Methods of Artificial Intelligence

Kai-Uwe Kühnberger & Nico Potyka

Theorem Proving Advanced

UNIVERSITÄT OSNABRÜCK

Tentative Schedule of this Course

1.	23.10.2017	Introduction
	Part I: Foundations & Planning	
2.	30.10.2017	Local Search
3.	06.11.2017	Constraint Satisfaction Problems Advanced
4.	13.11.2017	Theorem Proving Advanced
5.	20.11.2017	Planning
	Part II: Reasoning & Learning	
6.	27.11.2017	Knowledge Representation
7.	04.12.2017	Midterm
8.	11.12.2017	Reasoning over Space and Time
	18.12.2017	Uncertain Reasoning and Learning Basics
9.	08.01.2018	ML 1: SVMs and Random Forests
10.	15.01.2018	ML 2: Reinforcement Learning
	Part III: Cognitive Architectures & Games	
11.	22.01.2018	Cognitive Architectures
12.	29.01.2018	Games Advanced
13.	05.01.2018	Repetition
14.	06.02.2018	Final Exam

Overview

- Organizational Remarks
- Forms of Reasoning: Deduction, Induction, Abduction
- Repetition: Propositional and First-Order Logic
- Prove Techniques in Mathematics
  - Direct Proofs
  - Indirect Proofs
- Something to Relax: A Logic of Chords
- Resolution Calculus
  - Clause Form
  - Examples of Resolution
- Potential Problems of Using Logic for Reasoning
- Possible Cognitive Mechanisms for Reasoning

Organization: Midterm Examination

- The midterm exam is scheduled for December 5th, 2017.
  - Rooms: 15/323 and 15/324
  - Time: Two groups, first group 12 o’clock, second group 13 o’clock.
- Features of the midterm:
  - The exam will be mostly a multiple-choice test.
  - Wrong answers result in negative credits. For each question with *n* many possible answers you **cannot** earn negative credits.
  - There will be probably a few questions where you have to type in something.
  - Some of the questions will be taken from the weekly quizzes.
  - You will have 50 minutes to solve the problems.
  - You can expect approximately 30 questions.
- There will be a test exam online one week before the exam. Notice that it will be not sufficient to practice only the test exam.

Organization: 27th and 28th of November

- On the 27th and 28th of November I and Nico Potyka will be busy with the hiring committee for the professorship “Cognitive Natural Language Processing and Communication”
- We plan the following:
  - On 27th of November: we would ask you to attend at least one scientific talk and one lecture of a candidate for the professorship (Schedule: see next slide)
  - On the 28th of November: Lucas Bechberger will present work on conceptual spaces
- Important: the talks of the candidates for the professorship will not be relevant for the exams.
- Lucas Bechberger’s presentation on conceptual spaces will be relevant for exams.

Organization: 27th and 28th of November

Zeitplan Anhörungen BV Cognitive Natural Language Processing & Communication

Montag, 27.11.2017 **Raum 11/E08**

09:00 – 09:45 Uhr Vortrag **Schneider** (Konstanz/Winterthur)

10:00 – 10:45 Uhr Probelehrveranstaltung **Schneider**

11:00 – 12:00 Uhr Gespräch mit der Berufungskommission

12:30 – 13:15 Uhr Vortrag **Markert** (Hannover)

13:30 – 14:15 Uhr Probelehrveranstaltung **Markert**

14:30 – 15:30 Uhr Gespräch mit der Kommission

15:30 – 16:15 Uhr Vortrag **Staudte** (Saarbrücken)

16:30 – 17:15 Uhr Probelehrveranstaltung **Staudte**

17:30 – 18:30 Uhr Gespräch mit der Kommission

Organization: 27th and 28th of November

<u>Zeitplan Anhörungen BV Cognitive Natural Language Processing & Communication</u>

<u>Dienstag, 28.11.2017</u> **<u>Raum 11/E08</u>**

09:00 – 09:45 Uhr Vortrag **Rettinger** (Karlsruhe)

10:00 – 10:45 Uhr Probelehrveranstaltung **Rettinger**

11:00 – 12:00 Uhr Gespräch mit der Berufungskommission

12:30 – 13:15 Uhr Vortrag **Bergmann** (Bielefeld)

13:30 – 14:15 Uhr Probelehrveranstaltung **Bergmann**

14:30 – 15:30 Uhr Gespräch mit der Kommission

15:30 – 16:15 Uhr Vortrag **Franke** (Tübingen)

16:30 – 17:15 Uhr Probelehrveranstaltung **Franke**

17:30 – 18:30 Uhr Gespräch mit der Kommission

Forms of Reasoning:
Deduction, Abduction, Induction

Theorem Proving,
Sherlock Holmes,
and All Swans are White...

Basic Types of Inferences: Deduction

- Deduction: Derive a conclusion from given axioms (“knowledge”) and facts (“observations”).

- Example:

All humans are mortal.	(axiom)
Socrates is a human.	(fact/ premise)
Therefore, it follows that Socrates is mortal.	(conclusion)

- Remarks
  - The conclusion can be derived by applying the *modus ponens* inference rule (Aristotelian logic).
  - Theorem proving is based on deductive reasoning techniques.

Basic Types of Inferences: Induction

- Induction: Derive a general rule (axiom) from background knowledge and observations.
- Example:

Socrates is a human	(background knowledge)
Socrates is mortal	(observation/ example)
. . .	
Therefore, I hypothesize that all humans are mortal	(generalization)

- Remark
  - Induction means to infer generalized knowledge from example observations. Therefore, induction is *the* inference mechanism for (machine) learning.

Basic Types of Inferences: Abduction

- Abduction: From a known axiom (theory) and some observation, derive a premise (inference to the best explanation).
- Example:

All humans are mortal	(theory)
Socrates is mortal	(observation)
Therefore, Socrates must have been a human	(diagnosis)

- Remarks:
  - Abduction is typical for diagnostic and expert systems.
  - Simple medical diagnosis:

    If one has the flue, one has moderate fever.

    Patient X has moderate fever.

    Therefore, he has the flue.

Remark Concerning Reasoning

- Deduction, induction, and abduction are not the only possible reasoning mechanisms. In the following list more are mentioned:

Types of Reasoning	Corresponding Formalisms
Deduction	Classical Logic
Induction	Inductive Logic Programming / Machine Learning
Abduction	Extensions of Logic Programming
Analogical Reasoning	SME, LISA, AMBR, HDTP
Non-Monotonic Reasoning	Answer Set Programming
Frequency-Based Reasoning	Bayesian Reasoning
Vague and Uncertain Reasoning	Fuzzy and Probabilistic Reasoning
Reasoning about Possibility and Necessity	Modal Logic
Etc.	Etc.

- Further difficulties for classical reasoning paradigms: complexity problems, robustness, integration.

Deduction

Deductive inferences are also called theorem proving or logical inference.

- Deduction is truth preserving: If the premises (axioms and facts) are true, then the conclusion (theorem) is true.

To perform deductive inferences on a machine, a calculus is needed:

- A calculus is a set of syntactical rewriting rules defined for some (formal) language. These rules must be sound and complete.

We will focus on the resolution calculus for first-order logic (FOL).

- $\Rightarrow$ Syntax of FOL.
- $\Rightarrow$ Semantics of FOL.

Repetition: Propositional Logic and First-Order Predicate Logic

Some rather Abstract Stuff…

Repetition: Propositional Logic

Formulas:

Given is a countable set of atomic propositions *AtProp* = {*p*,*q*,*r*,...}. The set of well-formed formulas *Form* of propositional logic is the smallest class such that it holds:

$\forall p \in AtProp$:	$p \in Form$
$\forall \varphi, \psi \in Form$:	$\varphi \wedge \psi \in Form$
$\forall \varphi, \psi \in Form$:	$\varphi \vee \psi \in Form$
$\forall \varphi \in Form$:	$\neg\varphi \in Form$

Semantics:

Truth tables (compare the course "Foundations of Logic I")

A formula φ is valid if φ is true for all possible assignments of the atomic propositions occurring in φ.

A formula φ is satisfiable if φ is true for some assignment of the atomic propositions occurring in φ.

Models of propositional logic are specified by Boolean algebras.

Repetition: First-Order Predicate Logic

A signature

$$\Sigma = (c_1,\ldots,c_n,f_1,\ldots,f_m,R_1,\ldots,R_l)$$

defines the ‘vocabulary’ of a first-order logical language

$c_1,\ldots,c_n$	refer to constants / names
$f_1,\ldots,f_m$,	refer to function symbols (+ *arity*)
$R_1,\ldots,R_l$	refer to relation symbols (+ *arity*)

We can view *arity* as a function

$$\mathit{arity}: \{f_1,\ldots,f_m,R_1,\ldots,R_l\} \rightarrow \mathbb{N}$$

Examples:

- 2 + 3 = 5 — “2” and “5” are constants, “+” is a binary function symbol, “=” a binary relation symbol
- *give(mary,john,book)* — “Mary gives John a book”: “mary”, “john”, and “book” are constants, “give” is a three-ary relation

Repetition: First-Order Predicate Logic

- Syntactically well-formed first-order formulas for a signature $\Sigma = (c_1,\ldots,c_n,f_1,\ldots,f_m,R_1,\ldots,R_l)$ are inductively defined
  - This means, they are defined as the smallest class of formulas that is closed under certain construction conditions
- The set T_Σ of Terms is the smallest class such that:
  - A variable $x \in Var$ is a term, where Var is a countable set of variables
  - A constant $c_i \in \{c_1,\ldots,c_n\}$ is a term.
  - If $f_i \in \{f_1,\ldots,f_m\}$ is a function symbol of arity r and $t_1,\ldots,t_r$ are terms then $f_i(t_1,\ldots,t_r)$ is a term.
- Examples:
  - 2 + 3 is a term
  - 2 +3 = 5 is not a term (why?)

