Logics for Knowledge Representation

0. General introduction to logic

Luciano Serafini

Fondazione Bruno Kessler, Trento, Italy

September 20, 2012

Representing and reasoning about knowledge

- Often we want to describe and reason about real world phenomena:
  - Providing a complete description of the real world is clearly impossible, and maybe also useless.
  - Typically one is interested in a portion of the world, e.g., a particular physical phenomenon, a social aspect, or modeling rationality of people, . . .

Representing and reasoning about knowledge

- Often we want to describe and reason about real world phenomena:
  - Providing a complete description of the real world is clearly impossible, and maybe also useless.
  - Typically one is interested in a portion of the world, e.g., a particular physical phenomenon, a social aspect, or modeling rationality of people, ...
- We use sentences of a language to describe objects of the real world, their properties, and facts that hold.
  - The language can be:
    - informal (natural lang., graphical lang., icons, ...) or
    - formal (logical lang., programming lang., mathematical lang., ...)
    - mixed languages, i.e., languages with parts that are formal, and others that are informal (e.g., UML class diagrams)

Representing and reasoning about knowledge

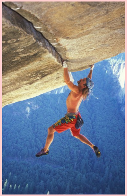
- Often we want to describe and reason about real world phenomena:
  - Providing a complete description of the real world is clearly impossible, and maybe also useless.
  - Typically one is interested in a portion of the world, e.g., a particular physical phenomenon, a social aspect, or modeling rationality of people, . . .
- We use sentences of a language to describe objects of the real world, their properties, and facts that hold.
  - The language can be:
    - informal (natural lang., graphical lang., icons, . . .) or
    - formal (logical lang., programming lang., mathematical lang., . . .)
    - mixed languages, i.e., languages with parts that are formal, and others that are informal (e.g., UML class diagrams)
- If we are also interested in a more rigorous description of the phenomena, we provide a mathematical model:
  - Is an abstraction of the portion of the real world we are interested in.
  - It represents real world entities in the form of mathematical objects, such as sets, relations, functions, . . .

Language, real world, and math. model: Example

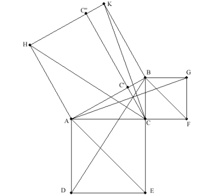
Language

In any right triangle, the area of the square whose side is the hypotenuse (the side opposite the right angle) is equal to the sum of the areas of the squares whose sides are the two legs (the two sides that meet at a right angle).

Real world



Mathematical model



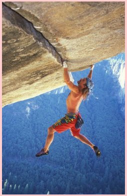
Facts about euclidean geometry can be expressed in terms of natural language, and they can refer to one or more real world situations. (In the picture it refers to the composition of the forces in free climbing). However, the importance of the theorem lays in the fact that it describes a general property that holds in many different situations. All these different situations can be abstracted in the mathematical structure which is the euclidean geometry. So indeed the sentence can be interpreted directly in the mathematical structure. In this example the language is informal but it has an interpretation in a mathematical structure.

Language, real world, and math. model: Example 2

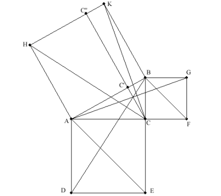
Language

In a triangle ABC, if $\widehat{BAC}$ is right, then
$\overline{AB}^2 + \overline{AC}^2 = \overline{BC}^2$.

Real world



Mathematical model



This example is obtained from the previous one by taking a language that is "more formal". Indeed the language mixes informal statements (e.g., "if ... then ..." or "is right") with some formal notation.
E.g., $\widehat{BAC}$ is an unambiguous and compact way to denote an angle.
Similarly $\overline{AB}^2 + \overline{AC}^2 = \overline{BC}^2$ is a rigorous description of an equation that holds between the lengths of the triangle sides.

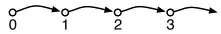
Language, real world, and math. model: Example 3

Language

$$x - 2y + 3$$
$$x + y = 0$$

Real world

Mathematical model



In this example the language is purely formal, i.e., the language of arithmetic.
This abstract language is used to represent many situations in the real world (in the primary school we have many examples about apples, pears, and how they cost, which are used by teachers to explain to kids the intuitive meaning of the basic operations on numbers).
The mathematical model in this case is the structure of natural numbers.

What is Logic?

Main objective

- The main objective of a logic (there is not a unique logic but many) is to express by means of a formal language the knowledge (the truth) about a certain phenomena or a certain portion of the world
- ...and to codify by means of an proof system that allow to rigorously demonstrate what are the other facts (truths) that follows from a set of hypothetical truths.

What is Logic?

