CHAPTER

Introduction

1

This book is about commonsense reasoning, the sort of reasoning people perform in daily life. Here are some examples of commonsense reasoning:

1. In the living room, Lisa picked up a newspaper and walked into the kitchen. Where did the newspaper end up? It ended up in the kitchen.
2. Kate set a book on a coffee table and left the living room. When she returned, the book was gone. What happened to the book? Someone must have taken it.
3. Jamie walks to the kitchen sink, puts the stopper in the drain, turns on the faucet, and leaves the kitchen. What will happen as a result? The water level will increase until it reaches the rim of the sink. Then the water will start spilling onto the floor.
4. Kimberly turns on a fan. What will happen? The fan will start turning. What if the fan is not plugged in? Then the fan will not start turning.
5. A hungry cat saw some food on a nearby table. The cat jumped onto a chair near the table. What was the cat about to do? The cat was about to jump from the chair onto the table in order to eat the food.

This book is concerned with understanding and describing commonsense reasoning to such a level of detail that it can be *automated*, or performed automatically by a machine such as a computer. It reviews methods for commonsense reasoning and describes in detail a method for commonsense reasoning using the event calculus, which is based on classical predicate logic.

1.1 WHAT IS COMMONSENSE REASONING?

Commonsense reasoning is a process that involves taking information about certain aspects of a scenario in the world and making inferences about other aspects of the scenario based on our *commonsense knowledge*, or knowledge of how the world works. Commonsense reasoning is essential to intelligent behavior and thought. It allows us to fill in the blanks, to reconstruct missing portions of a scenario, to figure out what happened, and to predict what might happen next. Commonsense reasoning stands in contrast to various types of expert reasoning such as economic, legal, mathematical, medical, and scientific reasoning.

1.2 KEY ISSUES OF COMMONSENSE REASONING

Although commonsense reasoning comes naturally to us and appears to be simple, it is actually a complex process. In this section, we examine the previously mentioned examples of commonsense reasoning in detail. We introduce fundamental concepts and point out some of the key issues that must be addressed by any method for commonsense reasoning.

Consider the first scenario.

Representation

In the living room, Lisa picked up a newspaper …

In order to automate commonsense reasoning about a scenario such as this, we must first build a representation of the scenario. A *representation* is something that resembles something else. For the purpose of automating commonsense reasoning, the representation should be a data structure or a sentence of a language defined by a formal syntax, and the representation should facilitate automated reasoning.

Objects, properties, events, and time

Several fundamental entities must be represented. First, we must represent objects in the world and agents such as persons and animals; we must represent Lisa, the newspaper, and the living room. Second, we must represent properties of the world that change over time; we need to represent the time-varying locations of Lisa and the newspaper. Third, we must represent events or actions that occur in the world; we need to represent the event of Lisa picking up the newspaper. Fourth, we must represent time; we must represent that Lisa picked up the newspaper when she and the newspaper were in the living room.

Object identity

We must represent the identities of objects; we must represent the fact that Lisa and the newspaper are not the same object.

Reasoning

Having formed a representation of the scenario, we can then perform commonsense *reasoning* or *inference*. Because our goal is automation, the method of reasoning should be expressed as an algorithm or formal rule that takes representations as input and produces representations as output.

Representations of commonsense knowledge

We must construct representations of commonsense knowledge that can be used by the reasoning method to reason about this scenario as well as other scenarios.

Effects of events

We must be able to represent and reason about the effects of events on world properties. We must be able to reason from a specific event and general knowledge about the effects of events to the specific effects of the specific event. We should be able to represent that, if a person picks up an object, then the person will be holding that object. Given that Lisa picked up the newspaper, and this piece of commonsense knowledge, we should be able to infer that Lisa was then holding the newspaper.

Context-sensitive effects

We must be able to represent and reason about the context-sensitive effects of events. We should be able to represent that, if a person picks up a slippery object and is not careful, then the person will not be holding the object.

Nondeterministic effects

We must also be able to represent and reason about events with nondeterministic effects. We should be able to represent that, if a person picks up a slippery object, then the person may or may not be holding the object.

Concurrent events

We must be able to represent and reason about concurrent events. We should be able to represent that certain concurrent events are impossible; for example, a person cannot walk into two rooms simultaneously. We must be able to reason about concurrent events with cumulative or canceling effects. For example, if a shopping cart is pushed, it moves forward. If it is pulled, it moves backward. But, if it is simultaneously pulled and pushed, then it moves neither forward nor backward; instead, it spins around.

