Handling Exceptions in Knowledge Representation: A Brief Introduction to Nonmonotonic Reasoning

Gerhard Brewka

Computer Science Institute
University of Leipzig
brewka@informatik.uni-leipzig.de

Outline

Part I: Background and Simple Forms of Nonmon

1. Background and Motivation
2. Closed World Assumption
3. Argumentation Frameworks

Outline

Background and Simple Forms of Nonmon

1. Background and Motivation

- Classical logic allows us to represent universal statements:

$$\forall x. PhDstudent(x) \rightarrow Student(x)$$

- Useful, e.g. for concept definitions or in mathematics

1. Background and Motivation

- Classical logic allows us to represent universal statements:

$$\forall x.PhDstudent(x) \rightarrow Student(x)$$

- Useful, e.g. for concept definitions or in mathematics
- Less useful to represent generic statements which may have exceptions:

1. Background and Motivation

- Classical logic allows us to represent universal statements:

$$\forall x. PhDstudent(x) \rightarrow Student(x)$$

- Useful, e.g. for concept definitions or in mathematics
- Less useful to represent generic statements which may have exceptions:
    - *Professors teach ...*

1. Background and Motivation

- Classical logic allows us to represent universal statements:

$$\forall x. PhDstudent(x) \rightarrow Student(x)$$

- Useful, e.g. for concept definitions or in mathematics
- Less useful to represent generic statements which may have exceptions:
  - *Professors teach ... unless they are on sabbatical.*

1. Background and Motivation

- Classical logic allows us to represent universal statements:

$$\forall x.PhDstudent(x) \rightarrow Student(x)$$

- Useful, e.g. for concept definitions or in mathematics
- Less useful to represent generic statements which may have exceptions:
  - *Professors teach ... unless they are on sabbatical.*
  - *Birds fly ...*

1. Background and Motivation

- Classical logic allows us to represent universal statements:

$$\forall x. PhDstudent(x) \rightarrow Student(x)$$

- Useful, e.g. for concept definitions or in mathematics
- Less useful to represent generic statements which may have exceptions:
  - *Professors teach ... unless they are on sabbatical.*
  - *Birds fly ... unless they are penguins.*

1. Background and Motivation

- Classical logic allows us to represent universal statements:

$$\forall x.PhDstudent(x) \rightarrow Student(x)$$

- Useful, e.g. for concept definitions or in mathematics
- Less useful to represent generic statements which may have exceptions:
  - *Professors teach ... unless they are on sabbatical.*
  - *Birds fly ... unless they are penguins.*
  - *Owls hunt at night ...*

1. Background and Motivation

- Classical logic allows us to represent universal statements:

$$\forall x.PhDstudent(x) \rightarrow Student(x)$$

- Useful, e.g. for concept definitions or in mathematics
- Less useful to represent generic statements which may have exceptions:
    - *Professors teach ... unless they are on sabbatical.*
    - *Birds fly ... unless they are penguins.*
    - *Owls hunt at night ... unless they live in a zoo.*

1. Background and Motivation

- Classical logic allows us to represent universal statements:

$$\forall x.PhDstudent(x) \rightarrow Student(x)$$

- Useful, e.g. for concept definitions or in mathematics
- Less useful to represent generic statements which may have exceptions:
  - *Professors teach ... unless they are on sabbatical.*
  - *Birds fly ... unless they are penguins.*
  - *Owls hunt at night ... unless they live in a zoo.*
  - *Students hate theoretical computer science ...*

1. Background and Motivation

- Classical logic allows us to represent universal statements:

$$\forall x.PhDstudent(x) \rightarrow Student(x)$$

- Useful, e.g. for concept definitions or in mathematics
- Less useful to represent generic statements which may have exceptions:
    - *Professors teach ... unless they are on sabbatical.*
    - *Birds fly ... unless they are penguins.*
    - *Owls hunt at night ... unless they live in a zoo.*
    - *Students hate theoretical computer science ... unless they are very clever.*

1. Background and Motivation

- Classical logic allows us to represent universal statements:

$$\forall x. PhDstudent(x) \rightarrow Student(x)$$

- Useful, e.g. for concept definitions or in mathematics
- Less useful to represent generic statements which may have exceptions:
  - *Professors teach ... unless they are on sabbatical.*
  - *Birds fly ... unless they are penguins.*
  - *Owls hunt at night ... unless they live in a zoo.*
  - *Students hate theoretical computer science ... unless they are very clever.*
  - *After spending 2 hours in the doctor's waiting room patients get angry ...*

