TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Faculty of Computer Science Chair of Automata Theory

INTRODUCTION TO NONMONOTONIC REASONING

Anni-Yasmin Turhan

Dresden, WS 2019/20

Course Material

- Book "Nonmonotonic Reasoning" by Grigoris Antoniou
- Book "Nonmonotonic Reasoning" by G. Brewka, J. Dix, K. Konolige
- available on course website:
  - Slides
  - Exercise Sheets
- Things written on the blackboard

Contact Information

- `anni-yasmin.turhan@tu-dresden.de`
- course web page: `https://tu-dresden.de/ing/informatik/thi/lat/studium/lehrveranstaltungen/wintersemester-2019-2020/introduction-to-non-monotonic-reasoning`

Exams

Oral exams at the end of the semester or during semester break

Plan of sessions

Week	Mon	Tuesday	Wed	Thursday	Fri	
14. - 18.10.		VL		VL		
21. - 25.10.		VL		Tut-1		
28. - 01.11.		VL		(holiday)		
04. - 08.11.		Tut-2		VL		
11. - 15.11.		-		Tut-3		
18. - 22.11.		VL	(holiday)	Tut-4		
25. - 29.11.		VL		Tut-5		
02. - 06.12.		VL		Tut-6		
09. - 13.12.		VL		Tut-7		
16. - 20.12.		VL		Tut-8		
break						
06. - 10.01.		VL		Tut-9		
13. - 17.01.		VL		Tut-10		
20. - 24.01.		VL		Tut-11		
27. - 31.01.		VL		Tut-12		
03. - 07.02.		VL		Tut-13		"+ 2 VL"

Section 1

Introduction

A non-technical one!

Nonmonotonic Reasoning

is a long-standing area of knowledge representation and thus of AI

Classical Logic

is monotone in the following sense:
Whenever a formula φ is a logical consequence of a set of formulas T,
then φ is also a logical consequence of an arbitrary superset of T.

Common sense reasoning

works differently:
We draw plausible conclusions based on the assumption the world is normal and works as expected.

Best we can do under incomplete information.
Sometimes the normality assumption goes wrong.

Nonmonotonic Reasoning

: additional information may invalidate conclusions drawn earlier

An Example

Assume you want to meet Prof. Petterson.

She usually is in the CS building in the afternoon.

You assume that this is the case today and go to the CS department in the afternoon.

Many people stand in front of the building—a while ago a fire alarm has gone off.

This new piece of information invalidates the normality assumption and so the conclusion about the present location of Prof. Petterson, too!

Major Application Areas

Databases

Suppose we want to build a DB about airline flights.

- put positive facts into DB:
  "Flight from DRS to LHR, by StarAir, 8.00 am, Oct. 16th"
- assumption: DB contains all relevant facts
  - But: impossible to put *all* negative facts!
  - But: inconvenient for updates!
- solution proposed by R. Reiter '78: capture the assumption formally

  Closed world assumption (CWA) :
  "Statement that is true is also known to be true and, conversely, what is not known to be true, is false."

  $\leadsto$ Assumes complete knowledge!

Major Application Areas

Puzzles

- For example: "Missionaries and cannibals":
  3 missionaries and 3 cannibals come to a river. A row boat with 2 seats is available. If the cannibals outnumber the missionaries, on a bank of the river, the missionaries will be eaten.
  How shall they cross the river?
- easy: stating positive facts, e.g. "the boat carries 2 people."
- difficult: stating negative assumptions, e.g. "the river can only be crossed by boat."
- formal approach for dealing with unstated assumptions: circumscription by John McCarthy
  Idea: introduce a predicate for “abnormality” and do reasoning in regard of preferred models.

