Knowledge Representation (Overview)

Marek Sergot

October 2017

Knowledge Representation*

* includes reasoning

- a huge sub-field of AI
- a variety of representation/modelling formalisms, mostly (these days, always) based on logic
- assorted representation problems

So these days, more or less: applied (computational) logic

ERIK MUELLER

COMMONSENSE REASONING

An Event Calculus Based Approach

MK

SECOND EDITION

KR 2016: 15th International Conference on Principles of Knowledge Representation and Reasoning, Cape Town

- Argumentation
- Belief revision and update, belief merging, etc.
- Commonsense reasoning
- Contextual reasoning
- Description logics
- Diagnosis, abduction, explanation
- Inconsistency- and exception tolerant reasoning, paraconsistent logics
- KR and autonomous agents: intelligent agents, cognitive robotics, multi-agent systems
- KR and data management, data analytics
- KR and decision making, game theory, social choice
- KR and machine learning, inductive logic programming, knowledge discovery and acquisition
- KR and natural language processing
- KR and the Web, Semantic Web

KR 2016: 15th International Conference on Principles of Knowledge Representation and Reasoning, Cape Town

- Logic programming, answer set programming, constraint logic programming
- Nonmonotonic logics, default logics, conditional logics
- Ontology formalisms and models
- Philosophical foundations of KR
- Preferences: modeling and representation, preference-based reasoning
- Reasoning about action and change: action languages, situation calculus, causality
- Reasoning about knowledge and belief, dynamic epistemic logic, epistemic and doxastic logics
- Reasoning systems and solvers, knowledge compilation
- Spatial and temporal reasoning, qualitative reasoning
- Uncertainty, vagueness, many-valued and fuzzy logics

KR 2016: Programme

- KR and Data Management 1
  Argumentation 1
  Short Papers: Automated Reasoning – Logic prog/inconsistency
- Temporal and Spatial Reasoning 1
  Automated Reasoning and Computation 1
  Short Papers: Reasoning about Action – Uncertainty
- Planning and Strategies
  KR and Data Management 2
- Description Logic 1
  Epistemic Reasoning 1
  Short Papers: Description Logic – Argumentation
- Automated Reasoning and Computation 2
  Decision Theory, Rationality, and Uncertainty
  KR and Data Management 3
  Belief Revision and Nonmonotonicity
- Description Logic 2
  Reasoning about Action, Causality
  Argumentation 2
  Epistemic Reasoning 2
- Argumentation 3
  Temporal and Spatial Reasoning 2

Aims of this course

- Logic
  Logic $\neq$ classical (propositional) logic !!

Aims of this course

- Logic
  Logic $\neq$ classical (propositional) logic !!

- Computational logic
  Logic programming $\neq$ Prolog !!

Aims of this course

- Logic
  Logic $\neq$ classical (propositional) logic !!
- Computational logic
  Logic programming $\neq$ Prolog !!
- Non-monotonic logics (methods and examples)

Aims of this course

- Logic
  Logic $\neq$ classical (propositional) logic !!

- Computational logic
  Logic programming $\neq$ Prolog !!

- Non-monotonic logics (methods and examples)

- Some examples
  - defeasible (non-monotonic) rules
  - action + ‘inertia’ + causality
  - priorities (preferences)
  - ‘practical reasoning’: *what should I do?*

From *SOLE* 2014 . . .

The most interesting of all the courses offered . . . My only suggestion for improvement would be to offer this course in the first term and . . .

From *SOLE* 2014 . . .

The most interesting of all the courses offered . . . My only suggestion for improvement would be to offer this course in the first term and . . .

Prof Marek Sergot is the most lucid, patient, engaging, humourous, enthusiastic and approachable lecturer one could ever hope to have. It is a privilege to encounter such a lecturer.

From *SOLE* 2014 . . .

The most interesting of all the courses offered . . . My only suggestion for improvement would be to offer this course in the first term and . . .

Prof Marek Sergot is the most lucid, patient, engaging, humourous, enthusiastic and approachable lecturer one could ever hope to have. It is a privilege to encounter such a lecturer.

This happened on several occasions and I believe it is not acceptable.

Logic of conditionals (‘if ... then ...’)

