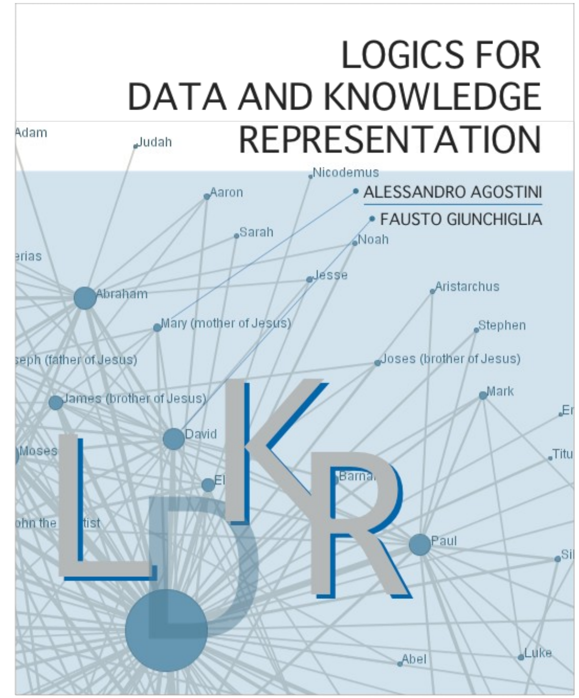


Logics for Data and Knowledge Representation

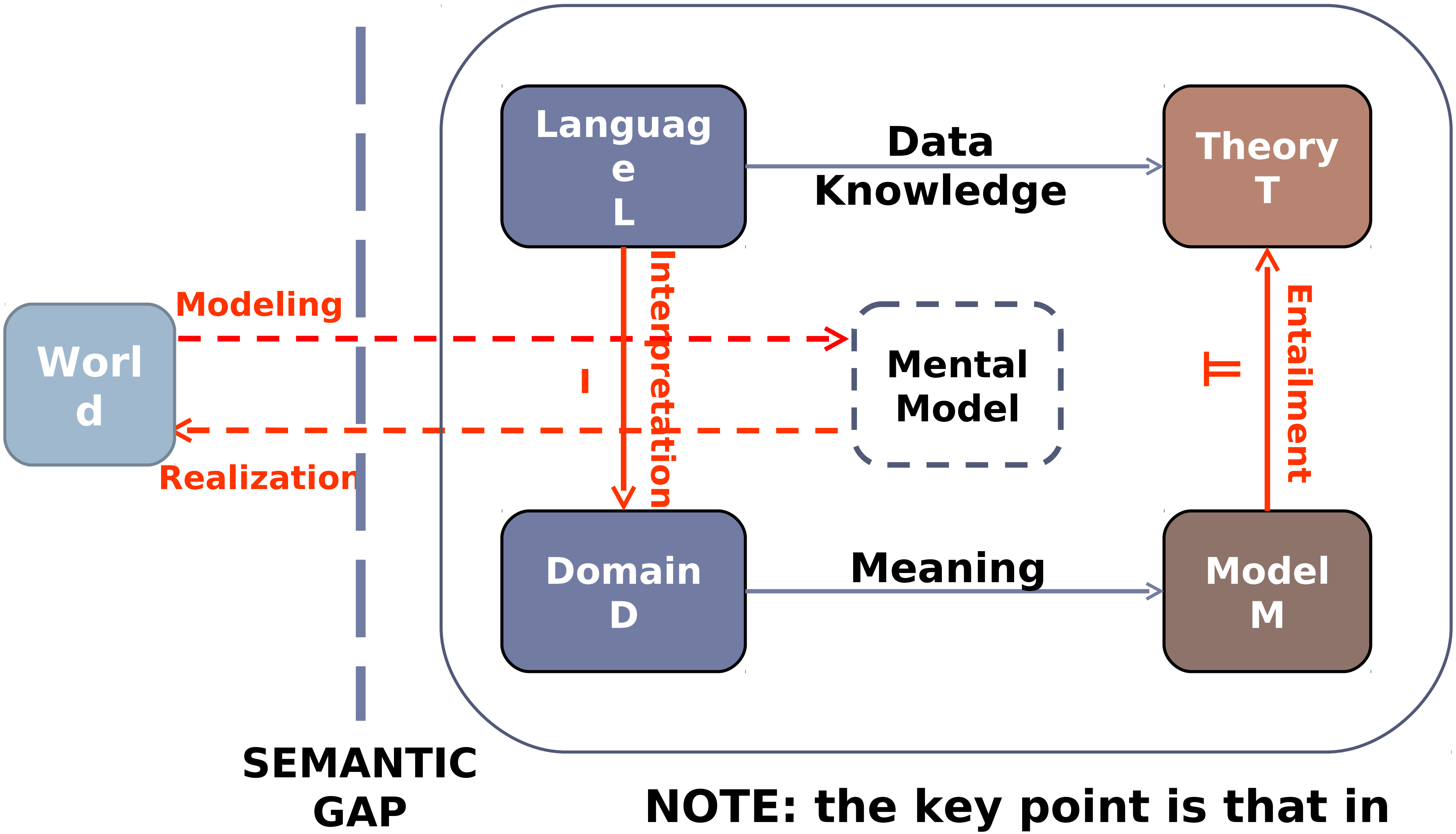
Propositional Logic

Originally by Alessandro Agostini and Fausto Giunchiglia
Modified by Fausto Giunchiglia, Rui Zhang and Vincenzo Maltese

Outline

- Syntax
- Semantics
- Entailment and logical implication
- Reasoning Services

Logical Modeling



NOTE: the key point is that in logical modeling we have **formal semantics**

Language (Syntax)

- The first step in setting up a formal language is to list the symbols of the alphabet

Σ_0

Descriptive

Constants
one proposition only
A, B, C …

Variables
they can be substituted by any proposition or formula
P, Q, ψ …

Logical
$\wedge, \vee, \neg, \ldots$

- Auxiliary symbols: parentheses: ( )
- Defined symbols:

  ⊥ (falsehood symbol, false, bottom) $\bot =_{df} P \wedge \neg P$

  T (truth symbol, true, top) $T =_{df} \neg \bot$

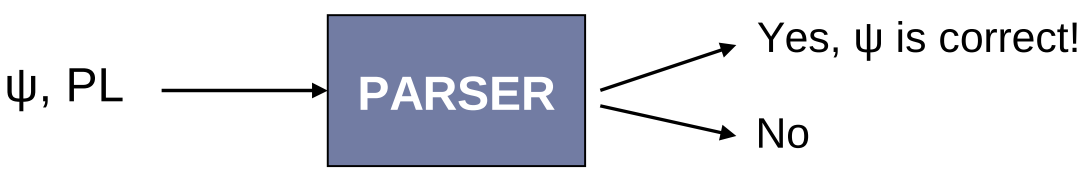
Formation Rules (FR): well formed formulas

- Well formed formulas (wff) in PL can be described by the following BNF grammar (codifying the rules):

  <Atomic Formula> ::= A | B | ... | P | Q | ... | ⊥ | ⊤

  <wff> ::= <Atomic Formula> | ¬<wff> | <wff>∧ <wff> | <wff> ∨ <wff>

- Atomic formulas are also called atomic propositions
- Wff are propositional formulas (or just propositions)
- A formula is correct if and only if it is a wff



- Σ_0 + FR define a propositional language

Propositional Theory

- Propositional (or sentential) theory
  - A set of propositions
  - It is a (propositional) knowledge base (true facts)
  - It corresponds to a TBox (terminology) only, where no meaning is specified yet: it is a syntactic notion

Semantics: formal model

- **Intensional interpretation**

