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. Mental, computational and logical models
3. Language
4. Logical modeling
5. Why Logic? Formal and informal languages/ models

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

    $$I: L \rightarrow D$$
  - Domain D always (in this class) as a set of elements.
- Informal syntax/semantics
  - The opposite of formal, namely the absence of the elements above

Formal language = formal syntax + formal semantics

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

Formal language – example 1

Language L

- Alphabet = $\{A, B, \wedge, \vee\}$
- Syntactic constructors:
  - If A, B formulas in L then $A \wedge B$ is also a formula in L
  - If A, B formulas of L then $A \vee B$ is also a formula in L

- Domain D – in our course always set theory
  - $D = \{T, F\}$

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

- Examples:
  - Take A as “Fausto is younger than 25” then I(A) = F
  - Take B as “Fausto is a professor of AI” then I(B) = T
  - Furthermore, $I(A \wedge B) = F$, $I(A \vee B) = T$

Formal language – example 2

Language L

- Alphabet = {A, B, $\wedge$, $\vee$}
- Syntactic constructors:
  - If A, B formulas in L then A $\wedge$ B is also a formula in L
  - If A, B formulas of L then A $\vee$ B is also a formula in L

- Domain D – In our course always set theory
  - D = {Fausto, Maria, John}

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

- Examples:
  - Take A as "Male" then I(A) = {Fausto, John}
  - Take B as "Italian" then I(B) = {Fausto, Maria}
  - I(A$\wedge$B) = {Fausto}, I(A$\vee$B) = {Fausto, Maria, John}

Formal language – example 3

Language L

- Alphabet = {A, x, $\wedge$, forall}
- Syntactic constructors:
  - A(x) is a formula in L
  - If A(x) is a formula in L then forall x. A(x) is a formula in L
  - If A is a formula in L then A $\wedge$ A is also a formula in L

- Domain D – In our course always set theory
  - D = {Fausto, Maria, John}

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

- Examples:
  - Take A as "Male" then I(A) = {John, Fausto}
  - Take A as "Male" then I(A(x)) = {John} or {Fausto}
  - Take A as "Male" then forall x.A(x) is F (not an element of D)

Logic

A logic has three fundamental components

- L, a formal language - defines what can be (correctly) said
- I, an interpretation function - defines how the atomic elements of the language are to be interpreted in the intended domain and model
- ⊨, a entailment relation – used to define (compute) two relations
  - Satisfiability - when a formula is true (in the intended model)
  - Logical consequence - when the truth a formula follows from the truth of a set of formulas (in the intended model)

Given L, I, ⊨, a logic allows to define two components

- Theory - the (usually infinite) set of sentences in L which are assumed true in the intended Model, as computed via entailment starting from finite set (called itself theory, or finite presentation of theory)
- Model - the (usually infinite) set of facts expressed in the language describing the mental model (the part of the world observed), in agreement with the theory and the interpretation function,

Using logics in Practice

- Define a logic
  - most often by reseachers (almost never in real world applications)
  - Done once for all (very hard. Examples. PL, FOL, DL, ML, …)
- Choose the right logic for the problem (L, I, $\models$)
  - 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 (making sure it complies to intended mental model)
- Different uses (see above)
  - Use theory as basis for agreement developer/customer
  - Use theory to guarantee semantic interoperability (e.g., as in data integration, program composition, …)
  - Use reasoning (entailment) to make sure program does what it is supposed to do
  - Use reasoning to implement AI
  - …

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 expressiveness and automation efficiency

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

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
...		

Decidability of reasoning

- A logic is decidable if there is an effective method for determining whether a formula is included in a theory
- A decision procedure is an algorithm that, given a decision problem, terminates with the correct yes/no answer.
- All logics in this course but First Order Logic are decidable

(Computational) complexity of reasoning (given decidability)

- The difficulty to compute a reasoning task in a given logic
- The logical languages are classified according to varying "degrees of complexity" (P, NP, Pspace, ...)

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