- material implication $(A \to B \; = \neg A \vee B)$
- ‘strict implication’
- causal conditionals
- counterfactuals
- conditional obligations
- defeasible (non-monotonic) conditionals

$\vdots$

Example

A recent article about the Semantic Web was critical about the use of logic for performing useful inferences in the Semantic Web, citing the following example, among others:

> *'People who live in Brooklyn speak with a Brooklyn accent. I live in Brooklyn. Yet I do not speak with a Brooklyn accent.'*

Example

A recent article about the Semantic Web was critical about the use of logic for performing useful inferences in the Semantic Web, citing the following example, among others:

> *‘People who live in Brooklyn speak with a Brooklyn accent. I live in Brooklyn. Yet I do not speak with a Brooklyn accent.’*

According to the author,

> *‘each of these statements is true, but each is true in a different way. The first is a generalization that can only be understood in context.’*

Example

A recent article about the Semantic Web was critical about the use of logic for performing useful inferences in the Semantic Web, citing the following example, among others:

> *'People who live in Brooklyn speak with a Brooklyn accent. I live in Brooklyn. Yet I do not speak with a Brooklyn accent.'*

According to the author,

> *'each of these statements is true, but each is true in a different way. The first is a generalization that can only be understood in context.'*

The article was doubtful that there are any practical ways of representing such statements.

`www.shirky.com/writings/semantic syllogism.html`.

His point (the classical syllogism)

$$\frac{\begin{array}{l}\forall x\,(p(x) \rightarrow q(x))\\ p(\mathtt{a})\end{array}}{q(\mathtt{a})}$$

In logic programming notation:

$$\frac{\begin{array}{l}q(x) \leftarrow p(x)\\ p(\mathtt{a})\end{array}}{q(\mathtt{a})}$$

Solution

We need either or both of:

- a new kind of conditional $\rightsquigarrow$
- a special kind of defeasible entailment

$$\frac{\begin{array}{l}\forall x\,(p(x) \rightsquigarrow q(x))\\ p(\mathtt{a})\end{array}}{q(\mathtt{a})}$$

There is a huge amount of work on this in AI!

This is the main technical core of the course

Non-monotonic logics

Classical logic is monotonic:

$$\text{If} \quad KB \models \alpha \ \text{ then } \ KB \cup X \models \alpha$$

New information X always preserves old conclusions α.

Non-monotonic logics

Classical logic is monotonic:

$$\text{If} \quad KB \models \alpha \;\; \text{then} \;\; KB \cup X \models \alpha$$

New information X always preserves old conclusions α.

Default reasoning is typically non-monotonic. Can have:

$$KB \models_{\Delta} \alpha \;\; \text{but} \;\; KB \cup X \not\models_{\Delta} \alpha$$

Non-monotonic logics

Classical logic is monotonic:

$$\text{If} \quad KB \models \alpha \;\; \text{then} \;\; KB \cup X \models \alpha$$

New information X always preserves old conclusions α.

Default reasoning is typically non-monotonic. Can have:

$$KB \models_{\Delta} \alpha \;\; \text{but} \;\; KB \cup X \not\models_{\Delta} \alpha$$

BIRDS $\cup$ {bird(frank)} $\models_{\Delta}$ flies(frank)

But

BIRDS $\cup$ {bird(frank)} $\cup$ {penguin(frank)} $\not\models_{\Delta}$ flies(frank)

Can Susan Vote in the US?

$$
\begin{array}{lrcll}
 & res\_Cuba & \leftarrow & & \\
 & res\_NAmerica & \leftarrow & res\_Cuba & \\
\\
\delta_1 : & cit\_US & \leftharpoonup\!\!\!\leadsto & res\_NAmerica & \delta_3 > \delta_2 > \delta_1 \\
\delta_2 : & cit\_Cuba & \leftharpoonup\!\!\!\leadsto & res\_Cuba & \\
\delta_3 : & vote\_US & \leftharpoonup\!\!\!\leadsto & cit\_US & \\
\\
 & \neg cit\_US & \leftarrow & cit\_Cuba & \% \; \neg(\, cit\_Cuba \wedge cit\_US \,) \\
 & \neg cit\_Cuba & \leftarrow & cit\_US & \\
 & \neg vote\_US & \leftarrow & cit\_Cuba & \% \; \neg(\, cit\_Cuba \wedge vote\_US \,)
\end{array}
$$

Multiple extensions: “The Nixon diamond”

- Quakers are typically pacifists.
- Republicans are typically not pacifists.

