Mathematical Logics
Introduction*

Fausto Giunchiglia and Mattia Fumagallli

University of Trento

**Originally by Luciano Serafini and Chiara Ghidini*
Modified by Fausto Giunchiglia and Mattia Fumagalli

Outline

1. Course Description
2. Modeling
3. Logical Modeling
4. Languages
5. Using Logic

Course Description - People

- Fausto Giunchiglia
  website: http://disi.unitn.it/~fausto/
  email: fausto@dit.unitn.it

- Mattia Fumagalli
  email: mattia.fumagalli@unitn.it

Course Description - WebSite

http://disi.unitn.it/~ldkr/ml2017/index.html

Course Description – Mailing List Subscription

➤ Go to website home page

of logic in the effective and efficient modeling of data and knowledge. The course will have succeeded if it stimulates the interested students to continue their career with higher interest into logic-based models for data and knowledge representation in their own field of expertise, and to produce computer-processable solutions of relevant problems.

COURSE DESIGNERS

Fausto Giunchiglia

Mattia Fumagalli

Latest News

Midterm exam schedules coming soon!

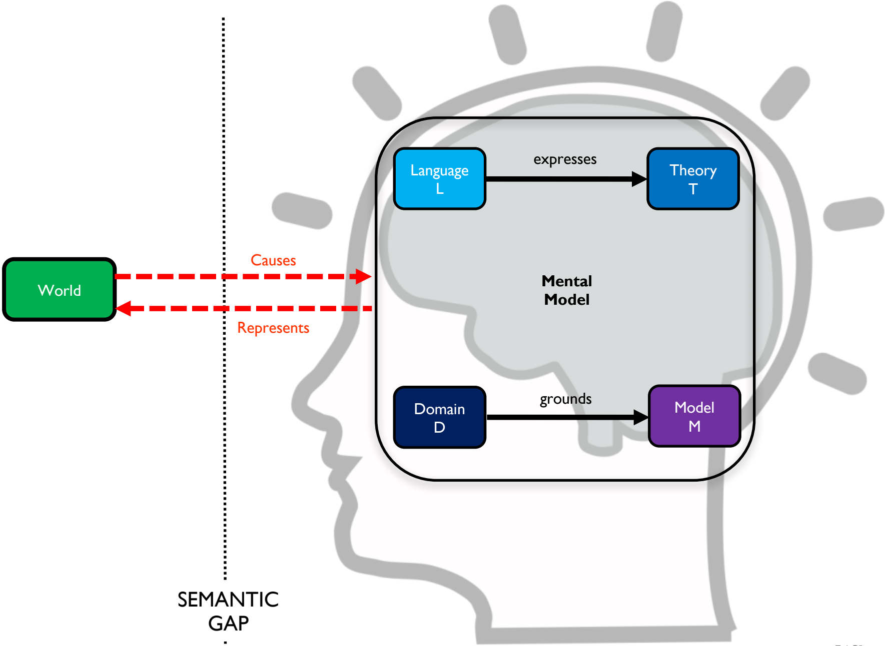
To subscribe to the mailing list:

1. Put your email here.
2. Wait to be invited

click here

Please apply with your UNITN account, otherwise the application will not be approved.

Mental Model

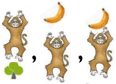


Mental Model

World | Mental Model | Language L | Domain D | Theory T | Model M

L: "*MCT, MAT, MBeBa, MNBa, MCR, MGBa, …*"*

T: "*MAT, MNBa, MGBa*"

D: {  }

M: "  "

SEMANTIC GAP

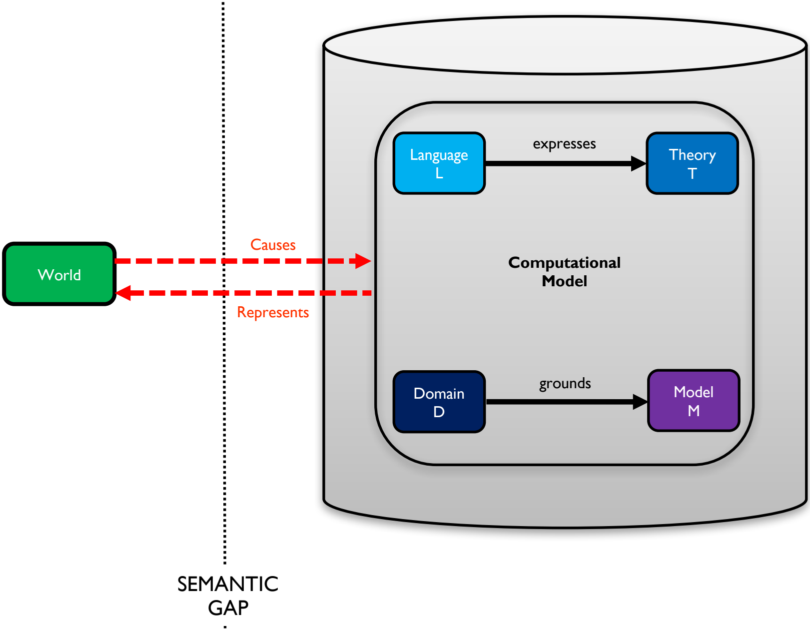
* *monkey(M), climb(C), tree(T), aboe(A), below(Be), near(N), banana(Ba), rock/R), get(G).*

