Knowledge Representation and Reasoning CS616

Time: 17–21 March 2003, MSc A (lower ground floor).

www: Do check the website regularly.

Lecture Course comes in 4 parts:
Part I (Dix,3) Predicate Logic
Part II (Schmidt,15) Modal Logic
Part III (Dix,17) Answer Set Programming
Part IV (Schmidt,5) Description Logic

Exam: Friday, 4th April, 10-12, LF 12

Overview

1. **Predicate Logic**
2. **Modal Logic**
3. **Answer Set Programming**
4. **Description Logics**

Detailed Schedule

Monday:

9:30-12:30 Dix, Predicate Logic (Chapter 1)
12:30-13:30 Lunch
13:30-17:00 Schmidt, Modal Logic (Chap. 2)

Tuesday:

9:30-12:30 Schmidt, Modal Logic (Chap. 2)
12:30-13:30 Lunch
13:30-17:00 Schmidt, Modal Logic (Chap. 2)

Wednesday:

9:30-11:30 Schmidt, Modal Logic (Chap. 2)
11:30-12:30 Dix, Answer Set Progr. (Chap. 3 (3.1))
12:30-13:30 Lunch
13:30-16:00 Dix, Answer Set Progr. (Chap. 3 (3.1-3.2))
16:00-17:30 Dix, Lab

Thursday:

9:30-11:00 Dix, Answer Set Progr. (Chap.3 (3.3-3.4))
11:00-12:30 Dix, Lab
12:30-13:30 Lunch
13:30-16:00 Dix, Answer Set Progr. (Chap. 3 (3.5-3.6))
16:00-17:30 Dix, Lab

Friday:

9:30-11:00 Dix, Answer Set Progr. (Chap. 3 (3.7))
11:00-12:30 Schmidt, Description Logic (Chapter 4)
12:30-13:30 Lunch
13:30-17:00 Schmidt, Description Logic (Chapter 4)

Pre-course work

1.1 Elementary set theory

What is a set, a relation, a function, set operations (intersection, union, etc), properties of binary relations (reflexivity, symmetry, transitivity, etc).

Reference: Any book on the mathematical foundations of CS or discrete mathematics.

Chapter 5 in (Cormen, Leiserson, and Rivest 1989) (details below).

1.2 Propositional logic

Very simple representation language; expressively weak; modal logics and description logics are natural extensions.

Reference: (Kelly).

1.3 Constraint satisfaction problems

What is a CSP and what is its structure? N-Queens and Schur as instances. Problem representation via logic

Reference: Appropriate chapters in (Russell and Norvig 1995).

1.4 Graph theoretical problems

What is k-colourability, the hamiltonian circuit problem, k-cliques and vertex cover?

Reference: any introductory textbook on algorithms, e.g. (Cormen, Leiserson, and Rivest 1989).

Exercise sheet will be made available.

Reading material

Module does not follow a specific book: copies of the slides are made available. Part III follows partly some chapters of (Baral 2003)

Recommended reading material: listed on the course description webpage. List is not final and may change until the start of the module.

With the exception of books not as yet published, the books on the webpage are available in the Resources Centre or the Departmental Research Library.

No need to buy a book for this module.

Copies of any additional papers will be made available on the web.

Coursework

Exercises and assignments are of varying difficulty – those in the teaching week are aimed to consolidate the material of the lectures and are thus easier.

Some exercises and assignments are to be done with pencil and paper, some will require the use of tools (SPASS for the ML & DL part, DLV for the ASP part).

1 First Order (Predicate) Logic (FOL)

1.1 Propositional logic (PL)
1.2 First Order Logic (FOL)
1.3 Equality, function symbols: Yes or no?
1.4 Herbrand models
1.5 Some Examples
1.6 Undecidability

For the **post-course work** you will be given a selection of topics from which you choose one.
This work may involve writing a program, formalising problems, using reasoning tools for solving such problems, a case study on some research in one of the areas, or a mixture of these.

Assessment

5% assignments in the pre course week, 25% assignments in the teaching week, 30% post-course work, 40% exam

1.1 Propositional Logic

How to formalise problems in propositional logic?

Definition 1.1 (Propositional Logic $\mathcal{L}_{PL}$, Language $\mathcal{L} \subseteq \mathcal{L}_{PL}$)
The language $\mathcal{L}_{PL}$ of propositional logic consists of

- $\bot$ and $\top$: falsum and verum,
- $p, q, r, s_1, s_2, \ldots s_n, \ldots$: a countably infinite set $\mathcal{AT}$ of (propositional) constants,
- $\neg, \wedge, \vee, \rightarrow$: the connectives ($\neg$ is unary, all others binary),
- $(,)$: the parentheses, for unique readability.

Usually we are just considering finitely many constants. A subset of $\mathcal{AT}$ uniquely determines a language $\mathcal{L} \subseteq \mathcal{L}_{PL}$. This results (inductively) in the set of all $\mathcal{L}$-formulae $Fml_{\mathcal{L}}$.

Definition 1.2 (Semantics, valuation, Model)
A **valuation** v for a language $\mathcal{L} \subseteq \mathcal{L}_{PL}$ is a **mapping** from the set of propositional constants (determined by $\mathcal{L}$) and $\{\top, \bot\}$ in the set $\{\mathbf{true}, \mathbf{false}\}$ with $v(\bot) = \mathbf{false}$, $v(\top) = \mathbf{true}$.

