Knowledge Representation, Reasoning, and Design of Intelligent Agents

Michael Gelfond and Yulia Kahl

Texas Tech University

Aug 2015
• **Question:** How to build software components of rational agents capable of reasoning and acting in a changing environment?

• **Hypothesis:** To exhibit intelligent behavior, an agent should have a mathematical model of its environment and its own capabilities and goals, as well as algorithms for achieving these goals.

• **Goal:** Discover such models and algorithms, and learn how to use them to build practical intelligent systems.
Why is this important?

- Gives insight into cognitive processes and the nature of rationality. Helps to learn rational modes of reasoning and acting outside of mathematics.

- Helps to discover new design principles and paradigms (i.e. philosophical and theoretical frameworks of a scientific school or discipline within which theories, laws, and generalizations and the experiments performed in support of them are formulated).

- Helps to create very useful software systems.
Agent – entity which observes and acts upon an environment and directs its activity towards achieving goals.

Model of an intelligent agent normally consists of

- A language(s) for representing the agent’s knowledge.

- Reasoning algorithms which use this knowledge to perform intelligent tasks, including planning, diagnostics, learning, etc.

- An agent architecture which is the structure combining different sub-models of an agent (normally related to different reasoning tasks) in one coherent whole.
Research Directions

• Design of knowledge representation languages.

• Design and development of general purpose reasoning algorithms and systems.

• Axiomatizations of knowledge.

• Reduction of Computational Tasks to Reasoning.

• Applications.
Languages differ according to the type of information their designers want to communicate to computers. Two basic types:

ALGORITHMIC languages describe sequences of actions for a computer to perform.

DECLARATIVE languages describe properties of objects in a given domain.

We concentrate on declarative languages.
father(john, sam).
mother(alice, sam).
gender_of(john, male).
gender_of(sam, male).
gender_of(alice, female).

To test the program’s understanding we ask it some questions:
mother(alice, sam)? yes
mother(X, sam)? X = alice
The agent maintains a knowledge base containing knowledge about environment and its own capabilities and goals, and executes the following loop:

- observes the world, checks that its observations are consistent with its expectations, and updates its knowledge base;
- selects an appropriate goal $G$;
- searches for a plan, (a sequence of actions), to achieve $G$;
- executes some initial part of the plan, updates the knowledge base, and goes back to step (1).
We study the logic-based approach which proposes to

- use a declarative language to describe the domain,

- express various tasks (which may include requests to find plans or explanations of unexpected observations) as queries to the resulting program, and

- use an inference engine, i.e. a collection of reasoning algorithms, to answer these queries.