1. Background and Motivation

- Classical logic allows us to represent universal statements:

$$\forall x. PhDstudent(x) \rightarrow Student(x)$$

- Useful, e.g. for concept definitions or in mathematics
- Less useful to represent generic statements which may have exceptions:
  - *Professors teach ... unless they are on sabbatical.*
  - *Birds fly ... unless they are penguins.*
  - *Owls hunt at night ... unless they live in a zoo.*
  - *Students hate theoretical computer science ... unless they are very clever.*
  - *After spending 2 hours in the doctor's waiting room patients get angry ... unless they are close to finishing a proof.*
  - ...

A solution?

- Most of our commonsense knowledge is of this kind
- What can we do to represent it adequately?

A solution?

- Most of our commonsense knowledge is of this kind
- What can we do to represent it adequately?
- What if instead of $\forall x. Bird(x) \rightarrow Flies(x)$ we use

$$\forall x. Bird(x) \wedge \neg Ab(x) \rightarrow Flies(x)$$

A solution?

- Most of our commonsense knowledge is of this kind
- What can we do to represent it adequately?
- What if instead of $\forall x.Bird(x) \rightarrow Flies(x)$ we use

$$\forall x.Bird(x) \wedge \neg Ab(x) \rightarrow Flies(x)$$

and add

$$\forall x.Ab(x) \leftrightarrow Penguin(x) \vee Ostrich(x) \vee Injured(x) \vee \ldots$$

A solution?

- Most of our commonsense knowledge is of this kind
- What can we do to represent it adequately?
- What if instead of $\forall x. Bird(x) \rightarrow Flies(x)$ we use

$$\forall x. Bird(x) \land \neg Ab(x) \rightarrow Flies(x)$$

and add

$$\forall x. Ab(x) \leftrightarrow Penguin(x) \lor Ostrich(x) \lor Injured(x) \lor \ldots$$

- Problem 1: no exhaustive list of abnormalities.

A solution?

- Most of our commonsense knowledge is of this kind
- What can we do to represent it adequately?
- What if instead of $\forall x.Bird(x) \rightarrow Flies(x)$ we use

$$\forall x.Bird(x) \land \neg Ab(x) \rightarrow Flies(x)$$

  and add

$$\forall x.Ab(x) \leftrightarrow Penguin(x) \lor Ostrich(x) \lor Injured(x) \lor \ldots$$

- Problem 1: no exhaustive list of abnormalities.
- Problem 2: does not give us *Flies*(*tweety*) unless *tweety* is known not to be an exception.

How to use generic information

- Want to draw conclusions from generic information *as long as nothing indicates an exception*.
- If additional information tells us something is abnormal, retract former conclusion.

How to use generic information

- Want to draw conclusions from generic information *as long as nothing indicates an exception*.
- If additional information tells us something is abnormal, retract former conclusion.

  $\Rightarrow$ *Conclusions do not grow monotonically with premises.*

How to use generic information

- Want to draw conclusions from generic information *as long as nothing indicates an exception*.
- If additional information tells us something is abnormal, retract former conclusion.

  $\Rightarrow$ *Conclusions do not grow monotonically with premises.*
- Classical logic cannot model this, as it is monotonic:

$$X \subseteq Y \Rightarrow Th(X) \subseteq Th(Y).$$

- Why? q follows from X if q holds in all models of X. Models of Y a subset, thus q holds in all of them as well.

How to use generic information

- Want to draw conclusions from generic information *as long as nothing indicates an exception*.
- If additional information tells us something is abnormal, retract former conclusion.

  $\Rightarrow$ *Conclusions do not grow monotonically with premises.*
- Classical logic cannot model this, as it is monotonic:

$$X \subseteq Y \Rightarrow Th(X) \subseteq Th(Y).$$

- Why? q follows from X if q holds in all models of X. Models of Y a subset, thus q holds in all of them as well.
- Observation led to the AI field of nonmonotonic reasoning, active for over 30 years.

