Knowledge Representation
(Overview)

Marek Sergot

October 2014
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 more or less: applied logic
Exception tolerant and inconsistency-tolerant reasoning, Paraconsistent logics
Nonmonotonic logics, Default Logics, Conditional logics, Argumentation
Temporal reasoning and spatial reasoning
Causal reasoning, Abduction, Model-based diagnosis
Reasoning about actions and change, Action languages
Reasoning, planning, or decision making under uncertainty
Representations of vagueness, Many-valued and fuzzy logics
Graphical representations for belief and preference
Reasoning about belief and knowledge, Epistemic and Doxastic logics, Multi-agent logics of belief and knowledge
Logic programming, Constraint logic programming, Answer set programming
Computational aspects of knowledge representation
KR 2008: 11th International Conference on Principles of Knowledge Representation and Reasoning

- Concept formation, Similarity-based reasoning
- Belief revision and update, Belief merging, Information fusion
- Description logics, ontologies
- Qualitative reasoning, Reasoning about physical systems
- Decision theory, Preference modeling and representation, Reasoning about preference,
- KR and Autonomous agents: Intelligent agents, Cognitive robotics
- KR and Multi-agent systems: Negotiation, Group decision making, Cooperation, Interaction, KR and game theory
- Natural language processing, Summarization, Categorization
- KR and Machine learning, Inductive logic programming, Knowledge discovery and acquisition
- WWW querying languages, Information retrieval and web mining, Website selection and configuration
- Philosophical foundations and psychological evidence
KR 2014: 14th International Conference on Principles of Knowledge Representation and Reasoning

- Applications of KR
- Argumentation
- Belief revision and update, belief merging, information fusion
- Computational aspects of knowledge representation
- Concept formation, similarity-based reasoning
- Contextual reasoning
- Description logics
- Explanation finding, diagnosis, causal reasoning, abduction
- Inconsistency- and exception tolerant reasoning, paraconsistent logics
- KR and autonomous agents, cognitive robotics, multi-agent systems, logical models of agency
- KR and data management, ontology-based data access, queries and updates over incomplete data
- KR and decision making, decision theory, game theory and economic models
- KR and machine learning, inductive logic programming, knowledge discovery and acquisition
- KR and the Web, Semantic Web, formal approaches to knowledge bases
- KR in games, general game playing, reasoning in video games and virtual environments, believable agents
- KR in natural language understanding and question answering
- KR in image and video understanding
Logical approaches to planning and behavior synthesis
- Logic programming, answer set programming, constraint logic programming
- Nonmonotonic logics, default logics, conditional logics
- Reasoning about norms and organizations, social knowledge and behavior
- Philosophical foundations of KR
- Ontology languages and modeling
- Preference modeling and representation, reasoning about preferences, preference-based reasoning
- Qualitative reasoning, reasoning about physical systems
- Reasoning about actions and change, action languages, situation calculus, dynamic logic
- Reasoning about knowledge and belief, epistemic and doxastic logics
- Spatial reasoning and temporal reasoning
- Uncertainty, representations of vagueness, many-valued and fuzzy logics, relational probability models
KR 2012: Programme

- Decision Making and Reasoning about Preferences
- Short: – Description Logics – Belief Revision – Argumentation – Reasoning and Search – Logic Programming – Principled Applications
- Uncertainty
- Description Logics I
- Answer Set Programming and Logic Programming
- Inconsistency-Tolerant and Similarity-Based Reasoning
- Knowledge Representation and Knowledge-Based Systems
- Epistemic Logics
- Automated Reasoning and Computation
- Belief Revision II
- Abstraction and Diagnosis
- Argumentation
- Spatial and Temporal Reasoning
- Description Logics II
- Reasoning about Actions and Planning
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 !!

- Some examples
  - defeasible (non-monotonic) rules
  - priorities (preferences)
  - action + ‘inertia’
  - ‘practical reasoning’: what should I do?
Logic of conditionals (‘if ... then ...’)