Diagnosis

- Diagnosis is a nonmonotonic task!
- Suppose an emergency case is brought in.
  The doctor treats the patient immediately without knowing the cause of the symptoms
  $\leadsto$ makes assumptions about the "most plausible" and "worst possible" cause.
- Reasoning task: find "most plausible cause" of the problem

Natural language understanding

- natural language is ambiguous $\leadsto$ yields competing interpretations
- if choice was wrong, working hypothesis may be revised and new conclusions drawn
- nonmonotonic reasoning provides mechanisms to support these operations

Legal reasoning

- Most legal regulations are rules with exceptions
- different legal regulations can hold at the same time. Which overwrites which?
  E.g.: state law can overwrite local law
- defaults can represent this naturally

Rules with exceptions

Most rules we use in commonsense reasoning such as:

"Professors teach."
"Birds fly."
"Well tested software is reliable."

describe what normally holds, but not necessarily without exception.

This is in contrast to formulas in first order logic:

$$\forall x\big(\mathit{prof}(x) \longrightarrow \mathit{teaches}(x)\big)$$
$$\forall x\big(\mathit{bird}(x) \longrightarrow \mathit{flies}(x)\big)$$
$$\forall x\big(\mathit{Software}(x) \wedge \mathit{well\text{-}tested}(x) \longrightarrow \mathit{reliable}(x)\big)$$

To apply such a rule we need to know whether the concrete instance is exceptional. How to derive this?

The Frame Problem

- To express effects of actions (and reason about changes they cause), one has to indicate under what circumstances a proposition whose truth value may vary does hold.
- E.g. in situation calculus, effects of actions can easily be described. It is more problematic to describe what does not change when an event occurs.
- The frame problem asks how to represent the large amount of non-changes when reasoning about action.

  Idea: use a persistence rule such as

  > "what holds in a situation typically holds in the situation after an action was performed, unless it contradicts the description of the effects of the actions."

  This rule is nonmonotonic!
- The frame problem has provided a major impetus to research of nonmonotonic reasoning

Main topics

- Default logic
- Autoepistemic Logic
- Circumscription
- Belief Revision
- Nonmonotonic inference rules

Default Logics — covered in Section 3

- introduced by Ray Reiter in 1980
- default logics distinguish:
  facts (or axioms) from defaults ("rules of thumb")
  E.g.:

$$\frac{\mathit{bird}(x) : \mathit{flies}(x)}{\mathit{flies}(x)}$$

- default theory:
  - set of facts: certain, but incomplete information about the world
  - set of defaults: sanction plausible but not necessarily true conclusions
- reasoning under the closed world assumption
- operational semantics by extensions (:beliefs that may hold about the domain)
- goal: compute sets of acceptable beliefs
- We'll consider: variants of default logics

Autoepistemic Logics — covered in Section 4

- proposed by Robert C. Moore in 1985
- "autoepistemic": reflection upon self-knowledge
- autoepistemic logics can express the lack of facts
- an ideally rational agent forms belief sets given initial assumption
- closely related to Modal logics
- emphasis on inference relations
- compute expansions — specifies which formulae are true and which ones are false
- computation of set of preferred models ($\approx$ "normal models"): minimization of the extension of some predicates
  New information changes this set!
- We'll consider:
  - stable sets
  - computational properties

Circumscription — covered in Section 5

- introduced by John McCarthy, refined and formalized by Vladimir Lifschitz
- objective: formalize common sense reasoning used in dealing with everyday problems
- implicit assumption of inertia: things do not change unless otherwise specified.
- also based on preferred models (minimizing the extent of some predicates)
- We'll consider:
  - computational properties
  - relation to default logic

Nonmonotonic inference rules — covered in Section 7

- orthogonal view:
  which postulates should a “good” nonmonotonic inference rule fulfill?
- interaction of logical connectives
- preferential models
- formal properties of inference rules characterized by postulates

Belief Revision — covered in Section 6

- investigated by Alchourrón, Gärdenfors, Makinson
- provides operations to model changes of a knowledge base
- inconsistent knowledge: which facts to give up, which to keep?
- computational model: change of finite theory basis and iterated revision
- minimality principle, AGM postulates
- computational properties

Section 2

Preliminaries

A technical brush-up!

