The FO($\cdot$) Knowledge Base System project

Marc Denecker
Knowledge Representation & Reasoning (KRR)
Katholieke Universiteit Leuven

May 25, 2016

Introduction: the FO($\cdot$) KBS project

A small course:

- Introduction: motivation, definition of terms
- Lecture 1: the FO(.) language framework
- Lecture 2: the knowledge base system IDP
- Lecture 3: advanced topics

Introduction: the FO($\cdot$) KBS project

The fundamental KRR research question

- Humans experts possess (declarative) knowledge.
- They use it
  - to accomplish tasks
  - to solve problems
  - or to build programs that do this for us (computer science)
- How does this work?

The fundamental KRR research question

- Humans experts possess (declarative) knowledge.
- They use it
  - to accomplish tasks
  - to solve problems
  - or to build programs that do this for us (computer science)
- How does this work?

- Inherently a KRR research question.
  - (KRR: Knowledge Representation and Reasoning)
- If we ever want to be able to build software systems in a principled way, we will NEED to understand this.
- This places KRR at the foundations of computer science.

State of the art

- About every area in Computational logic is involved in aspects of this question.
- Scientific understanding is partial and scattered over the many fields of computational logic and declarative problem solving.

State of the art

- About every area in Computational logic is involved in aspects of this question.
- Scientific understanding is partial and scattered over the many fields of computational logic and declarative problem solving.
- One issue that fragments computational logic more than anything else:

the reasoning/inference task

State of the art

- Logics are entangled with a reasoning task:
  - Classical first order logic (FO):
    deduction
  - Deductive Databases (SQL, Datalog):
    query answering & other database operations
  - Answer set Programming (ASP):
    answer set computation
  - Abductive Logic Programming:
    abduction
  - Constraint Programming (CP):
    constraint solving
  - Description logics:
    subsumption $\subseteq$ deduction, other forms of inference
  - Planning languages PDDL :
    planning
  - Temporal logics :
    model checking
  - ...

State of the art

A declarative proposition in natural language:

Each lecturer teaches at least one course in the first bachelor

What is its purpose?
What task is it to be used for?

State of the art

A declarative proposition in natural language:

Each lecturer teaches at least one course in the first bachelor

What is its purpose?
What task is it to be used for?

- It could be a query to a database.

State of the art

A declarative proposition in natural language:

Each lecturer teaches at least one course in the first bachelor

What is its purpose?
What task is it to be used for?

- It could be a query to a database.
- It could be a constraint in a course assignment problem.

State of the art

A declarative proposition in natural language:

Each lecturer teaches at least one course in the first bachelor

What is its purpose?
What task is it to be used for?

- It could be a query to a database.
- It could be a constraint in a course assignment problem.
- It could be a desired property, to be proven from a formal specification of the course assignment domain.
- ...

State of the art

A declarative proposition in natural language:

Each lecturer teaches at least one course in the first bachelor

What is its purpose?
What task is it to be used for?

- It could be a query to a database.
- It could be a constraint in a course assignment problem.
- It could be a desired property, to be proven from a formal specification of the course assignment domain.
- ...

In the current state of the art, depending on the task to be solved, a different language and system is needed to represent this proposition.

Is declarative knowledge not independent of the task (and hence, of a specific form of inference) ?

An illustration

Question: How to solve a graph coloring problem in a declarative way?

- Step 1: What is the knowledge? What is the central proposition of a correct graph coloring?

An illustration

Question: How to solve a graph coloring problem in a declarative way?

- Step 1: What is the knowledge? What is the central proposition of a correct graph coloring?
  - No two adjacent vertices have the same color.
  - In FO (classical first order logic):

$$\forall x \forall y (G(x, y) \Rightarrow Col(x) \neq Col(y))$$

An illustration

Question: How to solve a graph coloring problem in a declarative way?