Each valuation v can be uniquely extended to a function $\bar{v} : Fml_{\mathcal{L}} \to \{\mathbf{true}, \mathbf{false}\}$ such that:

- $\bar{v}(\neg p) = \begin{cases} \mathbf{true}, & \text{if } \bar{v}(p) = \mathbf{false}, \\ \mathbf{false}, & \text{if } \bar{v}(p) = \mathbf{true}. \end{cases}$
- $\bar{v}(\varphi \wedge \gamma) = \begin{cases} \mathbf{true}, & \text{if } \bar{v}(\varphi) = \mathbf{true} \text{ and } \bar{v}(\gamma) = \mathbf{true}, \\ \mathbf{false}, & \text{else.} \end{cases}$

- $\bar{v}(\varphi \vee \gamma) = \begin{cases} \mathbf{true}, & \text{if } \bar{v}(\varphi) = \mathbf{true} \text{ or } \bar{v}(\gamma) = \mathbf{true}, \\ \mathbf{false}, & \text{else.} \end{cases}$
- $\bar{v}(\varphi \to \gamma) =$
  $\begin{cases} \mathbf{true}, & \text{if } \bar{v}(\varphi) = \mathbf{false} \text{ or } (\bar{v}(\varphi) = \mathbf{true} \text{ and } \bar{v}(\gamma) = \mathbf{true}), \\ \mathbf{false}, & \text{else.} \end{cases}$

A valuation v therefore uniquely determines a $\bar{v}$. We call $\bar{v}$ a **$\mathcal{L}$-Structure** or simply **model** $\mathcal{A}_v$.

Thus a model determines for each formula whether it is true or false. In the following, we often use model and valuation synonymously.

Definition 1.3 (Truth of a formula, Tautology)

1. A formula $\varphi \in Fml_{\mathcal{L}}$ is **true under the valuation** v, if $\bar{v}(\varphi) = \mathbf{true}$. We write $\bar{v} \models \varphi$ or simply $v \models \varphi$.
2. A **theory** is a set of formulae: $T \subseteq Fml_{\mathcal{L}}$. v satisfies T if $\bar{v}(\varphi) = \mathbf{true}$ for all $\varphi \subseteq T$. We write $v \models T$.
3. A $\mathcal{L}$-formula φ is a **$\mathcal{L}$-tautology**, if for all valuations v on $\mathcal{L}$: $v \models \varphi$.

We often drop the underlying language if it is clear from context. But it is important to know what belongs to it.

Definition 1.4 (Consequences $Cn(T)$)
A formula φ **follows** from T, if all valuations v with $v \models T$ also satisfy $v \models \varphi$. We write: $T \models \varphi$.

We denote by $Cn_{\mathcal{L}}(T) := \{\varphi \in Fml_{\mathcal{L}} : T \models \varphi\}$, or simply $Cn(T)$, the **semantical consequence operator.**

Lemma 1.1 (Properties of $Cn(T)$)
The semantical consequence operator satisfies the following:

1. **T-Extendability:** $T \subseteq Cn(T)$,
2. **Monotony:** $T \subseteq T' \Rightarrow Cn(T) \subseteq Cn(T')$,
3. **Closure:** $Cn(Cn(T)) = Cn(T)$.

Lemma 1.2 ($\varphi \notin$ Cn(T))
$\varphi \notin Cn(T)$ *if and only if* there is a v with $v \models T$ and $\bar{v}(\varphi) = \mathbf{false}$.

Definition 1.5 (MOD(T), $Cn(\mathcal{U})$)
For $T \subseteq Fml_{\mathcal{L}}$, we denote by MOD($T$) the set of all $\mathcal{L}$-structures $\mathcal{A}$ that are models of T:

$$\text{MOD}(T) := \{\mathcal{A} : \mathcal{A} \models T\}.$$

If $\mathcal{U}$ is a set of models (valuations), then we are often interested in all formulae that are true in all models. We denote this set, by abuse of language, also by $Cn(\mathcal{U})$:

$$Cn(\mathcal{U}) := \{\varphi \in Fml_{\mathcal{L}} : \forall v \in \mathcal{U} : \bar{v}(\varphi) = \mathbf{true}\}.$$

MOD and Cn are obviously **dual** to each other:

$$Cn(\text{MOD}(T)) = Cn(T), \quad \text{MOD}(Cn(T)) = \text{MOD}(T).$$

Definition 1.6 (Completeness of a theory T)
T is called **complete**, if for each formula $\varphi \in Fml$: $T \models \varphi$ or $T \models \neg\varphi$.

This is not to be confused with the appropriate property of a valuation v: Each valuation is of course complete.

Definition 1.7 (Consistency of a theory)
T is called **consistent**, if there is a valuation v with $\bar{v}(\varphi) = \mathbf{true}$ for all $\varphi \in T$.

Lemma 1.3 (Ex Falso Quodlibet)
T is consistent *if and only if* $Cn(T) \neq Fml_{\mathcal{L}}$.

Calculi for PL

Definition 1.8 (Axiom, Inference rule)
Axioms of PL are the following formulae (to be read as **schemata**: ϕ, ψ can be arbitrary formulae):

- $\phi \rightarrow \top$, $\bot \rightarrow \phi$, $\neg\top \rightarrow \bot$, $\bot \rightarrow \neg\top$,
- $(\phi \wedge \psi) \rightarrow \phi$, $(\phi \wedge \psi) \rightarrow \psi$,
- $\phi \rightarrow (\phi \vee \psi)$, $\psi \rightarrow (\phi \vee \psi)$,
- $\neg\neg\phi \rightarrow \phi$, $(\phi \rightarrow \psi) \rightarrow ((\phi \rightarrow \neg\psi) \rightarrow \neg\phi)$,
- $\phi \rightarrow (\psi \rightarrow \phi)$, $\phi \rightarrow (\psi \rightarrow (\phi \wedge \psi))$.