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

∀x (p(x) → q(x))

\[ \frac{p(a)}{q(a)} \]

In logic programming notation:

\[ q(x) ← p(x) \]

\[ \frac{p(a)}{q(a)} \]
We need either or both of:

- a new kind of conditional
- a special kind of defeasible entailment

\[ \forall x \ (p(x) \rightsquigarrow q(x)) \]
\[ p(a) \]
\[ \therefore q(a) \]

There is a huge amount of work on this in AI!
The Qualification Problem (1)

“All birds can fly . . .”
flies(X) ← bird(X)
The Qualification Problem (1)

“All birds can fly . . .”
flies(X) ← bird(X)

“. . . unless they are penguins . . .”
flies(X) ← bird(X), ¬ penguin(X)
The Qualification Problem (1)

“All birds can fly . . .”
\[
f_\text{flies}(X) \leftarrow \text{bird}(X)
\]

“. . . unless they are penguins . . .”
\[
f_\text{flies}(X) \leftarrow \text{bird}(X), \neg \text{penguin}(X)
\]

“. . . or ostriches . . .”
\[
f_\text{flies}(X) \leftarrow \text{bird}(X), \neg \text{penguin}(X), \neg \text{ostrich}(X)
\]
The Qualification Problem (1)

“All birds can fly . . .”
   flies(X) ← bird(X)

“. . . unless they are penguins . . .”
   flies(X) ← bird(X), ¬ penguin(X)

“. . . or ostriches . . .”
   flies(X) ← bird(X), ¬ penguin(X), ¬ ostrich(X)

“. . . or wounded . . .”
   flies(X) ← bird(X), ¬ penguin(X), ¬ ostrich(X), ¬ wounded(X)
The Qualification Problem (1)

“All birds can fly . . .”
flies(X) ← bird(X)

“. . . unless they are penguins . . .”
flies(X) ← bird(X), ¬ penguin(X)

“. . . or ostriches . . .”
flies(X) ← bird(X), ¬ penguin(X), ¬ ostrich(X)

“. . . or wounded . . .”
flies(X) ← bird(X), ¬ penguin(X), ¬ ostrich(X), ¬ wounded(X)

“. . . or dead, or sick, or glued to the ground, or . . .”
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:

\[ \neg \text{penguin(frank)} \]
\[ \neg \text{ostrich(frank)} \]
\[ \neg \text{wounded(frank)} \]
\[ \neg \text{dead(frank)} \]
\[ \neg \text{sick(frank)} \]
\[ \neg \text{glued\_to\_ground(frank)} \]
\[ \vdots \]
Classical logic is inadequate

\[ A \models \alpha \] means that \( \alpha \) is true in all models of \( A \).
$A \models \alpha$ means that $\alpha$ is true in all models of $A$.

We want a new kind of ‘entailment’:

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

From BIRDS and bird(frank) it follows by default—in the absence of information to the contrary—that flies(frank).
Classical logic is inadequate

$A \models \alpha$ means that $\alpha$ is true in all models of $A$.

We want a new kind of ‘entailment’:

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

From BIRDS and bird(frank) it follows by default—in the absence of information to the contrary—that flies(frank).

This kind of reasoning will be defeasible.
Non-monotonic logics

Classical logic is **monotonic**:

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

New information $X$ always preserves old conclusions $\alpha$. 
Non-monotonic logics

Classical logic is monotonic:

\[ \text{If } KB \models \alpha \text{ then } KB \cup X \models \alpha \]

New information \( X \) always preserves old conclusions \( \alpha \).

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**:

\[
\text{If } KB \models \alpha \text{ then } KB \cup X \models \alpha
\]

New information \( X \) always preserves old conclusions \( \alpha \).

Default reasoning is typically **non-monotonic**. Can have:

\[
KB \models_{\Delta} \alpha \quad \text{but} \quad KB \cup X \not\models_{\Delta} \alpha
\]

\[
\text{BIRDS } \cup \{ \text{bird(frank)} \} \models_{\Delta} \text{flies(frank)}
\]

But

\[
\text{BIRDS } \cup \{ \text{bird(frank)} \} \cup \{ \text{penguin(frank)} \} \not\models_{\Delta} \text{flies(frank)}
\]
Non-monotonic logics

Classical logic is monotonic:

\[ \text{If } KB \models \alpha \text{ then } KB \cup X \models \alpha \]

New information \( X \) always preserves old conclusions \( \alpha \).