Space

…and walked into the kitchen.

In order to automate commonsense reasoning, we must be able to deal with space. We must represent the knowledge that, after a person walks into a room, the person will be in that room. From this knowledge and the fact that Lisa walked into the kitchen, we should be able to infer that afterward Lisa was in the kitchen.

Indirect effects

Where did the newspaper end up? It ended up in the kitchen.

In order to make this inference, we must be able to reason about the indirect effects or ramifications of events. We know that, if a person is holding an object, then the object moves along with the person.

Next, we consider the second scenario.

Kate set a book on a coffee table …

We must represent the effect of setting an object on another object, and we should be able to reason that, after Kate set the book on the coffee table, the book was on the coffee table.

Preconditions

We should also be able to infer that before Kate set the book on the table, she was holding the book and she was near the table; we must be able to represent and reason about the preconditions of actions or events. We need to represent two preconditions of a person placing an object onto a surface: (1) the person must be holding the object, and (2) the person must be near the surface.

…and left the living room.

We should be able to infer that, after Kate left the living room, she was no longer in the living room.

Commonsense law of inertia

We should also be able to infer that the book was still on the table in the living room after she left.

When she returned, …

We should be able to infer that, after Kate returned, the book was probably still on the table in the living room. That is, unless a person or animal moved the book, or some natural phenomenon such as an earthquake occurred, the book was where Kate left it. This property of the commonsense world, that things tend to stay the same unless affected by some event, is known as the *commonsense law of inertia*.

But we learn that the book was no longer in the living room:

…the book was gone.

In this case we should be able to infer that someone took the book out of the room (or a natural phenomenon occurred):

What happened to the book? Someone must have taken it.

Next, we consider the third scenario.

Delayed effects and continuous change

Jamie walks to the kitchen sink, puts the stopper in the drain, turns on the faucet, and leaves the kitchen.

We have so far seen that it is necessary for us to be able to represent and reason about the immediate effects of events, such as putting a stopper in a drain and turning on a faucet. Thus, we should be able to infer that the stopper is in the drain, the faucet is running, and the sink is filling. In addition, we should be able to represent and reason about the delayed effects of events:

What will happen as a result? The water level will increase until it reaches the rim of the sink. Then the water will start spilling onto the floor.

Making these inferences involves representing and reasoning about continuous change. We should be able to represent that, if a faucet is turned on with the stopper in place, then the water level will increase with time.

Release from the commonsense law of inertia

Recall that the commonsense law of inertia states that things stay the same unless affected by some event. But notice that the water level continues to change after the event of turning on the faucet. Therefore we must be able to represent that, after the faucet is turned on, the water level is released from the commonsense law of inertia and is permitted to vary. We must further represent that the water level is proportional to the time elapsed since the faucet was turned on.

Triggered events

In order to reason about this scenario, we must also be able to represent and reason about triggered events. The water level does not increase endlessly. When a sink is filling and the water reaches the rim of the sink, the sink will overflow. We should be able to represent and reason that, when a sink overflows, the water starts spilling onto the floor and the water level stops increasing. At this point, the water level will again be subject to the commonsense law of inertia.

Consider the fourth scenario.

Default reasoning

When we perform commonsense reasoning, we rarely have complete information. We are unlikely to know the state of affairs down to the last detail, everything about the events that are occurring, or everything about the way the world works. Therefore, when we perform commonsense reasoning, we must jump to conclusions. Yet, if new information becomes available that invalidates those conclusions, then we must also be able to take them back. Reasoning in which we reach conclusions and retract those conclusions when warranted is known as *default reasoning*. In the fourth scenario,

> *Kimberly turns on a fan. What will happen? The fan will start turning.*

How can the method for commonsense reasoning conclude that the fan will start turning? In fact, the fan might not start turning if the fan is broken, if the fan is not plugged in, and so on. The world is filled with exceptions such as these. The method must be able to assume that things are as normal and conclude that the fan will start turning. If it is later learned that the fan is not plugged in, then the conclusion should be revised:

> *What if the fan is not plugged in? Then the fan will not start turning.*

Two special cases of default reasoning are required for reasoning about events. First, although we are told that a fan is turned on, we do not know what other events occur. We do not know whether, for example, some other person is simultaneously attempting to turn off the fan. The method for commonsense reasoning must assume that this is not the case; that is, it must be assumed by default that unexpected events do not occur.

