Jou, D., & Casas-Vázquez, J. (2008) *Understanding non-equilibrium thermodynamics: foundations, applications, frontiers.* Springer.

Lee, J., Lifschitz, V. & Yang, F. (2013) Action Language BC: Preliminary Report. *IJCAI-13*, 3-9.

Lenat, D., Prakash, M., & Shepherd, M. (1986). CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Magazine, 6*:4 65-85.

Levesque, H. (1996). What is planning in the presence of sensing? *AAAI-96*, 1139-1146.

Levesque, H. & Lakemeyer, G. (2008). Cognitive robotics. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 869-886.

Levesque, H. & Reiter, R. (1999). High-level robotic control: Beyond planning. A position paper. *AAAI Spring Symposium: Integrating Robotics: Taking the Next Big Leap.*

Liang, P. (2016). Learning executable semantic parsers for natural language understanding. *Comm. ACM, 59:*(9) 68-76.

Lifschitz, V. (Ed.) (1990) *Formalizing Common Sense: Papers by John McCarthy.* Ablex.

Lifschitz, V., Morgenstern, L. & Plaisted, D. (2008). Knowledge representation and classical logic. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 3-88

Ligozat, G. (1998). Reasoning about cardinal direction. *Journal of Visual Languages and Computing, 9:* 23-44.

Lin, F. (2008). Situation calculus. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 649-670.

Lin, F. & Reiter, R. (1994). State constraints revisited. *Journal of Logic and Computation, 4*(5) 655-678.

Marr, D. (1982). *Vision.* W.H. Freeman.

Marsella, S., Gratch, J., & Petta, P. (2010). Computational models of emotion. In Scherer, K. R., Bänziger, T., & Roesch, E. (Eds.). *A Blueprint for Affective Computing:* A sourcebook and manual. Oxford University Press.

Mates, B. (1972). *Elementary Logic.* Oxford University Press.

McCarthy, J. (1959). Programs with common sense. In *Proc. Symposium on Mechanisation of Thought Processes* 1.

McCarthy, J. (1963). Situations, actions, and causal laws. Stanford AI Project Memo #2.

McCarthy, J. (1968). Programs with Common Sense. In Minsky, M. (ed.) *Semantic Information Processing.* MIT Press.

McCarthy, J. (1977). Epistemological problems of artificial intelligence. *IJCAI-77.*

McCarthy, J. (1980). Circumscription — A form of nonmonotonic logic. *Artificial Intelligence. 13:*27-39.

McCarthy, J. & Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., & Michie, D. (Eds.). *Machine Intelligence 4.* Edinburgh University Press.

McDermott, D. (1978). Tarskian semantics, or No notation without denotation! *Cognitive Science, 2*:3, 277-282.

McDermott, D. (1982). A temporal logic for reasoning about processes and plans. *Cognitive Science, 6*: 101-155.

McDermott, D. (1987). A critique of pure reason. *Computational Intelligence, 3*: 151-160.

McIlraith, S. A. (2000). Integrating actions and state constraints: A closed-form solution to the ramification problem (sometimes). *Artificial Intelligence, 116:*(1), 87-121.

Mendelson, E. (1979). *Introduction to Mathematical Logic.* Van Nostrand.

Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D. L., & Kolobov, A. (2007). BLOG: Probabilistic models with unknown objects. In Getoor, L. (ed.) *Introduction to Statistical Relational Learning*, 373-398.

Miller, G. (1995). WordNet:A lexical database for English. *Comm. ACM*, *38*(11), 39-41.

Miller, R. & Shanahan, M. (1999) The event calculus in classical logic — alternative axiomatisations. *Linköping Electronic Articles in Computer and Information Science, 4*(16).

Minsky, M. (1975). A framework for representing knowledge. In P. H. Winston (Ed.) *The Psychology of Computer Vision,* McGraw-Hill, pp. 157-209

Mitchell, T. et al. (2015). Never Ending Learning. *AAAI-15.*

Moore, R.C. (1982). The role of logic in knowledge representation and commonsense reasoning. *AAAI-82,* 428-433.

Moore, R.C. (1985a). A formal theory of knowledge. In Hobbs, J. & Moore, R. (Eds.). *Formal Theories of the Commonsense World.* ABLEX Publishing

Moore, R.C. (1985b). Semantical considerations on nonmonotonic logic. *Artificial Intelligence, 25:*(1), 75-94.

Morgenstern, L. (1988). *Foundations of a Logic of Knowledge, Action, and Communication.* Ph.D. Thesis, New York University.

Morgenstern, L. (1991). Knowledge and the frame problem. *International Journal of Expert Systems, 3:*(4), 309-343.

Mueller, E.T. (2006). *Commmonsense Reasoning.* Morgan Kaufmann.

Mueller, E.T. (2008). Event calculus. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 671-708.