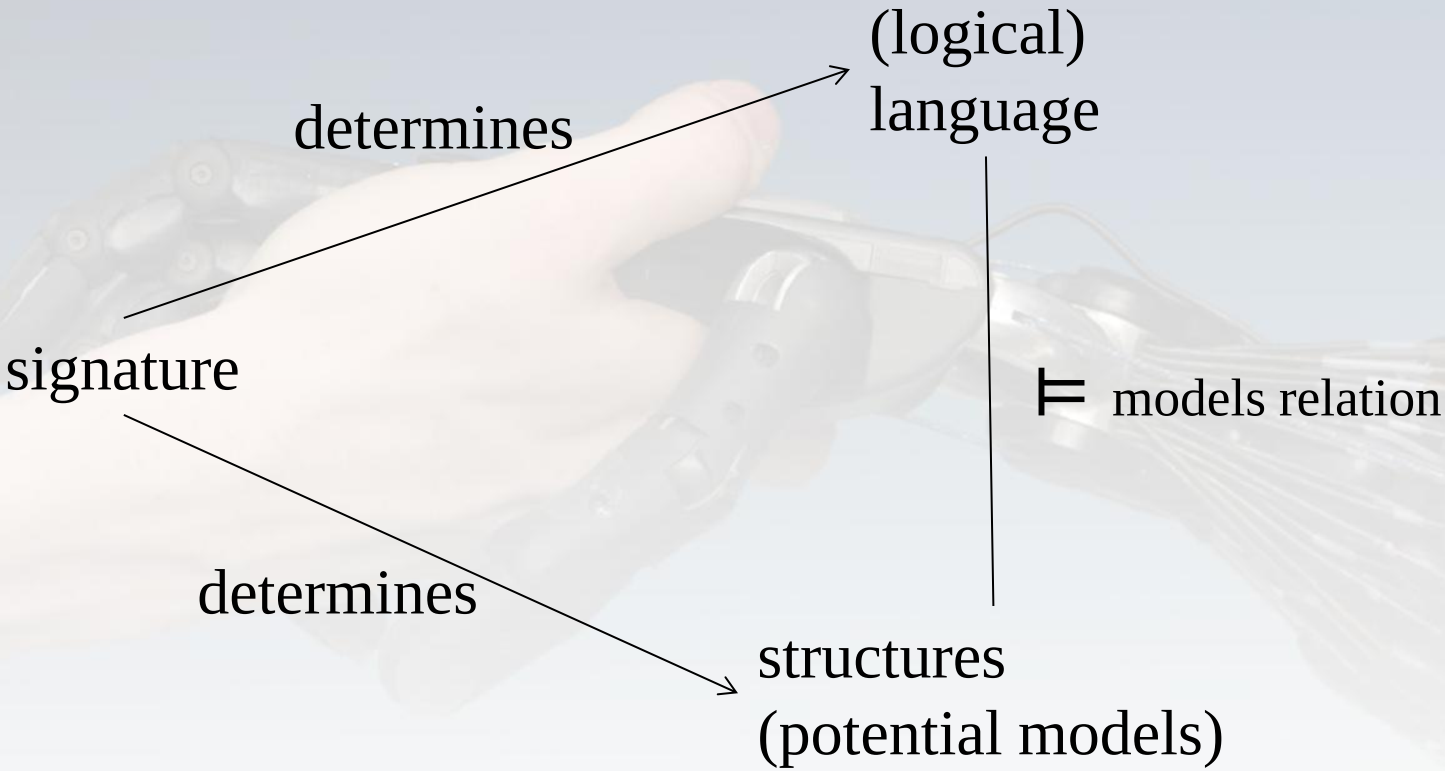
Repetition: First-Order Predicate Logic

- The set F_Σ of Formulas is the smallest class such that:
  - If R_j is a predicate symbol of arity r and $t_1,\ldots,t_r$ are terms, then $R_j(t_1,\ldots,t_r)$ is a formula (atomic formula or literal).
  - For all formulas φ and ψ: $\varphi \wedge \psi$, $\varphi \vee \psi$, $\neg\varphi$, $\varphi \rightarrow \psi$, $\varphi \leftrightarrow \psi$ are formulas.
  - If $x \in Var$ and φ is a formula, then $\forall x\varphi$ and $\exists x\varphi$ are formulas.
- Notice that “term” and “formula” are rather different concepts. Terms are used to define formulas and not vice versa.

- Examples:
  - 2 + 3 = 5 is a formula
  - $\forall x(x > 0 \rightarrow \exists y(y > 0 \wedge \forall b \forall a(|a-b| < y \rightarrow |f(a) - f(b)| < x)))$

    is a formula

Repetition: First-Order Predicate Logic



Repetition: First-Order Predicate Logic

Semantics (meaning, models) of FOL formulas.

Expressions of a FOL language of a given signature Σ are interpreted in relational structures of the same signature (or algebra)

$$\mathcal{M} = (\mathcal{U}, (c'_1,\ldots,c'_n,f'_1,\ldots,f'_m,R'_1,\ldots,R'_l))$$

where

c'_i is an element of the universe $\mathcal{U}$

f'_l is a function

R'_l is a relation

An *interpretation function* [[.]] maps:

terms to elements of the universe: $[[\,.\,]]: T_\Sigma \rightarrow \mathcal{U}$

formulas to truth-values: $[[\,.\,]]: F_\Sigma \rightarrow \{\mathit{true}, \mathit{false}\}$

Repetition: First-Order Predicate Logic

Semantics (meaning) of FOL formulas.

Recursive definition for interpreting terms and evaluating truth values of formulas:

for $c \in \{c_1,\ldots,c_n\}$:	$[[c]] = c'$	
for $x \in Var$:	$[[x]] \in \mathcal{U}$	
$[[f_i(t_1,\ldots,t_r)]] = f'([[t_1]],\ldots,[[t_r]])$		
$[[R(t_1,\ldots,t_r)]] = \text{true}$	iff	$<[[t_1]],\ldots,[[t_r]]> \in R'$
$[[\varphi \wedge \psi]] = \text{true}$	iff	$[[\varphi]] = \text{true}$ and $[[\psi]] = \text{true}$
$[[\varphi \vee \psi]] = \text{true}$	iff	$[[\varphi]] = \text{true}$ or $[[\psi]] = \text{true}$
$[[\neg\varphi]] = \text{true}$	iff	$[[\varphi]] = \text{false}$
$[[\forall x\varphi(x)]] = \text{true}$	iff	for all $d \in \mathcal{U}$: $[[\varphi(x)]]_{x\|d} = \text{true}$
$[[\exists x\varphi(x)]] = \text{true}$	iff	exists $d \in \mathcal{U}$: $[[\varphi(x)]]_{x\|d} = \text{true}$

Repetition: First-Order Predicate Logic

Examples: Structures (algebras)

Natural Numbers:	$(\mathbb{N},0,suc)$
Boolean algebras:	$(D,\wedge,\vee,\neg,\top,\bot)$
Field of Real Numbers:	$(\mathbb{R},0,1,+,*)$
Field of Complex Numbers:	$(\mathbb{C},0,1,+,*)$
Vector Space over a field K:	$(V,\oplus,\otimes)$, where $\oplus$ is vector addition and $\otimes$ is scalar product

Examples: Languages

Arithmetic expressions: $F_{(0,1,+,*,=)}$

Boolean expression over two atoms $\{p,q\}$: $F_{(p,q)}$

Repetition: First-Order Predicate Logic

Semantics

Model

If the interpretation of a formula φ with respect to a structure $\mathcal{M}$ results in the truth value *true*, $\mathcal{M}$ is called a model for φ (formal: $\mathcal{M} \vDash \varphi$).

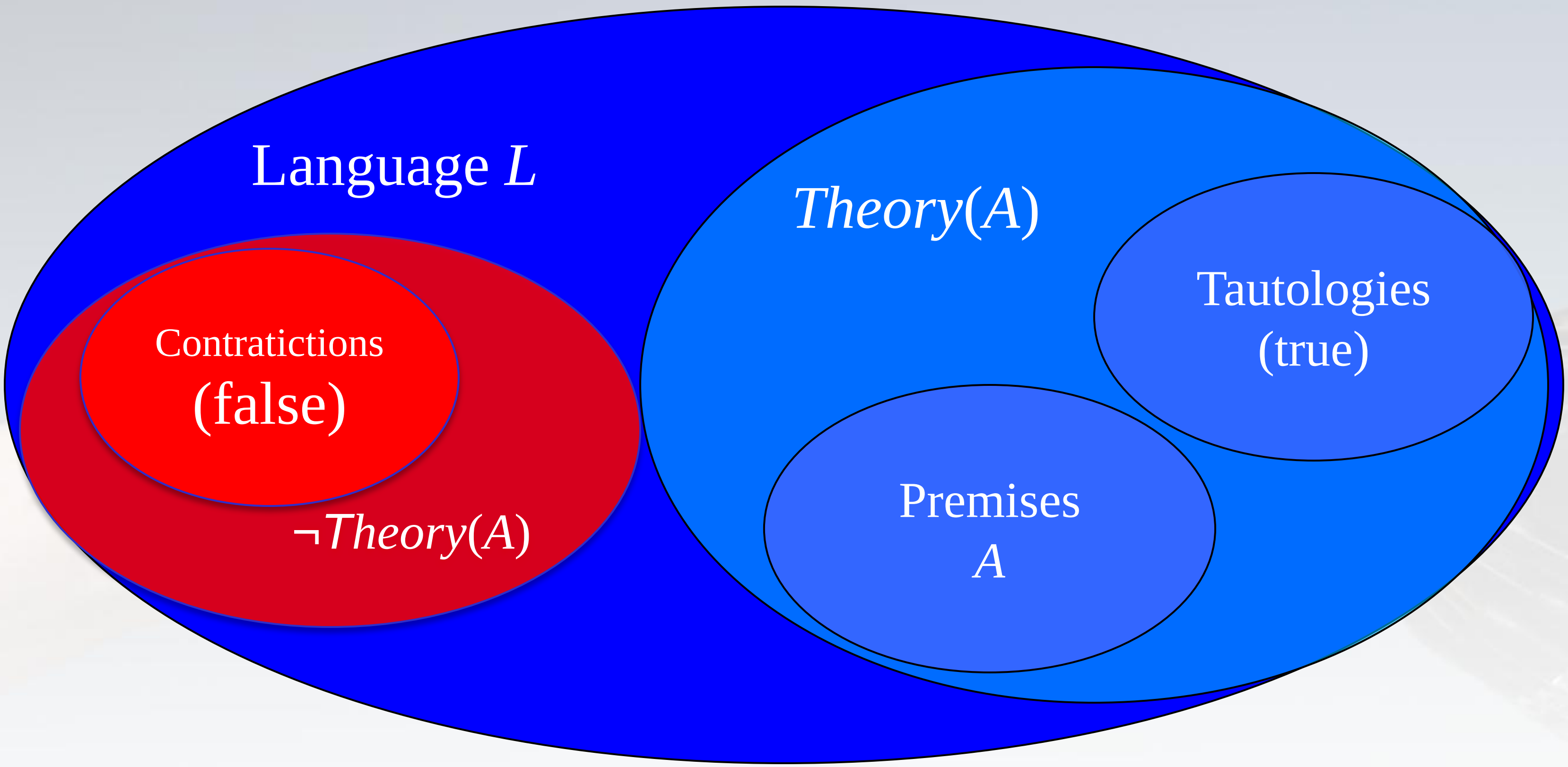
Validity

If every structure $\mathcal{M}$ is a model for φ we call φ valid (and write $\vDash \varphi$).