The sole inference rule of PL is the following (called **Modus Ponens**):

$$MP : Fml \times Fml \to Fml : (\varphi, \varphi \to \psi) \mapsto \psi.$$

We also write:

$$\textbf{(MP)} \quad \frac{\varphi,\ \varphi \to \psi}{\psi}.$$

Here, φ, ψ can be arbitrary compound formulae.

Logical puzzles.

Example 1.1 (Who let the zebra out)
Five different people:

1. Having different **nationalities** (Norway, Ukraine, UK, Japan, Spain).
2. Owning houses of different **colours** (red, yellow, blue, ivory, green).
3. Having different **pets** (fox, horse, snails, dog, zebra).
4. Drinking different **drinks** (water, tea, milk, oj, coffee).
5. Having a **favourite sport** (soccer, cricket, wrestling, skiing, ice hockey).

Here are the known facts:

1. The Englishman lives in the red house.
2. The person living in the yellow house loves cricket.
3. The Ukrainian drinks tea.
4. The Spaniard owns the dog.
5. The Norwegian lives in the first house on the left.
6. The yellow house owner loves wrestling.
7. The man who loves soccer lives in the house next to the man with the fox.
8. The Norwegian lives next to the blue house.
9. The cricket lover owns snails.
10. The ice-hockey fan drinks orange juice.

11. The Japanese loves skiing.
12. The wrestling fan lives next to the house where the horse is kept.
13. Coffee is drunk in the green house.
14. The green house is to the right of the ivory house.
15. Milk is drunk in the middle house.

To which house belongs the zebra?
How many solutions are there (if any)?

1.2 First Order Logic

Why first order logic? What is missing in propositional logic?
What additional power give us quantifiers and variables?

Terms: Easier to express statements in the form $drink(curacao)$, $likes_sport(peter, cricket)$. Or to define new predicates: $sum(X, Y+1, Z) \leftrightarrow (sum(X, Y, Z') \wedge Z = Z'+1)$.

Variables: No need to write down all ground instances.

Quantification: Statements about all objects. **For all x there exists a y** such that something holds.

Expressibility: First Order Logic is much more expressible (in fact, undecidable in general).

Definition 1.9 (FOL $\mathcal{L}_{PL1}$, language $\mathcal{L} \subseteq \mathcal{L}_{PL1}$)
The language $\mathcal{L}_{PL1}$ of first order logic consists of

- $X, Y, Z, X_1, X_2, \ldots, x_n, \ldots$: a countable set $\mathcal{VAR}$ of variables,
- for all $k \in \mathbb{N}$: $p^k_1, p^k_2, \ldots, p^k_n, \ldots$ a countable set $\mathcal{PRED}^k$ of k-ary predicate symbols[a],
- for all $k \in \mathbb{N}$: $f^k_1, f^k_2, \ldots, f^k_n, \ldots$ a countable set $\mathcal{FUNCT}^k$ of k-ary function symbols,
- $\neg$, $\wedge$, $\vee$: the connectives ($\neg$ is unary, all others binary),
- (,): the parentheses,
- $\forall$, $\exists$: the **quantifiers**.

[a] The nullary predicate symbols are exactly the propositional constants $\mathcal{AT}$ of $\mathcal{L}_{PL}$. We assume, that $\bot$ and $\top$ are available.

Nullary function symbols are called **constants**, we omit the parentheses for better readability. In general, for a particular application, we only need a subset of all these predicate and function symbols. These will then determine the underlying language $\mathcal{L} \subseteq \mathcal{L}_{PL1}$ (as in Definition 1.1).

The set of underlying function and predicate symbols is called **signature** Σ. As was the case in propositional logic, we define inductively the notion of a $\mathcal{L}$-formula φ. But in contrast to propositional logic, we also define the notion of a $\mathcal{L}$-term t:

Term: $\mathcal{L}$-terms t are inductively defined as follows:

1. Each variable is a $\mathcal{L}$-term.
2. If f^k is a function symbol with arity k from $\mathcal{L}$ and $t_1, \ldots, t_k$ are $\mathcal{L}$-terms, then $f^k(t_1, \ldots, t_k)$ is a $\mathcal{L}$-term.

The set of all $\mathcal{L}$-terms built up from a set $X \subseteq \mathcal{VAR}$ is denoted by $Term_{\mathcal{L}}(X)$ or $Term_{\Sigma}(X)$. For $X = \emptyset$ we get the set of ground terms $Term_{\mathcal{L}}(\emptyset)$ or, shortly, $Term_{\mathcal{L}}$.

Formula: $\mathcal{L}$-formulae φ are also inductively defined:

1. If p^k is a predicate symbol with arity k from $\mathcal{L}$ and $t_1,\ldots,t_k$ are $\mathcal{L}$-terms, then $p^k(t_1,\ldots,t_k)$ is a $\mathcal{L}$-formula.
2. For each $\mathcal{L}$-formula φ the string $(\neg\varphi)$ is also a $\mathcal{L}$-formula.
3. For each $\mathcal{L}$-formula φ and ψ the strings $(\varphi\wedge\psi)$ and $(\varphi\vee\psi)$ are also $\mathcal{L}$-formulae.
4. For a variable X and a $\mathcal{L}$-formula φ, the following are also $\mathcal{L}$- formulae: $(\exists X\,\varphi)$ and $(\forall X\,\varphi)$.