First Order Logic (FOL)
a.k.a. Predicate logic

Predicate logic

- is a classical logic
- classical reasoning in First Order Logic is monotone
- basic formalism for the approaches discussed in this lecture

Syntax of First Order Logic

Symbols:

- special symbols:
  - junctors: $\wedge$ (conjunction) , $\vee$ (disjunction), $\neg$ (negation), $\longrightarrow$ (implication), $\longleftrightarrow$ (equivalence),
  - quantifiers: $\exists$ (existential quantifier), $\forall$ (universal quantifier),
  - auxiliary symbols: (,),
  - countable set of variables: $V_1, \ldots, V_n$
- signature Σ
  Intuitively, Σ contains the predicate symbols and function symbols — each associated with an arity.
  E.g.: $\Sigma = \{(+, 3), (inc, 1), \ldots\}$

A function symbol of arity 0 is called a constant.

A predicate symbol of arity 0 is called an atom (or proposition).

Syntax of First Order Logic: Terms

(Consider a fixed signature Σ.)

Definition 2.1 (Terms)

FOL terms are defined as:

- Every variable or constant is a term.
- If f is a function of arity n and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.
- There is no other way of building terms.

A term is called ground iff[1] it does not contain any variables.

A term t' is a subterm of a term t, if it is a sub-string of t.

Example: From $\Sigma = \{(+, 3), (inc, 1), (=, 2), 0, 1, 2, \ldots\}$, where $0, 1, 2, \ldots$ are constants, we can build the

- term: $= \Big(+ (V_1, 2, V_3), +\big(inc(inc(V_1), 0, V_3)\big)\Big)$
- ground term: $= (inc(2), (inc(inc(1)))$

[1] short-hand for "if and only if".

Definition 2.2 (Formulae)

Let p be a predicate symbol of arity n, $t_1, \ldots, t_n$ are terms, φ and ψ be formulae and X be a variable.

FOL formulae are defined as:

- $p(t_1, \ldots, t_n)$ is an (atomic) formula
- the following are (complex) formulae:
  - $\neg\varphi$
  - $(\varphi \vee \psi)$
  - $(\varphi \wedge \psi)$
  - $(\varphi \longrightarrow \psi)$
  - $(\varphi \longleftrightarrow \psi)$
  - $(\exists X \varphi)$
  - $(\forall X \varphi)$
- There is no other way of building formulae.

If in $p(t_1, \ldots, t_n)$ all terms t_i ($1 \leq i \leq n$) are ground, then $p(t_1, \ldots, t_n)$ is a ground atomic formula.

If Σ contains only atoms (propositions!) and there are no variables in a formula φ, then φ is a formula of propositional logic.

NB: We may omit occasionally parentheses.

Examples of First Order Logic formulae

"Good students are studious and intelligent."

$$\forall X \; (\mathit{Student}(X) \land \mathit{Good}(X)) \longrightarrow ((\mathit{Studious}(X) \land \mathit{Intelligent}(X)))$$

"For each student there is a tutor whom the student consults."

$$\forall X \; (\mathit{Student}(X) \land \exists Y \; (\mathit{Tutor}(Y) \land \mathit{Consults}(X, Y)))$$

"For the student Cecilia there is a tutor whom she consults."

$$\mathit{Student}(\mathit{cecilia}) \land \exists Y \; (\mathit{Tutor}(Y) \land \mathit{Consults}(\mathit{cecilia}, Y))$$

Syntax of First Order Logic: terminology for parts of formulae, kinds of variables

How do we call specific variables or formulae?

For an occurrence of $\forall X \varphi$ or $\exists X \varphi$ within a formula ψ, the scope of the quantification $\forall X \varphi$ resp. $\exists X \varphi$ is φ.

An occurrence of a variable X in a formula ψ is called bound iff it is included in the scope of a quantification $\forall X$ or $\exists X$; otherwise the variable is free.

The variables for which there exists at least one free occurrence in a formula ψ are the free variables in ψ.

A formula is closed if it has no free variables, otherwise it is called open.

Closed formulae are called sentences.

For every open formula ψ we define $\forall(\psi)$, the universal closure of ψ, to be the formula
$\forall X_1 \ldots \forall X_n\ \psi$, where $X_1, \ldots, X_n$ are all the free variables in ψ.
(Existential closure is defined analogously using the $\exists$ quantifier.)