Satisfiability

If there exists a model $\mathcal{M}$ for φ we call φ satisfiable.

Repetition: First-Order Predicate Logic



Repetition: First-Order Predicate Logic

Semantics

Examples:

$\forall x \forall y\, (R(x) \wedge R(y) \rightarrow R(x) \vee R(y))$ [valid]

„If *x* and *y* are rich then either *x* is rich or *y* is rich“

„If *x* and *y* are even then either *x* is even or *y* is even“

$\exists x (N(x) \wedge P(x,c))$ [satisfiable]

„There is a natural number that is smaller than 17.“

„There exists someone who is a student and likes logic.“

Notice that there are interpretations which make the statement false.

Repetition: First-Order Predicate Logic

Semantics

Examples:

$\forall x\ (planet(x) \wedge electron(y) \rightarrow bigger(x, y))$ [???]

“Planets are bigger than electrons”

$\neg\exists x(N(x) \wedge x < 0)$ [???]

“There is no natural number smaller than 0”

Repetition: First-Order Predicate Logic

Semantics

Logical consequence

A formula φ is a logical consequence (or a logical entailment) of $A = \{A_1,\ldots,A_n\}$, if each model for A is also a model for φ.

We write $A \vDash \varphi$.

Notice: $A \vDash \varphi$ can mean that A is a model for φ or that φ is a logical consequence of *A.*

Therefore people usually use different alphabets or fonts to make this difference visible.

Theory

given axioms $A = \{A_1,\ldots,A_n\}$

$$Th(A) = \{\varphi \mid A \vDash \varphi\}$$

Propositional Logic vs. First-Order Logic

- Differences and Similarities.
  - In propositional logic, formulas are interpreted *directly* by truth values.

    Potential “models” are distributions of truth values over atoms.
  - In FOL, first, we must map the components of a formula into a structure.
    - Constant symbols are mapped to objects in the domain.
    - Interpreted functions applied to terms are mapped to objects in the domain.
    - Formulas defined on terms are mapped to truth values.
    - Mapping of atomic formulas (literals) to truth values: First, the arguments of a predicate symbol are interpreted, afterwards it is determined whether the corresponding relation holds in the structure.
    - For an existentially quantified formula there needs to be at least one element in the domain for which the formula is true.
    - For universally quantified formulas, the formula must be true for all objects in the domain.
    - If all atomic formulas are associated with truth values more complex formulas can be interpreted.

Some Questions

- What are the relations between axioms and a theory?
- How big is a theory?
- What is the relation between atoms in propositional logic and expression in FOL?

Repetition: First-order Logic

Semantic equivalences

Two formulas φ and ψ are semantically equivalent (we write $\varphi \equiv \psi$) if for all interpretations of φ and ψ it holds:

$\mathcal{M}$ is a model for φ iff $\mathcal{M}$ is a model for ψ.

Examples:

$$\varphi \wedge \varphi \equiv \varphi$$

$$\varphi \wedge \psi \equiv \psi \wedge \varphi$$

$$\varphi \wedge (\psi \vee \chi) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$$

Etc.

Repetition: First-order Logic

The following expressions are semantically equivalent:

- G is a logical consequence of $\{A_1,...,A_n\}$.
- $A_1 \wedge ... \wedge A_n \rightarrow G$ is *valid*.
  - Every structure is a model for this expression.
- $A_1 \wedge ... \wedge A_n \wedge \neg G$ is *not satisfiable*.
  - There is no structure making this expression true.

This can be used in the *refutation* based provers:

- If an expression $A_1 \wedge ... \wedge A_n \wedge \neg G$ is *semantically* not satisfiable, then *false* can be derived *syntactically*.
- If the reason for this is $\neg G$ (meaning $\{A_1 ,... , A_n\}$ has a model)
- then G must be true

Repetition: Semantic Equivalences

Here is a list of semantic equivalences

- $(\varphi \wedge \psi) \equiv (\psi \wedge \varphi)$, $(\varphi \vee \psi) \equiv (\psi \vee \varphi)$ (commutativity)
- $(\varphi \wedge \psi) \wedge \chi \equiv \varphi \wedge (\psi \wedge \chi)$, $(\varphi \vee \psi) \vee \chi \equiv \varphi \vee (\psi \vee \chi)$ (associativity)
- $(\varphi \wedge (\varphi \vee \psi)) \equiv \varphi$, $(\varphi \vee (\varphi \wedge \psi)) \equiv \varphi$ (absorption)
- $(\varphi \wedge (\psi \vee \chi)) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$ (distributivity)
- $(\varphi \vee (\psi \wedge \chi)) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \chi)$ (distributivity)
- $\neg\neg\varphi \equiv \varphi$ (double negation)
- $\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)$, $\neg(\varphi \vee \psi) \equiv (\neg\varphi \wedge \neg\psi)$ (deMorgan)
- $(\perp \wedge \varphi) \equiv \perp$, $(\perp \vee \varphi) \equiv \varphi$
- $(\mathrm{T} \wedge \varphi) \equiv \varphi$, $(\mathrm{T} \vee \varphi) \equiv \mathrm{T}$

Here are some more semantic equivalences

- $(\varphi \wedge \varphi) \equiv \varphi$, $(\varphi \vee \varphi) \equiv \varphi$ (idempotency)
- $\varphi \vee \neg\varphi \equiv \mathrm{T}$ (tautology)
- $\varphi \wedge \neg\varphi \equiv \perp$ (contradiction)
- $\neg\forall x\varphi \equiv \exists x\neg\varphi$, $\neg\exists x\varphi \equiv \forall x\neg\varphi$ (quantifiers)
- $(\forall x\ \varphi \wedge \psi) \equiv \forall x\ (\varphi \wedge \psi)$, $(\forall x\ \varphi \vee \psi) \equiv \forall x\ (\varphi \vee \psi)$
- $\forall x(\varphi \wedge \psi) \equiv (\forall x\varphi \wedge \forall x\psi)$
- Etc.

Properties of Logical Systems

Soundness

- A calculus is sound, if only such conclusions can be derived which also hold in the model.
  - In other words: Everything that can be derived is semantically true.
  - The introduced deductive calculi are sound, abduction and induction are not sound

Completeness

- A calculus is complete, if all conclusions can be derived which hold in the model.
  - In other words: Everything that is semantically true can syntactically be derived.

Decidability

- A calculus is decidable if there is an algorithm that calculates effectively for every formula whether such a formula is a theorem or not.

Usually people are interested in completeness results and decidability results.

We say a *logic* is sound/complete/decidable if there exists a *calculus* with these properties.

Some Properties of Classical Logic

- Propositional Logic:
  - Sound and complete, i.e. everything that can be proven is valid and everything that is valid can be proven.
  - Decidable, i.e. there is an algorithm that decides for every input whether this input is a theorem or not.
- First-order logic:
  - Complete (famous paper of Kurt Gödel 1930).
  - Undecidable, i.e. no algorithm exists that decides for every input whether this input is a theorem or not (Church 1936).
    - More precisely FOL is semi-decidable.
- Higher-order logic:
  - Situation is complex, in general not complete and not decidable
- Models
  - The classical models for FOL are Boolean algebras.

Proof Techniques in Mathematics

Direct, Indirect Proofs, Mathematical Induction

The Logic of Direct Proofs

Here is a quite simple reasoning method:

If A_1 is true and it holds

$A_1 \Rightarrow A_2, A_2 \Rightarrow A_3, \ldots, A_{n-1} \Rightarrow A_n$

then we know that it holds: $A_1 \Rightarrow A_n$ (i.e. then we can deduce that A_n must be true)

This kind of proof uses logically the transitivity of the implication „$\Rightarrow$“ and Modus Ponens

Example:

Claim: If $0 < x < y$ then $0 < 1/y < 1/x$.

Proof: If $0 < x < y$ then $0 < 1 < y/x$ [Multiplication with $1/x$].

If $0 < 1 < y/x$ then $0 < 1/y < 1/x$ [Multiplication with $1/y$].

Hence: $0 < x < y$ implies that $0 < 1/y < 1/x$.

We assumed that we know that $0 < x < y$ implies $0 < 1/x$ and $0 < 1/y$.

Notice: a direct proof is in general not simply a single application of Modus Ponens.

Examples of Direct Proofs

Complexity of breadth-first search

Many of the steps to show that the time complexity of BFS is $O(b^d)$ are related to direct proofs.

Proofs in a calculus

Sequences of axioms and statements that are deduced by axioms and Modus Ponens are prototypical examples for direct proofs.

Etc.

Example of a Direct Proof

Example:

Axioms = {*raining* $\vee$ *sprinkler_on* $\rightarrow$ *wet*,
sprinkler_on,
wet $\rightarrow$ *slippery*}

Is *slippery* true?

Proof:

1. *raining* $\vee$ *sprinkler_on* $\rightarrow$ *wet*
2. *sprinkler_on*
3. *wet* $\rightarrow$ *slippery*
4. *raining* $\vee$ *sprinkler_on* (*A* / *B* $\vee$ *A*, 2)
5. *wet* (Modus Ponens, 1, 4)
6. *slippery* (Modus Ponens, 3, 5)

Cases in direct proofs

Sometimes it is necessary to split a direct proof into different cases.

- Each case must be separately proven in order to establish the claim.
- Make sure that no case is left over.
  - If you miss a case your proof is not valid.