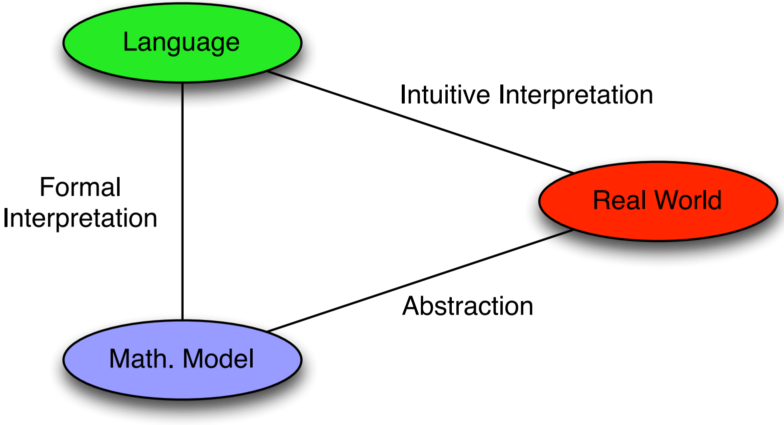
Main objective

- The main objective of a logic (there is not a unique logic but many) is to express by means of a formal language the knowledge (the truth) about a certain phenomena or a certain portion of the world
- ...and to codify by means of an proof system that allow to rigorously demonstrate what are the other facts (truths) that follows from a set of hypothetical truths.

Additional components ...

- Logical languages usually have a formal semantics which maps logical expressions into objects or propositions of a mathematical structure, which abstractly represents the domain of discourse.
- Proof system can be encoded in a set of inference rules that can be successively applied in order to infer all the possible truth from the initial hypothesis, or they can be encoded in an algorithm (usually called Decision procedure) that can check if a certain truth follows from the truth of other facts.

Connections between language, world, and math. model



Connections between language, world, and math. model

Intuitive interpretation (or informal semantics)

Every element of a logical langauge should be associate to any element of the language an interpretation in the real world. This is called the intuitive interpretation (or informal semantics). E.g., in learning a new programming language, you need to understand what is the effect in terms of execution of all the languages construct. For this reason the manual, typically, reports in natural language and with examples, the behavior of the language primitives.

Formal interpretation (or formal semantics)

Is a function that maps the elements of the language (i.e., symbols, words, complex sentences, ...) into one or more elements of the mathematical structure. It is indeed the formalization of the intuitive interpretation (or the intuitive semantics).

Abstraction

Is the link that connects the real world with it's mathematical and abstract representation into a mathematical structure. If a certain situation is supposed to be abstractly described by a given structure, then the abstraction connects the elements that participate to the situation, with the components of the mathematical structure, and the properties that hold in the situation with the mathematical properties that hold in the structure

Logic

In the modern are a logic is usually defined by specifying the following main components:

- The logical language, which must be a formal langauge
- The formal interpretation, i.e., a mapping from the language to a (class of) mathematical structures that allow to formally define some notion of truth.
- The notion of logical consequence between formulas.
  I.e., the conditions under which, if a set Γ of formulas are true then also φ is true.

Formal language

- We are given a non-empty set Σ of symbols called alphabet.
- A formal language (over Σ) is a subset L of Σ^*, i.e., a set of finite strings of symbols in Σ.
- The elements of L are called well formed phrases.
- Formal languages can be specified by means of a grammar, i.e., a set of formation rules that allow one to build complex well formed phrases starting from simpler ones.

Logical language

A language of a logic, i.e., a logical language is a formal language that has the following characteristics:

- The alphabet typically contains basic symbols that are used to indicate the basic (atomic) components of the (part of the) world the logic is supposed to describe.
  Examples of such atomic objects are, individuals, functions, operators, truth-values, propositions, ...
- The grammar of a logical language defines all the possible ways to construct complex phrases starting from simpler ones.
  - A logical grammar always specifies how to build formulas, which are phrases that denotes propositions, i.e., objects that can assume some truth value (e.g., true, false, true in certain situations, true with probability of 3%, true/false in a period of time, ...).
  - Another important family of phrases which are usually defined in logic are terms which usually denote objects of the world (e.g., cats, dogs, time points, quantities, ...).

Alphabet

The alphabet of a logical language is composed of two classes of symbols:

- Logical constants, whose formal interpretation is constant and fixed by the logic (e.g., $\wedge$, $\forall$, $=$, $\ldots$).
- Non logical symbols, whose formal interpretation is not fixed by the logic, and must be defined by the "user".

Alphabet

The alphabet of a logical language is composed of two classes of symbols:

- Logical constants, whose formal interpretation is constant and fixed by the logic (e.g., $\wedge$, $\forall$, $=$, $\ldots$).
- Non logical symbols, whose formal interpretation is not fixed by the logic, and must be defined by the "user".

We can make an analogy with programming languages (say C, C++, python):

- Logical constants correspond to reserved words (whose meaning is fixed by the interpreter/compiler).
- Non logical symbols correspond to the identifiers that are introduced by the programmer for defining functions, variables, procedures, classes, attributes, methods, ...
  The meaning of these symbols is fixed by the programmer.

Alphabet: Logical constants – Example

The logical constants depend on the logic we are considering:

- *Propositional logic*: $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), $\supset$ (implication), $\equiv$ (equivalence), $\bot$ (falsity).
  These are usually called propositional connectives.