Muller, P. (1998). A qualitative theory of motion based on spatio-temporal primitives. *KR-98.*

Murphy, G. (2002). *The Big Book of Concepts.* MIT Press.

Murphy, K. (2012) *Machine Learning: A Probabilistic Perspective.* MIT Press.

Nau, D., Muñoz-Avila, H., Cao, Y., Lotem, A. & Mitchell, S. (2001). Total-Order Planning with Partially-Ordered Subtasks. *IJCAI-01.*

Nayak, P. (1994). Causal approximations. *Artificial Intelligence, 70:*277-334.

Nebel, B. & Bürckert, H.J. (1995). Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. *Journal of the ACM, 42:* 43-66.

Newell, A. (1981) The knowledge level. *AI Magazine, 2*:2, 1-20.

Nilsson, N. J. (1986). Probabilistic logic. *Artificial intelligence, 28:*(1), 71-87.

Ortony, A., Clore, G. L., & Collins, A. (1990). *The Cognitive Structure of Emotions.* Cambridge University Press.

Palmer M., Kingsbury P. & Gildea D (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, *31*:1 71106.

Parikh, R. (1994). Logical omniscience. *Logic and Computational Complexity.* 22-29.

Parikh, R. (2005). Logical omniscience and common knowledge: WHAT do we know and what do WE know? *Theoretical Aspects of Reasoning about Knowledge* 62-77.

Pauly. M. (2002). A modal logic for coalitional power in games. *Journal of Logic and Computation, 12:*(1) 305-324.

Pauly, M. & Parikh, R. (2003). Game logic: An overview. *Studia Logica, 75:*(2) 165-182.

Pearl, J. (1987). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann.

Peppas, P. (2008). Belief revision. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 317-360.

Pinto, J. (1997). Integrating discrete and continuous change in a logical framework. *Computational Intelligence, 14:*(1).

Poslad, S. (2007). Specifying Protocols for Multi-Agent System Interactions. *ACM Transactions on Autonomous and Adaptive Systems, 2*:4, 15.

Prasad, M. R., Biere, A., & Gupta, A. (2005). A survey of recent advances in SAT-based formal verification. *International Journal on Software Tools for Technology Transfer, 7*(2), 156-173.

Pratt, I. (1999). Qualitative spatial representation languages with convexity. *Spatial Cognition and Computation, 1:* 181-204.

Pratt-Hartmann, I. (2007). First-order mereotopology. In Aiello, M., Pratt-Hartmann, I., & van Benthem, J. (Eds.) *Handbook of Spatial Logics,* Springer-Verlag.

Pustejovsky, J., et al. (2003). TimeML: Robust specification of event and temporal expressions in text. *New Directions in Question Answering, 3*, 28-34.

Pylyshyn, Z. (Ed.) (1987). *The Robot's Dilemma: The Frame Problem in Artificial Intelligence.* Ablex.

Quillian, R. (1968). Semantic memory. In Minsky, M. *Semantic Information Processing.* MIT Press. 227-270

Raiman, O. (1990). Order of magnitude reasoning. In Weld, D. & de Kleer, J. (Eds.) *Readings on Qualitative Reasoning about Physical Systems,* Morgan Kaufmann.

Randell, D.A. & Cohn, A.G (1989). Modelling topological and metrical properties of physical processes. *KR-89*

Randell, D.A., Cui, Z. & Cohn, A.G (1992). A spatial logic based on regions and connection. *KR-92.*

Rao, A. S., & Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. *KR-91*, 473-484.

Reece, J.B. et al. (2010). *Campbell Biology*, 9th edition. Benjamin Cummings.

Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence. 13:* 81-132.

Reiter, R. (2001). *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems.* MIT Press.

Renz, J. & Mitra, D. (2004). Qualitative direction calculi with arbitrary granularity. *Proc. 8th Pacific Rim Intl. Conf. on Artificial Intelligence,* 65-74.

Richens, R.H. (1956). Preprogramming for mechanical translation, *Mechanical Translation 3*:1, 20-25.

Rossi, F., van Beek, P. & Walsh, T. (2008). Constraint Programming. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 181-212.

Sachan, M. & Xing, E. (2016). Machine comprehension using rich semantic representations. *ACL-16.*

Sandewall, E. (1994). *Features and Fluents: A Systematic Approach to the Representation of Knowledge about Dynamical Systems.* Oxford University Press.

Schank, R. & Abelson, R. (1977). *Scripts, Plans, Goals, and Understanding.* Lawrence Erlbaum.

Scherl, R.B & Levesque, H. (2003). Knowledge, action, and the frame problem. *Artificial Intelligence, 144.*

Schubert, L.K. (1990). Monotonic solution of the frame problem in the situation calculus. In Kyburg, H., Loui, R., & Carlson, G., *Knowledge Representation and Defeasible Reasoning.* Kluwer.

Scotti, R.A. (2009). The story behind the theft of the Mona Lisa. *Times of London,* Sunday, April 12, 2009

Shafer, G. & Pearl, J. (1990). *Readings in Uncertain Reasoning.* Morgan Kaufmann.