Example: To show that the subset relation $\subseteq$ is a partial order relation on $\wp(X)$ for a given set X consider cases:

- Show reflexivity
  - $\forall a \in \wp(X)\colon a \subseteq a$
- Show antisymmetry
  - $\forall a \in \wp(X)\ \forall b \in \wp(X)\colon (a \subseteq b \land b \subseteq a \rightarrow a = b)$
- Show transitivity
  - $\forall a \in \wp(X)\ \forall b \in \wp(X)\ \forall c \in \wp(X)\colon (a \subseteq b \land b \subseteq c \rightarrow a \subseteq c)$

Equivalences

The proof of an equivalence between two statements is essentially a characterization.

A certain property is characterized by an equivalent property

An equivalence „A $\Leftrightarrow$ B" consist of two directions

"$\Rightarrow$": The left-to-right direction

"$\Leftarrow$": The right-to-left direction

Example: Show that $a \wedge b = a \Leftrightarrow a \leq b$ (given a lattice with $a \wedge b = \inf(a,b)$ and $\leq$ is the smaller than relation, i.e. a partial order relation)

"$\Rightarrow$" Assume $a \wedge b = a$. Then by definition a is a lower bound of a and b. Hence $a \leq b$

"$\Leftarrow$" Assume $a \leq b$. Then a is a lower bound of b and a is a lower bound of a itself (reflexivity). Because a is the greatest lower bound of a we have $a \wedge b = a$

Proof Methods: Indirect Proofs

- The logical principle behind one type of indirect proof techniques is the following one:
  - If it is true that p implies q, then it also holds: $\neg q$ implies $\neg p$.
  - This holds if we work in classical logic.

- Remark: There are alternative logical systems where this principle does not hold.
  - If you want to reject this principle:
  - Try to change the properties of the negation.
  - For example, reject: $\neg\neg p \leftrightarrow p$.

Indirect Proofs

- Examples
  - If every philosophical problem is not solvable,
    then a problem that is solvable cannot be a philosophical one.
  - Completeness proofs for logic systems work quite often this way:
    - We do not proof that: If $\vDash \varphi$ then $\vdash \varphi$. It is usually easier to show:
    - If $\nvdash \varphi$ then $\nvDash \varphi$.
- There is a related proof technique
  - Proof by contradiction (Proof by refutation).
  - If you want to prove that $p \rightarrow q$, show that it holds: if we assume that $p \wedge \neg q$, then we can deduce a contradiction.
  - Resolution is an example of this proof technique.

Indirect Proofs

- Quite often it happens that we do not get a direct contradiction to the assumptions of the theorem, but a contradiction to facts we already know about the theory.
- We want to prove the theorem „$A \Rightarrow B$“.
- We extend the premise „A“ with the axioms and facts about our theory C:

$$(A \wedge C) \Rightarrow B$$

- Assume „$\neg B$“.
- Show a contradiction to „A“ or to „C“ according to

$$\neg B \Rightarrow (\neg A \vee \neg C)$$

Example

- We want to prove the theorem:

$$\text{For all } x \in \mathbb{R}\text{: } (x > 0) \Rightarrow (1/x > 0).$$

- Proof:
  - Assume $1/x \leq 0$.
  - Either (a) $1/x = 0$ or (b) $1/x < 0$.
  - If $1/x = 0$, then $1 = x \cdot 1/x = 0$ which contradicts the axiom that $0 \neq 1$.
  - If $1/x < 0$ and $x > 0$, then $1 = x \cdot 1/x < 0 \cdot x = 0$ which contradicts the fact that $0 < 1$.
    - [The principle $(a < b \wedge c > 0) \rightarrow (ac < bc)$ is used here.]
  - Conclude: the theorem holds.

Example for a Proof by Contradiction

We want to prove that

$$R = \{x \mid x \notin x\} \text{ is not a set.}$$

Proof:

Assume R is given as above and R *is* a set.

Then $R \in R$ or $R \notin R$.

If $R \in R$, then $R \in \{x \mid x \notin x\}$.

Hence: $R \notin R$ (contradiction).

If $R \notin R$, then $R \notin \{x \mid x \notin x\}$.

Hence: $R \in R$ (contradiction).

Conclude: R is not a set.

Historically this simple proof showed that naïve set theory is inconsistent by constructing the so-called Russell set.

Proof by Refutation

- The following statements are equivalent (based on the deduction theorem):
  - G is a logical consequence of $\{A_1,\ldots,A_n\}$
  - $\{A_1,\ldots,A_n\} \models G$
  - $A_1 \wedge \ldots \wedge A_n \rightarrow G$ is valid (i.e. every structure is a model for this expression: $\models A_1 \wedge \ldots \wedge A_n \rightarrow G$)
  - $A_1 \wedge \ldots \wedge A_n \wedge \neg G$ is not satisfiable (i.e. there is no structure making this expression *true*)
- This can be used in the resolution calculus: If an expression $A_1 \wedge \ldots \wedge A_n \wedge \neg G$ is not satisfiable, then *false* can be derived syntactically.

Proof by Example / Counterexample

Logical structure.

You want to prove either (a) $\exists x$: $\varphi(x)$ or
(b) $\neg\forall x$: $\varphi(x)$.

Method:

(a) Find a c such that $\varphi(c)$ holds.

(b) Rewrite $\neg\forall x$: $\varphi(x)$ to $\exists x$: $\neg\varphi(x)$ and find c such that $\neg\varphi(c)$.

Example:

Show that for a given prime number p there is a prime number $p‘$ such that $p < p‘$.

Enumerate all prime numbers up to p, e.g. 2, 3, 5, 7, ..., p.

Calculate the product of these numbers and add 1.

$p‘ = (2 \cdot 3 \cdot 5 \cdot \ldots \cdot p) + 1.$

Show that $p’$ is a prime number.

Proof by Mathematical Induction

- Mathematical induction
  - Can be applied to problems where we need to establish a fact for all natural numbers $n \in \mathbb{N}$.
  - Prove a statement for the base case or the base cases (often for $n = 0$ or $n = 1$) and show that if the claim holds for a number n, then it holds for $n + 1$ as well.
- More precisely and more generally: mathematical induction can be used for „inductive sets“.
  - Compare the inductive definition of well-formed terms, formulas etc.

Proof by Mathematical Induction

Here is a **false** proof.

Theorem: All students have the same hair color.

Proof: Proof by induction. Let $P(n)$ be the proposition that in any set of n many students all students have the same hair color.

Base case: $P(1)$ is obviously true, because there is only one student in the set.

$n \to n+1$: Assume $P(n)$ holds for $n \in \mathbb{N}$. Consider $P(n+1)$, i.e. consider an arbitrary set $\{s_1,\ldots,s_{n+1}\}$ of $n+1$ many students. By induction hypothesis $s_1,\ldots,s_n$ have the same hair color. Likewise $s_2,\ldots,s_{n+1}$. Therefore $s_1,\ldots,s_{n+1}$ must have the same hair color.

What is wrong with this "proof"?

Some Invalid Proof Techniques

- Proof by example
  - Showing that a general statement holds for $n = 3$
- Proof by omission
  - „The other cases are similar“
  - „The proof is an easy exercise and left to the reader“
- Proof by picture
  - Sometimes quite convincing but not a proof
- Proof by cumbersome notation
  - Use at least four alphabets and some special symbols

Some invalid Proof Techniques

- Proof by intimidation
  - „Trivial“
- Proof by eminent authority
  - „I saw the nobel prize winner downstairs and she said the statement is probably true“
- Proof by obfuscation
  - A long plotless sequence of true and/or meaningless syntactically related statements
- Proof by the reference of inaccessible literature
  - The author cites a corollary of a theorem to be found in a privately circulated memoir of the Maldive Islands Philological Society, 1883.

Something to Relax

A Logic of Chords

Ingolf Max: Logic of Chords

- Idea of representing chords:

$$\left\{ \begin{matrix} 8^1 \\ 3^1 \\ 11^0 \\ 5^0 \end{matrix} \begin{bmatrix} +5^0 \\ +4^0 \\ +6^0 \end{bmatrix}^1 \begin{bmatrix} +9^0 \\ +10^0 \end{bmatrix}^2 [+3^1]^3 \right\}$$

- The tones are absolutely represented as intervals to a key tone (C^0 here) and the distance between all pairs of tones are specified.
  - Result: 5^0 -> F, 11^0 -> B, 3^1 -> D#, 8^1 -> G#

Tristan chord

Chopin op. 23

Tristan chord

- The bracketed sequence of numbers is the type of a chord (having different instantiations) and builds the basis for a logic of chords.

Ingolf Max: Logic of chords

What can be done in a logic of chords?

Negation(s): inverse the order of the length of basic intervals

$$\begin{Bmatrix} \begin{matrix} 10^0 \\ 7^0 \\ 4^0 \\ 0^0 \end{matrix} & \begin{bmatrix} +3^0 \\ +3^0 \\ +4^0 \end{bmatrix}^1 & \begin{bmatrix} +6^0 \\ +7^0 \end{bmatrix}^2 & [+10^0]^3 \end{Bmatrix} \text{ vs. } \begin{Bmatrix} \begin{matrix} 10^0 \\ 6^0 \\ 3^0 \\ 0^0 \end{matrix} & \begin{bmatrix} +4^0 \\ +3^0 \\ +3^0 \end{bmatrix}^1 & \begin{bmatrix} +7^0 \\ +6^0 \end{bmatrix}^2 & [+10^0]^3 \end{Bmatrix}$$

Inferences

- If the tones constituting chord B is a subset of the tones constituting A, then B follows tonally from A

$$\begin{Bmatrix} \begin{matrix} 0^1 \\ 7^0 \\ 4^0 \\ 0^0 \end{matrix} & \begin{bmatrix} +5^0 \\ +3^0 \\ +4^0 \end{bmatrix}^1 & \begin{bmatrix} +8^0 \\ +7^0 \end{bmatrix}^2 & [+0^1]^3 \end{Bmatrix} \models_T \begin{Bmatrix} \begin{matrix} 0^1 \\ 7^0 \\ 0^0 \end{matrix} & \begin{bmatrix} +5^0 \\ +7^0 \end{bmatrix}^1 & [+0^1]^2 \end{Bmatrix}$$