Mental Model

- World: the phenomenon that we perceive and that we want to describe
- Mental Model: what we have in mind. Our representation of the world (subject to the semantic gap)
- Semantic gap: the difference between the world and our representation of the world

- Language: a set of words and rules we use to build sentences used to describe our mental model
- Theory: a set of true sentences in our mental model
- Domain: mental representations of the world used to represent what is the case in the world
- Model: the set of *true mental representations* (facts)

Computational Model



Computational Model

World | Computational Model | Language L | Domain D | Theory T | Model M

SEMANTIC GAP

MONKEY

I	A	N	G
M	T #1	Ba #2	Ba #3
…	…	…	…

L: "*MCT, MAT, MBeBa, MNBa, MCR, MGBa, …*"

T: "*MAT, MNBa, MGBa*" (abstraction of the table)

D: {#1, #2, #3} (memory location)

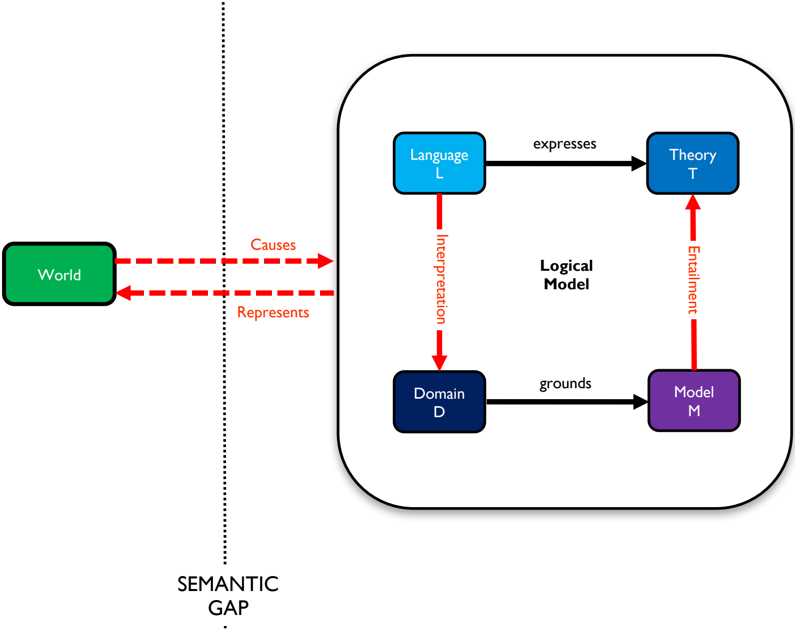
M: "*#1, #3*"

Computational Model

- World: the phenomenon that we perceive and that we want to describe
- Computational Model: an abstract model that organizes elements of data and standardizes how they relate to one another and to properties of the real world objects (or state of affairs) – (machine equivalent of the mental model)
- Semantic gap: the difference between the world and our representation of the world

- Language: a set of words and rules we use to build sentences used to describe our computational model
- Theory: a set of true sentences in our computational model
- Domain: tuples used to represent what is the case in the world
- Model: the set of true facts (e.g., as stored in a data base)

Logical Model



Logical Model

World | Logical Model | Language L | Domain D | Theory T | Model M

MONKEY			
/	A	N	G
M	T #1	Ba #2	Ba #3
…	…	…	…

SEMANTIC GAP

L = "*MCT, MAT, MBeBa, MNBa, MCR, MGBa*, ∧, ∨, ¬, →, …"

T = "*MGBa* → (*MAT* ∨ *MNBa*)"

D: {#1, #2, #3}

I: "I(MAT) = #1, I(MNBa) = #2, I(MGBa) = #3"

M: "*#1*, *#2*, *#3*"

M ⊨ *MAT*

M ⊨ *MNBa*

M ⊨ *MAT* ∨ *MNBa*"