Alphabet: Logical constants – Example

The logical constants depend on the logic we are considering:

- *Propositional logic*: $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), $\supset$ (implication), $\equiv$ (equivalence), $\perp$ (falsity).
  These are usually called propositional connectives.
- *Predicate logic*: in addition to the propositional connectives, we have quantifiers:
  - universal quantifier $\forall$, standing for "every object is such that ..."
  - existential quantifier $\exists$, standing for "there is some object that ..."

Alphabet: Logical constants – Example

The logical constants depend on the logic we are considering:

- *Propositional logic*: $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), $\supset$ (implication), $\equiv$ (equivalence), $\perp$ (falsity).
  These are usually called propositional connectives.
- *Predicate logic*: in addition to the propositional connectives, we have quantifiers:
  - universal quantifier $\forall$, standing for "every object is such that . . ."
  - existential quantifier $\exists$, standing for "there is some object that . . ."
- *Modal logic*: in addition to the propositional connectives, we have modal operators:
  - $\Box$, standing for "it is necessarily true that . . ."
  - $\Diamond$, standing for "it is possibly true that . . .".

Alphabet: Non-logical symbols – Example

- *Propositional logic*: non logical symbols are called propositional variables, and represent (i.e., have intuitive interpretation) propositions.
  The proposition associated to each propositional variable is not fixed by the logic.

Alphabet: Non-logical symbols – Example

- *Propositional logic*: non logical symbols are called propositional variables, and represent (i.e., have intuitive interpretation) propositions.
  The proposition associated to each propositional variable is not fixed by the logic.
- *Predicate logic*: there are four families of non logical symbols:
  - Variable symbols, which represent any object.
  - Constant symbols, which represent specific objects.
  - Function symbols, which represent transformations on objects.
  - Predicate symbols, which represent relations between objects.

Alphabet: Non-logical symbols – Example

- *Propositional logic*: non logical symbols are called propositional variables, and represent (i.e., have intuitive interpretation) propositions.
  The proposition associated to each propositional variable is not fixed by the logic.
- *Predicate logic*: there are four families of non logical symbols:
  - Variable symbols, which represent any object.
  - Constant symbols, which represent specific objects.
  - Function symbols, which represent transformations on objects.
  - Predicate symbols, which represent relations between objects.
- *Modal logic*: non logical symbols are the same as in propositional logic, i.e., propositional variables.

Example of grammar: Language of propositional logic

Grammar of propositional logic

Allows one to define the unique class of phrases, called formulas (or well formed formulas), which denote propositions.

$$
\begin{array}{lcll}
\texttt{Formula} & \longrightarrow & P & (P \text{ is a propositional variable}) \\
 & \mid & (\texttt{Formula} \wedge \texttt{Formula}) & \\
 & \mid & (\texttt{Formula} \vee \texttt{Formula}) & \\
 & \mid & (\texttt{Formula} \rightarrow \texttt{Formula}) & \\
 & \mid & (\neg \texttt{Formula}) &
\end{array}
$$

Example of grammar: Language of propositional logic

Grammar of propositional logic

Allows one to define the unique class of phrases, called formulas (or well formed formulas), which denote propositions.

$$
\begin{array}{lcll}
\texttt{Formula} & \longrightarrow & P & (P \text{ is a propositional variable}) \\
 & | & (\texttt{Formula} \land \texttt{Formula}) & \\
 & | & (\texttt{Formula} \lor \texttt{Formula}) & \\
 & | & (\texttt{Formula} \rightarrow \texttt{Formula}) & \\
 & | & (\neg\texttt{Formula}) &
\end{array}
$$

Example (Well formed formulas)

$$(P \land (Q \rightarrow R)) \qquad ((P \rightarrow (Q \rightarrow R)) \lor P)$$

These formulas are well formed, because there is a sequence of applications of grammar rules that

Example of grammar: Language of propositional logic

Grammar of propositional logic

Allows one to define the unique class of phrases, called formulas (or well formed formulas), which denote propositions.

$$
\begin{array}{lcll}
\texttt{Formula} & \longrightarrow & P & (P \text{ is a propositional variable}) \\
 & \mid & (\texttt{Formula} \wedge \texttt{Formula}) & \\
 & \mid & (\texttt{Formula} \vee \texttt{Formula}) & \\
 & \mid & (\texttt{Formula} \rightarrow \texttt{Formula}) & \\
 & \mid & (\neg\texttt{Formula}) &
\end{array}
$$

Example (Well formed formulas)

$$(P \wedge (Q \rightarrow R)) \qquad ((P \rightarrow (Q \rightarrow R)) \vee P)$$

These formulas are well formed, because there is a sequence of applications of grammar rules that