Conflicting Defaults

- Defaults may give rise to conflicting conclusions:

  (1) Quakers normally are pacifists.
  (2) Republicans normally are not pacifists.
  (3) Nixon is a quaker and a republican.

- (1) and (2) conflicting.
- Nothing wrong with the defaults!

Conflicting Defaults

- Defaults may give rise to conflicting conclusions:

  (1) Quakers normally are pacifists.
  (2) Republicans normally are not pacifists.
  (3) Nixon is a quaker and a republican.

- (1) and (2) conflicting.
- Nothing wrong with the defaults!
- Different approaches to deal with this:
  - some apply none of the conflicting defaults,
  - most generate different acceptable belief sets (extensions) leave open whether to use them sceptically (p true in all of them) or credulously (p true in some of them, or in a particular one).

- Check the QuantLA Spring School time table
  - *Question: Is Franz teaching on Friday?*
  - *Your answer (presumably): No*

2. The Closed World Assumption

- Check the QuantLA Spring School time table
  - *Question: Is Franz teaching on Friday?*
  - *Your answer (presumably): No*
- Why is this answer correct?
- Does not follow from the explicit information in the time table

2. The Closed World Assumption

- Check the QuantLA Spring School time table
  - *Question: Is Franz teaching on Friday?*
  - *Your answer (presumably): No*
- Why is this answer correct?
- Does not follow from the explicit information in the time table
- But: follows from this information *assuming that the list of courses is complete*
- You (presumably) used this assumption, and do so in many everyday contexts

The Closed World Assumption, ctd.

- In many situations way more negative than positive facts.

The Closed World Assumption, ctd.

- In many situations way more negative than positive facts.
- Communication convention: represent the latter only, leave the former implicit.
  - train/flight schedules
  - TV programs
  - library catalogues
  - list of lectures at a spring school

The Closed World Assumption, ctd.

- In many situations way more negative than positive facts.
- Communication convention: represent the latter only, leave the former implicit.
  - train/flight schedules
  - TV programs
  - library catalogues
  - list of lectures at a spring school
- Know how to infer negative information based on completeness assumption.

Reiter's formalization

- Let *KB* be a set of formulas, define new form of entailment under CWA:

$$KB \models_c \alpha \text{ iff } KB \cup Negs \models \alpha$$

where $Negs = \{\neg p \mid p \text{ atomic and } KB \not\models p\}$

Reiter's formalization

- Let *KB* be a set of formulas, define new form of entailment under CWA:

$$KB \models_c \alpha \text{ iff } KB \cup Negs \models \alpha$$

  where $Negs = \{\neg p \mid p \text{ atomic and } KB \not\models p\}$

- $\models_c$ nonmonotonic, for instance $\{a\} \models_c \neg b$ whereas $\{a, b\} \not\models_c \neg b$

Reiter's formalization

- Let *KB* be a set of formulas, define new form of entailment under CWA:

$$KB \models_c \alpha \text{ iff } KB \cup Negs \models \alpha$$

  where $Negs = \{\neg p \mid p \text{ atomic and } KB \not\models p\}$

- $\models_c$ nonmonotonic, for instance $\{a\} \models_c \neg b$ whereas $\{a, b\} \not\models_c \neg b$
- CWA makes knowledge complete: for arbitrary α (without quantifiers) we have $KB \models_c \alpha$ or $KB \models_c \neg\alpha$.
- Recursive query evaluation; queries reduced to atomic case.

Reiter's formalization

- Let *KB* be a set of formulas, define new form of entailment under CWA:

$$KB \models_c \alpha \text{ iff } KB \cup Negs \models \alpha$$

  where $Negs = \{\neg p \mid p \text{ atomic and } KB \not\models p\}$

- $\models_c$ nonmonotonic, for instance $\{a\} \models_c \neg b$ whereas $\{a, b\} \not\models_c \neg b$
- CWA makes knowledge complete: for arbitrary α (without quantifiers) we have $KB \models_c \alpha$ or $KB \models_c \neg\alpha$.
- Recursive query evaluation; queries reduced to atomic case.
- Results extend to quantified formulas if we add *domain closure assumption* (each object named by constant) and *unique names assumption* (different constants denote different objects).

A major problem

- Works for simple cases only, e.g. KB a set of atoms.
- Assume $KB \models (p \lor q)$, but $KB \not\models p$ and $KB \not\models q$.
- Now $\neg p \in Negs$ and $\neg q \in Negs$, thus $KB \cup Negs$ inconsistent.