Atomic $\mathcal{L}$-formulae are those built according to 1. We denote them by $At_{\mathcal{L}}(\mathcal{X})$ ($\mathcal{X}\subseteq\mathcal{VAR}$). The set of all $\mathcal{L}$-formulae wrt. $\mathcal{X}$ is denoted by $Fml_{\mathcal{L}}(\mathcal{X})$.

Positive formulae, $Fml^{+}_{\mathcal{L}}(\mathcal{X})$ are those, built with *1.*, *3.* and *4.*.

If a $\mathcal{L}$- formula φ is part of another $\mathcal{L}$- formula ψ, we call φ subformula of ψ

Example 1.2 (From semigroups to rings)

We consider $\mathcal{L}=\{0,1,+,\cdot,\leq,=\}$, where $0,1$ are constants, $+,\cdot$ binary function symbols and $\leq,=$ binary relation symbols. What is expressible in this language?

Ax 1:	$\forall X\forall Y\forall Z$	$X+(Y+Z)=(X+Y)+Z$
Ax 2:	$\forall X$	$X+0=0+X$
Ax 3:	$\forall X\forall Y$	$X+Y=0\ \wedge\ Y+X=0$
Ax 4:	$\forall X\forall Y$	$X+Y=Y+X$
Ax 5:	$\forall X\forall Y\forall Z$	$X\cdot(Y\cdot Z)=(X\cdot Y)\cdot Z$
Ax 6:	$\forall X\forall Y\forall Z$	$X\cdot(Y+Z)=X\cdot Y+X\cdot Z$
Ax 7:	$\forall X\forall Y\forall Z$	$X\cdot(Y+Z)\cdot X=Y\cdot X+Z\cdot X$

Ax 1 describes a *semi group*, Ax 1-2 a *monoid*, Ax 1-3 a *group*, Ax 1-4 a *abelian group*, Ax 1-7 a *ring*.

Definition 1.10 ($\mathcal{L}$-Structure $\mathcal{A}=(U_{\mathcal{A}},I_{\mathcal{A}})$)

A **$\mathcal{L}$-structure** (also called $\mathcal{L}$-Interpretation) is a pair $\mathcal{A}:=(U_{\mathcal{A}},I_{\mathcal{A}})$, where $U_{\mathcal{A}}$ is an arbitrary nonempty set, called the **universe** of $\mathcal{A}$. $I_{\mathcal{A}}$ is a mapping which assigns

- to each k-ary predicate symbol P^k from $\mathcal{L}$ a k-are predicate over $U_{\mathcal{A}}$,
- to each k-ary function symbol f^k from $\mathcal{L}$ a k-ary function on $U_{\mathcal{A}}$.

In other words: $I_{\mathcal{A}}$ is defined on the set of all predicate and function symbols of $\mathcal{L}$. The range of $I_{\mathcal{A}}$ is the set of predicates and functions over $U_{\mathcal{A}}$. We write:

$$I_{\mathcal{A}}(P)=P^{\mathcal{A}},\ I_{\mathcal{A}}(f)=f^{\mathcal{A}}.$$

Let φ be a $\mathcal{L}_1$-formula and $\mathcal{A} := (U_\mathcal{A}, I_\mathcal{A})$ a $\mathcal{L}$-structure. $\mathcal{A}$ is compatible to φ if $I_\mathcal{A}$ is defined for all symbols occurring in φ: $\mathcal{L}_1 \subseteq \mathcal{L}$.

Definition 1.11 (Variable Assignment ρ)

A **variable assignment** ρ over a $\mathcal{L}$-structure $\mathcal{A} = (U_\mathcal{A}, I_\mathcal{A})$ is a function

$$\rho : \mathcal{VAR} \to U_\mathcal{A};\ x \mapsto \rho(x).$$

Definition 1.12 (Semantics of FOL)

Let φ a formula, $\mathcal{A}$ a structure compatible to φ and ρ a variable assignment over $\mathcal{A}$. For each term t (built from parts from φ) we define inductively the **value of t in the structure $\mathcal{A}$**. We denote this value by $\mathcal{A}(t)$.

1. For a variable X: $\mathcal{A}(X) := \rho(X)$.
2. If t has the form $t = f^k(t_1, \ldots, t_k)$, where $t_1, \ldots, t_k$ are terms and

f^k is a k-ary function symbol, we define $\mathcal{A}(t) := f^\mathcal{A}(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k))$.

We are now defining the truthvalue of a formula φ in a structure $\mathcal{A}$:

1. If $\varphi := P^k(t_1, \ldots, t_k)$ with terms $t_1, \ldots, t_k$ and k-ary predicate symbol P^k, then

$$\mathcal{A}(\varphi) := \begin{cases} \textbf{true}, & \text{if } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)) \in P^\mathcal{A}, \\ \textbf{false}, & \text{else.} \end{cases}$$

2. If $\varphi := \neg\psi$, then

$$\mathcal{A}(\varphi) := \begin{cases} \textbf{true}, & \text{if } \mathcal{A}(\psi) = \textbf{false}, \\ \textbf{false}, & \text{else.} \end{cases}$$

3. If $\varphi := (\psi \wedge \eta)$, then

$$\mathcal{A}(\varphi) := \begin{cases} \textbf{true}, & \text{if } \mathcal{A}(\psi) = \textbf{true} \textit{ and } \mathcal{A}(\eta) = \textbf{true}, \\ \textbf{false}, & \text{else.} \end{cases}$$

