# Introduction to Artificial Intelligence

Summary of Chapter 7

Blanka Hasior

June 12, 2018

## Contents

# 1 Agents use in Artificial Intelligence

## 1.1 Introduction

Artificial intelligence is intelligence demonstrated by machines, in contrast to the natural intelligence (NI) displayed by humans and other animals. In computer science AI research is defined as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Colloquially, the term "artificial intelligence" is applied when a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving". The scope of AI is disputed: as machines become increasingly capable, tasks considered as requiring "intelligence" are often removed from the definition, a phenomenon known as the AI effect, leading to the quip, "AI is whatever hasn't been done yet." For instance, optical character recognition is frequently excluded from "artificial intelligence", having become a routine technology. Capabilities generally classified as AI as of 2017include successfully understanding human speech, competing at the highest level in strategic game systems autonomous cars, intelligent routing in content delivery network and military simulations. Artificial intelligence was founded as an academic discipline in 1956, and in the years since has experienced several waves of optimism, followed by disappointment and the loss of funding followed by new approaches, success and renewed funding. For most of its history, AI research has been divided into subfields that often fail to communicate with each other. These sub-fields are based on technical considerations, such as particular goals (e.g. "robotics" or "machine learning"), the use of particular tools ("logic" or artificial neural networks), or deep philosophical differences. Subfields have also been based on social factors (particular institutions or the work of particular researchers). The traditional problems (or goals) of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception and the ability to move and manipulate objects. General intelligence is among the field's long-term goals. Approaches include statistical methods, computational intelligence, and traditional symbolic AI. Many tools are used in AI, including versions of search and mathematical optimization, artificial neural networks, and methods based on statistics, probability and economics. The AI field draws upon computer science, mathematics, psychology, linguistics, philosophy and many others. The field was founded on the claim that human intelligence "can be so precisely described that a machine can be made to simulate it". This raises philosophical arguments about the nature of the mind and the ethics of creating artificial beings endowed with human-like intelligence which are issues that have been explored by myth, fiction and philosophy since antiquity. Some people also consider AI to be a danger to humanity if it progresses unabatedly. Others believe that AI, unlike previous technological revolutions, will create a risk of mass unemployment. In the twenty-first century, AI techniques have experienced a resurgence following concurrent advances in computer power, large amounts of data, and theoretical understanding; and AI techniques have become an essential part of the technology industry, helping to solve many challenging problems in computer science.

## 1.2 Knowledge-based agents

The central component of a knowledge – based agent is its knowledge base. The knowledge base is frequently called sentence. Each sentence represents some assertion about the world. There must be a way to add new sentence to the knowledge base and way to query what is known. Very frequently there is a phenomenon interference. Word interference in artificial network tell that new sentence deriving old sentence. The standard names for these operation are tell and ask. The agent maintains a knowledge base which may initially contain some background knowledge. Agent program consist of three things. The first it TELLs it is the knowledge base what it perceives. Second thing it ASKs the knowledge base what action it should platform. In the process of answering this query, extensive reasoning may be done about the current state of the world, about the outcomes of possible action sequences, and so on. Third, the agent program TELLs the knowledge base which action was chosen, and the agent executes the action. The details of the representation language are hidden inside three functions that implement the interface between the sensors and actuators on one side and the core representation and reasoning system on the other. MAKE-PERCEPT-SENTENCE constructs a sentence asserting that the agent perceived the given percept at the given time. MAKE-ACTION-QUERY constructs a sentence that asks what action should be done at the current time. Finally, MAKE-ACTION-SENTENCE constructs a sentence asserting that the chosen action was executed. The details of the inference mechanisms are hidden inside TELL and ASK. Later sections will reveal these details. A knowledge-based agent can be built simply by TELLing it what it needs to know. Starting with an empty knowledge base, the agent designer can TELL sentences one by one DECLARATIVE until the agent knows how to operate in its environment. We now understand that a successful agent often combines both declarative and procedural elements in its design, and that declarative knowledge can often be compiled into more efficient procedural code.

## 1.3 Logical

As we know well knowledge bases consist of sentences. These sentences are expressed according to the syntax of the representation language, which specifies all the sentences that are well formed. The notion of syntax is clear enough in ordinary arithmetic:is a well-formed sentence, whereas is not. A logic must also define the semantics or meaning of sentences. The semantics defines the truth of each sentence with respect to each possible world. When we need to be precise, we use the term model in place of "possible world." Whereas possible worlds might be thought of as real environments that the agent might or might not be in, models are mathematical abstractions, each of which simply fixes the truth or falsehood of every relevant sentence. If a sentence is true in model m, we say that m satisfies or sometimes m is a model of . We use the notation M to mean the set of all models of . Now that we have a notion of truth, we are ready to talk about logical reasoning. This involves the relation of logical entailment between sentences the idea that a sentence follows logically from another sentence. An inference algorithm that derives only entailed sentences is called sound or truth preserving. Soundness is a highly desirable property. An unsound inference procedure essentially makes things up as it goes along—it announces the discovery of nonexistent needles. It is easy to see that model checking, when it is applicable,4 is a sound procedure. The property of completeness is also desirable: an inference algorithm is complete if it can derive any sentence that is entailed. For real haystacks, which are finite in extent, it seems obvious that a systematic examination can always decide whether the needle is in the haystack. For many knowledge bases, however, the haystack of consequences is infinite, and completeness becomes an important issue. Fortunately, there are complete inference procedures for logics that are sufficiently expressive to handle many knowledge bases. So, while an inference process operates on "syntax"—internal physical configurations such as bits in registers or patterns of electrical blips in brains—the process corresponds. To the real-world relationship whereby some aspect of the real world is the case6 by virtue of other aspects of the real world being the case. This correspondence between world and representation is illustrated under picture (Figure 2).