A major problem

- Works for simple cases only, e.g. KB a set of atoms.
- Assume $KB \models (p \vee q)$, but $KB \not\models p$ and $KB \not\models q$.
- Now $\neg p \in Negs$ and $\neg q \in Negs$, thus $KB \cup Negs$ inconsistent.
- Weaker versions of CWA avoiding inconsistency were proposed.
- CWA best viewed as a method for restricted contexts (e.g. databases).

A major problem

- Works for simple cases only, e.g. KB a set of atoms.
- Assume $KB \models (p \vee q)$, but $KB \not\models p$ and $KB \not\models q$.
- Now $\neg p \in Negs$ and $\neg q \in Negs$, thus $KB \cup Negs$ inconsistent.
- Weaker versions of CWA avoiding inconsistency were proposed.
- CWA best viewed as a method for restricted contexts (e.g. databases).

Standard Reference:

Reiter, Raymond (1978). *On Closed World Data Bases*. In Gallaire, H.; Minker, J., Logic and Data Bases. Plenum Press. pp. 119-140.

2. Argumentation

- Argumentation highly active area in AI.
- Idea: to construct acceptable set(s) of beliefs from given KB:
  1. construct arguments (beliefs with associated reasons),
  2. determine jointly acceptable arguments (extensions),
  3. accept their conclusions.
- Assumption: step 2 can be done independently and abstractly.
- Dung's Abstract Argumentation Frameworks widely used tool.

Argumentation Frameworks

Abstract Argumentation

- Arguments “atomic”, their structure irrelevant.
- All that matters are attacks among arguments.
- Argumentation frameworks (AFs) represent attack relations.
- Semantics formalize different intuitions about how to solve conflicts and how to pick acceptable arguments.
- Semantics map an AF to subsets of its arguments (extensions).
- Nonmonotonic: new argument may throw out what was accepted.

Definition

Argumentation Frameworks

An argumentation framework (AF) is a pair (A, R) where

- A is a set of arguments,
- $R \subseteq A \times A$ is a relation representing "attacks". ("defeats")

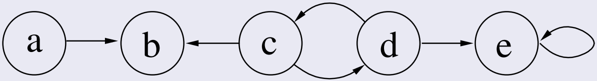
Definition

Argumentation Frameworks

An argumentation framework (AF) is a pair (A, R) where

- A is a set of arguments,
- $R \subseteq A \times A$ is a relation representing "attacks". ("defeats")

Example

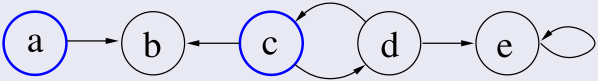


Semantics: minimal requirement no conflicts

Conflict-Free Set

Given an AF $F = (A, R)$.
A set $S \subseteq A$ is conflict-free in F, if, for each $a, b \in S$, $(a, b) \notin R$.

Example



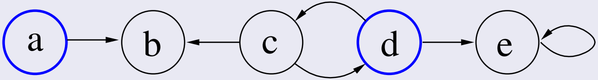
$cf(F) = \{\{a, c\},$

Semantics: minimal requirement no conflicts

Conflict-Free Set

Given an AF $F = (A, R)$.
A set $S \subseteq A$ is conflict-free in F, if, for each $a, b \in S$, $(a, b) \notin R$.

Example



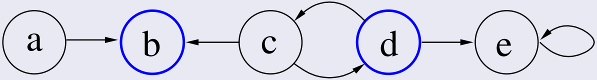
$cf(F) = \{\{a, c\}, \{a, d\},$

Semantics: minimal requirement no conflicts

Conflict-Free Set

Given an AF $F = (A, R)$.
A set $S \subseteq A$ is conflict-free in F, if, for each $a, b \in S$, $(a, b) \notin R$.

Example



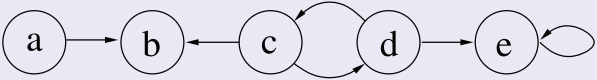
$cf(F) = \{\{a, c\}, \{a, d\}, \{b, d\},$

Semantics: minimal requirement no conflicts

Conflict-Free Set

Given an AF $F = (A, R)$.
A set $S \subseteq A$ is conflict-free in F, if, for each $a, b \in S$, $(a, b) \notin R$.