Multiple extensions: “The Nixon diamond”

- Quakers are typically pacifists.
- Republicans are typically not pacifists.
- Richard Nixon is a Quaker.
- Richard Nixon is a Republican

Is Nixon is a pacifist or not?

Defeasible conditional imperatives

$$F \rightsquigarrow\ !\alpha$$

law: $\rightsquigarrow\ !\neg(drink \land drive)$
wife: $\rightsquigarrow\ !drive$
friends: $\rightsquigarrow\ !drink$

law > *wife*   *law* > *friends*

Defeasible conditional imperatives

$$F \rightsquigarrow !\,\alpha$$

law: $\rightsquigarrow !\,\neg(drink \wedge drive)$
wife: $\rightsquigarrow !\,drive$
friends: $\rightsquigarrow !\,drink$

law > *wife*   *law* > *friends*

wife > *friends*: $\{drive, \neg drink\}$

friends > *wife*: $\{drink, \neg drive\}$

Example: a problem of practical moral reasoning

(Katie Atkinson and Trevor Bench-Capon)

Hal, a diabetic, has no insulin. Without insulin he will die.

Carla, also a diabetic, has (plenty of) insulin.

Should Hal take Carla's insulin? (Is he so justified?)

If he takes it, should he leave money to compensate?

Suppose Hal *does not know* whether Carla needs all her insulin.
Is he still justified in taking it?
Should he compensate her?

(Why?)

Hal, Carla and Dave

$$
\begin{aligned}
&has\_insulin(Carla) \\
&has\_insulin(Dave) \\
&diabetic(Dave)
\end{aligned}
$$

$$
\begin{aligned}
has\_insulin(X) \rightsquigarrow\ !\,take\_from(X) &\quad :: \quad life(Hal) \\
\rightsquigarrow\ !\,\neg take\_from(X) &\quad :: \quad property(X) \\
diabetic(X) \rightsquigarrow\ !\,\neg take\_from(X) &\quad :: \quad life(X) \\
!take\_from(X) \rightsquigarrow\ !\,pay(X) &\quad :: \quad property(X) \\
\rightsquigarrow\ !\,\neg pay(X) &\quad :: \quad property(Hal)
\end{aligned}
$$

$$
\begin{aligned}
\neg take\_from(X) &\leftarrow not\ \ has\_insulin(X) \\
\neg take\_from(X) &\leftarrow take\_from(Y),\ X \neq Y
\end{aligned}
$$

Hal, Carla and Dave

$$has\_insulin(Carla)$$
$$has\_insulin(Dave)$$
$$diabetic(Dave)$$

Altruistic Hal: $life(X) > life(Hal) > property(Y) > property(Hal)$

$$\{\neg take\_from(Dave), take\_from(Carla), \neg pay(Dave), pay(Carla)\}$$

Hal, Carla and Dave

$$has\_insulin(Carla)$$
$$has\_insulin(Dave)$$
$$diabetic(Dave)$$

Altruistic Hal: $life(X) > life(Hal) > property(Y) > property(Hal)$

$$\{\neg take\_from(Dave), take\_from(Carla), \neg pay(Dave), pay(Carla)\}$$

Selfish Hal: $life(Hal) > life(X) > property(Hal) > property(Y)$

$$\{\neg take\_from(Dave), take\_from(Carla), \neg pay(Dave), \neg pay(Carla)\}$$
$$\{take\_from(Dave), \neg take\_from(Carla), \neg pay(Dave), \neg pay(Carla)\}$$

Hal, Carla and Dave

$$has\_insulin(Carla)$$
$$has\_insulin(Dave)$$
$$diabetic(Dave)$$

Altruistic Hal: $life(X) > life(Hal) > property(Y) > property(Hal)$

$$\{\neg take\_from(Dave), take\_from(Carla), \neg pay(Dave), pay(Carla)\}$$

Selfish Hal: $life(Hal) > life(X) > property(Hal) > property(Y)$

$$\{\neg take\_from(Dave), take\_from(Carla), \neg pay(Dave), \neg pay(Carla)\}$$
$$\{take\_from(Dave), \neg take\_from(Carla), \neg pay(Dave), \neg pay(Carla)\}$$