Example (Non well formed formulas)

$$P(Q \rightarrow R)$$
$$(P \rightarrow \vee P)$$

Example of grammar: Language of first order logic

Grammar of first order logic

Term	$\leftrightarrow$	x	(x is a variable symbol)
	|	c	(c is a constant symbol)
	|	$f(\texttt{Term}, \dots, \texttt{Term})$	(f is a function symbol)
Formula	$\leftrightarrow$	$P(\texttt{Term}, \dots, \texttt{Term})$	(P is a predicate symbol)
	|	$\texttt{Formula} \wedge \texttt{Formula}$	
	|	$\texttt{Formula} \vee \texttt{Formula}$	
	|	$\texttt{Formula} \rightarrow \texttt{Formula}$	
	|	$\neg\texttt{Formula}$	
	|	$\forall x(\texttt{Formula})$	(x is a variable symbol)
	|	$\exists x(\texttt{Formula})$	(x is a variable symbol)

The rules define two types of phrases:

- terms denote objects (they are like noun phrases in natural language)
- formulas denote propositions (they are like sentences in natural language)

Exercise

Give examples of terms and formulas, and of phrases that are neither of

Example of grammar: Language of a description logic

Grammar of the description logic $\mathcal{ALC}$

$$
\begin{array}{lcll}
\texttt{Concept} & \leftrightarrow & A & (A \text{ is a concept symbol}) \\
 & | & \texttt{Concept} \sqcup \texttt{Concept} & \\
 & | & \texttt{Concept} \sqcap \texttt{Concept} & \\
 & | & \neg\texttt{Concept} & \\
 & | & \exists\texttt{RoleConcept} & \\
 & | & \forall\texttt{RoleConcept} & \\
\texttt{Role} & \leftrightarrow & R & (R \text{ is a role symbol}) \\
\texttt{Individual} & \leftrightarrow & a & (a \text{ is an individual symbol}) \\
\texttt{Formula} & \leftrightarrow & \texttt{Concept} \sqsubseteq \texttt{Concept} & \\
 & | & \texttt{Concept}(\texttt{Individual}) & \\
 & | & \texttt{Role}(\texttt{Individual}, \texttt{Individual}) &
\end{array}
$$

Example (Concepts and formulas of the DL $\mathcal{ALC}$)

- Concepts: $A \sqcap B$, $A \sqcup \exists R.C$, $\forall S.(C \sqcup \forall R.D) \sqcup \neg A$
- Formulas: $A \sqsubseteq B$, $A \sqsubseteq \exists R.B$, $A(a)$, $R(a, b)$, $\exists R.C(a)$

Intuitive interpretation of a logical language

While, non logical symbols do not have a fixed formal interpretation, they usually have a fixed intuitive interpretation. Consider for instance:

Type	Symbol	Intuitive interpretation
propositional variable	$rain$	it is raining
constant symbol	$MobyDick$	the whale of a novel by Melville
function symbol	$color(x)$	the color of the object x
predicate symbol	$Friends(x, y)$	x and y are friends

Intuitive interpretation of a logical language

While, non logical symbols do not have a fixed formal interpretation, they usually have a fixed intuitive interpretation. Consider for instance:

Type	Symbol	Intuitive interpretation
propositional variable	$rain$	it is raining
constant symbol	$MobyDick$	the whale of a novel by Melville
function symbol	$color(x)$	the color of the object x
predicate symbol	$Friends(x, y)$	x and y are friends

The intuitive interpretation of the non logical symbols does not affect the logic itself.

- In other words, changing the intuitive interpretation does not affect the properties that will be proved in the logic.
- Similarly, replacing these logical symbols with less evocative ones, like r, M, $c(x)$, $F(x, y)$ will not affect the logic.

Interpretation of complex formulas

The intuitive interpretation of complex formulas is done by combining the intuitive interpretations of the components of the formulas.

Example

Consider the propositional formula:

$$(\texttt{raining} \lor \texttt{snowing}) \rightarrow \neg \texttt{go\_to\_the\_beach}$$

If the intuitive interpretations of the symbols are:

symbol	**intuitive meaning**
`raining`	it is raining
`snowing`	it is snowing
`go_to_the_beach`	we go to the beach
$\lor$	either ... or ...
$\rightarrow$	if ... then ...
$\neg$	it is not the case that ...

then the above formula intuitively represent the proposition:

if (it is raining or it is snowing) then it is not the case that (we go to the beach)

Formal model

- Class of models: The models in which a logic is *formally interpreted* are the members of a class of algebraic structures, each of which is an abstract representation of the relevant aspects of the (portion of the) world we want to formalize with this logic.
- Models represent only the components and aspects of the world which are relevant to a certain analysis, and abstract away from irrelevant facts. Example: if we are interested in the everage temperature of each day, we can represent time with the natural numbers and use a function that associates to each natural number a floating point number (the average temperature of the day corresponding to the point).
- **Applicability of a model:** Since the real world is complex, in the construction of the formal model, we usually do simplifying assumptions that bound the usability of the logic to the cases in which these assumptions are verified.
  Example: if we take integers as formal model of time, then this model is not applicable to represent continuous change.
- Each model represents a single possible (or impossible) state of the world. The class of models of a logic will represent all the (im)possible states of the world.