Second, although we know that a fan will start turning after it is turned on, we do not know what the other results of turning on the fan might be. Perhaps turning on the fan also unlocks a nearby door. The method for commonsense reasoning must assume by default that events do not have unexpected effects.

Next, we consider the fifth scenario.

Mental states

A hungry cat saw some food on a nearby table.

We must represent the piece of commonsense knowledge that, if an agent has an unsatisfied goal, then the agent will form a plan to achieve that goal. In this case, if an animal has the goal to eat and has the belief that food is nearby, then the animal will form the plan to go to the food and eat it.

The cat jumped onto a chair near the table.

We must further represent that agents act on their plans. We should be able to infer that jumping onto a chair is part of the cat's plan to eat.

What was the cat about to do? The cat was about to jump from the chair onto the table in order to eat the food.

Based on the knowledge that agents act on their plans, we should be able to infer that the cat will complete the plan. After the cat eats the food, we should infer that the plan is completed. We may also then infer that the goal to eat is satisfied.

Reasoning types

A method for automated commonsense reasoning must support several types of commonsense reasoning. The first is *temporal projection* or *prediction*, in which we start with an initial state and some events and then reason about the state that results from the events. The examples of Lisa walking into the kitchen, the kitchen sink overflowing, and the cat eating the food all involve temporal projection. The second type of reasoning is *abduction*, in which we start with an initial state and a final state and then reason about the events that lead from the initial state to the final state. The example of Kate's book disappearing involves abduction. The third type of reasoning is *postdiction*, in which we start with some events that lead to a state and then reason about the state prior to the events. If we are told that Lisa picked up a newspaper and was then holding the newspaper, we may reason that Lisa was not previously holding the newspaper.

1.2.1 SUMMARY

Any method for automated commonsense reasoning must address the following.

Representation. The method must represent scenarios in the world and must represent commonsense knowledge about the world.

Commonsense entities. The method must represent objects, agents, time-varying properties, events, and time.

Commonsense domains. The method must represent and reason about time, space, and mental states. The method must deal with object identity.
Commonsense phenomena. The method must address the commonsense law of inertia, release from the commonsense law of inertia, concurrent events with cumulative and canceling effects, context-sensitive effects, continuous change, delayed effects, indirect effects, nondeterministic effects, preconditions, and triggered events.
Reasoning. The method must specify processes for reasoning using representations of scenarios and representations of commonsense knowledge. The method must support default reasoning, temporal projection, abduction, and postdiction.

1.3 BRIEF HISTORY OF COMMONSENSE REASONING

Artificial intelligence researchers have been trying to invent ways of automating commonsense reasoning since the inception of the field in 1956. Work on commonsense reasoning can be divided into two categories: logical and nonlogical. Logical methods are reviewed in detail in Chapter 16, and nonlogical methods are reviewed in Chapters 17–19. In this section, we present a brief history of work on logical and nonlogical methods.

1.3.1 LOGICAL METHODS

In 1958, John McCarthy proposed to use logic to give computer programs common sense. In the 1960s, he and Patrick J. Hayes introduced the situation calculus, a logical formalism for commonsense reasoning. In the 1970s, the crucial role of defaults in commonsense reasoning was recognized, and researchers began to formalize methods for default reasoning. Important formalisms for default reasoning such as circumscription and default logic appeared around 1980.

Taking their inspiration from the situation calculus, Robert Kowalski and Marek Sergot introduced the event calculus in 1986. In the late 1980s, several other logical formalisms began to appear, including the features and fluents framework, action languages, and the fluent calculus.

Since the early 1990s, logic-based commonsense reasoning has been the focus of intense activity. Researchers proposed a number of benchmark problems designed to expose issues of commonsense reasoning not yet addressed by the available formalisms. This led to a considerable evolution of the formalisms.

In this book, we use a version of the event calculus developed in the 1990s by Murray Shanahan and Rob Miller, which we call EC, and a version that is equivalent for integer time, called the discrete event calculus (DEC). The event calculus has benefited enormously from the investigation of benchmark problems. Table 1.1 shows some of the benchmark problems that led to the addition of features to the event calculus.