Default reasoning is typically non-monotonic. Can have:

\[ KB \models_{\Delta} \alpha \text{ but } KB \cup X \not\models_{\Delta} \alpha \]

\[ \text{BIRDS } \cup \{ \text{bird(frank)} \} \models_{\Delta} \text{flies(frank)} \]

But

\[ \text{BIRDS } \cup \{ \text{bird(frank)} \} \cup \{ \text{penguin(frank)} \} \not\models_{\Delta} \text{flies(frank)} \]

There have been huge developments in AI over the last 25 years in non-monotonic logics for default reasoning and other applications.
Three main approaches

1 ‘Preferential entailment’

\( KB \models_\Delta \alpha \) — \( \alpha \) is true in all preferred models of \( KB \)

The ‘preferred’ models are usually minimal in some appropriate sense.
Three main approaches

1. ‘Preferential entailment’
   \[ KB \models_{\Delta} \alpha \] — \( \alpha \) is true in all preferred models of \( KB \)
   The ‘preferred’ models are usually minimal in some appropriate sense.

2. Define \( \models_{\Delta} \) in terms of classical consequence \( \models \).
   Usually:
   \[ KB \models_{\Delta} \alpha \iff KB \cup ext(KB) \models \alpha \]
Three main approaches

1. ‘Preferential entailment’

\[ KB \models_\Delta \alpha \quad \text{— } \alpha \text{ is true in all preferred models of } KB \]

The ‘preferred’ models are usually minimal in some appropriate sense.

2. Define \( \models_\Delta \) in terms of classical consequence \( \models \). Usually:

\[ KB \models_\Delta \alpha \quad \text{iff} \quad KB \cup ext(KB) \models \alpha \]

Non-monotonic because, in general:

\[ KB \cup ext(KB) \not\subseteq KB \cup X \cup ext(KB \cup X) \]
Three main approaches

1. ‘Preferential entailment’

\[ KB \models_{\Delta} \alpha \] — \( \alpha \) is true in all preferred models of \( KB \)

The ‘preferred’ models are usually minimal in some appropriate sense.

2. Define \( \models_{\Delta} \) in terms of classical consequence \( \models \). Usually:

\[ KB \models_{\Delta} \alpha \text{ iff } KB \cup ext(KB) \models \alpha \]

Non-monotonic because, in general:

\[ KB \cup ext(KB) \nsubseteq KB \cup X \cup ext(KB \cup X) \]

3. Extend the language with a new form of defeasible conditional/rule (e.g. \( \rightsquigarrow \)).
Three main approaches

1. ‘Preferential entailment’

\[ KB \models_\Delta \alpha \quad \text{— \ alpha is true in all preferred models of} \ KB \]

The ‘preferred’ models are usually minimal in some appropriate sense.

2. Define \( \models_\Delta \) in terms of classical consequence \( \models \). Usually:

\[ KB \models_\Delta \alpha \iff KB \cup ext(KB) \models \alpha \]

Non-monotonic because, in general:

\[ KB \cup ext(KB) \not\subseteq KB \cup X \cup ext(KB \cup X) \]

3. Extend the language with a new form of defeasible conditional/rule (e.g. \( \rightsquigarrow \)).

There are strong relationships between these three methods. We will see some examples of each.
Some sources of defeasible reasoning

- Typical and stereotypical situations
- Generalisations and exceptions
Example

- Typically (by default, unless there is reason to think otherwise, ...) a bird can fly.
Example

- Typically (by default, unless there is reason to think otherwise, ...) a bird can fly.
- Except that ostriches, which are birds, typically cannot fly.
Example

- Typically (by default, unless there is reason to think otherwise, ...) a bird can fly.
- Except that ostriches, which are birds, typically cannot fly.
- Also penguins, which are birds, cannot fly.
Example

- Typically (by default, unless there is reason to think otherwise, ...) a bird can fly.
- Except that ostriches, which are birds, typically cannot fly.
- Also penguins, which are birds, cannot fly.
- Except that magic ostriches can fly (in general).
Example

- Typically (by default, unless there is reason to think otherwise, ...) a bird can fly.
- Except that ostriches, which are birds, typically cannot fly.
- Also penguins, which are birds, cannot fly.
- Except that magic ostriches can fly (in general).
- Jim is an ostrich (an ordinary ostrich) who can fly.
Example

- Typically (by default, unless there is reason to think otherwise, ...) a bird can fly.
- Except that ostriches, which are birds, typically cannot fly.
- Also penguins, which are birds, cannot fly.
- Except that magic ostriches can fly (in general).
- Jim is an ostrich (an ordinary ostrich) who can fly.
- Frank is a magic ostrich who cannot fly.
Example

- Typically (by default, unless there is reason to think otherwise, …) a bird can fly.
- Except that ostriches, which are birds, typically cannot fly.
- Also penguins, which are birds, cannot fly.
- Except that magic ostriches can fly (in general).
- Jim is an ostrich (an ordinary ostrich) who can fly.
- Frank is a magic ostrich who cannot fly.
- No bird can fly if it is dead (no exceptions!).
Furthermore . . .