4. If $\varphi := (\psi \vee \eta)$, then

$$\mathcal{A}(\varphi) := \begin{cases} \textbf{true}, & \text{if } \mathcal{A}(\psi) = \textbf{true} \textit{ or } \mathcal{A}(\eta) = \textbf{true}, \\ \textbf{false}, & \text{else.} \end{cases}$$

5. If $\varphi := \forall X\, \psi$, then

$$\mathcal{A}(\varphi) := \begin{cases} \textbf{true}, & \text{if for } \textit{all } d \in U_\mathcal{A} : \mathcal{A}_{[X/d]}(\psi) = \textbf{true}, \\ \textbf{false}, & \text{else.} \end{cases}$$

6. If $\varphi := \exists X\, \psi$, then

$$\mathcal{A}(\varphi) := \begin{cases} \textbf{true}, & \text{if } \textit{there is a } d \in U_\mathcal{A} : \mathcal{A}_{[X/d]}(\psi) = \textbf{true}, \\ \textbf{false}, & \text{else.} \end{cases}$$

In cases 5. and 6. we use the notation $[X/d]$, which is defined as follows. For $d \in U_\mathcal{A}$ let $\mathcal{A}_{[X/d]}$ the structure $\mathcal{A}'$, which is almost everywhere identical to $\mathcal{A}$, except for the definition of $X^{\mathcal{A}'}$: $X^{\mathcal{A}'} := d$ (independently of whether $I_\mathcal{A}$ is defined for X or not). We write:

- $\mathcal{A} \models \varphi[\rho]$ for $\mathcal{A}(\varphi) = \textbf{true}$: $\mathcal{A}$ is a model for φ wrt. ρ.
- If φ does not contain any free variables, then $\mathcal{A} \models \varphi[\rho]$ independently of ρ. In that case we omit ρ.
- If there is at least one model of φ, we call φ satisfiable (or consistent).

Definition 1.13 (Truth of a formula, Tautology)

1. A formula $\varphi \in Fml_{\mathcal{L}}$ is **true wrt. the assignment ρ in the structure** $\mathcal{A}$, if $\varphi[\rho] = \mathbf{true}$. We write $\mathcal{A} \models \varphi[\rho]$ or simply $\mathcal{A} \models \varphi$ if φ does not contain any free variables.
2. A **theory** is a set of formulae: $T \subseteq Fml_{\mathcal{L}}$. $\mathcal{A}$ satisfies T if $\varphi[\rho] = \mathbf{true}$ in $\mathcal{A}$ for all $\varphi \subseteq T$ and all variable assignments ρ. Often, then theories we consider do not contain any free variables. then the variable assignments ρ do not matter. We write $\mathcal{A} \models T$.
3. A $\mathcal{L}$-formula φ is a **$\mathcal{L}$-tautology**, if for all $\mathcal{L}$-structures $\mathcal{A}$: $\mathcal{A} \models \varphi$.

We often drop the underlying language if it is clear from context. But it is important to know what belongs to it.

Homework 1 (Equivalence)

Give equivalent but simpler formulae:

1. $\neg p(c) \rightarrow (p(c) \vee p(b))$,
2. $(\neg p(c) \wedge p(b)) \rightarrow p(c)$,
3. $(\neg p(c) \wedge \neg p(b)) \rightarrow \bot$,
4. $p(c) \rightarrow (\neg p(c) \vee p(b))$,
5. $(\neg placed(I) \wedge queen(I)) \rightarrow \bot$.

Definition 1.14 (Consequences $Cn(T)$)

A formula φ **follows** from T, if for all structures $\mathcal{A}$ with $\mathcal{A} \models T$ also $\mathcal{A} \models \varphi$ holds. We write: $T \models \varphi$.

We denote by $Cn_{\mathcal{L}}(T) := \{\varphi \in Fml_{\mathcal{L}} : T \models \varphi\}$, or simply $Cn(T)$, the **semantical consequence operator.**

Lemma 1.4 (Properties of $Cn(T)$)

The semantical consequence operator satisfies the following:

1. **T-Extendability:** $T \subseteq Cn(T)$,
2. **Monotony:** $T \subseteq T' \Rightarrow Cn(T) \subseteq Cn(T')$,
3. **Closure:** $Cn(Cn(T)) = Cn(T)$.

Lemma 1.5 ($\varphi \notin$ Cn(T))

$\varphi \notin Cn(T)$ *if and only if* there is a $\mathcal{A}$ with $\mathcal{A} \models T$ and $\mathcal{A} \models \neg\varphi$.

Definition 1.15 (MOD(T), $Cn(\mathcal{U})$)
For $T \subseteq Fml_{\mathcal{L}}$, we denote by MOD($T$) the set of all $\mathcal{L}$-structures $\mathcal{A}$ that are models of T:

$$\text{MOD}(T) := \{\mathcal{A} : \mathcal{A} \models T\}.$$

If $\mathcal{U}$ is a set of models, then we are often interested in all formulae that are true in all models. We denote this set, by abuse of language, also by $Cn(\mathcal{U})$:

$$Cn(\mathcal{U}) := \{\varphi \in Fml_{\mathcal{L}} : \forall v \in \mathcal{U} : \bar{v}(\varphi) = \textbf{true}\}.$$

MOD and Cn are obviously **dual** to each other:

$$Cn(\text{MOD}(T)) = Cn(T), \quad \text{MOD}(Cn(T)) = \text{MOD}(T).$$

Definition 1.16 (Completeness of a theory T)
T is called **complete**, if for each formula $\varphi \in Fml$: $T \models \varphi$ or $T \models \neg\varphi$.

This is not to be confused with the appropriate property of a structure $\mathcal{A}$: Each structure is of course complete. Any sentence is either true or false in a structure.