Table 1.1 Benchmark Problems Leading to the Addition of Features to the Event Calculus

Benchmark Problem	Commonsense Phenomena	Event Calculus Features Added
Bus ride scenario (Kartha, 1994)	Nondeterministic effects	Disjunctive event axioms (Shanahan, 1997b)
Chessboard scenario due to Raymond Reiter (Kartha & Lifschitz, 1994)	Nondeterministic effects	Determining fluents (Shanahan, 1997b)
Commuter scenario (Shanahan, 1999a)	Compound events	Three-argument *Happens*
Kitchen sink scenario (Shanahan, 1990)	Continuous change, triggered events	Trajectory axioms, trigger axioms
Russian turkey scenario (Sandewall, 1994)	Nondeterministic effects	Release axioms (Shanahan, 1997b)
Shopping outlet scenario (R. Miller, Morgenstern, & Patkos, 2013)	Knowledge	Possible worlds
Soup bowl scenario (Gelfond, Lifschitz, & Rabinov, 1991)	Concurrent events	Cumulative effect axioms (R. Miller & Shanahan, 1999)
Stolen car scenario (Kautz, 1986)	Explanation	Abduction (Shanahan, 1997b)
Stuffy room scenario (Ginsberg & Smith, 1988a)	Indirect effects	State constraints, primitive and derived fluents (Shanahan, 1997b)
Thielscher's circuit (Thielscher, 1996)	Indirect effects	Causal constraints (Shanahan, 1999b)
Walking turkey scenario due to Matthew L. Ginsberg (Baker, 1991)	Indirect effects	Effect constraints (Shanahan, 1997b)
Yale shooting scenario (Hanks & McDermott, 1987)	Commonsense law of inertia	Circumscription, forced separation (Shanahan, 1997b)

1.3.2 NONLOGICAL METHODS

Starting in the early 1970s, Roger Schank, Robert Abelson, and their colleagues and students developed a number of knowledge structures and inference methods for use in natural language understanding systems. A notation known as conceptual dependency (CD) was proposed for representing actions and states. Knowledge structures called scripts were introduced to represent stereotypical activities such as attending a birthday party. A taxonomy of human plans and goals was developed, and representations for the themes of stories were introduced.

Starting in the late 1970s, researchers working in the area of qualitative reasoning developed techniques for automated reasoning about physical mechanisms such as bathtubs, clocks, electrical circuits, pressure regulators, sliding blocks, and water tanks. Physical devices are described using a modeling language, and simulation algorithms are used to perform reasoning. These techniques are useful for commonsense reasoning in the physical domain. Some of the techniques have been recast in first-order logic.

Beginning in the early 1980s, researchers developed methods for analogical processing. These methods can be used to find an analogy between a familiar domain and a novel domain and then to use the analogy to generate candidate inferences about the novel domain. Analogical processing is not a complete method for commonsense reasoning, because candidate inferences must still be evaluated and repaired using other commonsense reasoning techniques.

Probability theory has a long history and is well suited to default reasoning. Since the late 1980s, some work has been performed on using probabilistic reasoning for reasoning about action and change, but this approach is not as well developed as the logical approach. The logical and probabilistic approaches are closely related, and the integration of logic and probability theory is an active area of research.

Starting in the early 1970s, the society of mind theory was developed by Marvin Minsky and his colleagues and students. This theory views human common sense as a vast collection of skills involving multiple representation and reasoning techniques. Unlike most other approaches, this approach places a great emphasis on procedural representations of knowledge and on the ways that procedures can monitor and influence one another.

In the 2000s, after Push Singh's introduction of the Open Mind Common Sense knowledge base and Marti A. Hearst's publication of influential papers on mining WordNet relations from text, there was much activity on crowdsourcing commonsense knowledge and mining commonsense knowledge. In 2011, IBM's Watson system won a Jeopardy! television game show match against two Jeopardy! grand champions, which demonstrated the power of using unstructured information and machine learning for solving difficult artificial intelligence problems.

1.4 THE EVENT CALCULUS

The event calculus addresses all the key issues of commonsense reasoning described in Section 1.2. Using the event calculus we can represent commonsense knowledge, represent scenarios, and use the knowledge to reason about the scenarios.