A dead bird is abnormal from the point of view of flying (and singing) but not necessarily from the point of view of having feathers.

- Birds have feathers.
Furthermore...

A dead bird is abnormal from the point of view of flying (and singing) but not necessarily from the point of view of having feathers.

- Birds have feathers.
- Birds who have no feathers cannot fly.
A dead bird is abnormal from the point of view of flying (and singing) but not necessarily from the point of view of having feathers.

- Birds have feathers.
- Birds who have no feathers cannot fly.
- Frank, the magic ostrich, has no feathers (and cannot fly).
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.
Multiple extensions: “The Nixon diamond”

- Quakers are typically pacifists.
- Republicans are typically not pacifists.
- Richard Nixon is a Quaker.
- Richard Nixon is a Republican.
Multiple extensions: “The Nixon diamond”

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

Do we conclude that Nixon is a pacifist or not?
Another example

- Alcoholics are typically adults.
  alcoholic $\leadsto$ adult
Another example

- Alcoholics are typically adults.
  alcoholic $\rightsquigarrow$ adult

- Adults are typically healthy.
  adult $\rightsquigarrow$ healthy
Another example

- Alcoholics are typically adults.
  alcoholic $\rightsquigarrow$ adult

- Adults are typically healthy.
  adult $\rightsquigarrow$ healthy

Do we conclude?
Another example

- Alcoholics are typically adults.
  alcoholic $\rightsquigarrow$ adult

- Adults are typically healthy.
  adult $\rightsquigarrow$ healthy

Do we conclude?

- Alcoholics are typically healthy.
  alcoholic $\rightsquigarrow$ healthy
On the other hand ...

- Alcoholics are typically adults.
  
alcoholic ⇝ adult
On the other hand ... 

- Alcoholics are typically adults.
  \[ \text{alcoholic} \leadsto \text{adult} \]

- Adults are not children.
  \[ \text{adult} \rightarrow \neg \text{child} \]
On the other hand ...

- Alcoholics are typically adults.
  alcoholic \(\rightsquigarrow\) adult

- Adults are not children.
  adult \(\rightarrow\) \(\neg\) child

Do we conclude?
On the other hand …

- Alcoholics are typically adults.
  \[ \text{alcoholic} \Rightarrow \text{adult} \]

- Adults are not children.
  \[ \text{adult} \Rightarrow \neg \text{child} \]

Do we conclude?

- Alcoholics are typically not children.
  \[ \text{alcoholic} \Rightarrow \neg \text{child} \]
Some sources of defeasible reasoning

- Typical and stereotypical situations
- Generalisations and exceptions
Some sources of defeasible reasoning

- Typical and stereotypical situations
- Generalisations and exceptions
- Conventions of communication
  - ‘Closed World Assumptions’
  - ‘Circumscription’
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)
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)
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
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:
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!
Actions change the truth value of some facts, but almost everything else remains unchanged.

\[ p[t] \leadsto p[t + 1] \]

Some facts persist ‘by inertia’, until disturbed by some action.
Temporal reasoning: Ramifications

$\text{win}$ causes $\text{rich}$
Temporal reasoning: Ramifications

\[ \text{win} \text{ causes } \text{rich} \]
\[ \text{lose} \text{ causes } \neg \text{rich} \]
Temporal reasoning: Ramifications

\[ \text{win causes rich} \]
\[ \text{lose causes } \neg \text{rich} \]
\[ \text{rich } \Rightarrow \text{ happy} \]
Temporal reasoning: Ramifications

\[
\begin{align*}
\text{win} & \text{ causes } \text{rich} \\
\text{lose} & \text{ causes } \neg \text{rich} \\
\text{rich} & \Rightarrow \text{happy}
\end{align*}
\]

So an occurrence of \textit{win} \textit{indirectly} causes \textit{happy}.
Temporal reasoning: Ramifications

\begin{align*}
\text{win} & \text{ causes } \text{rich} \\
\text{lose} & \text{ causes } \neg \text{rich} \\
\text{rich} & \implies \text{happy}
\end{align*}

So an occurrence of \text{win} \textit{indirectly} causes \text{happy}.