Formal interpretation

- Given a structure S and a logical language L, the formal interpretation in S of L is a function that associates an element of S to any non logical symbol of the alphabet.
- The formal interpretation in the algebraic structure is the parallel counterpart (or better, the formalization) of the intuitive interpretation in the real world.
- The formal interpretation is specified only for the non logical symbols.
- Instead, the formal interpretation of the logical symbols is fixed by the logic.
- The formal interpretation of a complex expression e, obtained as a combination of the sub-expressions $e_1, \ldots, e_n$, is uniquely determined as a function of the formal interpretation of the sub-components $e_1, \ldots, e_n$.

Truth in a structure: Models

- As said, the goal of logic is the formalization of what is true/false in a particular world. The particular world is formalized by a structure, also called an interpretation.
- The main objective of the formal interpretation is that it allows to define when a formula is true in an interpretation.
- Every logic therefore defines the satisfiability relation (denoted by $\models$) between interpretations and formulas.
- If $\mathcal{I}$ is an interpretation and φ a formula, then

$$\mathcal{I} \models \varphi$$

  stands for the fact that $\mathcal{I}$ satisfies φ, or equivalently that φ is true in $\mathcal{I}$.
- An interpretation $\mathcal{M}$ such that $\mathcal{M} \models \varphi$ is called a model of φ.

(Un)satisfiability and validity

On the basis of truth in an interpretation ($\models$) the following notions are defined in any logic:

- φ is satisfiable if it has model, i.e., if there is a structure $\mathcal{M}$ such that $\mathcal{M} \models \varphi$.
- φ is un-satisfiable if it is not satisfiable, i.e., it has no models.
- φ is valid, denoted $\models \varphi$, if is true in all interpretations.

Logical consequence (or implication)

- The notion of logical consequence (or implication) is defined on the basis of the notion of truth in an interpretation.
- Intuitively, a formula φ is a logical consequence of a set of formulas (sometimes called assumptions) Γ (denoted $\Gamma \models \varphi$) if such a formula is true under this set of assumptions.
- Formally, $\Gamma \models \varphi$ holds when:

  $$\text{For all interpretations } \mathcal{I}\text{, if } \mathcal{I} \models \Gamma \text{ then } \mathcal{I} \models \varphi.$$

  In words: φ is true in all the possible situations in which all the formulas in Γ are true.
- Notice that the two relations, "truth in a model" and "logical consequence" are denoted by the same symbol $\models$ (this should remind you that they are tightly connected).

Difference between $\models$ and implication ($\rightarrow$)

At a first glance $\models$ looks like implication (usually denoted by $\rightarrow$ or $\supset$). Indeed in most of the cases they represent the same relation between formulas.

Similarity

- For instance, in propositional logic (but not only) the fact that φ is a logical consequence of the singleton set $\{\psi\}$, i.e., $\{\psi\} \models \varphi$, can be encoded in the formula $\psi \rightarrow \varphi$.
- Similarly, the fact that φ is a logical consequence of the set of formulas $\{\varphi_1, \ldots, \varphi_n\}$, i.e., $\{\varphi_1, \ldots, \varphi_n\} \models \varphi$ can be encoded by the formula $\varphi_1 \wedge \cdots \wedge \varphi_n \rightarrow \varphi$.

Difference

- When $\Gamma = \{\gamma_1, \gamma_2, \ldots\}$ is an infinite set of formulas, the fact that φ is a logical consequence of Γ cannot be represented with a formula $\gamma_1 \wedge \gamma_2 \wedge \cdots \rightarrow \varphi$ because this would be infinite, and in logic all the formulas are finite. (Actually there are logics, called infinitary logics, where formulas can have infinite size.)

Logical consequence, validity and (un)satisfiability

Exercise

Show that if $\Gamma = \emptyset$, then $\Gamma \models \varphi \iff \varphi$ is valid.

Logical consequence, validity and (un)satisfiability

Exercise

Show that if $\Gamma = \emptyset$, then $\Gamma \models \varphi \iff \varphi$ is valid.

Solution

($\Longrightarrow$) *Since* Γ *is empty, every interpretation* $\mathcal{I}$ *satisfies all the formulas in* Γ*. Therefore, if* $\Gamma \models \varphi$*, then every interpretation* $\mathcal{I}$ *must satisfy* φ*, hence* φ *is valid.*

($\Longleftarrow$) *If* φ *is valid, then every* $\mathcal{I}$ *is such that* $\mathcal{I} \models \varphi$*. Hence, whatever* Γ *is (in particular, when* $\Gamma = \emptyset$*), every model of* Γ *is also a model of* φ*, and so* $\Gamma \models \varphi$*.*

Logical consequence, validity and (un)satisfiability

Exercise

Show that if $\Gamma = \emptyset$, then $\Gamma \models \varphi \iff \varphi$ is valid.