- Step 1: What is the knowledge? What is the central proposition of a correct graph coloring?
  - No two adjacent vertices have the same color.
  - In FO (classical first order logic):

$$\forall x \forall y (G(x, y) \Rightarrow Col(x) \neq Col(y))$$

- Step 2: What is the problem (Input/Output):
  - Input: a pair of a graph $G(Vertex, Vertex)$ and a set $Color$ of colours; this is a structure $\mathfrak{A}$
  - Output: a function $Col(Vertex) : Color$ satisfying the proposition

An illustration

Question: How to solve a graph coloring problem in a declarative way?

- Step 1: What is the knowledge? What is the central proposition of a correct graph coloring?
  - No two adjacent vertices have the same color.
  - In FO (classical first order logic):

$$\forall x \forall y (G(x,y) \Rightarrow Col(x) \neq Col(y))$$

- Step 2: What is the problem (Input/Output):
  - Input: a pair of a graph $G(Vertex, Vertex)$ and a set $Color$ of colours; this is a structure $\mathfrak{A}$
  - Output: a function $Col(Vertex) : Color$ satisfying the proposition
- Step 3: What kind of inference do we need to solve the problem?

An illustration

Question: How to solve a graph coloring problem in a declarative way?

- Step 1: What is the knowledge? What is the central proposition of a correct graph coloring?
  - No two adjacent vertices have the same color.
  - In FO (classical first order logic):

$$\forall x \forall y (G(x, y) \Rightarrow Col(x) \neq Col(y))$$

- Step 2: What is the problem (Input/Output):
  - Input: a pair of a graph $G(Vertex, Vertex)$ and a set $Color$ of colours; this is a structure $\mathfrak{A}$
  - Output: a function $Col(Vertex) : Color$ satisfying the proposition
- Step 3: What kind of inference do we need to solve the problem?
  - model generation: expand $\mathfrak{A}$ to a model; return $Col^{\mathfrak{A}}$

"What kind of inference do we need to solve this problem?"

- A question that until recently, for FO, was not even asked let alone addressed.

"What kind of inference do we need to solve this problem?"

- A question that until recently, for FO, was not even asked let alone addressed.
- FO was seen exclusively as the logic of deductive reasoning.
  - In some fields, this is still the dominating view.

"What kind of inference do we need to solve this problem?"

- A question that until recently, for FO, was not even asked let alone addressed.
- FO was seen exclusively as the logic of deductive reasoning.
  - In some fields, this is still the dominating view.
- Deduction is utterly useless for solving the graph coloring problem.

"What kind of inference do we need to solve this problem?"

- A question that until recently, for FO, was not even asked let alone addressed.
- FO was seen exclusively as the logic of deductive reasoning.
  - In some fields, this is still the dominating view.
- Deduction is utterly useless for solving the graph coloring problem.
- Instead, people developed new logics to handle problems like this:
  - Constraint Programming Languages
    - Ilog, Zinc, Constraint Logic Programming, ...
  - Answer Set Programming (ASP)

Why do we need all these syntaxes for expressing the same information?

Isn't it possible to solve multiple types of tasks using the same language?

Spread out over all disciplines of computational logic, there is an enormous expertise about KR and inference.

If only we could bundle what is known about KR and inference in a coherent scientific framework!?

The FO($\cdot$)-KBS project: an integration project

On the logical level: FO($\cdot$)

> Knowledge exists, it can be studied through the methods of formal empirical science

Study "knowledge" by principled development of expressive KR languages:

- Clear informal semantics
- Expressive languages, rich enough so that the information, relevant to solve a problem CAN be represented.
  - (We ignore the "Tractibility/Expressivity" trade-off)
- Classical first-order logic (FO) as foundation, extended where necessary.

The FO($\cdot$)-KBS project: an integration project

On the logical level: FO($\cdot$)

> Knowledge exists, it can be studied through the methods of formal empirical science

Study "knowledge" by principled development of expressive KR languages:

- Clear informal semantics
- Expressive languages, rich enough so that the information, relevant to solve a problem CAN be represented.
  - (We ignore the "Tractibility/Expressivity" trade-off)
- Classical first-order logic (FO) as foundation, extended where necessary.