- There is also the concept of inference with respect to interval length

Ingolf Max: Logic of chords

chord	traditional name	result of negation −	traditional name	tonal correlation
$\left\{\begin{matrix}7^0\\4^0\\0^0\end{matrix}\begin{bmatrix}+3^0\\+4^0\end{bmatrix}[+7^0]\right\}$	3-tone-c-major-chord in root position	$\left\{\begin{matrix}7^0\\3^0\\0^0\end{matrix}\begin{bmatrix}+4^0\\+3^0\end{bmatrix}[+7^0]\right\}$	3-tone-c-minor-chord in root position	major–minor in same position
$\left\{\begin{matrix}0^1\\7^0\\4^0\end{matrix}\begin{bmatrix}+5^0\\+3^0\end{bmatrix}[+8^0]\right\}$	3-tone-c-major-chord in first inversion	$\left\{\begin{matrix}0^1\\9^0\\4^0\end{matrix}\begin{bmatrix}+3^0\\+5^0\end{bmatrix}[+8^0]\right\}$	3-tone-a-minor-chord in second inversion	relative minor in next inversion
$\left\{\begin{matrix}4^1\\0^1\\7^0\end{matrix}\begin{bmatrix}+4^0\\+5^0\end{bmatrix}[+9^0]\right\}$	3-tone-c-major-chord in second inversion	$\left\{\begin{matrix}4^1\\11^0\\7^0\end{matrix}\begin{bmatrix}+5^0\\+4^0\end{bmatrix}[+9^0]\right\}$	3-tone-e-minor-chord in first inversion	opposite relative minor in former inversion
$\left\{\begin{matrix}0^1\\7^0\\4^0\\0^0\end{matrix}\begin{bmatrix}+5^0\\+3^0\\+4^0\end{bmatrix}\begin{bmatrix}+8^0\\+7^0\end{bmatrix}[+0^1]\right\}$	4-tone-c-major-chord in root position	$\left\{\begin{matrix}0^1\\8^0\\5^0\\0^0\end{matrix}\begin{bmatrix}+4^0\\+3^0\\+5^0\end{bmatrix}\begin{bmatrix}+7^0\\+8^0\end{bmatrix}[+0^1]\right\}$	4-tone-f-minor-chord in second position	minor-X-dominant next but one inversion
$\left\{\begin{matrix}4^1\\0^1\\7^0\\4^0\end{matrix}\begin{bmatrix}+5^0\\+3^0\\+4^0\end{bmatrix}\begin{bmatrix}+8^0\\+7^0\end{bmatrix}[+0^1]\right\}$	4-tone-c-major-chord in first position	$\left\{\begin{matrix}4^1\\0^1\\9^0\\4^0\end{matrix}\begin{bmatrix}+4^0\\+3^0\\+5^0\end{bmatrix}\begin{bmatrix}+8^0\\+7^0\end{bmatrix}[+0^1]\right\}$	4-tone-a-minor-chord in second position	relative minor in next inversion
$\left\{\begin{matrix}7^1\\4^1\\0^1\\7^0\end{matrix}\begin{bmatrix}+3^0\\+4^0\\+5^0\end{bmatrix}\begin{bmatrix}+8^0\\+7^0\end{bmatrix}[+0^1]\right\}$	4-tone-c-major-chord in second position	$\left\{\begin{matrix}7^1\\2^1\\10^0\\7^0\end{matrix}\begin{bmatrix}+5^0\\+4^0\\+3^0\end{bmatrix}\begin{bmatrix}+8^0\\+7^0\end{bmatrix}[+0^1]\right\}$	4-tone-g-minor-chord in root position	minor-X-dominant in next-to -last-position

Ingolf Max: Logic of chords

A difficult question is: what are conjunctions and disjunctions?

- Is it possible to identify conjunctions of chords with a blending process?

Are there more general notions of inference?

Pros

- Rather concrete approach
- Allows to represent a large variety of musical styles and traditions

Cons

- It is unclear to which extent the approach is really a logic: (no proper conjunction, disjunction, inference)?
- How is it possible to construct more general concepts from chord representations (e.g. cadences)

Resolution

Clause Form, Substitution, Unification, Resolution, Examples

Normal Forms

- If there are a lot of different representations of the same statement the question arises:
  - Are there “simple” ones?
  - Are there “normal forms”?
- Different normal forms for FOL
  - *Negation normal form*: Only negations of atomic formulas
  - *Prenex normal form*: No embedded quantifiers
  - *Conjunctive normal form*: Only conjunctions of disjunctions
  - *Disjunctive normal form*: Only disjunctions of conjunctions
  - *Gentzen normal form*: Only implications where the condition is an atomic conjunction and the conclusion is an atomic disjunction

Clause Form

Conjunctive normal form.

We know: Every formula of propositional logic can be rewritten as a conjunction of disjunctions of atomic propositions.

Similarly, every formula of predicate logic can be rewritten as a conjunction of disjunctions of literals (modulo quantification).

A formula is in clause form, if it is rewritten as a set of disjunctions of (possibly negative) literals.

We will use the following example for an eight-step transformation to clause form:

$$\forall x\, [B(x) \rightarrow (\exists y\, [O(x,y) \wedge \neg P(y)] \wedge \neg\exists y\, [O(x,y) \wedge O(y,x)] \wedge \forall y\, [\neg B(y) \rightarrow \neg E(x,y)])]$$

Clause Form

(1) Remove Implications.

$$\forall x[\neg B(x) \vee (\exists y[O(x,y) \wedge \neg P(y)] \wedge \neg \exists y[O(x,y) \wedge O(y,x)]$$
$$\wedge \forall y[\neg[\neg B(y)] \vee \neg E(x,y)])]$$

(2) Reduce scope of negations.

$$\forall x[\neg B(x) \vee (\exists y[O(x,y) \wedge \neg P(y)] \wedge \forall y[\neg O(x,y) \vee \neg O(y,x)]$$
$$\wedge \forall y[B(y)] \vee \neg E(x,y)])]$$

(3) Skolemization (remove existential quantifiers).

If an existentially quantified variable is in the scope of a universally quantified variable it is replaced by a function symbol dependent on this variable.

$$\forall x[\neg B(x) \vee ([O(x,f(x)) \wedge \neg P(f(x))] \wedge \forall y[\neg O(x,y) \vee \neg O(y,x)]$$
$$\wedge \forall y[B(y) \vee \neg E(x,y)])]$$

The used Skolem function is a *choice function* => axiom of choice.

Clause Form

(4) Standardize (rename) variables.

$\forall x[\neg B(x) \vee ([O(x,f(x)) \wedge \neg P(f(x))] \wedge \forall y[\neg O(x,y) \vee \neg O(y,x)] \wedge \forall z[B(z) \vee \neg E(x,z)])]$

A variable bound by a quantifier is a “dummy” and can be renamed without changing the semantics.

(5) Prenex-form.

$\forall x \forall y \forall z[\neg B(x) \vee ([O(x,f(x)) \wedge \neg P(f(x))] \wedge [\neg O(x,y) \vee \neg O(y,x)] \wedge [B(z) \vee \neg E(x,z)])]$

Apply laws governing quantifiers.

(6) Conjunctive normal form (CNF).

Apply the distributive laws (perhaps they need to be applied several times).

$\forall x \forall y \forall z[(\neg B(x) \vee O(x,f(x))) \wedge (\neg B(x) \vee \neg P(f(x))) \wedge$
$(\neg B(x) \vee \neg O(x,y) \vee \neg O(y,x)) \wedge (\neg B(x) \vee B(z) \vee \neg E(x,z))]$

Clause Form

(7) Eliminate conjunctions and rename variables if necessary. Each conjunct should have a different set of variables.

$\forall x\ [\neg B(x) \lor O(x,f(x))]$, $\forall w\ [\neg B(w) \lor \neg P(f(w))]$,

$\forall u \forall y\ [\neg B(u) \lor \neg O(u,y) \lor \neg O(y,u)]$, $\forall v \forall z\ [\neg B(v) \lor B(z) \lor \neg E(v,z)]$

(8) Eliminate universal quantifiers.

$M = \{\ \neg B(x) \lor O(x,f(x)),\quad \neg B(w) \lor \neg P(f(w)),$

$\neg B(u) \lor \neg O(u,y) \lor \neg O(y,u),\quad \neg B(v) \lor B(z) \lor \neg E(v,z)\ \}$

Why clause form at all?

Because of the resolution calculus.

Resolution steps are defined on pairs of clauses by unifying literals.

Repetition: First-order Logic

- The following statements are equivalent:
  - G is a logical consequence of $\{A_1,\ldots,A_n\}$.
  - $A_1 \wedge \ldots \wedge A_n \rightarrow G$ is *valid*. (Every structure is a model for this expression)
  - $A_1 \wedge \ldots \wedge A_n \wedge \neg G$ is *not satisfiable*.

    (There is no structure making this expression true)

- This can be used in *refutation*-based theorem provers:
  - If an expression $A_1 \wedge \ldots \wedge A_n \wedge \neg G$ is not satisfiable, then *false* can be derived *syntactically*.
  - If the reason for this is $\neg G$ (means $\{A_1 ,\ldots , A_n\}$ has a model), then G must be true

Resolution Calculus

Idea: The resolution calculus consists of a single rule. It does not possess any axioms

Resolution is defined for clauses not for arbitrary FOL formulas.

Clause: each formula is a disjunction of positive and negative literals.

All formulas must hold corresponding to a conjunction of clauses.

Proof by refutation, exploiting the already mentioned equivalence.

If $A_1 \wedge \ldots \wedge A_n \wedge \neg G$ is not satisfiable,

then *false* (the empty clause) can be derived in the calculus:

$$A_1 \wedge \ldots \wedge A_n \wedge \neg G \vdash \bot$$

Using substitutions to compute specializations of (universally quantified) clauses

More precisely, computing most general unifiers

Using the schema:

$(B \vee C) \wedge (\neg B \vee D) \rightarrow C \vee D$

Substitution

A substitution $\Theta = \{v_1 \leftarrow t_1,\ldots,v_n \leftarrow t_n\}$
replaces variables v_i by terms t_i in a formula.

Example:

Application of a substitution $\Theta = \{x \leftarrow C\}$ to an expression of the form

$$P(x) \vee (\neg Q(x,y) \wedge P(f(x)))$$

results in

$$P(C) \vee (\neg Q(C,y) \wedge P(f(C)))$$

If we apply a substitution Θ to an expression φ
we call the result of the substitution $\varphi' = \varphi\Theta$ an instance.