Callous Hal: $life(Hal) > property(Hal) > life(X) > property(Y)$

$$\{\neg take\_from(Dave), take\_from(Carla), \neg pay(Dave), \neg pay(Carla)\}$$
$$\{take\_from(Dave), \neg take\_from(Carla), \neg pay(Dave), \neg pay(Carla)\}$$

Some sources of defeasible reasoning

- Typical and stereotypical situations
- Generalisations and exceptions

⋮

The Qualification Problem (1)

"All birds can fly . . . "

$\text{flies}(X) \leftarrow \text{bird}(X)$

The Qualification Problem (1)

"All birds can fly . . . "

$\text{flies}(X) \leftarrow \text{bird}(X)$

". . . unless they are penguins . . . "

$\text{flies}(X) \leftarrow \text{bird}(X), \neg\ \text{penguin}(X)$

The Qualification Problem (1)

"All birds can fly ..."

$flies(X) \leftarrow bird(X)$

"... unless they are penguins ..."

$flies(X) \leftarrow bird(X), \neg\, penguin(X)$

"... or ostriches ..."

$flies(X) \leftarrow bird(X), \neg\, penguin(X), \neg\, ostrich(X)$

The Qualification Problem (1)

"All birds can fly . . ."

$\text{flies}(X) \leftarrow \text{bird}(X)$

". . . unless they are penguins . . ."

$\text{flies}(X) \leftarrow \text{bird}(X), \neg\ \text{penguin}(X)$

". . . or ostriches . . ."

$\text{flies}(X) \leftarrow \text{bird}(X), \neg\ \text{penguin}(X), \neg\ \text{ostrich}(X)$

". . . or wounded . . ."

$\text{flies}(X) \leftarrow \text{bird}(X), \neg\ \text{penguin}(X), \neg\ \text{ostrich}(X), \neg\ \text{wounded}(X)$

The Qualification Problem (1)

"All birds can fly . . ."

$\text{flies}(X) \leftarrow \text{bird}(X)$

". . . unless they are penguins . . ."

$\text{flies}(X) \leftarrow \text{bird}(X), \neg\ \text{penguin}(X)$

". . . or ostriches . . ."

$\text{flies}(X) \leftarrow \text{bird}(X), \neg\ \text{penguin}(X), \neg\ \text{ostrich}(X)$

". . . or wounded . . ."

$\text{flies}(X) \leftarrow \text{bird}(X), \neg\ \text{penguin}(X), \neg\ \text{ostrich}(X), \neg\ \text{wounded}(X)$

". . . or dead, or sick, or glued to the ground, or . . ."

The Qualification Problem (2)

Let BIRDS be the set of rules about flying birds.
Even if we could list all these exceptions, classical logic would still not allow

$$\text{BIRDS} \cup \{\text{bird(frank)}\} \models \text{flies(frank)}$$

The Qualification Problem (2)

Let BIRDS be the set of rules about flying birds.
Even if we could list all these exceptions, classical logic would still not allow

$$\text{BIRDS} \cup \{\text{bird(frank)}\} \models \text{flies(frank)}$$

We would also have to affirm all the *qualifications*:

The Qualification Problem (2)

Let BIRDS be the set of rules about flying birds.
Even if we could list all these exceptions, classical logic would still not allow

$$\text{BIRDS} \cup \{\text{bird(frank)}\} \models \text{flies(frank)}$$

We would also have to affirm all the *qualifications*:

$\neg$ penguin(frank)
$\neg$ ostrich(frank)
$\neg$ wounded(frank)
$\neg$ dead(frank)
$\neg$ sick(frank)
$\neg$ glued_to_ground(frank)
$\vdots$

Some sources of defeasible reasoning

- Typical and stereotypical situations
- Generalisations and exceptions
- Conventions of communication
  - ‘Closed World Assumptions’
  - ‘Circumscription’
- Autoepistemic reasoning (reasoning about your own beliefs)
- Burdens of proof (e.g. in legal reasoning)
- Persistence and change in temporal reasoning

⋮

Temporal reasoning: The Frame Problem

Actions change the truth value of some facts, but almost everything else remains unchanged.

Painting my house pink changes the colour of the house to pink . . .

but does not change:

Temporal reasoning: The Frame Problem

Actions change the truth value of some facts, but almost everything else remains unchanged.