Definition 1.17 (Consistency of a theory)
T is called **consistent**, if there is a structure $\mathcal{A}$ with $\mathcal{A} \models \varphi$ for all $\varphi \in T$.

Lemma 1.6 (Ex Falso Quodlibet)
T is consistent *if and only if* $Cn(T) \neq Fml_{\mathcal{L}}$.

How can one formalise Example 1.1 in FOL?

1. unary predicates $country(\cdot)$, $color(\cdot)$, $nationality(\cdot)$, $pet(\cdot)$, $drink(\cdot)$, $sport(\cdot)$, together with constants representing the countries, sports etc.
2. binary predicate $next_house(\cdot,\cdot)$,
3. 6-ary predicate $together(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot,)$ (number, sport, country, color, pet, drink)
4. Axioms describing the facts:
   $\exists X_1,X_2,X_3,X_4\ together(X_1,X_2,english,red,X_3,X_4,)$, ... Statements (5), (7), (8), (12), (14) and (15) require the ordering of the houses.

$$\exists X_1,X_2,X_3,X_4 \quad together(1,X_1,norwegian,X_2,X_3,X_4)$$
$$\forall X,X' \quad (\quad (\exists X_1,X_2,X_3,X_4\ together(X,soccer,X_1,X_2,X_3,X_4) \wedge next_house(X,X'))$$
$$\rightarrow \exists X_1,X_2,X_3,X_4\ together(X',X_1,X_2,X_3,fox,X_4) \ )$$

How to formalise the *next_house* predicate? We can have constants $1,2,3,4,5,6$ (1 is leftmost, etc) and a function $+$ with $1+1=2$, $2+1=3$, $3+1=4$, $4+1=5$, $5+1=6$, $6+1=6$.

Why is this approach (with the predicate *together*) not the optimal one?

Better use one parameter as **anchor** and group the others around it.

1. Introduce constants and predicates for the numbers, animals, drinks, etc. E.g.:

$$animal(snail),\ animal(zebra),\ \ldots$$
$$house(c_1), house(c_2),\ \ldots$$
$$drink(tea), drink(oj),\ \ldots$$
$$country(uk),\ \ldots$$
$$color(red),\ \ldots$$

2. Formalise that exactly one color, one country, one pet etc

belongs to each house. E.g.:

$$\begin{aligned} has_color(H,C) &\rightarrow house(H) \wedge color(C) \\ has_color(H,red) \vee \ldots \vee has_color(H,green) &\leftrightarrow house(H) \\ has_color(c_1,C) \vee \ldots \vee has_color(c_5,C) &\leftrightarrow color(C) \\ \bot &\leftrightarrow has_color(H,C) \wedge \\ &\quad \wedge has_color(H,C) \wedge \neg C \doteq C' \end{aligned}$$

3. Formalise the statements. E.g.:

$house(H) \wedge has_country(H,uk) \wedge has_color(H,red)$

$has_country(c_1,norway)$

$house(H) \wedge house(H') \wedge next(H,H') \wedge has_sport(H,wrestling) \wedge has_animal(H$

Example 1.3 (The natural numbers in different languages)

- $\mathcal{N}_{Pr} = (\mathbb{N}, 0^{\mathcal{N}}, +^{\mathcal{N}}, =^{\mathcal{N}})$ (Presburger Arithmetic),
- $\mathcal{N}_{PA} = (\mathbb{N}, 0^{\mathcal{N}}, +^{\mathcal{N}}, \cdot^{\mathcal{N}}, =^{\mathcal{N}})$ (**Peano Arithmetic**),
- $\mathcal{N}_{PA'} = (\mathbb{N}, 0^{\mathcal{N}}, 1^{\mathcal{N}}, +^{\mathcal{N}}, \cdot^{\mathcal{N}}, =^{\mathcal{N}})$ (Variant of Peano Arithmetic).

The object is the same (natural numbers) but looked at in different languages $\mathcal{L}_{Pr}$, $\mathcal{L}_{PA}$, $\mathcal{L}_{PA'}$.

The bigger the language, the more can be expressed.

Can one express strictly more with $\mathcal{L}_{PA'}$ than with $\mathcal{L}_{PA}$?

No, because we can define $1^{\mathcal{N}}$ with a $\mathcal{L}_{PA}$ formula: there is a $\mathcal{L}_{PA}$ formula $\phi(\mathbf{x})$ s.t. for all variable assignments ρ:

$$\mathcal{N}_{PA'} \models_\rho \phi(\mathbf{X}) \text{ iff } \rho(\mathbf{X}) = 1^{\mathbb{N}}$$

One can choose for $\phi(\mathbf{X})$ the formula $\forall Z(Z \cdot \mathbf{X} = Z)$.

Note that:

1. Any calculus for FOL derives, for a given set of formulae (axioms) the set of formulae that are true **in all models** of the axioms. And there are very many of these models.
2. We cannot restrict to just one model (like the natural numbers) and expect our calculus to compute us all formulae true in this model (see subsection 1.6).

3. In order to implement the derivability relation, we have to somehow deal with ground terms. The fact that a formula follows from a theory can be equivalently stated as the **unsatisfiability** of a set of formulae. This means we have to attempt to construct a model and to show that this is not possible (see subsection 1.3).
4. Efficient methods in automated reasoning require that the input formulae are in a particular form, **clausal normal form**. But transforming formulae in this normalform is expensive (most probably exponential).

Homework 2 (Properties of relations)
How can one express the properties of symmetry, transitivity, anti-symmetry, reflexivity of a binary relation?