Composition of substitutions is possible.
Notice: composition of substitution is not necessarily commutative.

Unification is used to unify expressions in the resolution calculus.

Unification / Unifier

Let $\{E_1,E_2,...E_n\}$ be a set of expressions

A substitution Θ is a unifier of $E_1,E_2,...,E_n$ if it holds:

$E_1\Theta = E_2\Theta =...= E_n\Theta$

A unifier Θ is called the most general unifier (mgu)

if for all other unifiers σ of $E_1,E_2,...,E_n$

there exists a substitution γ, such that $\sigma = \Theta\gamma$.

Theorem: If there exists a unifier

then there exists a most general unifier as well.

There are a lot of unification algorithms, for example, developed by John Alan Robinson.

Examples:

$\{P(x), P(A)\}$	$\Theta = \{x \leftarrow A\}$
$\{P(f(x),y,g(y)), P(f(x),z,g(x))\}$	$\Theta = \{y \leftarrow x, z \leftarrow x\}$
$\{P(x,f(y),B), P(x,f(B),B)\}$	$\sigma = \{x \leftarrow A, y \leftarrow B\}$
	$\Theta = \{y \leftarrow B\}$

Resolution: Example

An example of a resolution step.

$C_1 = \{P(f(x)), \neg Q(z), P(z)\}$ $\sigma_1 = \{\}$

$C_2 = \{\neg P(x), R(g(x)), A\}$ $\sigma_2 = \{x \leftarrow u\}$ (renaming variables)

$\Theta = \{z \leftarrow f(x), u \leftarrow f(x)\}$

$R = \{\neg Q(f(x)), R(g(f(x))), A\}$

In a simple diagram:

$\{P(f(x)), \neg Q(z), P(z)\}$ $\{\neg P(u), R(g(u)), A\}$

$\{z \leftarrow f(x)\}$ $\{u \leftarrow f(x)\}$

$\{\neg Q(f(x)), R(g(f(x))), A\}$

Resolution: The Abstract Definition

The abstract definition:

We write a clause $C = L_1 \vee \ldots \vee L_n$ as a set $C = \{L_1,\ldots,L_n\}$.

Let C_1, C_2 and R be clauses

R is called the resolvent of C_1 and C_2 if the following conditions hold:

1. There are substitutions σ_1 and σ_2 such that $C_1\sigma_1$ and $C_2\sigma_2$ have no common variables.

2. There exist literals $L_1,\ldots,L_m \in C_1\sigma_1$ and literals $l_1,\ldots l_n \in C_2\sigma_2$ such that
$$L = \{\neg L_1,\ldots,\neg L_m, l_1,\ldots,l_n\}$$
is unifiable with Θ as most general unifier of L.

3. R has the following form:
$$R = ((C_1\sigma_1 \setminus \{L_1,\ldots,L_m\}) \cup (C_2\sigma_2 \setminus \{l_1,\ldots,l_n\}))\Theta$$

Resolution

Proof by resolution:

To prove that a formula φ follows logically from a set of axioms $A_1,\ldots,A_n$:

Include the negated formula φ in the set of axioms and try to derive a contradiction (empty clause).

Notice: we do not prove a logical consequence directly, rather we show that the negation of the consequent results in a contradiction.

The theoretical basis is the following fact.

> Theorem: A set of clauses is not satisfiable if the empty clause can be derived in a resolution proof.

What we get is a so-called refutation tree.

What we want to infer is a contradiction: $C_1 = A$, $C_2 = \neg A$ stands for $A \wedge \neg A$ and we get $A \wedge \neg A \vdash \bot$.

Example

The classical example:

Axiom: All humans are mortal.

Fact: Socrates is human.

To show: Socrates is mortal.

Formalization:

First-order logic

$\forall x$: human(x) $\rightarrow$ mortal(x)

human(S)

$\neg$mortal(S)

Clauses

$\neg$human(x) $\vee$ mortal(x)

human(S)

$\neg$mortal(S)

$\neg$mortal(S) $\neg$human(x) $\vee$ mortal(x)

$\neg$human(S)   human(S)

$\perp$

Properties of Resolution

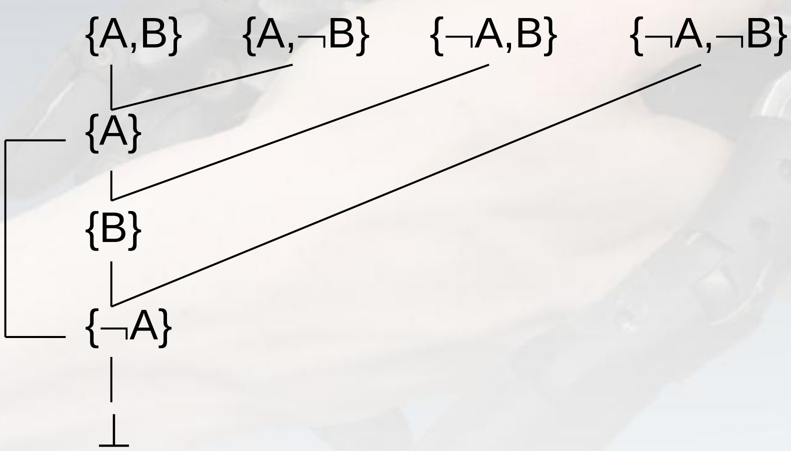
- In general, there are many possibilities, to find two clauses, which are resolvable.
  - Of the many alternatives, there are possibly only a few which help to derive the empty clause: combinatorial explosion.
- For feasible algorithms: use a resolution strategy.
  - Use heuristic rules that specify in which order to resolve the clauses.
    - Theoretically not well understood why such strategies do (or should) work.
  - Example:
    - Preference for resolution with unit clauses, that is, clauses with one literal.
- Well known efficient strategy: SLD-Resolution (*linear resolution with selection function for definite clauses*)
  (e.g. used in Prolog).

SLD Resolution

Linearity: Use a sequence of clauses ($C_0,\ldots,C_n$) starting with the negated assertion C_0 ending with the empty clause C_n

Each C_i is generated as resolvent by $C_{i\text{-}1}$ and a clause $B_{j\text{-}1}$ where $B_{j\text{-}1}$ is either an axiom (input resolution) or a clause C_j with $j < i\text{-}1$

Notice: In Prolog just input resolution is used



Selection function identifies the literal which is chosen in the next resolution step

Example: Use a sequence instead of a set of literals and choose the first one

Definite Horn clauses: A Horn clause contains maximally one positive literal; a definite Horn clause contains exactly one positive literal (Prolog rule)

Example

The following set of clauses are given:

$$F = \{\{T(a)\}, \{\neg P(x), Q(x), R(x,f(x))\}, \{P(a)\}, \{\neg P(x), Q(x), S(f(x))\},$$
$$\{\neg T(y), \neg S(y)\}, \{\neg R(a,z), T(z)\}, \{\neg T(x), \neg Q(x)\}\}$$

Show that F is not satisfiable.

$\{\neg P(x), Q(x), S(f(x))\}$ $[x \leftarrow a]$ $\qquad \{\neg T(y), \neg S(y)\}$ $\qquad [y \leftarrow f(a)]$

$\{\neg P(a), Q(a), \neg T(f(a))\}$ $\qquad \{\neg R(a,z), T(z)\}$ $\qquad [z \leftarrow f(a)]$

$\{\neg P(a), Q(a), \neg R(a,f(a))\}$ $\qquad \{\neg P(x), Q(x), R(x,f(x))\}$ $\qquad [x \leftarrow a]$

$\{\neg P(a), Q(a)\}$ $\qquad \{P(a)\}$

$\{Q(a)\}$ $\qquad \{\neg T(x), \neg Q(x)\}$ $\qquad [x \leftarrow a]$

$\{\neg T(a)\}$ $\qquad \{T(a)\}$

$\perp$

Example

	PROLOG	Logic	
Fact	isa(fish,animal). isa(trout,fish).	isa(Fish,Animal) isa(Trout,Fish)	Positive literal
Rule	is(X,Y) :- isa(X,Y). is(X,Z) :- isa(X,Y), is(Y,Z).	$is(x,y) \vee \neg isa(x,y)$ $is(x,z) \vee \neg isa(x,y) \vee \neg is(y,z)$	Definite clause
Query	?-is(trout,animal). ?-is(fish,X).	$\neg is(Trout,Animal)$ $\neg is(Fish,x)$	Assertion

Denotations

:- denotes as usual the reversed implication

$$isa(x,y) \wedge is(y,z) \rightarrow is(x,z) \quad \equiv \quad \neg(isa(x,y) \wedge is(y,z)) \vee (is(x,z))$$

$$\equiv \quad \neg isa(x,y) \vee \neg is(y,z) \vee is(x,z)$$

Variables which occur in the head of a clause are implicitly universally quantified.

$$\forall x \forall y \forall z: \neg isa(x,y) \vee \neg is(y,z) \vee is(x,z)$$

Variables which occur only in the body are existentially quantified (in the body).