> *Painting my house pink changes the colour of the house to pink . . .*

but does not change:

the age of my house is 93 years

Temporal reasoning: The Frame Problem

Actions change the truth value of some facts, but almost everything else remains unchanged.

> *Painting my house pink changes the colour of the house to pink . . .*

but does not change:

the age of my house is 93 years
the father of Brian is Bill

Temporal reasoning: The Frame Problem

Actions change the truth value of some facts, but almost everything else remains unchanged.

Painting my house pink changes the colour of the house to pink . . .

but does not change:

the age of my house is 93 years
the father of Brian is Bill
the capital of France is Paris

⋮

Temporal reasoning: The Frame Problem

Actions change the truth value of some facts, but almost everything else remains unchanged.

> *Painting my house pink changes the colour of the house to pink . . .*

but does not change:

the age of my house is 93 years
the father of Brian is Bill
the capital of France is Paris

⋮

Qualification problems!

Temporal reasoning: Default Persistence (‘Inertia’)

Actions change the truth value of some facts, but almost everything else remains unchanged.

$$p[t] \rightsquigarrow p[t+1]$$

Some facts persist ‘by inertia’, until disturbed by some action.

Closely connected to forms of causality

Temporal reasoning: Ramifications

win causes *rich*

Temporal reasoning: Ramifications

win causes *rich*
lose causes $\neg$*rich*

Temporal reasoning: Ramifications

win causes *rich*
lose causes $\neg$*rich*
rich $\Rightarrow$ *happy*

Temporal reasoning: Ramifications

win causes *rich*
lose causes $\neg$*rich*
rich $\Rightarrow$ *happy*

So an occurrence of *win* indirectly causes *happy*.

Material implication

Everyone in Ward 16 has cancer.

$$\forall x\,(\,in\_ward\_16(x) \;\rightarrow\; has\_cancer(x)\,)$$

But compare:

Material implication

Everyone in Ward 16 has cancer.

$$\forall x\,(\,in\_ward\_16(x) \;\rightarrow\; has\_cancer(x)\,)$$

But compare:

$$\forall x\,(\,in\_ward\_16(x) \;\Rightarrow\; has\_cancer(x)\,)$$

Being in Ward 16 causes you to have cancer.
x has cancer because x is in Ward 16.

The ‘paradoxes of material implication’

- $A \rightarrow (B \rightarrow A)$
- $\neg A \rightarrow (A \rightarrow B)$
- $(\neg A \wedge A) \rightarrow B$

- $((A \wedge B) \rightarrow C) \rightarrow ((A \rightarrow C) \vee (B \rightarrow C))$

- $(A \rightarrow B) \vee (B \rightarrow A)$

Logic of conditionals (‘if ... then ...’)

- material implication (classical $\rightarrow$)
- ‘strict implication’
- intuitionistic implication
- causal conditionals
- counterfactuals
- conditional obligations
- defeasible (non-monotonic) conditionals

$\vdots$

A favourite topic — action

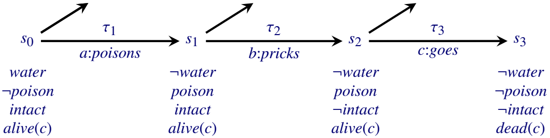
Action

- state change/transition
- agency + causality
- what is it 'to act'?

'Actual cause'

- something happened
- *who* caused it?
- *what* caused it?

Agency: an example of 'proximate cause'



J.A. McLaughlin. Proximate Cause. *Harvard Law Review* 39(2):149–199 (Dec. 1925)

Aims

- Logic
  Logic $\neq$ classical (propositional) logic !!
- Computational logic
  Logic programming $\neq$ Prolog !!
- Non-monotonic logics (core methods and examples)
- Some examples
  (temporal reasoning, action + causality, 'practical reasoning', ...)

Contents (not necessarily in this order)

- Logic: models, theories, consequence relations
- Logic databases/knowledge bases (in general)
- Defeasible reasoning, defaults, non-monotonic logics, non-monotonic consequence
- Some specific non-monotonic formalisms
  - normal logic programs, extended logic programs, Reiter default logic, . . . , 'nonmonotonic causal theories', . . . Answer Set Programming
  - priorities and preferences
- Temporal reasoning: action, change, persistence (and various related concepts)
- If time permits, examples from
  - 'practical reasoning', action, norms . . .
  - more about priorities and preferences

Assumed knowledge

- Basic logic: syntax and semantics; propositional and first-order logic.
- Elementary set theory
- Basic logic programming: syntax and semantics, inference and procedural readings (Prolog), negation as failure — helpful but not essential
- Previous AI course(s) — definitely not essential.