1.4.1 EVENTS, FLUENTS, AND TIMEPOINTS

The basic notions of the event calculus are as follows. An *event* represents an event or action that may occur in the world, such as a person picking up a glass. We use the words *event* and *action* interchangeably. A *fluent* represents a time-varying property of the world, such as the location of a physical object. A *timepoint* represents an instant of time, such as 9:30 AM Greenwich Mean Time on November 13, 2017. The event calculus uses *linear time*, in which time is considered to be a line, rather than the *branching time* of the situation calculus, in which time is considered to be a tree.

An event may occur or happen at a timepoint. A fluent has a truth value at a timepoint or over a timepoint interval; the possible truth values are true and false. After an event occurs, the truth values of the fluents may change. We have commonsense knowledge about the effects of events on fluents. Specifically, we have knowledge about events that initiate fluents and events that terminate fluents. For example, we know that the event of picking up a glass initiates the fluent of holding the glass and that the event of setting down a glass terminates the fluent of holding the glass. We represent these notions in first-order logic as follows.

$HoldsAt(f, t)$ represents that fluent f is true at timepoint t.
$Happens(e, t)$ represents that event e occurs at timepoint t.
$Initiates(e, f, t)$ represents that, if event e occurs at timepoint t, then fluent f will be true after t.
$Terminates(e, f, t)$ represents that, if event e occurs at timepoint t, then fluent f will be false after t.

1.4.2 A SIMPLE EXAMPLE

Here is a simple example of how the event calculus works. We use a simplified version of the event calculus, consisting of the following single axiom:

$$\begin{gathered}(Happens(e, t_1) \land Initiates(e, f, t_1) \land t_1 < t_2 \land \\ \neg\exists e, t\,(Happens(e, t) \land t_1 < t \land t < t_2 \land Terminates(e, f, t))) \Rightarrow \\ HoldsAt(f, t_2)\end{gathered} \tag{1.1}$$

This axiom states that, if an event occurs and the event is known to initiate a particular fluent, then that fluent will be true from just after the moment the event occurs, until and including the moment an event occurs that is known to terminate the fluent.

Now let us see how this axiom can be used to solve a simple commonsense reasoning problem, using one event and one fluent. The event $WakeUp(p)$ represents that person p wakes up, and the fluent $Awake(p)$ represents that person p is awake. Our commonsense knowledge consists of the following.

If a person wakes up, then the person will be awake:

$$Initiates(e, f, t) \Leftrightarrow \exists p\,(e = WakeUp(p) \land f = Awake(p)) \tag{1.2}$$

If a person falls asleep, then the person will no longer be awake:

$$Terminates(e, f, t) \Leftrightarrow \exists p\,(e = FallAsleep(p) \land f = Awake(p)) \tag{1.3}$$

Now suppose we have the following scenario. Nathan is not awake at timepoint 0:

$$\neg HoldsAt(Awake(Nathan), 0) \tag{1.4}$$

The only known event is that Nathan wakes up at timepoint 1:

$$Happens(e,t) \Leftrightarrow e = WakeUp(Nathan) \wedge t = 1 \tag{1.5}$$

From this commonsense knowledge and scenario, and event calculus axiom (1.1), we can prove that Nathan is awake at timepoint 3:

$$(1.1) \wedge (1.2) \wedge (1.3) \wedge (1.4) \wedge (1.5) \vdash HoldsAt(Awake(Nathan), 3)$$

The proof runs as follows. From (1.3) and (1.5), we have $\neg\exists e, t\ (Happens(e,t) \wedge 1 < t \wedge t < 3 \wedge Terminates(e, Awake(Nathan), t))$. From this, (1.1), (1.2), (1.5), and $1 < 3$, we have $HoldsAt(Awake(Nathan), 3)$, as required.

Adding commonsense knowledge and event occurrences to this formalization requires us to modify (1.2), (1.3), and (1.5). Later, in Sections 2.6 and 2.7, we show how circumscription allows us to add commonsense knowledge and event occurrences simply by adding axioms.

1.4.3 AUTOMATED EVENT CALCULUS REASONING

In the logic-based approach to commonsense reasoning, knowledge is represented *declaratively* as logical formulas rather than *procedurally* as computer code. Using a declarative knowledge representation has two main advantages. First, the same knowledge can be used for different types of commonsense reasoning such as temporal projection, abduction, and postdiction. If a procedural knowledge representation is used, knowledge must often be duplicated for each type of commonsense reasoning. Second, using a declarative knowledge representation allows us to use the latest, off-the-shelf, automated theorem-proving techniques to solve reasoning problems. If a procedural knowledge representation is used, reasoning techniques must often be built from scratch or reinvented.