A literal L is either an atomic formula (positive literal) or its negation (negative literal).

Auxiliary sets

We define the following auxiliary sets:

N_{Var}	is the set of all variables
N_{Pred}	is the set of all predicates
N_{Func}	is the set of all relations
N^{Σ}_{Pred}	is the subset of all predicates in a given signature Σ, i.e., $N^{\Sigma}_{Pred} = N_{Pred} \cap \Sigma$
N^{Σ}_{Func}	is the subset of all relations in a given signature Σ, i.e., $N^{\Sigma}_{Func} = N_{Func} \cap \Sigma$

What to do with variables?

Definition 2.3 (Substitution)

A substitution σ is a finite set $\{X_1/t_1, \ldots, X_n/t_n\}$ s.t.[2]
$X_1 \ \ldots, X_n$ are different variables, and
t_i is a term different from X_i (for all $1 \leq i \leq n$).

If all terms are ground, then σ is a ground substitution.

Intuition:
The result of applying a substitution σ to a term t (denoted $t\sigma$) is replacing all occurrences of X_i in t by t_i simultaneously.

For example:
Let $\sigma_{ex} = \{V_1/p'(), V_2/q(V_1), V_3/q'(V_2, p'())\}$ and $t = p\big(V_2, f(V_1, V_2), V_3\big)$, then

$$\begin{aligned} t\sigma &= p\big(q(V_1), f(p'(), q(V_1)), q'(V_2, p'())\big) \\ (t\sigma)\sigma &= p\big(q(p'()), f(p'(), q(p'())), q'(q(V_1), p'())\big) \\ ((t\sigma)\sigma)\sigma &= p\big(q(p'()), f(p'(), q(p'())), q'(q(p'()), p'())\big) \end{aligned}$$

[2] abbreviation for "such that"

Lifting substitutions to formulae:

The result of applying a substitution σ to a formula φ (denoted $\varphi\sigma$) is replacing all free occurrences of X_i in φ by t_i.

$\varphi\sigma$ is a ground instance of φ, if $\varphi\sigma$ contains no free variables.

$\varphi\sigma$ is admissible, if none of the variables of any t_i becomes bound after σ has been applied to ϕ.

Definition 2.4 (Interpretation)

An interpretation $\mathcal{I}$ consists of

- a non-empty set $dom(\mathcal{I})$ the interpretation domain (or universe),
- a function $f^{\mathcal{I}} : dom(\mathcal{I})^n \longrightarrow dom(\mathcal{I})$ for every function symbol f of arity n.
- a relation $p^{\mathcal{I}} \subseteq dom(\mathcal{I})^n$ for every predicate symbol p of arity n.

For example:
we can model the mathematical concept of graphs as a pair $G = (V, E)$, where V is the interpretation domain of vertices and E is the binary edge relation.

The state over an interpretation $\mathcal{I}$ is a function $sta : N_{Var} \longrightarrow dom(\mathcal{I})$.

Given: variable X and value $a \in dom(\mathcal{I})$.
The modified state, where X is substituted by a $sta\,[X/a]$ is as function sta, but now X is assigned to a.

Definition 2.5 (Value of a term)

Given an interpretation $\mathcal{I}$ and a state sta.
Then the value of a term t is defined inductively as:

- $val_{\mathcal{I},sta}(X) = sta(X)$
- $val_{\mathcal{I},sta}(f(t_1, \ldots, t_n)) = f^{\mathcal{I}}(val_{\mathcal{I},sta}(t_1), \ldots, val_{\mathcal{I},sta}(t_n))$.