How the fact that a partial order is dense and has two endpoints?

Homework 3 (Zebra)
Express Example 1.1 in FOL with predicates
$color, country, sport, pet, drink$ and $house, left, right, next$ **with the house number as anchor**.

Such that the satisfying models correspond to solutions of the puzzle.

1.3 Equality, Function symbols: Yes or no?

Equality is an important binary relation which has a unique interpretation in any universe. Terms are also important for representing knowledge. Do they complicate matters?

FOL with equality: equality is built-in as a symbol "$\doteq$" and interpreted as **true equality** in each structure.

FOL without equality: there is no binary relation that is interpreted as equality. But there might be a binary symbol eq together with some axioms describing it. **But it will never capture true equality completely.**

We introduce a binary relation-symbol *eq* and require that it satisfies the following axioms with respect to an underlying language $\mathcal{L}$:

$\forall X\ eq(X,X)$

for all function-symbols f of suitable arity:

$\forall X_1 \ldots X_n, Y_1 \ldots y_n$

$(eq(X_1,Y_1) \ldots eq(X_n,Y_n)) \rightarrow eq(f(X_1,\ldots,X_n), f(Y_1,\ldots,Y_n))$

for all predicate-symbols P of suitable arity:

$\forall X_1 \ldots X_n Y_1 \ldots Y_n$

$(eq(X_1,Y_1) \ldots eq(X_n,Y_n)) \rightarrow (p(X_1,\ldots,X_n) \rightarrow p(Y_1,\ldots,Y_n))$.

This set, is denoted by $\mathrm{EQ}_{\mathcal{L}}$. It can be shown that transitivity and symmetry of *eq* follow from these axioms.

What exactly is the influence of function symbols? Can't we get rid of them and only allow constants?

- Function symbols immediately introduce infinitely many terms and lead to many problems (actually, to undecidability).
- So why not allowing variables and constants, but no function symbols?

  This approach is taken in databases. It ensures that the set of ground terms is always finite.

Example 1.4 (Loops)
Example showing difficulties with detecting loops in evaluating expressions.

- $p(f(X)) \rightarrow p(X)$.
- $\neg even(X+1) \rightarrow even(X)$
  $even(0)$.
- $odd(X+1) \rightarrow even(X)$,
  $even(X+1) \rightarrow odd(X)$.

1.4 Herbrand models

Why not restricting the universes of the models we are interested in. Just finite models, or just models built up from the underlying language.

- Classical logic formalises the notion of a formula ϕ *entailed* by the theory T: truth in **all** models of T.
- Doesn't it make sense to look only at models generated by the underlying language?
- After all we are not interested in models that contain elements which are not representable as terms in our language.
- Can we enforce this by adding suitable axioms?

These requirements are usually called *unique names assumption* and *domain closure assumption*:

Definition 1.18 (UNA and DCA)
Let a language $\mathcal{L}$ be given. We understand by the **unique names assumption** the restriction to those models I, where syntactically different ground $\mathcal{L}$-terms t_1, t_2 are interpreted as nonidentical elements: t_1^I is not identical to t_2^I.

By the **domain closure assumption** we mean the restriction to those models I where for any element a in I there is a $\mathcal{L}$-term t that represents this element: $a = t^I$.

Often we are only interested in models whose universes consist of exactly the ground terms of the underlying language. These models have a special name:

Definition 1.19 (Herbrand-Model)
A model $\mathcal{A}$ is called Herbrand-Model wrt. a language $\mathcal{L}$, the universe of $\mathcal{A}$ consists of $Term_{\mathcal{L}}(\emptyset)$ and the function symbols f_i^k are interpreted as follows:

$$f_i^{k\mathcal{A}} : Term_{\mathcal{L}}(\emptyset) \times \ldots \times Term_{\mathcal{L}}(\emptyset) \rightarrow Term_{\mathcal{L}}(\emptyset);\ (t_1, \ldots, t_k) \mapsto f_i^k(t_1, \ldots, t_k)$$

We write $T \models_{Herb} \phi$ if each Herbrand model of T is also a model of ϕ.

Theorem 1.1
If T is universal and ϕ existential, then

$$T \models \phi \text{ if and only if } T \models_{Herb} \phi$$

Isn't $T \models_{Herb} \phi$ much easier, because we can restrict to Herbrand models?

No!

Homework 4
Show that the above theorem is not true when the assumption that T is universal is dropped.

The following theorem is the basis of many computational methods of automated reasoning. In a sense, it reduces first order logic to propositional logic.

Theorem 1.2 (Herbrand)
Let T universal and ϕ without quantifiers. Then

$$T \models \exists\phi \quad \text{iff} \quad \text{there are } t_1, \ldots, t_n \in Term_{\mathcal{L}}(\emptyset) \text{ s.t.:: } T \models \phi(t_1) \vee \ldots \vee \phi(t_n)$$

A slightly different formulation:

Let M be a set of clauses of FOL (formulae of the form $P_1(t_1) \vee \neg P_2(t_2) \vee \ldots \vee P_n(t_n)$ where $t_i \in Term_{\mathcal{L}}(X)$). Then:
M is unsatisfiable if and only if there is a finite unsatisfiable set M_{inst} of ground instances of M.

In automated reasoning one is often interested in tasks of the form

$$M \models \exists X_1, \ldots X_n \bigwedge_i \phi_i.$$

Then

$$M \cup \{\neg \exists X_1, \ldots X_n \bigwedge_i \phi_i\}$$

is a set of clauses which is unsatisfiable, if and only if the relation above holds.