Of course, entailment in first-order logic is *undecidable*: There is no algorithm that, given arbitrary formulas of first-order logic ψ and π, will eventually respond "yes" if ψ entails π and "no" if ψ does not entail π. First-order logic entailment is only *semidecidable*: There are algorithms that, given arbitrary formulas of first-order logic ψ and π, will respond "yes" if ψ entails π; if ψ does not entail π, the algorithms may eventually respond "no" or may never terminate. It turns out, however, that many real-world reasoning problems in the event calculus can be solved efficiently by computer programs.

Automated reasoning in the event calculus can be performed using several programs, as listed in Table 1.2. The Discrete Event Calculus Reasoner is discussed in Chapter 13. The F2LP program is discussed in Chapter 15. These programs rely on various solvers and provers, namely, satisfiability (SAT) solvers, logic programming languages, answer set grounders and solvers, and first-order automated theorem provers. Improving the efficiency of these solvers and provers is a major focus of activity. As better solvers and provers are developed, they can be plugged into event calculus reasoning programs.

The SAT approach is particularly effective. A SAT solver takes as input a set of Boolean variables and a propositional formula over those variables and produces as output zero or more models or satisfying truth assignments, truth assignments for the

Table 1.2 Event Calculus Reasoning Programs

Description	Technique	Reasoning Types
Event calculus planner (Shanahan, 2000)	Abductive logic programming	Abduction
Event calculus planner (Shanahan & Witkowski, 2004)	Propositional satisfiability	Abduction
Discrete Event Calculus Reasoner (Mueller, 2004a, 2004b)	Propositional satisfiability	Deduction, abduction, postdiction, model finding
Discrete event calculus theorem prover (Mueller & Sutcliffe, 2005b, 2005a; Sutcliffe & Suttner, 2014)	First-order logic automated theorem proving	Deduction
F2LP (Kim, Lee, & Palla, 2009; Lee & Palla, 2012)	Answer set programming	Deduction, abduction, postdiction, model finding

variables such that the formula is true. SAT solvers take a propositional formula in conjunctive normal form: a conjunction of clauses where each clause is a disjunction of literals and where each literal is a variable or a negated variable. In order to use a SAT solver to solve an event calculus problem, formulas of predicate logic must be transformed into formulas of propositional logic. This is accomplished by restricting the problem to a finite universe. Although entailment in propositional logic is decidable, it is *NP-complete*, or believed in the worst case to take a number of steps that is exponential on the size of the problem. But, in practice, large real-world SAT problems, some with as many as tens of millions of clauses and variables, can be solved efficiently.

BIBLIOGRAPHIC NOTES

Commonsense reasoning

People have long sought to describe and capture commonsense reasoning. Logic was developed to characterize valid reasoning (Kneale & Kneale, 1962, pp. 738 and 739). The advent of computers led to the field of artificial intelligence and to a call by McCarthy (1959) to use logic to build computer programs with common sense. Computer science and artificial intelligence drove the development of new logics (Gabbay & Guenthner, 2001, p. vii). Crevier (1993), Russell and Norvig (2009), and McCorduck (2004) provide histories of the field of artificial intelligence.

Book-length treatments of commonsense reasoning are provided by Hobbs and Moore (1985), Minsky (1986), E. Davis (1990), Lenat and Guha (1990), Lifschitz (1990a), Thanassas (1992), Reiter (2001), and Thielscher (2005a, 2008). Books on the related area of knowledge representation are by Reichgelt (1991), Baral (2003), Brachman and Levesque (2004), and van Harmelen, Lifschitz, and Porter (2008). Textbooks on artificial intelligence are by Nilsson (1998) and Russell and Norvig (2009).

The ability to perform commonsense reasoning starts to develop early in life. Several-month-old infants are able to reason using pieces of commonsense knowledge such as that a moving object must follow an unbroken path over time, an object cannot pass through another object, the parts of an object move together, and unsupported objects fall (Baillargeon, 1995; Spelke, Vishton, & von Hofsten, 1995).