## 1.4 Propositional logic

We cover the syntax of propositional logic and its semantics the way in which the truth of sentences is determined. Then we look at entailment the relation between a sentence and another sentence that follows from it and see how this leads to a simple algorithm for logical inference. The syntax of ATOMIC SENTENCES propositional logic defines the allowable sentences. The atomic sentences PROPOSITION consist of a single proposition symbol. Each such symbol stands for a proposition that can SYMBOL be true or false. We use symbols that start with an uppercase letter and may contain other letters or subscripts, for example: P, Q, R, W1,3 and North. Complex sentences are constructed from simpler sentences, using parentheses and logical connectives. There are five connectives in common use:

## 1.5 Semantics

Having specified the syntax of propositional logic, we now specify its semantics. The semantics defines the rules for determining the truth of a sentence with respect to a particular model. In propositional logic, a model simply fixes the truth value—true or false—for every proposition symbol. The semantics for propositional logic must specify how to compute the truth value of any sentence, given a model. This is done recursively. All sentences are constructed from atomic sentences and the five connectives; therefore, we need to specify how to compute the truth of atomic sentences and how to compute the truth of sentences formed with each of the five connectives. Atomic sentences are easy:

- True is true in every model and False is false in every model.

- The truth value of every other proposition symbol must be specified directly in the model. For example, in the model m1 given earlier, P1,2 is false. For complex sentences, we have five rules, which hold for any subsentences P and Q in any model m (here "if" means "if and only if"):

## 1.6 Propositional theorem proving

Before we plunge into the details of theorem-proving algorithms, we will need some additional concepts related to entailment. The first concept is logical equivalence: two sen- tences are logically equivalent if they are true in the same set of models. We write this as . For example, we can easily show (using truth tables) that P Q and Q P are logically equivalent; other equivalences are shown. These equivalences play much the same role in logic as arithmetic identities do in ordinary mathematics. An alternative definition of equivalence is as follows: any two sentences and are equivalent only if each of them entails the other. The second concept we will need is validity. A sentence is valid if it is true in all models. For example, the sentence is valid. Valid sentences are also known as tautologies they are necessarily true. Because the sentence True is true in all models, every valid sentence is logically equivalent to True. What good are valid sentences? From our definition of entailment, we can derive the deduction theorem, which was known to the ancient Greeks: For any sentences if and only if the sentence is valid. The final concept we will need is satisfiability. A sentence is satisfiable if it is true in, or satisfied by, some model. For example, the knowledge base given earlier Satisfiability can be checked by enumerating the possible models until one is found that satisfies the sentence. The problem of determining the satisfiability of sentences in propositional logic - the SAT problem - was the first problem proved to be NP-complete. Many problems in computer science are really satisfiability problems. For example, all the constraint satisfaction problems in Chapter 6 ask whether the constraints are satisfiable by some assignment. Validity and satisfiability are of course connected: is valid if is unsatisfiable; contrapositively, is satisfiable if is not valid. We also have the following useful result:

## 1.7 Inference and proofs

The best-known rule is called Modus Ponens. The notation means that, whenever any sentences of the form and are given, then the sentence can be inferred. Another useful inference rule is And Elimination, which says that, from a conjunction, any of the conjuncts can be inferred. By considering the possible truth values of and , one can show easily that Modus Ponens and And Elimination are sound once and for all. These rules can then be used in any particular instances where they apply, generating sound inferences without the need for enumerating models. We found this proof by hand, but we can apply any of the search algorithms to find a sequence of steps that constitutes a proof. We just need to define a proof problem as follows:

- INITIAL STATE: the initial knowledge base.
- ACTIONS: the set of actions consists of all the inference rules applied to all the sentences that match the top half of the inference rule.
- RESULT: the result of an action is to add the sentence in the bottom half of the inference rule.
- GOAL: the goal is a state that contains the sentence we are trying to prove.

Then, the resolution rule is applied to the resulting clauses. Each pair that contains complementary literals is resolved to produce a new clause, which is added to the set if it is not already present. The process continues until one of two things happens:

- there are no new clauses that can be added, in which case KB does not entail ;
- two clauses resolve to yield the empty clause, in which case KB entails .

The empty clause disjunction of no disjuncts is equivalent to false because a disjunction is true only if at least one of its disjuncts is true. Another way to see that an empty clause represents a contradiction is to observe that it arises only from resolving two complementary unit clauses such as P.

## References

[1] Artificial Intelligence Modern Approach, Peter Norvig, Stuart J. Russell.