Databases: Here only variables, but no function symbols are allowed. An additional requirement, **safety**, ensures that a **bottom-up approach** can be applied and terminates.

Grounding: When no function symbols are present, the complete **grounding** of the theory leads to a propositional theory. However, grounding should be avoided as it leads to huge theories. Special mechanisms dealing with the efficient handling of variables (instantiation-by-need) ensure this.

1.5 Some examples

What are the limitations of FOL in terms of handling non-trivial, nonlogical problems? Why can transitive closure not be handled in FOL?

Example 1.5 (N-Queens)
Given a chessboard of size N. How can one place N queens on it, so that they do not attack each other?

Example 1.6 (Graph Colouring)
Given a graph G. How many (if any) possibilities are there to colour it with a certain number of colours?

Example 1.7 (Transitive Closure)
Suppose you are given a binary relation r. Two elements e and e' in it are called connected, if there are elements $e_1, e_2, \ldots, e_n$ such that the following holds: $r(e, e_1)$, $r(e_1, e_2), \ldots,$ $r(e_n, e')$.

The **transitive closure** of a set M of elements in a binary relation is the set of all elements that are connected to elements in M.

Is the transitive closure expressible in FOL?

1.6 Undecidability of FOL

How expressive is logic? Which problems can be solved by theorem provers?

Let us consider the language of *arithmetic* $\mathcal{L}_{Ar}$ which contains: 0 (a constant), s (a unary function-symbol), eq (a two-ary relation-symbol) and $\oplus$, $\otimes$ (ternary relation-symbols).

We have in mind to axiomatise the theory of natural numbers. Before we do so we introduce the following abbreviation. The formula $\exists!Z\phi(Z)$ stands for

$$\exists Z(\phi(Z) \wedge \forall Y \phi(Y) \rightarrow eq(Y,Z)).$$

Definition 1.20 (Arithmetic Ar_{fin})

Ar_{fin} is the finite set consisting of $EQ_{\mathcal{L}_{Ar}}$ and the following axioms:

$$\begin{array}{ll} \forall X\forall Y\exists!Z & \oplus(X,Y,Z) \\ \forall X & \oplus(X,0,X) \\ \forall X\forall Y\forall Z & \oplus(X,Y,Z) \rightarrow \oplus(X,s(X),s(Z)) \end{array}$$

$$\begin{array}{ll} \forall X\forall Y\exists!Z & \otimes(X,Y,Z) \\ \forall X & \otimes(X,0,0) \\ \forall X\forall Y\forall Z\forall Z' & \otimes(X,Y,Z) \rightarrow (\otimes(X,s(Y),Z') \wedge \oplus(Z,X,Z')) \end{array}$$

The set of natural numbers $\mathcal{N} := (\mathbb{N}, 0^{\mathcal{N}}, s^{\mathcal{N}}, \oplus^{\mathcal{N}}, \otimes^{\mathcal{N}}, eq^{\mathcal{N}})$ is a model of Ar_{fin}. Here $0^{\mathcal{N}}$ is the "true" 0, $s^{\mathcal{N}}$ is the successor-function, $\oplus^{\mathcal{N}}$ is addition and $\otimes^{\mathcal{N}}$ is multiplication (viewed as relations), $eq^{\mathcal{N}}$ is identity.

We note the following facts:

1. The set $\{\phi : Ar_{fin} \vdash \phi\}$ is recursively enumerable but not recursive.
2. The set $\{\phi : \mathcal{N} \models \phi\}$ is not even recursively enumerable.

We even have

Theorem 1.3 (Gödel)

No set of formulae containing Ar_{fin} and having $\mathcal{N}$ as model, is recursive.

Every recursively enumerable set of formulae Φ that contains Ar_{fin} and has $\mathcal{N}$ as a model, is incomplete, i.e. there is ψ with: $\mathcal{N} \models \psi$ but $\Phi \not\models \psi$.

Therefore no complete axiomatisation of $\mathcal{N}$ is possible.

Note that, although $\mathcal{N}$ formally is not a Herbrand model, it is isomorphic to such a model. In fact, the axioms immediately imply that there is, up to isomorphy, only one single Herbrand-model of Ar_{fin} with respect to $\mathcal{L}_{Ar}$. Therefore to determine if a formula is true in all Herbrand-models of Ar_{fin} is just as complicated as the theory of $\mathcal{N}$ itself. $\mathcal{N}$ contains, for example, famous statements (or their negation) from number theory like Goldbach-conjecture or Fermat's last theorem.

Theorem 1.4 (Undecidability of FOL)

FOL is undecidable in general, when at least one binary predicate symbol is contained in the language.

FOL is decidable when there are only finitely many unary function symbols.

What is a way out of this dilemma? We want to be as expressive as possible, but we do not want to face an unsolvable problem.

Subsystems: Many decidable subsystems of FOL have been defined. By restricting quantification, function symbols.

Description Logics: Several constructs are restricted. Relativised quantification and predicates are unary or binary.

Databases: Variables allowed, but no function symbols. Restrictions to ensure the Herbrand universe is finite.

Modal Logic: In between propositional logic and FOL.

ASP: In general, also undecidable (even worse than FOL), because minimal models are of higher complexity. But the database version is decidable.

References

Baral, C. (2003). *Knowledge representation, reasoning and declarative problem solving with Answer sets*. Cambridge University Press.

Cormen, T. H., C. E. Leiserson, and R. L. Rivest (1989). *Introduction to Algorithms*. McGraw-Hill.

Kelly, J. *The essence of logic*. PHI.

Russell, S. J. and P. Norvig (1995). *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice-Hall.