Thagard (1996) reviews the basic notions of representations and computational processes that operate on those representations. Our list of key issues of commonsense reasoning follows those of other researchers. McCarthy (1984b, pp. 131-135) presents important aspects of commonsense capability. He considers reasoning about action and change to be a central aspect, writing that “the most salient commonsense knowledge concerns situations that change in time as a result of events” (p. 131). McCarthy also mentions other key aspects: knowledge about knowledge; knowledge about objects, space, beliefs, goals, intentions, and commonsense physics; logical inference; obtaining facts by observation; and default reasoning. For E. Davis (1990), the important areas are plausible reasoning, quantities and measurements, time, space, physics, minds, plans and goals, and society.

Logic and commonsense reasoning

The assumptions of logic-based artificial intelligence are elaborated by Nilsson (1991). Whether logic is the right approach to commonsense reasoning has been hotly debated (Birnbaum, 1991; Ginsberg, 1991; Hayes, 1977; Israel, 1985; Kolata, 1982; McCarthy & Lifschitz, 1987; McDermott, 1987; Minsky, 1974, 1986, 1991b; R. C. Moore, 1982; Nilsson, 1991). The following advantages of logic have been pointed out:

- Logic can be used to represent any domain (R. C. Moore, 1982, p. 430).
- Via model theory, logic provides an account of the meaning of logical formulas (Hayes, 1977, pp. 559-561; Nilsson, 1991, pp. 34-40).
- Logic allows the representation of incomplete information (R. C. Moore, 1995, p. 7).

The following alleged disadvantages of logic have been pointed out:

- Logic focuses on deductive reasoning, and not all reasoning is deductive (Birnbaum, 1991, pp. 59, 70 and 71; McDermott, 1987, pp. 151 and 152). But note that deduction is not the only type of reasoning that can be performed with logic; other types of reasoning such as abduction can be performed (Shanahan, 1997b, pp. xxxii and xxxiii).

- Logic is preoccupied with consistency, and anything can be deduced from a contradiction (Birnbaum, 1991, p. 63; Hewitt, 1987, pp. 185 and 186; Minsky, 1974, pp. 76-78). But note that logics have been developed without this property, such as paraconsistent logic (Priest, 2002) and active logic (Elgot-Drapkin & Perlis, 1990; Elgot-Drapkin, Kraus, Miller, Nirkhe, & Perlis, 1999). Also note that a logical theory can be revised by subtracting axioms as well as adding them (Hayes, 1979, pp. 54 and 55; Israel, 1980, p. 101; 1985, pp. 436 and 437). See also the discussion of Bibel and Nicolas (1989, pp. 18-22).
- Logical reasoning is computationally inefficient (Birnbaum, 1991, p. 72; Minsky, 1974, p. 76). But note that the efficiency of theorem-proving systems is constantly being improved (Balint, Belov, Heule, & Järvisalo, 2013; Sutcliffe, 2013).

The role of logic in human reasoning has been vigorously debated. Henle (1962), Braine (1978), Rips (1983, 1994), Braine and O'Brien (1998), and others argue that humans use a *mental logic* in which inference rules similar to those of the natural deduction of formal logic are applied, and they present experimental evidence supporting this theory. Other researchers disagree. Johnson-Laird (1983, 1993) proposes and presents experimental evidence for a *mental models* approach in which humans reason by building, updating, and evaluating models in the mind.

Logical methods for commonsense reasoning

McCarthy (1963, 1968) and McCarthy and Hayes (1969) introduced the situation calculus. Kowalski and Sergot (1986) introduced the original event calculus within the framework of logic programming (Kowalski, 1979). The event calculus was reformulated in classical predicate logic by Shanahan (1995a, 1996, 1997b, 1999a, 1999b). R. Miller and Shanahan (1999, 2002) introduced several alternative formulations of the classical logic event calculus. Mueller (2004a) introduced the discrete event calculus. Introductions to the classical logic event calculus are provided by Shanahan (1999a) and Mueller (2009). Symposia on logical formalizations of commonsense reasoning are regularly held (Kakas, 2013), as are conferences on knowledge representation and reasoning (Eiter, 2014). The Winograd Schema Challenge is a competition for commonsense reasoning.[1]

Benchmark problems

Lifschitz (1989) created a list of commonsense reasoning benchmark problems, following a suggestion by John McCarthy. E. Davis (1990, pp. 4-12) presents a methodology for formalization of commonsense reasoning based on the use of benchmark problems. Sandewall (1994) proposes a systematic methodology for assessing entailment methods, which we discuss in the Bibliographic notes of Chapter 16. McCarthy (1983, 1998a) argues that the field of artificial intelligence needs an equivalent to drosophilae, the fruit flies biologists use to study genetic mutations

[1] http://commonsensereasoning.org/winograd.html

because of their short generation time. A list of unsolved benchmark problems and a few solved ones is maintained by Leora Morgenstern (Morgenstern & Miller, 2014).