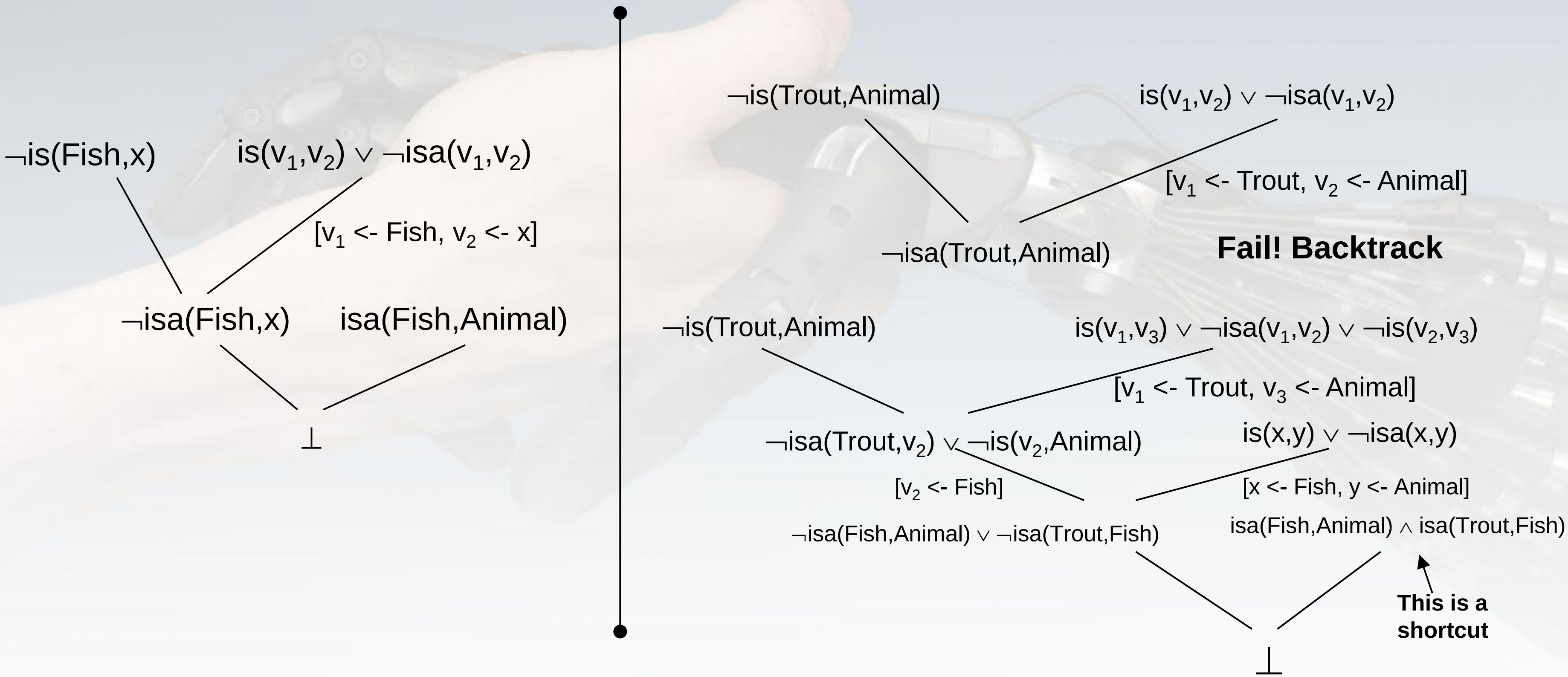
Example

Query: is(fish,X).

Stands for ∃x: is(Fish,x)

Negation of query: ¬∃x is(Fish,x) ≡ ∀x¬is(Fish,x)

SLD Resolution: (extract)



Theorem Provers

Theorem provers typically are more general than Prolog

- Not only Horn clauses but full FOL; no interleaving of logic and control (i.e. ordering of formulas has no effect on result)
- Theorem provers exist
  - For non-classical logics as well as for classical logic
  - For mathematics
  - For verification of hardware and software
  - For deductive program synthesis

All rule-based systems (production systems, planners, inference systems) can be realized using either forward-chaining or backward-chaining algorithms

- Forward chaining: Add a new fact to the knowledge base and derive all consequences (data-driven)
- Backward chaining: Start with a goal to be proved, find implication sentences that would allow to conclude the goal, attempt to prove the premises, etc.

Theorem Provers

- For evaluating theorem provers there are libraries with problems
  - TPTP: Thousands of Problems for Theorem Provers (http://www.cs.miami.edu/~tptp/)
  - At CADE (Conference on Automated Deduction) the world championships for automated theorem proving takes place on a yearly basis (http://www.cs.miami.edu/~tptp/CASC/).
    - Winners of the different competitions can be also found on this page.
  - Problem domains have a broad range (http://www.cs.miami.edu/~tptp/cgi-bin/SeeTPTP?Category=Problems):
    - Mathematics: Algebra, Analysis, Arithmetic, Boolean Algebras, Category Theory etc.
    - Natural Language Processing
    - Medicine
    - Hardware Verification
    - Commonsense Reasoning
    - Planning
    - Etc.

Properties of Logical Systems

- Propositional logic
  - Propositional logic is sound and complete.
  - Propositional logic is decidable.
- First-Order logic
  - First-order logic is sound and complete (Gödel 1930).
  - First-order logic is not decidable (Church 1936). [But: first-order logic is semi-decidable!]
  - First-order logic with predicate symbols of arity 1 is decidable (Löwenheim 1915).
- Resolution calculus
  - The resolution calculus is sound and (refutation) complete.
    - Refutation completeness means, that if a set of formula (clauses) is *not* satisfiable, then resolution will find a contradiction.
    - Resolution cannot be used to generate all logical consequences of a set of formulas, but it can establish that a given formula is entailed by the set.
    - Hence, it can be used to find all answers to a given question, using the "negated assumption" method.

Potential Problems of Using Logic for Reasoning

Challenges for Logic Approaches

Wason Selection Task

The Wason selection task

- 4 cards are given: On one side there is a number and on the other a letter printed.
- Rule: If there is a vowel at one side, there will be an even number at the other side.

The following situation is given:

A	D	4	7

The task is: Turn as few cards as possible to prove the rule.

- The correct answer is to turn A and 7.

Wason Selection Task

The experiment was executed in various versions.

One showed the following results:

A and 4:	46 %
A:	33 %
A and 7:	3 %
Others:	18 %

Modus Tollens:

If p, then q. And: *not q*. Therefore: *not p*.

It seems to be the case that humans do not think logically…

Wason Selection Task

New rule: Only people over 18 are allowed to drink alcohol.

Meaning: If for someone it is allowed to drink alcohol he/she must be over 18.

The new situation:

17	Water	Beer	22

The solution is to turn Beer and 17.

This version of the Wason selection task seems to be much easier to solve for humans.

Wason Selection Task

Some proposals for an explanation of these results:

- Humans do not think logical at all (Gigerenzer).
- Humans think in models not in terms of logical deductions (Johnson-Laird).
- Humans need to embed their reasoning in concrete situations. They have problems in reasoning in idealized situations, i.e. mental models do not reduce the problem to the idealized (abstracted) situation.
- Humans can solve such problems, if it is placed in a social context (evolutionary psychology).

Many theories were proposed to model these data.

- There are logic-based solutions as well as model-based solutions.

Wason Selection Task

Another important point to mention is the way to describe the task in natural language.

As a matter of fact, many logical connectives in natural language require a “more complex” interpretation than in classical logic.

“Peter is in the living room or in the kitchen.”

“Paul went to the university and gave a speech.” vs. “Paul gave a speech and went to the university.”

“If Jim works hard for the exam he will pass it.”

The standard version of the Wason selection task makes it plausible that a certain number of subjects interpret the implication as an equivalence.

Natural Language

Natural language shows many features that cannot be easily modeled with classical logical approaches. Here are some examples:

- "Many students read different books."
  - Generalized quantifiers require an extension of classical logic.
- "Could you tell me what time is it?"
  - Implicatures require a non-literal interpretation.
- "Yesterday John told me that in 150 years Germany will have a Mediterranean climate."
  - Temporal aspects require an extension of classical logic.
- "If I had been on holidays two weeks ago, I would not have a burnout now."
  - Counterfactuals
- "The king of France is bald."
  - Presuppositions extend the context in a non-trivial way, although there is nothing stated literally.
- "I am here."
  - Indexicals

San Diego vs. San Antonio

An experiment due to Goldstein & Gigerenzer (having to do with knowledge and rationality in general):

“Which city has more inhabitants: San Diego or San Antonio?”

This question was asked to American students and to German students.

Clearly German students knew little of San Diego, and many had never heard of San Antonio.

Results:

62% of the American students answered correctly: San Diego.

100% of the German students answered correctly: San Diego.

Gigerenzer proposes to use heuristics and cues to answer such questions resulting in a form of *bounded rationality*.

In any case, there is a certain tension between bounded rationality and classical logic and knowledge representation.

Probability and human intelligence

The Linda problem (Tversky & Kahneman, 1983)

Suppose Linda is 31 years old, single, outspoken, and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in antinuclear demonstrations.

Subjects are asked to rank the following statements by their probability:

1. Linda is a teacher in an elementary school
2. Linda works in a bookstore and takes Yoga classes
3. Linda is active in the feminist movement
4. Linda is a psychiatric social worker
5. Linda is a member of the League of Women Voters
6. Linda is a bank teller
7. Linda is an insurance salesperson
8. Linda is a bank teller and is active in the feminist movement.

Probability and human intelligence

Some remarks concerning the results of the study.

- In Tversky & Kahneman's (1983), 85% rated 8. as more probable than 6.
- Subjects' behavior is not in accordance to probability theory.
  - This is called the conjunction fallacy.
- Several proposals for explaining the behavior have been proposed.

Are Humans Stupid?

It seems to be the case that humans

- sometimes have problems to think logically,
- sometimes have problems to use properly their knowledge,
- sometimes have problems to decide according to the laws of probability theory.

Furthermore:

- Learning logic / mathematics in general seems to be hard for humans.
- Performing formal deductions seem to be even harder, than classical mathematical reasoning.

Does this exemplify that humans are stupid?

Actually, I think we are extremely smart. It is only that we are no deduction machines, we do not think in terms of Kolmogorov axioms etc.

We use other mechanisms for being intelligent.

Remarks on Possible Cognitive Mechanisms for Reasoning

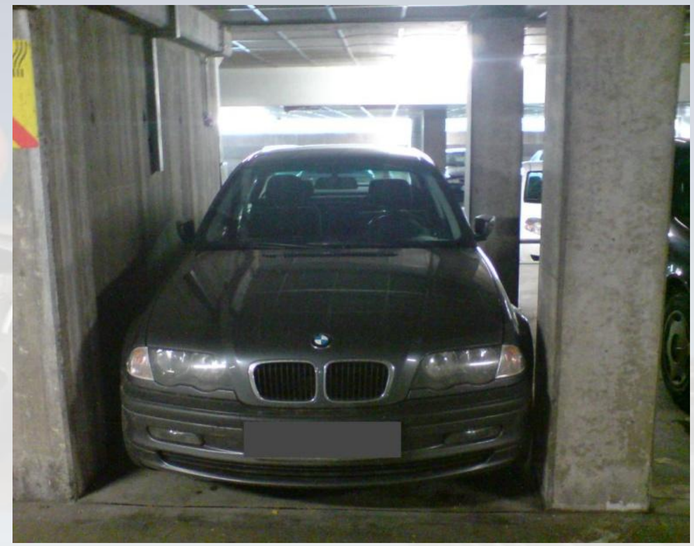
AI and Rationality

Although it is not really clear what human rationality is, I think that the mentioned psychological findings only show that classical logic is not appropriate for a variety of types of human behavior including types of behavior we would intuitively call rational.