Shanahan, M. (1997). *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia.* MIT Press.

Son, T. & Baral, C. (2001). Formalizing sensing actions — A transition function based approach. *Artificial Intelligence, 125:*(1-2) 19-91.

Sowa, John F. (Ed.) (1991). *Principles of Semantic Networks: Explorations in the Representation of Knowledge.* Morgan Kaufmann.

Sowa, J. (1992). Semantic networks. In Shapiro, S. (Ed.) *Encyclopedia of Artificial Intelligence,* 2nd edition, Wiley.

Sowa, J. (undated). Semantic networks. http://www.jfsowa.com/pubs/semnet.pdf.

Sowa, J. (2008). Conceptual graphs. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 181-212.

Steedman, M. (2012). Computational linguistic approaches to temporality. In Binnick, R. (Ed.) *Handbook of Tense and Aspect,* Oxford University Press, pp. 102-120.

Steunebrink, B.R., Dastani, M. & Meyer J.J. Ch (2007). A logic of emotion for intelligent agents. *AAAI-07*

Stein, L.A. & Morgenstern, L. (1994). Motivated action theory: A formal theory of causal reasoning. *Artificial Intelligence, 71:*(1) 1-42.

Sussman, G. (1975). *A Computational Model of Skill Acquisition.* American Elsevier.

Thielscher, M. (1997). Ramification and causality. *Artificial Intelligence, 89:*(1), 317-364.

Thielscher, M. (2000). Modeling actions with ramifications in nondeterministic, concurrent, and continuous domains — and a case study. *AAAI-00*, 497-502.

Thielscher, M. (2005). *Reasoning Robots: The Art and Science of Programming Robotic Agents.* Springer.

Thomason, R. (2003). Logic and artificial intelligence. *Stanford Encyclopedia of Philosophy.* Stanford University.

Touretzky, D. S. (1984). Implicit Ordering of Defaults in Inheritance Systems. *AAAI-84.*

Turner, R. (1984) *Logics for Artificial Intelligence.* Halsted Press.

van Benthem, J. (1983) *The Logic of Time: A Model-Theoretic Investigation.* Springer.

van Ditmarsch, H., van der Hoek, W. & Kooi, B. (2007). *Dynamic Epistemic Logic.* Springer.

van Ditmarsch, H., van der Hoek, W. & Kooi, B. (undated). Dynamic epistemic logic. *Internet Encyclopedia of Philosophy.* http://www.iep.utm.edu/de-logic/

van der Hoek, W. & Wooldridge, M. (2008). Multi-agent systems. In van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) *Handbook of Knowledge Representation.* Elsevier. 887-928.

van Harmelen, F., Lifschitz, V. & Porter, B. (Eds.) (2008). *Handbook of Knowledge Representation.* Elsevier.

van Linder, B., van der Hoek, W., and Meyer J.J.-Ch. (1998). Formalizing abilities and opportunities of agents. *Fundamentae Informaticae, 34*(1-2) 103-146.

Vardi, M. (1996). Why is modal logic so robustly decidable? *Descriptive logic and finite models, 31* 149-184.

Waltz, D, (1975). Understanding line drawings of scenes with shadows. In P. H. Winston (Ed.) *The Psychology of Computer Vision,* McGraw-Hill, pp. 157-209

Weld, D. (1990). Exaggeration. In Weld, D. & de Kleer, J. (Eds.) *Readings on Qualitative Reasoning about Physical Systems,* Morgan Kaufmann.

Weld, D. (1992). Approximation reformulations. *Artificial Intelligence, 56:* 255-300.

Weld, D. & de Kleer, J. (Eds.) *Readings on Qualitative Reasoning about Physical Systems,* Morgan Kaufmann.

Winston, P. H. (1975). Learning structural descriptions from examples, In P. H. Winston (Ed.) *The Psychology of Computer Vision,* McGraw-Hill, pp. 157-209

Winograd, T. (1972). *Understanding Natural Language.* Academic Press,

Wolfman, S. & Weld, D. (1999). The LPSAT engine and its application to resource planning. *IJCAI-99.*

Woods, W.A. (1975). What's in a link: Foundations for semantic networks. In Bobrow, D.G. & Collins, A.M. (Eds.) *Representation and Understanding: Studies in Cognitive Science.* Academic Press.

Wooldridge, M. (2009). *An Introduction to Multiagent Systems.* Wiley.

Wos, L., Overbeek, R., Lusk, E. & Boyle, J. (1984). *Automated Reasoning: Introduction and Applications.* Prentice Hall.

Wu, W., Li, H., Wang, H. & Zhu, K. (2012) Probase: A probabilistic taxonomy for text understanding. *SIGMOD-12.*

Zadeh, L. (1987). Fuzzy algorithms. *Information and Control, 12:* 94-102.