Semantics of FOL formulae

Definition 2.6

Given: an interpretation $\mathcal{I}$ and a state *sta*.
We define when a formula φ is true in $\mathcal{I}$ and *sta* (denoted $\mathcal{I} \models_{sta} \varphi$), inductively:

- $\mathcal{I} \models_{sta} p(t_1, \ldots, t_n)$ iff $(val_{\mathcal{I},sta}(t_1), \ldots, val_{\mathcal{I},sta}(t_n)) \in p^{\mathcal{I}}$
- $\mathcal{I} \models_{sta} \neg\psi$ iff $\mathcal{I} \not\models_{sta} \psi$
- $\mathcal{I} \models_{sta} (\psi_1 \vee \psi_2)$ iff $\mathcal{I} \models_{sta} \psi_1$ or $\mathcal{I} \models_{sta} \psi_2$
- $\mathcal{I} \models_{sta} (\psi_1 \wedge \psi_2)$ iff $\mathcal{I} \models_{sta} \psi_1$ and $\mathcal{I} \models_{sta} \psi_2$
- $\mathcal{I} \models_{sta} (\psi_1 \longrightarrow \psi_2)$ iff $\mathcal{I} \models_{sta} (\neg\psi_1 \vee \psi_2)$
- $\mathcal{I} \models_{sta} (\psi_1 \longleftrightarrow \psi_2)$ iff $\mathcal{I} \models_{sta} (\psi_1 \wedge \psi_2)$ or $\mathcal{I} \models_{sta} (\neg\psi_1 \wedge \neg\psi_2)$
- $\mathcal{I} \models_{sta} \forall X \psi$ iff $\mathcal{I} \models_{sta[X/a]} \psi$ for all $a \in dom(\mathcal{I})$
- $\mathcal{I} \models_{sta} \exists X \psi$ iff there is an $a \in dom(\mathcal{I})$ s.t. $\mathcal{I} \models_{sta[X/a]} \psi$

Note: state *sta* is irrelevant, if the formula is ground.
The truth-value depends only on $\mathcal{I}$.

We will mostly work with ground formulae!

Definition 2.7 (valid, model)

A FOL formula φ is valid (or true) in $\mathcal{I}$, if $\mathcal{I} \models \varphi$ for all states over $\mathcal{I}$.
In this case $\mathcal{I}$ is a model of φ (denoted $\mathcal{I} \models \varphi$).

Lifting this to sets of formulae:

$\mathcal{I}$ is a model of a set of formulae M ($\mathcal{I} \models M$),
iff it is a model of each formula in M.

A set of formulae M' follows (logically) from a set of formulae M (denoted $M \models M'$)
iff every model of M is also a model of M'.

Definition 2.8 (Deductive closure, theory)

Let M be a set of formulae. $Th(M)$ denotes the set of all formulae that follow from M (called the deductive closure of M).
If $M = Th(M)$, then M is called deductively closed.
A deductively closed set of closed formulae is called a Theory.

Definition 2.9 (Tautology, satisfiable)

A formula is a tautology (or valid), iff it is valid in every interpretation.

True or $\top$ denote tautologies. *False* and $\bot$ denote negations of tautologies.

A formula φ is satisfiable iff there is an interpretation $\mathcal{I}$ and a state *sta* s.t. $\mathcal{I} \models_{sta} \varphi$. A set of formulae M is satisfiable iff there is an interpretation $\mathcal{I}$ and a state *sta* s.t. $\mathcal{I} \models_{sta} M$.

The formulae φ and ψ are equivalent iff $\varphi \longleftrightarrow \psi$ is a tautology.

A set of formulae M is consistent iff M is satisfiable. A formula φ is consistent with M iff $M \cup \{\varphi\}$ is consistent.

Normal forms for FOL formulae

A FOL formula is in ...

Prenex normal form
if it has the form $Q_1 X_1 \cdots Q_n X_n \varphi$, where Q_i are quantifiers, X_i are variables, and φ a formula not containing any quantifiers.

Conjunctive normal form (CNF)
if it has the form $\bigwedge_{i=1}^{n} \bigvee_{j=1}^{m} L_{ij}$ with literals L_{ij}.

Disjunctive normal form (DNF)
if it has the form $\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} L_{ij}$ with literals L_{ij}.

Skolem normal form
if it has the form $\forall X_1 \cdots \forall X_n \varphi$, where φ is a quantifier-free formula in CNF.