"for each square \([x, y]\)." For any practical agent, these phrases have to be implemented by code that generates instances of the general sentence schema automatically for insertion into the knowledge base. For a wumpus world of reasonable size—one comparable to a smallish computer game—we might need a 100 x 100 board and 1000 time steps, leading to knowledge bases with tens or hundreds of millions of sentences. Not only does this become rather impractical, but it also illustrates a deeper problem: we know something about the wumpus world—namely, that the "physics" works the same way across all squares and all time steps—that we cannot express directly in the language of propositional logic. To solve this problem, we need a more expressive language, one in which phrases like "for each time \(t\)" and "for each square \([x, y]\)" can be written in a natural way. First-order logic, described in Chapter 8, is such a language; in first-order logic a wumpus world of any size and duration can be described in about ten sentences rather than ten million or ten trillion.

### 7.8 SUMMARY

We have introduced knowledge-based agents and have shown how to define a logic with which such agents can reason about the world. The main points are as follows:

- Intelligent agents need knowledge about the world in order to reach good decisions.
- Knowledge is contained in agents in the form of sentences in a knowledge representation language that are stored in a knowledge base.
- A knowledge-based agent is composed of a knowledge base and an inference mechanism. It operates by storing sentences about the world in its knowledge base, using the inference mechanism to infer new sentences, and using these sentences to decide what action to take.
- A representation language is defined by its syntax, which specifies the structure of sentences, and its semantics, which defines the truth of each sentence in each possible world or model.
- The relationship of entailment between sentences is crucial to our understanding of reasoning. A sentence \(\alpha\) entails another sentence \(\beta\) if \(\beta\) is true in all worlds where \(\alpha\) is true. Equivalent definitions include the validity of the sentence \(\alpha \models \beta\) and the unsatisfiability of the sentence \(\alpha \land \neg \beta\).
- Inference is the process of deriving new sentences from old ones. Sound inference algorithms derive only sentences that are entailed; complete algorithms derive all sentences that are entailed.
- Propositional logic is a simple language consisting of proposition symbols and logical connectives. It can handle propositions that are known true, known false, or completely unknown.
- The set of possible models, given a fixed propositional vocabulary, is finite, so entailment can be checked by enumerating models. Efficient model-checking inference algorithms for propositional logic include backtracking and local search methods and can often solve large problems quickly.
• Inference rules are patterns of sound inference that can be used to find proofs. The resolution rule yields a complete inference algorithm for knowledge bases that are expressed in conjunctive normal form. Forward chaining and backward chaining are very natural reasoning algorithms for knowledge bases in Horn form.

• Local search methods such as WALKSAT can be used to find solutions. Such algorithms are sound but not complete.

• Logical state estimation involves maintaining a logical sentence that describes the set of possible states consistent with the observation history. Each update step requires inference using the transition model of the environment, which is built from successor-state axioms that specify how each fluent changes.

• Decisions within a logical agent can be made by SAT solving: finding possible models specifying future action sequences that reach the goal. This approach works only for fully observable or sensorless environments.

• Propositional logic does not scale to environments of unbounded size because it lacks the expressive power to deal concisely with time, space, and universal patterns of relationships among objects.

BIBLIOGRAPHICAL AND HISTORICAL NOTES

John McCarthy’s paper “Programs with Common Sense” (McCarthy, 1958, 1968) promulgated the notion of agents that use logical reasoning to mediate between percepts and actions. It also raised the flag of declarativism, pointing out that telling an agent what it needs to know is an elegant way to build software. Allen Newell’s (1982) article “The Knowledge Level” makes the case that rational agents can be described and analyzed at an abstract level defined by the knowledge they possess rather than the programs they run. The declarative and procedural approaches to AI are analyzed in depth by Boden (1977). The debate was revived by, among others, Brooks (1991) and Nilsson (1991), and continues to this day (Shaparani et al., 2008). Meanwhile, the declarative approach has spread into other areas of computer science such as networking (Lao et al., 2006).

Logic itself had its origins in ancient Greek philosophy and mathematics. Various logical principles—principles connecting the syntactic structure of sentences with their truth and falsity, with their meaning, or with the validity of arguments in which they figure—are scattered in the works of Plato. The first known systematic study of logic was carried out by Aristotle, whose work was assembled by his students after his death in 322 B.C. as a treatise called the Organon. Aristotle’s syllogisms were what we would now call inference rules. Although the syllogisms included elements of both propositional and first-order logic, the system as a whole lacked the compositional properties required to handle sentences of arbitrary complexity.

The closely related Megarian and Stoic schools (originating in the fifth century B.C. and continuing for several centuries thereafter) began the systematic study of the basic logical connectives. The use of truth tables for defining connectives is due to Philo of Megara. The