The kitchen sink scenario is from Shanahan (1990; 1997b, pp. 326-329; 1999a, pp. 426-428). This scenario can be traced back to Siklóssy and Dreussi (1973, pp. 426, 429) and Hendrix (1973, pp. 149, 159-167), who used the example of filling a bucket with water. McDermott (1982, pp. 129-133, 135-138) used the example of water flowing into a tank, and Hayes (1985, pp. 99-103) used the example of filling a bath. The shopping cart example is from Shanahan (1997b, pp. 302-304). The hungry cat scenario is from Winikoff, Padgham, Harland, and Thangarajah (2002).

Nonlogical methods for commonsense reasoning

Conceptual dependency, scripts, plans, goals, and themes are discussed by Schank and Abelson (1977), Schank and Riesbeck (1981), Abelson (1981), Schank (1982), and Dyer (1983). Singh (2002) and Singh et al. (2002) describe Open Mind Common Sense, which was part of the larger Open Mind Initiative (Stork, 1999). Hearst (1992, 1998) describes methods for mining WordNet relations from text. WordNet is described by Fellbaum (1998). Watson is described by Ferrucci et al. (2010). Bibliographic notes for nonlogical methods are provided in Chapters 17–19.

Declarative/procedural distinction

The distinction between declarative and procedural (or imperative) knowledge representation is discussed by McCarthy (1959, pp. 79 and 80), Winograd (1975), Hayes (1977), Winston (1977, pp. 390-392), and Genesereth and Nilsson (1987, pp. 2-4). Winograd (1975, p. 186) and McCarthy (1988, p. 299) suggest that declarative representations are more flexible. They point out that a fact represented declaratively can be used for many purposes, even unanticipated ones, whereas a fact represented procedurally has to be represented differently for each purpose.

Automated reasoning and SAT

Automated theorem proving is treated in detail by Robinson and Voronkov (2001a, 2001b).

The undecidability of first-order logic entailment, satisfiability, and validity is due to Church (1936/2004) and Turing (1936/2004). The semidecidability of first-order logic entailment, unsatisfiability, and validity is due to Gödel (1930/1967). It is useful to keep in mind the following relationships among entailment, satisfiability, and validity:

- ψ entails π if and only if $\psi \wedge \neg\pi$ is unsatisfiable.
- ψ entails π if and only if $\neg\psi \vee \pi$ is valid.
- ψ entails π if and only if $\psi \Rightarrow \pi$ is valid.
- ψ is valid if and only if $\neg\psi$ is unsatisfiable.

Satisfiability (SAT) solvers are discussed by Du, Gu, and Pardalos (1997), Gomes, Kautz, Sabharwal, and Selman (2008), and Biere, Heule, van Maaren, and Walsh

(2009). Automated theorem-proving system competitions (Sutcliffe, 2013), SAT system competitions (Balint, Belov, Heule, & Järvisalo, 2013), and answer set programming system competitions (Calimeri, Ianni, & Ricca, 2014), are regularly held. The use of SAT solving for planning was proposed by Kautz and Selman (1992, 1996). The use of SAT solving in the event calculus was introduced by Shanahan and Witkowski (2004). Runtime statistics for the solution of some event calculus reasoning problems using SAT are given by Mueller (2003, 2004b, 2004c). NP-completeness is discussed by Garey and Johnson (1979). The NP-completeness of propositional satisfiability was proved by Cook (1971) and Levin (1973). The growth in the capabilities of SAT solvers is discussed by Selman, Kautz, and McAllester (1997), Kautz and Selman (2003), and Dechter (2003, pp. 186-188). The satisfiability of the `post-cbmc-zfcp-2.8-u2-noholes` problem, which has 10,950,109 variables and 32,697,150 clauses, was proved in under 30 s by several solvers (SAT Competition, 2011).