Example



$cf(F) = \{\{a, c\}, \{a, d\}, \{b, d\}, \{a\}, \{b\}, \{c\}, \{d\}, \emptyset\}$

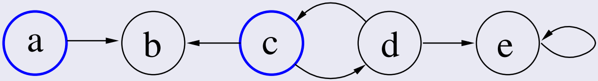
No undefended attacked arguments

Admissible Set

Given an AF $F = (A, R)$. A set $S \subseteq A$ is admissible in F, if

- S is conflict-free in F
- each $a \in S$ is defended by S in F,
  - $a \in A$ is defended by S in F, if for each $b \in A$ with $(b, a) \in R$, there exists a $c \in S$, such that $(c, b) \in R$.

Example



$adm(F) = \{\{a, c\},$

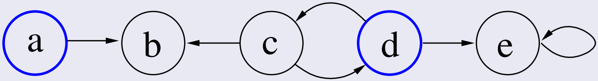
No undefended attacked arguments

Admissible Set

Given an AF $F = (A, R)$. A set $S \subseteq A$ is admissible in F, if

- S is conflict-free in F
- each $a \in S$ is defended by S in F,
  - $a \in A$ is defended by S in F, if for each $b \in A$ with $(b, a) \in R$, there exists a $c \in S$, such that $(c, b) \in R$.

Example



$adm(F) = \{\{a, c\}, \{a, d\},$

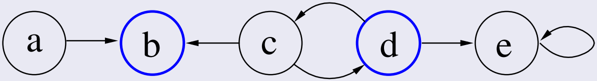
No undefended attacked arguments

Admissible Set

Given an AF $F = (A, R)$. A set $S \subseteq A$ is admissible in F, if

- S is conflict-free in F
- each $a \in S$ is defended by S in F
  - $a \in A$ is defended by S in F, if for each $b \in A$ with $(b, a) \in R$, there exists a $c \in S$, such that $(c, b) \in R$.

Example



$adm(F) = \{\{a, c\}, \{a, d\}, ~~\{b, d\},~~$

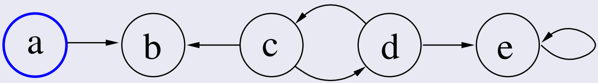
No undefended attacked arguments

Admissible Set

Given an AF $F = (A, R)$. A set $S \subseteq A$ is admissible in F, if

- S is conflict-free in F
- each $a \in S$ is defended by S in F,
  - $a \in A$ is defended by S in F, if for each $b \in A$ with $(b, a) \in R$, there exists a $c \in S$, such that $(c, b) \in R$.

Example



$adm(F) = \{\{a, c\}, \{a, d\},$ ~~$\{b, d\}$~~, $\{a\}$, ~~$\{b\}$~~, $\{c\}, \{d\}, \emptyset\}$

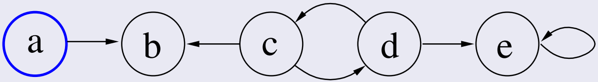
Want all defended arguments

Complete Set

Given an AF $F = (A, R)$. A set $S \subseteq A$ is complete in F, if

- S is admissible in F
- each $a \in A$ defended by S in F is contained in S
  - $a \in A$ is defended by S in F, if for each $b \in A$ with $(b, a) \in R$, there exists a $c \in S$, such that $(c, b) \in R$.

Example



$comp(F) = \{\{a, c\}, \{a, d\}, \{a\},$ ~~$\{c\}, \{d\}, \emptyset$~~ $\}$

An inherently skeptical approach

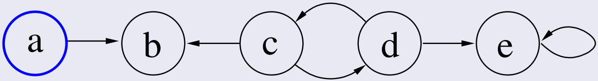
Grounded Extension

Given an AF $F = (A, R)$. A set $S \subseteq A$ is grounded in F, if

- S is complete in F
- for each $T \subseteq A$ complete in F, $T \not\subset S$

Proposition [Dung 95]: The grounded extension of an AF $F = (A, R)$ is given by the least fix-point of the operator $\Gamma_F : 2^A \rightarrow 2^A$, defined as $\Gamma_F(S) = \{a \in A \mid a \text{ is defended by } S \text{ in } F\}$

Example



$ground(F) = \{$~~$\{a, c\}, \{a, d\},$~~$\{a\}\}$

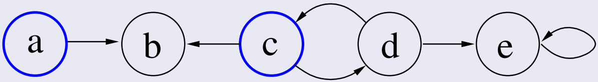
A credulous approach

Stable Extension

Given an AF $F = (A, R)$. A set $S \subseteq A$ is stable in F, if

- S is conflict-free in F
- for each $a \in A \setminus S$, there exists a $b \in S$, such that $(b, a) \in R$.