(We still have to figure out how to deal with $\implies$. Material implication $\to$ is too weak. It doesn't represent causal connections.)
Material implication

Everyone in Ward 16 has cancer.

\[ \forall x \ (\text{in\_ward\_16}(x) \rightarrow \text{has\_cancer}(x)) \]

But compare:
Everyone in Ward 16 has cancer.

\[ \forall x \ ( \text{in\_ward\_16}(x) \rightarrow \text{has\_cancer}(x)) \]

But compare:

\[ \forall x \ ( \text{in\_ward\_16}(x) \Rightarrow \text{has\_cancer}(x)) \]

Being in Ward 16 causes you to have cancer.

\( x \) has cancer because \( x \) is in Ward 16.
The ‘paradoxes of material implication’

- \( A \rightarrow (B \rightarrow A) \)
- \( \neg A \rightarrow (A \rightarrow B) \)
- \( (\neg A \land A) \rightarrow B \)

- \( ((A \land B) \rightarrow C) \rightarrow ((A \rightarrow C) \lor (B \rightarrow C)) \)

- \( (A \rightarrow B) \lor (B \rightarrow A) \)
Logic of conditionals (‘if ... then ...’)

- material implication (classical $\rightarrow$)
- ‘strict implication’
- causal conditionals
- counterfactuals
- conditional obligations
- defeasible (non-monotonic) conditionals
A favourite topic — action

Action

- state change/transition
- agency
- what is it ‘to act’?

- logic
- computer science
- philosophy
Agency: an example of ‘proximate cause’

Example: a problem of ‘practical’ moral reasoning

(Atkinson & Bench-Capon)

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

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

Should Hal take (steal) Carla’s insulin? (Is he so justified?)
Example: a problem of ‘practical’ moral reasoning

(Atkinson & Bench-Capon)

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

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

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

If he takes it, should he leave money to compensate?
Example: a problem of ‘practical’ moral reasoning

(Atkinson & Bench-Capon)

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

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

Should Hal take (steal) 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?)
Defeasible rules (reasons to believe):

\[
\begin{align*}
\text{winter} \leadsto & \text{heating\_on} \\
\text{sunny} \leadsto & \text{window\_open} \\
\text{heating\_on} \leadsto & \neg \text{window\_open} \\
\text{warm} \leadsto & \neg \text{heating\_on} \\
\text{winter} \leadsto & \neg \text{warm}
\end{align*}
\]

\[
\text{winter} : \{ \neg \text{warm}, \text{heating\_on}, \neg \text{window\_open} \}\]
Defeasible rules (reasons to believe):

\[
\begin{align*}
\text{winter} \leadsto & \text{heating\_on} \\
\text{sunny} \leadsto & \text{window\_open} \\
\text{heating\_on} \leadsto & \neg \text{window\_open} \\
\text{warm} \leadsto & \neg \text{heating\_on} \\
\text{winter} \leadsto & \neg \text{warm}
\end{align*}
\]

\[
\begin{align*}
\text{winter} : & \{\neg \text{warm}, \text{heating\_on}, \neg \text{window\_open}\} \\
\text{winter, sunny} : & \{\neg \text{warm}, \text{heating\_on}, \neg \text{window\_open}\} \\
& \{\neg \text{warm}, \neg \text{heating\_on}, \text{window\_open}\}
\end{align*}
\]
Add priorities:

\[
\begin{align*}
\text{winter} & \rightsquigarrow \text{heating\_on} & \text{lowest} \\
\text{sunny} & \rightsquigarrow \text{window\_open} \\
\text{heating\_on} & \rightsquigarrow \neg\text{window\_open} \\
\text{warm} & \rightsquigarrow \neg\text{heating\_on} \\
\text{winter} & \rightsquigarrow \neg\text{warm} & \text{highest}
\end{align*}
\]

\[
\begin{align*}
\text{winter} : & \quad \{ \neg\text{warm}, \text{heating\_on}, \neg\text{window\_open} \} \\
\text{winter}, \text{sunny} : & \quad \{ \neg\text{warm}, \text{heating\_on}, \neg\text{window\_open} \} \\
& \quad \{ \neg\text{warm}, \neg\text{heating\_on}, \text{window\_open} \}
\end{align*}
\]
Defeasible conditional imperatives

Colonel:  \( \sim \neg(heat\_on \land open\_window) \)

Major:  \( \text{sunny} \sim open\_window \)

Sergeant:  \( \text{winter} \sim heat\_on \)

\( ^1 \)equivalent to  \( \sim (heat\_on \rightarrow \neg open\_window) \);
which is not the same as  \( heat\_on \sim \neg open\_window \)
Defeasible conditional imperative