  We must make sure to assign the formal meanings out of our intended interpretation to the (symbols of the) language, so that formulas (propositions) really express what we intended.

- **The mental model: What we have in mind?**

  In our mind (mental model) we have a set of properties that we associate to propositions. We need to make explicit (as much as possible) what we mean.

- **The formal model**

  This is done by defining a formal model M. Technically: we have to define a pair $(M,\models)$ for our propositional language

- **Truth-values**

  In PL a sentence A is true (false) iff A denotes a formal object which satisfies (does not satisfy) the properties of the object in the real world.

Truth-values

- Definition: a truth valuation on a propositional language L is a mapping ν assigning to each formula A of L a truth value ν(A), namely in the domain D = {T, F}

- ν(A) = T or F according to the modeler, with A atomic
- ν(¬A) = T iff ν(A) = F
- ν(A∧B) = T iff ν(A) = T and ν(B) = T
- ν(A∨B) = T iff ν(A) = T or ν(B) = T

- ν(⊥) = F (since $\bot =_{df} P \wedge \neg P$)
- ν(⊤) = T (since $\top =_{df} \neg \bot$)

Truth Relation (Satisfaction Relation)

- Let ν be a truth valuation on language L, we define the truth-relation (or satisfaction-relation) ⊨ and write

$$\nu \models A$$

(read: ν satisfies A) iff ν(A) = True

- Given a set of propositions Γ, we define

$$\nu \models \Gamma$$

iff if ν ⊨ θ for all formulas θ ∈ Γ

Model, Satisfiability, truth and validity

- Let ν be a truth valuation on language L.
  - ν is a model of a proposition P (set of propositions Γ) iff ν satisfies P (Γ).
  - P (Γ) is satisfiable if there is some (at least one) truth valuation ν such that $\nu \models P$ ($\nu \models \Gamma$).