Solution

($\Longrightarrow$) *Since* Γ *is empty, every interpretation* $\mathcal{I}$ *satisfies all the formulas in* Γ*. Therefore, if* $\Gamma \models \varphi$*, then every interpretation* $\mathcal{I}$ *must satisfy* φ*, hence* φ *is valid.*

($\Longleftarrow$) *If* φ *is valid, then every* $\mathcal{I}$ *is such that* $\mathcal{I} \models \varphi$*. Hence, whatever* Γ *is (in particular, when* $\Gamma = \emptyset$*), every model of* Γ *is also a model of* φ*, and so* $\Gamma \models \varphi$*.*

Exercise

Show that if φ is unsatisfiable then $\{\varphi\} \models \psi$ for every formula ψ.

Logical consequence, validity and (un)satisfiability

Exercise

Show that if $\Gamma = \emptyset$, then $\Gamma \models \varphi \iff \varphi$ is valid.

Solution

($\Longrightarrow$) Since Γ is empty, every interpretation $\mathcal{I}$ satisfies all the formulas in Γ. Therefore, if $\Gamma \models \varphi$, then every interpretation $\mathcal{I}$ must satisfy φ, hence φ is valid.
($\Longleftarrow$) If φ is valid, then every $\mathcal{I}$ is such that $\mathcal{I} \models \varphi$. Hence, whatever Γ is (in particular, when $\Gamma = \emptyset$), every model of Γ is also a model of φ, and so $\Gamma \models \varphi$.

Exercise

Show that if φ is unsatisfiable then $\{\varphi\} \models \psi$ for every formula ψ.

Solution

If φ is unsatisfiable then it has no model, which implies that each interpretation that satisfies φ (namely, none) satisfies also ψ, independently from ψ.

Properties of logical consequence

[

Property] Show that the following properties hold for the logical consequence relation defined above:

Reflexivity: $\Gamma \cup \{\varphi\} \models \varphi$

Monotonicity: $\Gamma \models \varphi$ implies that $\Gamma \cup \Sigma \models \varphi$

Cut: $\Gamma \models \varphi$ and $\Sigma \cup \{\varphi\} \models \psi$ implies that $\Gamma \cup \Sigma \models \psi$

Properties of logical consequence

[

Property] Show that the following properties hold for the logical consequence relation defined above:

Reflexivity: $\Gamma \cup \{\varphi\} \models \varphi$

Monotonicity: $\Gamma \models \varphi$ implies that $\Gamma \cup \Sigma \models \varphi$

Cut: $\Gamma \models \varphi$ and $\Sigma \cup \{\varphi\} \models \psi$ implies that $\Gamma \cup \Sigma \models \psi$

Solution

Reflexivity: *If $\mathcal{I}$ satisfies all the formulas in $\Gamma \cup \{\varphi\}$ then it satisfies also φ, and therefore $\Gamma \cup \{\varphi\} \models \varphi$.*

Monotonicity: *Let $\mathcal{I}$ be an interpretation that satisfies all the formulas in $\Gamma \cup \Sigma$. Then it satisfies all the formulas in Γ, and if $\Gamma \models \varphi$, then $\mathcal{I} \models \varphi$. Therefore, we can conclude that $\Gamma \cup \Sigma \models \varphi$.*

Cut: *Let $\mathcal{I}$ be an interpretation that satisfies all the formulas in $\Gamma \cup \Sigma$. Then it satisfies all the formulas in Γ, and if $\Gamma \models \varphi$, then $\mathcal{I} \models \varphi$. This implies that $\mathcal{I}$ satisfies all the formulas in $\Sigma \cup \{\varphi\}$. Then, since $\Sigma \cup \{\varphi\} \models \psi$, we have that $M \models \psi$. Therefore we can conclude that $\Gamma \cup \Sigma \models \psi$.*

Checking logical consequence

Problem

Does there exist an algorithm that checks if a formula φ is a logical consequence of a set of formulas Γ?

Checking logical consequence

Problem

Does there exist an algorithm that checks if a formula φ is a logical consequence of a set of formulas Γ?

Solution 1: If Γ is finite and the set of models of the logic is finite, then it is possible to directly apply the definition by checking for every interpretation $\mathcal{I}$, that if $\mathcal{I} \models \Gamma$ then, $\mathcal{I} \models \varphi$.

Checking logical consequence

Problem

Does there exist an algorithm that checks if a formula φ is a logical consequence of a set of formulas Γ?

Solution 1: If Γ is finite and the set of models of the logic is finite, then it is possible to directly apply the definition by checking for every interpretation $\mathcal{I}$, that if $\mathcal{I} \models \Gamma$ then, $\mathcal{I} \models \varphi$.

Solution 2: If Γ is infinite or the set of models is infinite, then Solution 1 is not applicable as it would run forever. An alternative solution could be to generate, starting from Γ, all its logical consequences by applying a set of rules.