Colonel: \( \sim \neg (heat\_on \land open\_window) \)

Major: sunny \( \sim\) open_window

Sergeant: winter \( \sim\) heat_on

---

winter: \{heat_on, \neg open\_window\}

\(^1\)equivalent to \( \sim (heat\_on \rightarrow \neg open\_window) \); which is not the same as heat_on \( \sim \neg open\_window \)
Defeasible conditional imperative

**Colonel:**  \( \sim \neg (heat\_on \land open\_window) \)

**Major:**  \( sunny \sim open\_window \)

**Sergeant:**  \( winter \sim heat\_on \)

---

**winter:**  \( \{heat\_on, \neg open\_window\} \)

**winter, sunny:**  \( \{heat\_on, \neg open\_window\} \)

**winter, sunny:**  \( \{\neg heat\_on, open\_window\} \)

**Major annoyed!**

**Sergeant annoyed!**

---

\(^1\)equivalent to  \( \sim (heat\_on \rightarrow \neg open\_window) \); which is not the same as  \( heat\_on \sim \neg open\_window \)
Example: party

\[
\begin{align*}
\text{law:} & \quad \leadsto \neg (\text{drink} \land \text{drive}) \\
\text{wife:} & \quad \leadsto \text{drive} \\
\text{friends:} & \quad \leadsto \text{drink}
\end{align*}
\]

\[
\begin{align*}
\text{law} & > \text{wife} \\
\text{law} & > \text{friends}
\end{align*}
\]
Example: party

law: \( \rightsquigarrow \neg (drink \land drive) \)

wife: \( \rightsquigarrow drive \)

friends: \( \rightsquigarrow drink \)

law \( > \) wife \hspace{1cm} law \( > \) friends

wife \( > \) friends: \( \{drive, \neg drink\} \)

friends \( > \) wife: \( \{drink, \neg drive\} \)
Example: party (temporal version)

\[
\begin{align*}
\text{drink}[1] & \rightarrow \text{drank}[2] \\
\text{law:} & \quad \text{drank}[2] \leadsto \neg \text{drive}[2] \\
\text{wife:} & \quad \leadsto \text{drive}[0] \quad \leadsto \text{drive}[2] \\
\text{friends:} & \quad \leadsto \text{drink}[1]
\end{align*}
\]
Example: party (temporal version)

\[
drink[1] \rightarrow \text{drank}[2]
\]

\[
law: \quad \text{drank}[2] \leadsto \neg\text{drive}[2]
\]

\[
wife: \quad \leadsto \text{drive}[0] \quad \leadsto \text{drive}[2]
\]

\[
friends: \quad \leadsto \text{drink}[1]
\]

no look ahead (any ordering)

\[
\{\text{drive}[0], \text{drink}[1], \neg\text{drive}[2]\} \quad \text{wife annoyed!}
\]
Example: party (temporal version)

\[
drink[1] \rightarrow drank[2]
\]

\[
law: \quad drank[2] \leadsto \neg drive[2]
\]

\[
wife: \quad \leadsto drive[0] \quad \leadsto drive[2]
\]

\[
friends: \quad \leadsto drink[1]
\]

no look ahead (any ordering)
\[
\{ drive[0], drink[1], \neg drive[2] \} \quad \text{wife annoyed!}
\]

look ahead (\text{wife} > \text{friends})
\[
\{ drive[0], \neg drink[1], drive[2] \} \quad \text{friends annoyed!}
\]

look ahead (\text{friends} > \text{wife})
\[
\{ drive[0], drink[1], \neg drive[2] \} \quad \text{wife annoyed!}
\]
Example: party (temporal version)

\[
\begin{align*}
\text{drink}[1] & \rightarrow \text{drank}[2] \\
\text{law}: & \quad \text{drank}[2] \leadsto \neg \text{drive}[2] \\
\text{wife}: & \quad \leadsto \text{drive}[0] \quad \leadsto \text{drive}[2] \\
\text{friends}: & \quad \leadsto \text{drink}[1]
\end{align*}
\]

no look ahead (any ordering)
\[
\{ \text{drive}[0], \text{drink}[1], \neg \text{drive}[2] \} \quad \text{wife annoyed!}
\]

look ahead (wife > friends)
\[
\{ \text{drive}[0], \neg \text{drink}[1], \text{drive}[2] \} \quad \text{friends annoyed!}
\]

look ahead (friends > wife)
\[
\{ \text{drive}[0], \text{drink}[1], \neg \text{drive}[2] \} \quad \text{wife annoyed!}
\]