Assumed knowledge

- Basic logic: syntax and semantics; propositional and first-order logic.
- Elementary set theory
- Basic logic programming: syntax and semantics, inference and procedural readings (Prolog), negation as failure — helpful but not essential
- Previous AI course(s) — definitely not essential.

There is no (compulsory) practical lab work — though you are encouraged to implement/run the various examples that come up.

Assumed knowledge

- Basic logic: syntax and semantics; propositional and first-order logic.
- Elementary set theory
- Basic logic programming: syntax and semantics, inference and procedural readings (Prolog), negation as failure — helpful but not essential
- Previous AI course(s) — definitely not essential.

There is no (compulsory) practical lab work — though you are encouraged to implement/run the various examples that come up.

Recommended reading

References for specific topics will be given in the notes.

For background: any standard textbook on AI (not esssential)

Other possible topics, not covered in this course

- Assorted rule-based formalisms; procedural representations
- Structured representations (1) — old fashioned (frames, semantic nets, conceptual graphs), and their new manifestations
- Structured representations (2) — VERY fashionable
  - description logics (previously 'terminological logics')
    See e.g: `http://www.dl.kr.org`
- "Ontologies"
  - Develop 'ontology' for application X and world-fragment Y.
  - 'Ontology' as used in AI means 'conceptual framework'.

Other possible topics, not covered in this course

- Goals, plans, mentalistic structures (belief, desire, intention, . . .)
  - associated in particular with multi-agent systems.
- Belief system dynamics: belief revision – no time
- Argumentation
- Probabilistic approaches (various)

Some of these topics are covered in other MEng/MAC courses.

Description logic (example)

Bavaria $\sqsubseteq$ Germany
Person
Lager $\sqsubseteq$ Beer
Sam: Person

Description logic (example)

Bavaria $\sqsubseteq$ Germany
Person
Lager $\sqsubseteq$ Beer
Sam : Person

Person *drinks* Beer
Person *lives_in* Germany

Description logic (example)

Bavaria $\sqsubseteq$ Germany
Person
Lager $\sqsubseteq$ Beer
Sam: Person

Person *drinks* Beer
Person *lives_in* Germany

Person $\sqcap$ $\exists$*lives_in*.Bavaria
Sam: Person $\sqcap$ $\exists$*lives_in*.Bavaria

Description logic (example)

$\text{Bavaria} \sqsubseteq \text{Germany}$
Person
$\text{Lager} \sqsubseteq \text{Beer}$
$\mathit{Sam}\colon \text{Person}$

$\text{Person}\ \mathit{drinks}\ \text{Beer}$
$\text{Person}\ \mathit{lives\_in}\ \text{Germany}$

$\text{Person} \sqcap \exists \mathit{lives\_in}.\text{Bavaria}$
$\mathit{Sam}\colon \text{Person} \sqcap \exists \mathit{lives\_in}.\text{Bavaria}$

$\text{Person} \sqcap \exists \mathit{lives\_in}.\text{Bavaria} \quad \sqsubseteq \quad \text{Person} \sqcap \forall \mathit{drinks}.\text{Lager}$

Description logic (example)

$\text{Bavaria} \sqsubseteq \text{Germany}$
Person
$\text{Lager} \sqsubseteq \text{Beer}$
$\mathit{Sam} \colon \text{Person}$

Person *drinks* Beer
Person *lives_in* Germany

$\text{Person} \sqcap \exists \mathit{lives\_in}.\text{Bavaria}$
$\mathit{Sam} \colon \text{Person} \sqcap \exists \mathit{lives\_in}.\text{Bavaria}$

$$\text{Person} \sqcap \exists \mathit{lives\_in}.\text{Bavaria} \quad \sqsubseteq \quad \text{Person} \sqcap \forall \mathit{drinks}.\text{Lager}$$

Conclude:
$\mathit{Sam} \colon \text{Person} \sqcap \forall \mathit{drinks}.\text{Lager}$