If we do not want to fundamentally reject symbolic approaches, a natural strategy is to develop non-classical forms of reasoning to address these problems:

- Tense logic for temporal aspects of actions (similarly for spatial aspects of actions).
- Generalized quantifiers to extend types of quantification.
- Heuristic-based types of reasoning for bounded rationality.
- Reasoning with contexts for implicatures, presuppositions, and indexicals.
- Many-valued logic for dealing with underdetermined (or overdetermined) situations.
- Combination of logic and probability theory.
- Fuzzy logic for addressing vagueness.
- Neural-inspired approaches for reasoning.
- Etc.

Concept Formation



A mental approximations of the concept „narrow“ can emerge from different situations, e.g. where someone squeezed herself through a narrow gangway, where a car is parked in a narrow parking lot etc.
→ Similarity

Emergence of Abstract Concepts

According to Lakoff & Núñez (2000) mathematics origins from concrete domains of human activity (mathematical metaphors)

motion along a path	arithmetic
acts of moving along the path	arith. operations
a point location on the path	result of an operation; number
origin; beginning of the path	zero
unit location, a point location distinct from the origin	one
further from the origin than	greater
closer to the origin than	less
moving away from the origin a distance	addition
moving toward the origin a distance	subtraction

object collection	arithmetic
collections of objects of the same size	numbers
size of collection	number
bigger	greater
smaller	less
smallest collection	the unit (one)
putting collections together	addition
taking a smaller collection from a larger collection	subtraction

Lakoff & Núñez (2000) claim that there are more basic mathematical metaphors of the sketched type.

→ Metaphors containing action aspects.

Analogies

Metaphors:

"Gills are the lungs of fish."

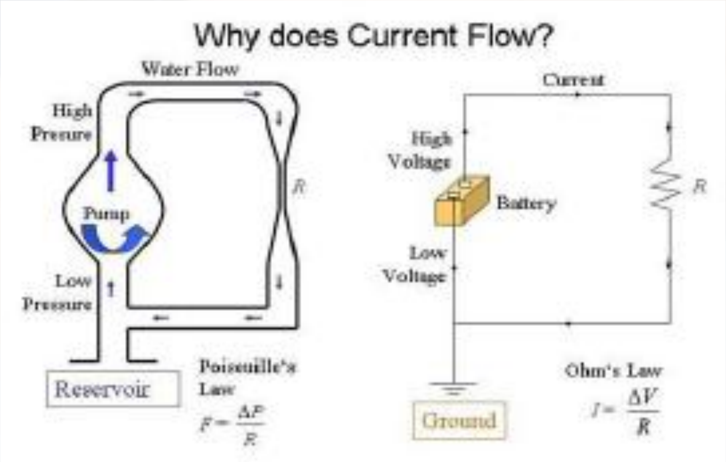
"Electrons are the planets of the nucleus."

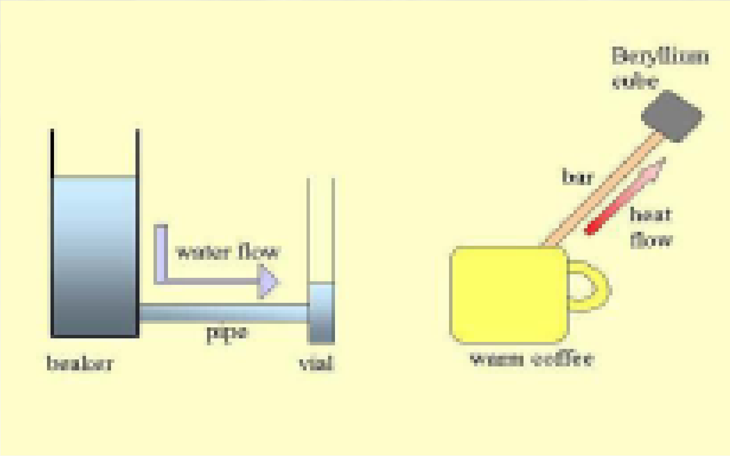
„Juliet is the sun."

„Lawyers are sharks."

Similarity of shape

Technical Concepts





Analogies in the Physics Domain

Analogies

Proportional analogies in the string domain (Hofstadter):

$$abba : cddc \approx effe : ??$$
$$abcabc : cbacba \approx defdef : ??$$
$$abc : abd \approx kji : ??$$

- Art
    - Jan van Eyck: The Arnolfini Marriage
    - This piece of art is full of metaphors.

Advertisement

Analogies

Analogies can be used (have been proposed) to explain phenomena like

- Creativity (art, advertisement)
- Learning of new concepts (physics domain)
- Development of abstract concepts (Emergence of numbers in children)
- Problem solving (proportional analogies in the string domain)
- Puzzles in the semantics of natural language (metaphors)
- Categorization of objects (perception)
- Etc.

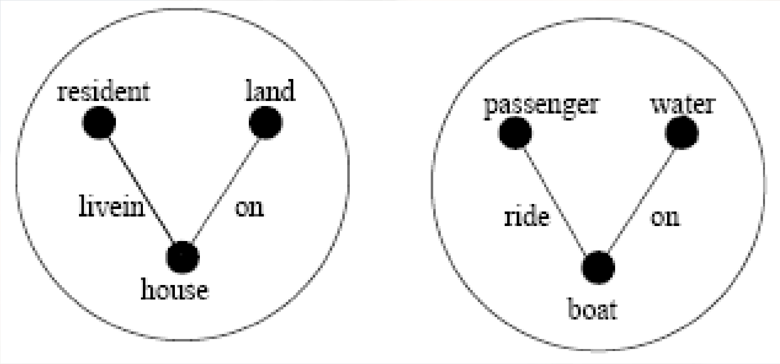
Notice: All these remarkable abilities are facets of intelligence.

→ **Analogy-making is a further mechanism for intelligence.**

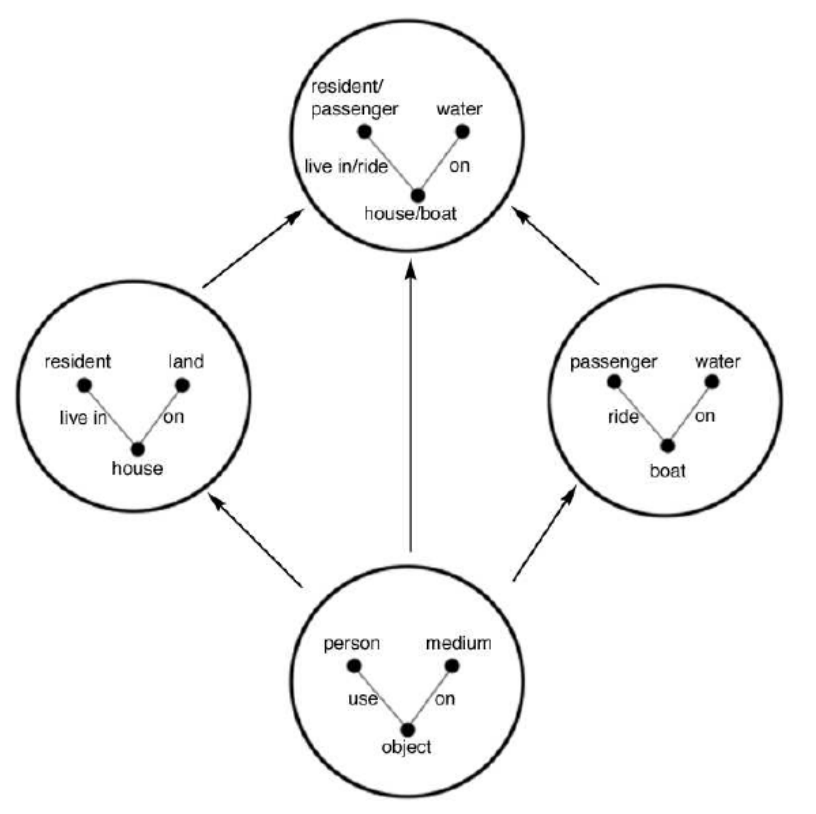
Conceptual blending

Example: semantics of “boathouse” and “houseboat”

Goguen’s (2006) blending theory allows to merge attributes and relations from different concepts to a new “blending space”.



The idea of mental spaces was discussed by Fauconnier, Gärdenfors, Rosch and others.



→ Conceptual blending is a further cognitive mechanism.

Conceptual Blending in Music

- We saw already in the first week of this course several examples of conceptual blending in music:
  - Blending to invent new chord progressions
    - Tritone substitution
    - Backdoor progression
  - Blending as a means to cross-fade chord progressions
    - Applications: modulations
- Blending does not only appear on the harmonization level.
  - Charles Ives: The Unanswered Question
    - https://www.youtube.com/watch?v=kkaOz48cq2g
  - It seems to be the case that three musical epochs are blended in one piece of music.

Tasks

- Build computational models for
  - Analogies
  - Similarity relations
  - Metaphors
  - Conceptual blends
  - Etc.
- Models for such cognitive mechanisms may be based on a variety of methods including logic-based methods, neuro-inspired frameworks, probabilistic theories etc.

Some Further Readings

- Resolution
  - U. Schöning: Logik für Informatiker, Spektrum, Heidelberg, Berlin, 2000.
  - U. Schöning: Logic for Computer Scientists, Birkhäuser, 1994.
- Rationality Puzzles
  - Wason, P. C.: Reasoning. In: B. Foss (ed.), New Horizons in psychology. Harmondsworth, Penguin, 1966.
  - Tversky, A., Kahneman, D.: Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment. Psychological Review 90(4) (1983) 293–315.
- Analogical Reasoning
  - Schwering, A., Krumnack, U., Kühnberger, K.-U., Gust, H.: Syntactic Principles of Heuristic-Driven Theory Projection. Cognitive Systems Research 10(3) (2009) 251–269.
- Further Reading
  - Eppe, M. et al. Computational Invention of Cadences and Chord Progressions by Conceptual Chord-Blending, International Joint Conference on Artificial Intelligence 2015.