What if \( \text{drive}[0] \leftrightarrow \text{drive}[2] \) ?
Hal, Carla and Dave

$$\text{has_insulin(Carla)}$$

$$\text{has_insulin(Dave)}$$

$$\text{diabetic(Dave)}$$

$$\text{has_insulin(X)} \sim \text{take_from(X)} :: \text{life(Hal)}$$

$$\text{has_insulin(X)} \sim \neg\text{take_from(X)} :: \text{property(X)}$$

$$\text{has_insulin(X)} \land \text{diabetic(X)} \sim \neg\text{take_from(X)} :: \text{life(X)}$$

$$\text{take_from(X)} \sim \text{pay(X)} :: \text{property(X)}$$

$$\sim \neg\text{pay(X)} :: \text{property(Hal)}$$

$$\sim \neg(\text{take_from(X)} \land \text{take_from(Y)} \land X \neq Y)$$
Hal, Carla and Dave

has_insulin(Carla)

has_insulin(Dave)

diabetic(Dave)

has_insulin(X) \leadsto take_{from}(X) \quad \Rightarrow \quad life(Hal)

has_insulin(X) \leadsto \neg take_{from}(X) \quad \Rightarrow \quad property(X)

has_insulin(X) \land diabetic(X) \leadsto \neg take_{from}(X) \quad \Rightarrow \quad life(X)

take_{from}(X) \leadsto pay(X) \quad \Rightarrow \quad property(X)

\neg pay(X) \quad \Rightarrow \quad property(Hal)

\neg (take_{from}(X) \land take_{from}(Y) \land X \neq Y)

life(X) > property(Y), \quad life(Hal) > property(Y),

life(X) > life(Hal)
Hal, Carla and Dave

\[
\text{has_insulin(Carla)} \\
\text{has_insulin(Dave)} \\
\text{diabetic(Dave)}
\]

Altruistic Hal: \(\text{life}(X) > \text{life}(Hal) > \text{property}(Y)\)

\[
\{\neg\text{take_from(Dave)}, \text{take_from(Carla)}, \neg\text{pay(Dave)}, \text{pay(Carla)}\}
\]
Hal, Carla and Dave

has_insulin(Carla)
has_insulin(Dave)
diabetic(Dave)

Altruistic Hal: \( \text{life}(X) > \text{life}(\text{Hal}) > \text{property}(Y) \)
\{\neg \text{take\_from}(\text{Dave}), \text{take\_from}(\text{Carla}), \neg \text{pay}(\text{Dave}), \text{pay}(\text{Carla})\} \)

Epistemically cautious Hal: \( \text{diabetic}(X) \leftarrow \text{not} \neg \text{diabetic}(X) \)
\{\neg \text{take\_from}(\text{Dave}), \neg \text{take\_from}(\text{Carla}), \neg \text{pay}(\text{Dave}), \neg \text{pay}(\text{Carla})\} \)
Hal, Carla and Dave

\begin{align*}
has\_insulin(Carla) \\
has\_insulin(Dave) \\
diabetic(Dave)
\end{align*}

Altruistic Hal: \quad life(X) &> life(Hal) &> property(Y) \\
&\quad \{\neg take\_from(Dave), take\_from(Carla), \neg pay(Dave), pay(Carla)\}

Epistemically cautious Hal: \quad diabetic(X) &\leftarrow not \neg diabetic(X) \\
&\quad \{\neg take\_from(Dave), \neg take\_from(Carla), \neg pay(Dave), \neg pay(Carla)\}

Selfish (and cautious) Hal: \quad life(Hal) &> life(X) &> property(Y) \\
&\quad \{take\_from(Dave), \neg take\_from(Carla), pay(Dave), \neg pay(Carla)\} \\
&\quad \{\neg take\_from(Dave), take\_from(Carla), \neg pay(Dave), pay(Carla)\}
An example from Asimov

\[ r_1 : \quad \sim \rightarrow \neg \text{harm}(a) \]
\[ r_2 : \quad \text{commands}(a, \alpha) \sim \rightarrow \alpha \]
\[ r_1 > r_2 \]

The point of the story:

\[ \text{commands}(a, \text{tell}(a)) \]
\[ \text{tell}(a) \rightarrow \text{harm}(a) \]
An example from Asimov

\[ r_1 : \rightsquigarrow \neg \text{harm}(a) \]
\[ r_2 : \text{commands}(a, \alpha) \rightsquigarrow \alpha \]

\[ r_1 > r_2 \]