(FO($\cdot$)= family of extensions of FO)

FO($\cdot$): turning FO into a practical KR language

- FO does not suffice for knowledge representation, modelling, specification

FO($\cdot$): turning FO into a practical KR language

- FO does not suffice for knowledge representation, modelling, specification

$\Rightarrow$ FO( )

FO($\cdot$): turning FO into a practical KR language

- FO does not suffice for knowledge representation, modelling, specification

$\Rightarrow$ FO(Types)

- Types

FO($\cdot$): turning FO into a practical KR language

- FO does not suffice for knowledge representation, modelling, specification

$\Rightarrow$ FO(Types,ID)

  - Types
  - (Inductive) Definitions

FO($\cdot$): turning FO into a practical KR language

- FO does not suffice for knowledge representation, modelling, specification

$\Rightarrow$ FO(Types,ID,Agg)

- Types
- (Inductive) Definitions
- Aggregates

FO($\cdot$): turning FO into a practical KR language

- FO does not suffice for knowledge representation, modelling, specification

$\Rightarrow$ FO(Types,ID,Agg,Arit)

  - Types
  - (Inductive) Definitions
  - Aggregates
  - Arithmetic

FO($\cdot$): turning FO into a practical KR language

- FO does not suffice for knowledge representation, modelling, specification

$\Rightarrow$ FO(Types,ID,Agg,Arit,FD)

  - Types
  - (Inductive) Definitions
  - Aggregates
  - Arithmetic
  - Coinductive Definitions

FO(·): turning FO into a practical KR language

- FO does not suffice for knowledge representation, modelling, specification

⇒ FO(Types,ID,Agg,Arit,FD,Mod)

  - Types
  - (Inductive) Definitions
  - Aggregates
  - Arithmetic
  - Coinductive Definitions
  - Modal operators

FO($\cdot$): turning FO into a practical KR language

- FO does not suffice for knowledge representation, modelling, specification

$\Rightarrow$ FO(Types,ID,Agg,Arit,FD,Mod,HO)

  - Types
  - (Inductive) Definitions
  - Aggregates
  - Arithmetic
  - Coinductive Definitions
  - Modal operators
  - Higher Order logic

FO($\cdot$): turning FO into a practical KR language

- FO does not suffice for knowledge representation, modelling, specification

$\Rightarrow$ FO(Types,ID,Agg,Arit,FD,Mod,HO,Caus,...)

  - Types
  - (Inductive) Definitions
  - Aggregates
  - Arithmetic
  - Coinductive Definitions
  - Modal operators
  - Higher Order logic
  - Causation
  - ...

The FO($\cdot$) language framework

The FO(·)-KBS project: an integration project

- On the inference level:
  - Building solvers for various forms of inference for FO(·)
  - Integrating various solving techniques from various declarative programming paradigms in one Knowledge Base System.

What do I mean with “inference”?

Definition

An inference problem is a computational problem:

- Input: a tuple of objects including a logic theory or formula
- Output: a set of computed objects
  - such that the output is invariant under replacing the input theory by a logically equivalent theory.

What do I mean with “inference”?

Definition

An inference problem is a computational problem:

- Input: a tuple of objects including a logic theory or formula
- Output: a set of computed objects
  - such that the output is invariant under replacing the input theory by a logically equivalent theory.

- E.g., Deduction inference
  - Input: theory T, sentence φ
  - Output : true if $T \models \varphi$

What do I mean with "inference"?

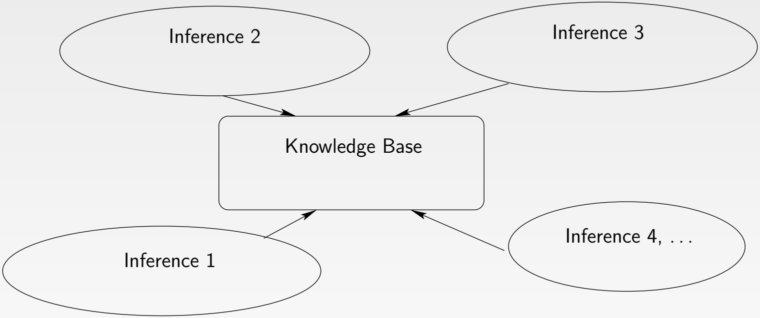
Definition

An inference problem is a computational problem:

- Input: a tuple of objects including a logic theory or formula
- Output: a set of computed objects
  - such that the output is invariant under replacing the input theory by a logically equivalent theory.