- Let ν be a truth valuation on language L.
  - P is true under ν if $\nu \models P$
  - P is valid if $\nu \models P$ for all ν (notation: $\models P$).

    P is called a tautology

Entailment and implication

- Propositional entailment: $\Gamma \models \psi$

  where $\Gamma = \{\theta_1, \ldots, \theta_n\}$ is a finite set of propositions

  $\nu \models \theta_i$ for all θ_i in Γ implies $\nu \models \psi$

- Entailment can be seen as the logical implication

  $(\theta_1 \wedge \theta_2 \wedge \ldots \wedge \theta_n) \rightarrow \psi$

  to be read $\theta_1 \wedge \theta_2 \wedge \ldots \wedge \theta_n$ logically implies ψ

  $\rightarrow$ is a new symbol that we add to the language

Implication and equivalence

- We extend our alphabet of symbols with the following *defined* logical constants:
  → (implication)
  ↔ (double implication or equivalence)

  <Atomic Formula> ::= A | B | ... | P | Q | ... | ⊥ | ⊤

  <wff> ::= <Atomic Formula> | ¬<wff> | <wff>∧ <wff> | <wff>∨ <wff> |

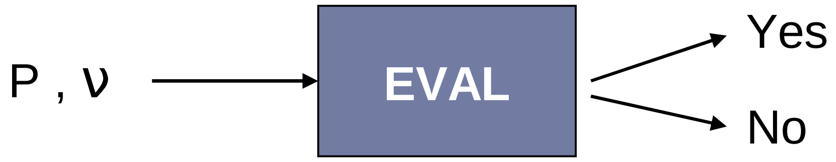
  <wff> → <wff> | <wff> ↔ <wff> **(new rules)**

- Let propositions ψ, θ, and *finite* set $\{\theta_1,\dots,\theta_n\}$ of propositions be given. We define:
  - $\models \theta \rightarrow \psi$ iff $\theta \models \psi$
  - $\models (\theta_1 \wedge \dots \wedge \theta_n) \rightarrow \psi$ iff $\{\theta_1,\dots,\theta_n\} \models \psi$
  - $\models \theta \leftrightarrow \psi$ iff $\theta \rightarrow \psi$ and $\psi \rightarrow \theta$

Reasoning Services

Model Checking (EVAL)

Is a proposition P true under a truth-valuation ν? Check $\nu \vDash P$



Satisfiability (SAT)

Is there a truth-valuation ν where P is true? find ν such that $\nu \vDash P$

Unsatisfiability (UnSAT)

the impossibility to find a truth-valuation ν

Reasoning Services

Validity (VAL)

Is P true according to all possible truth-valuation ν?

Check if $\nu \models P$ for all ν

Entailment (ENT)

All $\theta \in \Gamma$ true in ν (in all ν) implies ψ true in ν (in all ν).

check $\Gamma \models \psi$ in ν (in all ν) by checking that:

given that $\nu \models \theta$ for all $\theta \in \Gamma$ implies $\nu \models \psi$

Reasoning Services: properties

- EVAL is the easiest task. We just test one assignment.
- SAT is NP complete. We need to test in the worst case all the assignments. We stop when we find one which is true.
- UnSAT is CO-NP. We need to test in the worst case all the assignments. We stop when we find one which is true.
- VAL is CO-NP. We need to test all the assignments and verify that they are all true. We stop when we find one which is false.
- ENT is CO-NP. It can be computed using VAL (see next slide)

Using DPLL for reasoning tasks

- DPLL solves the CNFSAT-problem by searching a truth-assignment that satisfies all clauses θ_i in the input proposition $P = \theta_1 \wedge \ldots \wedge \theta_n$
- Model checking   Does ν satisfy P? ($\nu \vDash P$?)
  Check if $\nu(P) = true$
- Satisfiability   Is there any ν such that $\nu \vDash P$?
  Check that DPLL(P) succeeds and returns a ν
- Unsatisfiability   Is it true that there are no ν satisfying P?
  Check that DPLL(P) fails
- Validity   Is P a tautology? (true for all ν)
  Check that DPLL($\neg$P) fails