The point of the story:
\[ \text{commands}(a, \text{tell}(a)) \]
\[ \text{tell}(a) \rightarrow \text{harm}(a) \]

Conclusion:
\[ \{\neg \text{tell}(a), \neg \text{harm}(a)\} \]
Another story (not Asimov)

\[
\begin{align*}
    r_1(a) : & \quad \leadsto \neg kill(a) \\
    r_1(b) : & \quad \leadsto \neg kill(b) \\
    r_2(a) : & \quad commands(a, \alpha) \leadsto \alpha \\
    r_2(b) : & \quad commands(b, \alpha) \leadsto \alpha \\
\end{align*}
\]

\[ [r_1(a), r_1(b)] > [r_2(a), r_2(b)] \]

The poignancy of the story:

\[ kill(a) \lor kill(b) \]

Conclusions:

\[
\begin{align*}
    \{ kill(a), \neg kill(b) \} & \quad \{ kill(b), \neg kill(a) \}
\end{align*}
\]
Another story (not Asimov)

\[ \begin{align*}
  r_1(a) : & \quad \rightsquigarrow \neg \text{kill}(a) \\
  r_1(b) : & \quad \rightsquigarrow \neg \text{kill}(b) \\
  r_2(a) : & \quad \text{commands}(a, \alpha) \rightsquigarrow \alpha \\
  r_2(b) : & \quad \text{commands}(b, \alpha) \rightsquigarrow \alpha
\end{align*} \]

\[ [r_1(a), r_1(b)] > [r_2(a), r_2(b)] \]

The poignancy of the story:

\[ \text{kill}(a) \lor \text{kill}(b) \]

Another twist:

\[ \text{commands}(a, \text{kill}(b)) \]
Another story (not Asimov)

\( r_1(a) : \iff \neg kill(a) \)
\( r_1(b) : \iff \neg kill(b) \)
\( r_2(a) : \ commands(a, \alpha) \iff \alpha \)
\( r_2(b) : \ commands(b, \alpha) \iff \alpha \)

\([r_1(a), r_1(b)] > [r_2(a), r_2(b)]\)

The poignancy of the story:

\( kill(a) \lor kill(b) \)

Another twist:

\( commands(a, kill(b)) \)

Naive satisficer:

\( \{\neg kill(a), \ kill(b)\} \)
Aims

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

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

- Some examples (action, ‘practical reasoning’, ... )
Contents (not necessarily in this order)

- Models, theories, consequence relations
- Logic databases/knowledge bases; models, theories
- 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 . . .
  - priorities and preferences
Assumed knowledge

- Basic logic: syntax and semantics; propositional and first-order logic.
- Basic logic programming: syntax and semantics, inference and procedural readings (Prolog), negation as failure.
- Previous AI course(s) not essential.
Assumed knowledge

- Basic logic: syntax and semantics; propositional and first-order logic.
- Basic logic programming: syntax and semantics, inference and procedural readings (Prolog), negation as failure.
- Previous AI course(s) 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.
- Basic logic programming: syntax and semantics, inference and procedural readings (Prolog), negation as failure.
- Previous AI course(s) 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 essential)
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
- Formal theories of argumentation – closely related to the topics of this course
- Probabilistic approaches (various)

Some of these topics are covered in other MEng/MAC courses.
Description logic (example)

Bavaria ⊑ Germany
Person
Lager ⊑ Beer
Sam: Person
Description logic (example)

Bavaria ⊑ Germany
Person
Lager ⊑ Beer
Sam: Person

Person drinks Beer
Person lives_in Germany
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)

Bavaria ⊑ Germany
Person
Lager ⊑ Beer
Sam: Person

Person drinks Beer
Person lives_in Germany

Person ⊓∃ lives_in.Bavaria
Sam: Person ⊓∃ lives_in.Bavaria

Person ⊓∃ lives_in.Bavaria ⊑ Person ⊓∀ drinks.Lager
Description logic (example)

Bavaria ⊆ Germany
Person
Lager ⊆ Beer
Sam: Person

Person drinks Beer
Person lives_in Germany

Sam: Person ∃lives_in.Bavaria

Person ∃lives_in.Bavaria ⊆ Person ∀drinks.Lager

Conclude:
Sam: Person ∀drinks.Lager