- E.g., Deduction inference
  - Input: theory T, sentence φ
  - Output : true if $T \models \varphi$
- E.g., Query inference:
  - Input structure $\mathfrak{A}$, set expression $\{\bar{x} : \varphi\}$
  - Output $\{\bar{x} : \varphi\}^{\mathfrak{A}}$

A Knowledge Base System (KBS)



- Manages a declarative Knowledge Base (KB): a theory
- Equiped with different forms of inference

The FO($\cdot$)-KBS project: an integration project

- On the application level:
  - Towards a typology of tasks and computational problems in terms of (the same) logic and inference.
  - Eagerly searching for novel ways of using declarative specifications to solve problems.

Revised Knowledge Representation and Reasoning (Work) Hypothesis

Every computational task can be naturally solved by applying the right form of inference on the natural specification of the domain knowledge.

Revised Knowledge Representation and Reasoning (Work) Hypothesis

Every computational task can be naturally solved by applying the right form of inference on the natural specification of the domain knowledge.

A idealistic view of declarative problem solving in the context of an application

- For a specific problem domain, investigate what knowledge is available, how it could be represented in a natural way, and what sort of inference would solve the problem(s).
- As opposed to: investigate how to encode a computational problem in a given computational logic.

Revised Knowledge Representation and Reasoning (Work) Hypothesis

Every computational task can be naturally solved by applying the right form of inference on the natural specification of the domain knowledge.

A idealistic view of declarative problem solving in the context of an application

- For a specific problem domain, investigate what knowledge is available, how it could be represented in a natural way, and what sort of inference would solve the problem(s).
- As opposed to: investigate how to encode a computational problem in a given computational logic.

It is a Work Hypothesis: even failure is interesting.

Collected controversies

- Logic theories are not representations of problems?

Collected controversies

- Logic theories are not representations of problems?
- Logic does not have a built-in form of inference?

Collected controversies

- Logic theories are not representations of problems?
- Logic does not have a built-in form of inference?
- Knowledge can be studied with the methods of formal empirical science?

Collected controversies

- Logic theories are not representations of problems?
- Logic does not have a built-in form of inference?
- Knowledge can be studied with the methods of formal empirical science?
- Languages with clear informal semantics? With a clear "intuitive understanding"?

Collected controversies

- Logic theories are not representations of problems?
- Logic does not have a built-in form of inference?
- Knowledge can be studied with the methods of formal empirical science?
- Languages with clear informal semantics? With a clear "intuitive understanding"?
- FO? Isn't that language that failed in the seventies?

Collected controversies

- Logic theories are not representations of problems?
- Logic does not have a built-in form of inference?
- Knowledge can be studied with the methods of formal empirical science?
- Languages with clear informal semantics? With a clear "intuitive understanding"?
- FO? Isn't that language that failed in the seventies?
- The (revised) KRR hypothesis?

Collected controversies

- Logic theories are not representations of problems?
- Logic does not have a built-in form of inference?
- Knowledge can be studied with the methods of formal empirical science?
- Languages with clear informal semantics? With a clear "intuitive understanding"?
- FO? Isn't that language that failed in the seventies?
- The (revised) KRR hypothesis?
- Ignoring the "Tractibility/Expressivity" trade-off?

- The FO($\cdot$)-KBS project is the long-term research project of the KU Leuven KRR research group.
  - An attempt to set out clear fundamental research goals (this lecture)
  - At the same time, an effort to realize some of this in a practically useful language, system, methodology (next lecture)

IDP, IDP3 and the demos's are available via our webpage
http://krr.bitbucket.org
https://dtai.cs.kuleuven.be/software/idp/
Publications are on line via my webpage
http://people.cs.kuleuven.be/~marc.denecker/