Logical Model

- World: the phenomenon that we perceive and that we want to describe
- Logical Model: a formal model that organizes elements of data and standardizes how they relate to one another and to properties of the real world objects (or state of affairs)
- Semantic gap: the difference between the world and our representation of the world

- Language: a set of words and rules we use to build sentences used to describe our model
- Theory: a set of true sentences in our model
- Domain: tuples used to represent what is the case in the world
- Model: the set of true facts
- Interpretation: a function which associates each and any element of the language to one and only one element of the domain
- Truth-relation / logical entailment ($\models$): it connects what is true in the model with the elements of the theory. A sentence can be an element in a theory if and only if its interpretation is true in the model

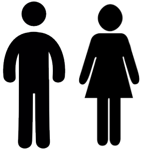
Language

- A (usually finite) set of words (elements of the alphabet) and formation rules to compose them to build "correct sentences". For instance, in logic:
  - Monkey and GetBanana are words
  - Monkey $\wedge$ GetBanana is a sentence (rule: A $\wedge$ B)
- There are many types of languages:
  - Natural languages (e.g., Italian, English, …)
  - Data languages (e.g., ER, UML, ...)
  - Programming languages (e.g., SQL, Java, C+, ...)
- All these languages have their own alphabet and formation rules

Syntax and Semantics

- **Syntax:** the way a language is written:
  - Syntax is determined by a set of rules saying how to construct the expressions of the language from the set of atomic tokens (i.e., terms, characters, symbols)
  - The set of atomic tokens is called alphabet of symbols, or simply the alphabet)
- **Semantics:** the way a language is interpreted:
  - It determines the meaning of the syntactic constructs (expressions), that is, the relationship between syntactic constructs and the elements of some universe of meanings, which may or may not be formalized.
  - Semantics are formalized in the formal models via the interpretation function

Suppose we want to represent the fact that Mary and Sara are near each other.

	ENGLISH Mary is near Tom.	informal syntax, informal semantics
	'SYMBOLIZED' ENGLISH near(M,T)	formal syntax, informal semantics
	LOGICS with an interpretation function I I (M) =  I (T) =  I(near) = ( , )	formal syntax, formal semantics

Formal Vs. Informal Languages

Language = Syntax (what we write) + Semantics (what we mean)

- Formal syntax
  - Infinite/finite (always recognizable) alphabet
  - Finite set of formal constructors and building rules for phrase construction
  - Algorithm for correctness checking (a phrase in a language)
- Formal Semantics
  - The relationship between syntactic constructs in a language L and the elements of an universe of meanings D is a (mathematical) function

$$I: L \rightarrow D$$

- Informal syntax/semantics
  - The opposite of formal, namely the absence of the elements above

NOTE: Formal semantics requires formal syntax (I is a mathematical function)

Levels of Formalization

Syntax and Semantics can be formal or informal.

NL	Diagrams	Programming Languages	**Logics**
Level$_1$	⟵	⟶	***Level$_n$***
English Italian Russian Hindi ...	ER UML ...	SQL C+ ...	PL FOL DL ...

FORMALIZATION ⟶

informal	**semi-formal**	**formal**

Let us try to recognize relevant entities, relations and properties in the NL text below

The Monkey-Bananas (MB) problem by McCarthy, 1969 *"There is a monkey in a laboratory with some bananas hanging out of reach from the ceiling. A box is available that will enable the monkey to reach the bananas if he climbs on it. The monkey and box have height Low, but if the monkey climbs onto the box he will have height High, the same as the bananas. [...]"*

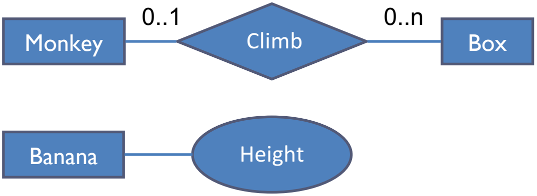
Question: How shall the monkey reach the bananas?

NOTE: Natural languages (e.g., English, Italian, ...) have informal syntax and informal semantics.

Languages with Formal Syntax

In the Entity-Relationship (ER) Model [Chen 1976] the alphabet is a set of graphical objects, that are used to construct schemas (the sentences).

Examples of ER sentences:



NOTE: Diagrams (e.g., UML, ER, EER, ...) have semi-formal syntax and semi-formal semantics.

