Knowledge-based agents

Humans, it seems, know things and what they know helps them to do things. These are not empty statements. They make strong claims about how the intelligence of humans is achieved, not by purely reflex mechanisms but by process of **reasoning** that operate on the internal **representation** of knowledge. In AI, this approach to intelligence is embodied in **Knowledge-based agents**.

In this lecture we develop **logic** as a general class of representations to support knowledge-base agents. Such agents can combine and recombine information to suit different purposes. Knowledge-based agents can accept new task in the form of explicitly described goals; they can achieve competence quickly by being told or learning new knowledge about the environment; and they can adapt to changes in the environment by updating the relevant knowledge.

In this lecture we explain the following:

- An overall agent designs.
- Illustrates the operation of knowledge-based agent.
- Explain the general principles of logic.

Logical Agents

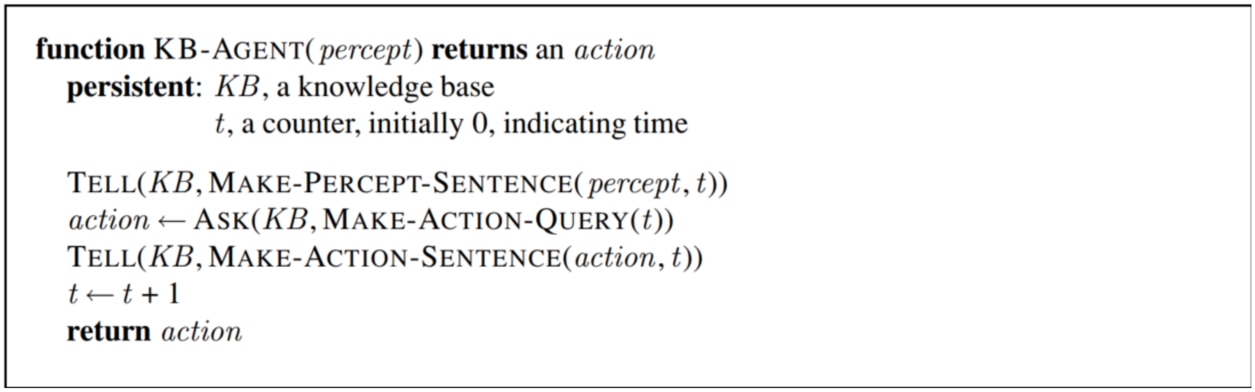
In which we design agents that can form representations of a complex world, use a process of inference to derive new representation about the world. And use this new representation to deduce what to do.

Knowledge-Based Agents

The central component of a knowledge-based agent is **Knowledge base (KB). A knowledge base** is a set of **sentences.** Each sentence is expressed in a language called a **knowledge representation language** and represents some assertion about the world. Sometimes we dignify a sentence with the name **axiom**, when the sentence is taken as give without being derived from other sentences.

There must be a way to add new sentence to the knowledge base and a way to query what is known. The standard names for these operations are **TELL** and **ASK**, respectively. Both operations may involve **inference**, that is deriving new sentences from old. Inference must obey the requirement that when one asks a question of the knowledge base, the answer should follow from what has been told to the knowledge base previously.

Figure below shows the outline of a knowledge-base agent program. Like all our agents, it takes a precept as input and returns an action. The agent maintains a knowledge base, KB, which may initially contain some **background knowledge**.



```
function KB-AGENT(percept) returns an action
  persistent: KB, a knowledge base
              t, a counter, initially 0, indicating time

  TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
  action ← ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE(action, t))
  t ← t + 1
  return action
```

Each time an agent program is called, it does three things.

1. It TELLs the knowledge base what it perceives.
2. It ASKS the knowledge base what action it should perform. In the process of answering this query, extensive reasoning may be done

E-mail: hasanein.y.m.alhrabi@uomus.edu.iq

about the current state of the world, about the outcomes of possible actin sequence, and so on.

3. The agent program TELLs the knowledge base which action was chosen, and agent executes the action.

Logic

This section summarizes the fundamental concepts of logical representation and reasoning. These beautiful ideas are independent of any logic's particular forms. We therefore postpone the technical details those forms until the next sections.

In the previous section we said that knowledge bases consist of sentences. these sentences are expressed according 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: X + Y = 4 is a well formed sentence, whereas X4Y+= 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**. For example, the semantics for arithmetic specifies that the sentence X + Y = 4 is true in a world where X is 2 and Y is 2. But false in a world where X is 1 and Y is 1. **In standard logic, every sentence must be either true or false in each possible world (model), there is no "in between"**.

When we need to be precise, we use the term **model** in place of possible world. Whereas possible worlds might be thought of as potentially real environments that the agent might or might not be in, **models are**

mathematical abstractions, **each of which simply fixes the truth or false of every relevant sentence**. **Formally, the possible models are just all possible assignments of real numbers to the variable x and y**. if a sentence **α** is true in model ***m***, we say that m **satisfies** α or sometimes m **is a model of** α. We use the notation ***M(α)*** to mean the set of all models of α.

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. In mathematical notation, we write.

$$\alpha \models \beta$$

to mean that the sentence α entails the sentence β. The formal definition of entailment is this: $\alpha \models \beta$ if and only if, in every model in which α is true, β is also true. Using the notation just introduced, we can write.

$$\alpha \models \beta \text{ if and only if } M(\alpha) \subseteq M(\beta)$$

Note the direction of the $\subseteq$ here: if $\alpha \models \beta$, then α is a stronger assertion than β. It rules out more possible worlds.

An inference algorithm that derives only entailed sentences is called **sound** or **truth preserving**. Soundness is a highly desirable property.

The property of **completeness** is also desirable, an inference algorithm is complete if it can derive any sentence that is entailed.

E-mail: hasanein.y.m.alhrabi@uomus.edu.iq

The final issue to consider is **grounding**, the connection between logical reasoning process and the real environment in which the agent exists. In particular, how do we know that KB is true in the real world? A simple answer is that the agent’s sensors cerate the connections.