Example



$stable(F) = \{$ ~~$\{a, c\},$~~

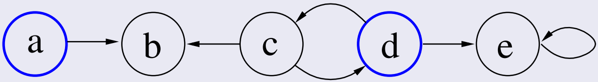
A credulous approach

Stable Extension

Given an AF $F = (A, R)$. A set $S \subseteq A$ is stable in F, if

- S is conflict-free in F
- for each $a \in A \setminus S$, there exists a $b \in S$, such that $(b, a) \in R$.

Example



$stable(F) = \{$~~$\{a, c\},$~~$\{a, d\},$

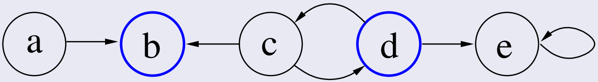
A credulous approach

Stable Extension

Given an AF $F = (A, R)$. A set $S \subseteq A$ is stable in F, if

- S is conflict-free in F
- for each $a \in A \setminus S$, there exists a $b \in S$, such that $(b, a) \in R$.

Example



$stable(F) = \{$~~$\{a, c\},$~~$\{a, d\},$ ~~$\{b, d\},$~~

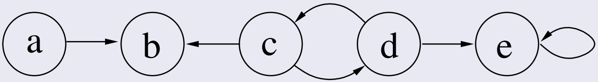
A credulous approach

Stable Extension

Given an AF $F = (A, R)$. A set $S \subseteq A$ is stable in F, if

- S is conflict-free in F
- for each $a \in A \setminus S$, there exists a $b \in S$, such that $(b, a) \in R$.

Example



$stable(F) = \{$ ~~$\{a, c\}$,~~ $\{a, d\}$, ~~$\{b, d\}$, $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\emptyset$~~ $\}$

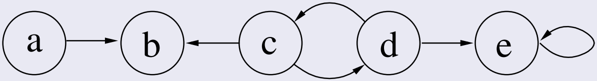
Guaranteeing existence of extensions

Preferred Extension

Given an AF $F = (A, R)$. A set $S \subseteq A$ is preferred in F, if

- S is admissible in F
- for each $T \subseteq A$ admissible in T, $S \not\subset T$

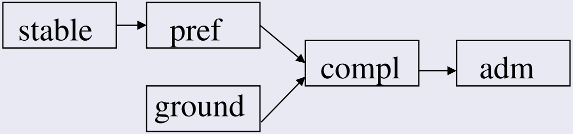
Example



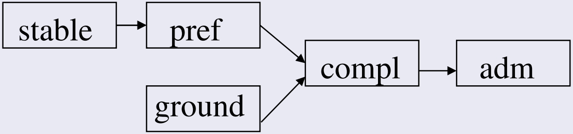
$pref(F) = \{\{a, c\}, \{a, d\},$ ~~$\{a\}, \{c\}, \{d\}, \emptyset$~~$\}$

Complexity

Relation between Semantics



Relation between Semantics



Complexity

	stable	*adm*	*pref*	*comp*	*ground*
Cred	NP-c	NP-c	NP-c	NP-c	in P
Skept	coNP-c	(trivial)	Π_2^P-c	in P	in P

[Dimopoulos & Torres 96; Dunne & Bench-Capon 02; Coste-Marquis *et al.* 05]

Further remarks

- AFs: simple graph representation of argumentation scenarios.
- Various semantics model different intuitions how to select reasonable argument sets.

BUT

- Fixed meaning of links: attack; fixed acceptance condition for args: no parent accepted.
- Want more flexibility:
  - Links supporting arguments/positions,
  - Nodes not accepted unless supported,
  - Flexible means of combining attack and support.
- Developed *Dialectical Frameworks* which can have arbitrary relations among args.
- Many options for adding quantities.