Checking logical consequence

Propositional logic: The method based on truth tables can be used to check logical consequence by enumerating all the interpretations of Γ and φ and checking if every time all the formulas in Γ are true then φ is also true.
This is possible because, when Γ is finite then there are a finite number of interpretations.
First order logic: A first order language in general has an infinite number of interpretations. Therefore, to check logical consequence, it is not possible to apply a method that enumerates all the possible interpretations, as in truth tables.
Modal logic: presents the same problem as first order logic.
In general for a set of formulas Γ, there is an infinite number of interpretations, which implies that a method that enumerates all the interpretations is not effective.

Checking logical consequence – Deductive methods

- An alternative method for determining if a formula is a logical consequence of a set of formulas is based on inference rules.
- An inference rule is a rewriting rule that takes a set of formulas and transforms it in another formulas.
- The following are examples of inference rules.

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi} \qquad \frac{\varphi \quad \psi}{\varphi \rightarrow \psi} \qquad \frac{\forall x.\varphi(x)}{\varphi(c)} \qquad \frac{\exists x.\varphi(x)}{\varphi(d)}$$

- Differently from truth tables, which apply a brute force exhaustive analysis not interpretable by humans, the deductive method simulates human argumentation and provides also an understandable explanation (i.e., a deduction) of the reason why a formula is a logical consequence of a set of formulas.

Inference rules to check logical consequence – Example

Let $\Gamma = \{p \to q, \neg p \to r, q \lor r \to s\}$.
[0.5] The following is a deduction (an explanation of) the fact that s is a logical consequence of Γ, i.e., that $\Gamma \models s$, which uses the following inference rules:

$$\frac{\varphi \to \psi \quad \neg\varphi \to \vartheta}{\psi \lor \vartheta}\ (*) \qquad\qquad \frac{\varphi \quad \varphi \to \psi}{\psi}\ (**)$$

Example of deduction

(1)	$p \to q$	Belongs to Γ.
(2)	$\neg p \to r$	Belongs to Γ.
(3)	$q \lor r$	By applying (*) to (1) and (2).
(4)	$q \lor r \to s$	Belongs to Γ.
(5)	s	By applying (**) to (3) and (4).

Hilbert-style inference methods

In a Hilbert-style deduction system, a formal deduction is a finite sequence of formulas

$$
\begin{array}{c}
\varphi_1 \\
\varphi_2 \\
\varphi_3 \\
\vdots \\
\varphi_n
\end{array}
$$

where each φ_i

- is either an axiom, or
- it is derived from previous formulas $\varphi_{j_1}, \ldots, \varphi_{j_k}$ with $j_1, \ldots, j_k < i$, by applying the inference rule

$$\frac{\varphi_{j_1}, \ \ldots, \ \varphi_{j_k}}{\varphi_i}$$

Hilbert axioms for classical propositional logic

Axioms

A1	$\varphi \to (\psi \to \varphi)$
A2	$(\varphi \to (\psi \to \theta)) \to ((\varphi \to \psi) \to (\varphi \to \theta))$
A3	$(\neg\psi \to \neg\varphi) \to ((\neg\psi \to \varphi) \to \psi)$

Inference rule(s)

MP $\dfrac{\varphi \quad \varphi \to \psi}{\psi}$

.5

Example (Proof of $A \to A$)

1.	A1	$A \to ((A \to A) \to A)$
2.	A2	$(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$
3.	MP(1,2)	$(A \to (A \to A)) \to (A \to A)$
4.	A1	$(A \to (A \to A))$
5.	MP(4,3)	$A \to A$

Natural Deduction

Can be used to show that $\Gamma \models \varphi$, i.e., that φ is a logical consequence of the formulas in Γ.

- Natural Deduction (ND) is called so because it mimics human reasoning in real life (in particular, in maths).
- A ND derivation of φ frrm Γ is a tree rooted at φ and with leaves in Γ.
- A ND proof is constructed starting from a set of assumptions (in Γ) by applying a set of inference rules.
- For every logical connective '$\circ$' there are two rules:
  - $\circ$I (introduction of $\circ$)
  - $\circ$E (elimination of $\circ$)

Propositional natural deduction

Example (Deduction in ND)

A natural deduction of $(P \wedge Q \to R) \to (P \to (Q \to R))$ is the following:

$$\dfrac{\dfrac{\dfrac{P \wedge Q \to R \qquad \dfrac{[P] \quad [Q]}{P \wedge Q}\wedge I}{R} \to E}{Q \to R} \to I}{P \to (Q \to R)} \to I$$

Natural deduction is not a decision procedure that "automatically checks" if a formula φ is a consequence of a set of formulas Γ. Instead it is a method for representing the reasoning done by humans. Usually ND proof are manually constructed.

Sequent calculus

The sequent calculus is an extension of ND calculus which is based on the notion of sequent, which is an expression of the form:

$$A_1, \dots, A_n \rightarrow B_1, \dots, B_m$$

It should be read as:

If $A_1, \dots, A_n$ are all true then at least one of the B_is is also true.

