Knowledge Representation and Reasoning (KRR)

Marina Anca Cidota
Faculty of Mathematics and Computer Science
University of Bucharest

cidota@fmi.unibuc.ro

References

1. Ronald J. Brachman, Hector J. Levesque. Knowledge representation and reasoning, Morgan Kaufmann, 2004.
2. Stuart J. Russell, Peter Norvig. Artificial Intelligence – a modern approach, 3rd edition, Pearson, 2010.
3. Ivan Bratko. Prolog Programming for Artificial Intelligence, Pearson Education Canada, 4th Edition, 2011.

Introduction

- Intelligence $\frac{explanatory}{dictionary}$ the ability to understand easily and well, to understand what is essential, to solve new situations or problems from previous experiences.

- The intelligent behavior is clearly conditioned by knowledge. Artificial intelligence (AI) studies the intelligent behavior acquired through computational means.

- KRR is an area of AI concerned with the study of how an agent (e.g., human, hardware, software) uses what it knows when it decides what to do. It is the study of thinking as a computational process.

An example of a problem

Tony, Mike and John belong to the Alpine Club. Every member of the Alpine Club who is not a skier is a mountain climber. Mountain climbers do not like rain, and anyone who does not like snow is not a skier. Mike dislikes whatever Tony likes, and likes whatever Tony dislikes. Tony likes rain and snow.

Is it true that there is a member of the Alpine Club who is a mountain climber but not a skier?

Knowledge

- In an informal description, knowledge is a relation between a knower and a proposition. The proposition is the idea expressed by a simple declarative sentence.

John knows that Mary will come to the party.

knower relation proposition

- Propositions are abstract entities that can be true or false, right or wrong.

- There may be different types of relationships between an agent and a proposition:

  John knows/hopes/doubts/regrets that Mary will come to the party.

- However, what matters about the propositions is their truth value.

Knowledge

- There are statements containing knowledge, which is not explicitly mentioned in propositions:

  John knows how to get there.

In this case, we can imagine the implicit propositions:

  John knows how to get to the party, he goes two blocks past the park, he turns left…

- On the other hand, in sentences like

  John knows George well.

the knowledge is vague.

- There are attitudes expressed by sentences like:

  John is absolutely certain/confident/of the opinion that…

These differ only in the level of conviction attributed to a fact. The judgement may not be always exact.

Representation

- It means a relationship between two domains, where the first takes the place of the second. The first one is more concrete, accessible than the second one.

- We shall consider the symbolic representation, that is a character or a group of characters from a predefined alphabet:

  7, VII, seven – represent number 7

  John – represents something concrete

  righteousness, truth – represent abstractions

  John loves Mary – symbolic representation of an abstract statement

- Knowledge representation is the field that studies the use of formal symbols to represent a collection of propositions made by an agent.

Reasoning

- It means the formal manipulation of the formal symbols that represent a collection of statements considered to be true, in order to produce representations of new symbols.

- Therefore, the symbols must be “accessible” enough to be able to manipulate them (i.e. move, take apart, copy, concatenate), so that representations of new propositions may be constructed.

- For instance, if we are given the propositions:

  John loves Mary.

  Mary comes to the party.

after a series of steps in reasoning, we can produce the sentence

  Someone John loves comes to the party.

- Reasoning is a form of calculation over symbols standing for propositions (rather that numbers in arithmetic) -- Gottfried Leibniz.

Why is KRR relevant for AI?

- Because in some situations it is useful to describe the behavior of a complex system in terms like believes, desires, intends, hopes etc.

For example, if we play chess against the computer – in order to help us make the next move, it would be useful to understand the program's behavior in terms of immediate goals pursued relative to its long-term intentions:

"It moved this way because it believed that the queen was vulnerable…"

Why is KRR relevant for AI?

- Because in some situations it is useful to describe the behavior of a complex system in terms like believes, desires, intends, hopes etc.

For example, if we play chess against the computer – in order to help us make the next move, it would be useful to understand the program's behavior in terms of immediate goals pursued relative to its long-term intentions:

"It moved this way because it believed that the queen was vulnerable…"

- A knowledge base (KB) is a collection of symbolic structures that represent the knowledge but also the reasoning made during the system operation.

The behavior of a knowledge-based system is conditioned not only by the represented facts (that can be retrieved like in a database). Through reasoning, the "beliefs" of the system go beyond these facts.

Why is KRR relevant for AI?

- There are AI systems that are fully knowledge-based. Expert systems are a classical example, but applications of knowledge-based systems can be found in the areas of language understanding, diagnosis, planning.

- Other AI systems are knowledge-based to a lesser extent – e.g. some games or high-level vision systems.

- There are also AI systems that are not knowledge-based at all – for example low-level speech, vision. These systems encode knowledge directly in the program.

- Open question: How much of the intelligent behavior should be knowledge-based?

Advantages of a knowledge-based system

Wouldn’t it be more efficient to “compile” the knowledge base (KB) and the distribute it to the procedures that need it? (this approach is known as procedural knowledge).

Why should we retrieve facts from the KB and make reasoning in the runtime to decide next actions?

Advantages of a knowledge-based system

Wouldn’t it be more efficient to “compile” the knowledge base (KB) and the distribute it to the procedures that need it? (this approach is known as procedural knowledge).

Why should we retrieve facts from the KB and make reasoning in the runtime to decide next actions?

By design, a knowledge-based system has the ability to learn facts about the world and, consequently, to adjust its behavior. The ability to make the behavior dependent on the knowledge is desirable when we cannot specify in advance how the knowledge will be used. This approach has some advantages:

- we can add new tasks and make them dependent on previous knowledge;
- we can extend the behavior by adding new beliefs;
- we can explain the behavior of the system.

Still …

Hurbert Dreyfus observed a paradox of expert systems:

These systems claim to be superior because they are knowledge-based. But experts do not reason – they just see the solution. Novices are the ones who reason. Dreyfus considers to be wrong the directions that attempts to copy the human intelligent behavior.

Still …

Hurbert Dreyfus observed a paradox of expert systems:

These systems claim to be superior because they are knowledge-based. But experts do not reason – they just see the solution. Novices are the ones who reason. Dreyfus considers to be wrong the directions that attempts to copy the human intelligent behavior.

The most serious challenge to KRR is the connectionist approach, that computes weights between artificial neurons. But the current machine-learning systems “learn in a very narrow way, they need much more data to learn a new task than [humans], they need humans to provide high-level concepts through labels, and they still make really stupid mistakes” – Dr. Yoshua Bengio

The Economist, June 13th 2020

The evaluation of the course

You will have to work individually on two projects, each one will consist of two parts:

- implementation (preferably in Prolog*, but any other programming language is allowed). If you do not use Prolog, you will pe penalized with 2 points.
- a written document (between 2-10 pages without the code); the code will be added at the end of the document, which will be verified for similarities with Turnitin.

The first project will be evaluated until the Christmas holiday, and the second one during the winter session.

The deadlines for the projects will be announced later.

Each project will be graded and the average of the two grades will be the final grade.

- If you miss both deadlines, you will be ABSENT in the catalog.
- If you miss only one deadline, you will get the grade 1 for that project.
- To pass the exam, the average of the two grades must be greater than or equal to 5.

Regarding the re-examinations: the projects will be the same and the grades greater than or equal to 5, obtained during the 1st semester, will be maintained.

*Peter Norvig's essay "Teach yourself programming in 10 years"