- Logics has two fundamental components:
  - L is a formal language (in syntax and semantics)
  - I is an interpretation function which maps sentences into a formal model M (over all possible ones) with a domain D

- Domain D = {T, F} or D' = $\{o_1, \ldots, o_n\}$

- Language L = $\{A, \wedge, \vee, \neg\}$

- Interpretation I: I: $L \rightarrow D$

- Theory: a set of sentences which are always true in the language (facts)

- Model: the set of true facts in the language describing the mental model (the part of the world observed), in agreement with the theory

Usages of Logic

The two purposes in modeling:

- Specification:
  - *Representation* of the problem at the proper level of abstraction
    Allow *informal/formal* syntax and *informal/formal* semantics
- Automation (Automated Reasoning):
  - *Computing consequences* or properties of the chosen specifications.
    It requires *formal* syntax and formal semantics

Why Natural Languages?

Used for	Advantages	Disadvantages
For informal specification	Often used at the very beginning of problem solving, when we need a direct, “flexible”, well-understood language and the problem is still largely unclear Useful to interact with users	Semantics is informal, largely ambiguous Pragmatically inefficient for automation

Why Diagrams?

Used for	Advantages	Disadvantages
To provide more structured and organized specification than natural languages Informal/formal syntax (depends on the kind of diagram)	Largely structured and organized; usually used in representation with unified languages when things are non-trivial or more precision is required w.r.t. Natural Language Useful to interact with users	Semantics is informal, largely ambiguous Pragmatically inefficient for automation

Why Logic?

Used for	Advantages	Disadvantages
Formal specification Automation	Well-understood with formal syntax and formal semantics: we can better specify and prove correctness Pragmatically efficient for automation exploiting the explicitly codified semantics: reasoning services	It can be hardly used to interact with users An exponential grow in cost (computational, man power)

- Logics provides a notion of deduction
  - Axioms, deductive machinery, theorem
- Deduction can be used to implement reasoners
  - Reasoners allow inferring conclusions from known facts (i.e., a set of “premises”, premises can be axioms or theorems).
  - From implicit knowledge to explicit knowledge
- Reasoning services (examples):
  - Model Checking (EVAL) Is a sentence ψ true in model M?
  - Satisfiability (SAT) Is there a model M where ψ is true?
  - Validity (VAL) Is ψ true according to all possible models?
  - Entailment (ENT) ψ_1 true implies ψ_2 true (in all models)

Logic in Practice

- Define a logic
  - most often by reseachers
  - once for all (not a trivial task!)
- Choose the right logic for the problem
  - Given a problem the computer scientist must choose the right logic, most often one of the many available
- Write the theory
  - The computer scientist writes a theory T
- Use reasoning services
  - Computer scientists use reasoning services to solve their programs

NOTE: same process as with programming languages

A Crucial Trade-Off

There is a trade-off between:

- expressive power (expressiveness) and
- computational efficiency provided by a (logical) language

This trade-off is a measure of the tension between specification and automation

To use logic for modeling, the modeler must find the right trade off between expressiveness in the language for more tractable forms of reasoning services.

Example of Expressiveness

Language	NL Sentence	Formula
Propositional logic	Fausto likes skiing I like skiing	Fausto-likes-skiing I-like-skiing
First-order logic	Every person likes skiing I like skiing Fausto likes skiing	$\forall$ person.like-skiing(person) like-skiing(I) like-skiing(Fausto)
Modal logic	I believe I like skiing	B(I-like-skiing)
Description Logic	Every person likes cars	person $\sqsubseteq$ $\exists$ likes.Car
…		

Efficiency vs. Complexity

❑ Efficiency

- ❑ Performing in the best possible manner; satisfactory and economical to use [Webster]
- ❑ In modeling it applies to reasoning
- ❑ In time, space, consumption of resources

❑ Complexity (or computational complexity) of reasoning

- ❑ The difficulty to compute a reasoning task expressed by using a logic
- ❑ With degrees of expressiveness, we may classify the logical languages according to some "degrees of complexity"

NOTE: When logic is used we always pay a performance price and therefore we use it when it is cost-effective

Decidability

The existence of an effective method to determine the validity of formulas in a logical language

A logic is decidable if there is an effective method to determine whether arbitrary formulas are included in a theory

A decision procedure is an algorithm that, given a decision problem, terminates with the correct yes/no answer.

Most logics in this course are decidable with one exception (First Order Logic)

Appendix: Greek letters

Αα	Alpha	Νν	Nu
Ββ	Beta	Ξξ	Xi
Γγ	Gamma	Οο	Omicron
Δδ	Delta	Ππ	Pi
Εε	Epsilon	Ρρ	Rho
Ζζ	Zeta	Σσς	Sigma
Ηη	Eta	Ττ	Tau
Θθ	Theta	Υυ	Upsilon
Ιι	Iota	Φφ	Phi
Κκ	Kappa	Χχ	Chi
Λλ	Lamdba	Ψψ	Psi
Μμ	Mu	Ωω	Omega