The informal understanding of the above sequent is the formula:

$$A_1 \wedge \cdots \wedge A_n \rightarrow B_1 \vee \cdots \vee B_m$$

Inference rules are of the form

$$\frac{\Gamma, A, B \rightarrow \Delta}{\Gamma, A \wedge B \rightarrow \Delta} \qquad \frac{\Gamma \rightarrow A, \Delta \quad \Gamma \rightarrow B, \Delta}{\Gamma \rightarrow A \wedge B, \Delta}$$

Propositional sequent calculus – Example

$$
\dfrac{\dfrac{\dfrac{\dfrac{A \to A}{A, B \to A} \quad \dfrac{B \to B}{A, B \to B}}{A, B \to A \wedge B}}{A \to B \to (A \wedge B)}}{\to A \to (B \to (A \wedge B))}
$$

Refutation

Reasoning by refutation is based on the principle of "Reductio ad absurdum".

Reductio ad absurdum

In order to show that a proposition φ is true, we assume that it is false (i.e., that $\neg\varphi$ holds) and try to infer a contradictory statement, such as $A \wedge \neg A$ (usually denoted by $\perp$, i.e., the false statement).

Reasoning by refutation is one of the most important principles for building automated decision procedures. This is mainly due to the fact that, proving a formula φ corresponds to the reduction of $\neg\varphi$ to $\perp$.

Propositional resolution

Propositional resolution is the most simple example of reasoning via refutation. The procedure can be described as follows:

Propositional resolution

INPUT: a propositional formula φ
OUTPUT: $\models \varphi$ or $\not\models \varphi$

1. Convert $\neg\varphi$ to conjunctive normal form, i.e., to a set C of formulas (called clauses) of the form

   [-1]

   $$p_1 \vee \cdots \vee p_k \vee \neg p_{k+1} \vee \cdots \vee \neg p_n$$

   that is logically equivalent to φ.

2. Apply exhaustively the following inference rule

   $$\frac{c \vee p \quad \neg p \vee c'}{c \vee c'} \; \textit{Resolution}$$

   [-1] and add $c \vee c'$ to C

3. if C contains two clauses p and $\neg p$ then return $\models \varphi$ otherwise return $\not\models \varphi$

Inference based on satisfiability checking

In order to show that $\models \varphi$ (i.e., that φ is valid) we search for a model of $\neg\varphi$, i.e., we show that $\neg\varphi$ is satisfiable.
If we are not able to find such a model, then we can conclude that there is no model of $\neg\varphi$, i.e., that all the models satisfy φ, which is: that φ is valid.

Inference based on satisfiability checking

There are two basic methods of searching for a model for φ:

SAT based decision procedures

- This method incrementally builds a model.
- At every stage it defines a "partial model" μ_i and does an early/lazy check if φ can be true in some extension of μ_i.
- At each point the algorithm has to decide how to extend μ_i to μ_{i+1} until constructs a full model for φ.

Tableaux based decision procedures

- This method builds the model of φ via a "top down" approach.
- I.e., φ is decomposed in its sub-formulas $\varphi_1, \ldots, \varphi_n$ and the algorithm recursively builds n models $M_1, \ldots, M_n$ for them.
- The model M of φ is obtained by a suitable combination of $M_1, \ldots, M_n$.

SAT based decision procedure – Example

We illustrate a SAT based decision procedure on a propositional logic example.
To find a model for $(p \vee q) \wedge \neg p$, we proceed as follows:

Partial model	lazy evaluation	result of lazy evaluation
$\mu_0 = \{p = \top\}$	$(\top \vee q) \wedge \neg \top$	$\bot$ (backtrack)
$\mu_1 = \{p = \bot\}$	$(\bot \vee q) \wedge \neg \bot$	p (continue)
$\mu_2 = \{p = \bot,\ q = \top\}$	$(\bot \vee \top) \wedge \neg \bot$	$\top$ (success!)

Soundness and Completeness

Let R be an inference method, and let $\vdash_R$ denote the corresponding inference relation.

Definition (Soundness of an inference method)

An inference method R is sound if

$$\begin{array}{rcll} \vdash_R \varphi & \Longrightarrow & \models \varphi & \\ \Gamma \vdash_R \varphi & \Longrightarrow & \Gamma \models \varphi & \qquad \text{(strongly sound)} \end{array}$$

Definition (Completeness of an inference method)

An inference method R is complete if

$$\begin{array}{rcll} \models \varphi & \Longrightarrow & \vdash_R \varphi & \\ \Gamma \models \varphi & \Longrightarrow & \Gamma \vdash_R \varphi & \qquad \text{(strongly complete)} \end{array}$$

Soundness and Completeness

Let R be an inference method, and let $\vdash_R$ denote the corresponding inference relation.

- An inference method R is sound if

$$\vdash_R \varphi \implies \models \varphi$$
$$\Gamma \vdash_R \varphi \implies \Gamma \models \varphi \quad \text{(strongly sound)}$$

- An inference method R is complete if:

$$\models \varphi \implies \Gamma \vdash_R \varphi$$
$$\Gamma \models \varphi \implies \Gamma \vdash_R \varphi \quad \text{(strongly complete)}$$