UNIVERSITY
OF TRENTO - Italy
Dipartimento di Ingegneria e Scienza dell'Informazione

LoP- The Logic of Propositions
Informal-to-Formal
(T2MP)

LoP – The Logic of Propositions

- **Introduction**
- NL to LoP + Modeling mistakes
- Informal EG to LoP

LoP Domain

Definition (Domain – facts).

$$D_{LoP} = < T, F >$$

Definition (Domain – percepts). Let

$$D_{LoDE} = < E, \{C\}, \{R\} >$$

be a LoDE domain of interpretation. Let $L_{LoDE} = \{a\}$ be a LoDE language for D_{LoP} where $\{a\}$ is the set of assertions in L_{LoDE}. Let $a_i \in L \subseteq L_{LoDE}$ be an assertion. Then

$$D_{LoP} = \{a^+, a^-\} = \{a_1^+, a_1^-, \ldots, a_N^+, a_N^-\} = \{T(a_1), F(a_1), \ldots, T(a_N), F(a_N)\}$$

where a^+, a^- are values of atomic propositions such that:

- $a^+ = T(a)$ = T if the LoDE assertion a is True
- $a^- = F(a)$ = T if the LoDE assertion a is False

Definition (Model). M is a set of atomic propositions $\{a^+, a^-\}$ such that, for each a, M contains one and only one between $a^+$and a^-.

$$M = \{f\} = \{a^+, a^-\} = \{\ldots, a_i^+, \ldots, a_j^-, \ldots\} \subseteq D_{LoP}.$$

Terminology (Model, atomic proposition). From now on, when no confusion arises, we talk of propositions meaning atomic propositions, the only propositions which belong to models.

Language (the same as LoE)

Definition (Assertional language)

$$L = < A , FR > = \{P\}$$

where:

- L is a **propositional language**, where $P \in \{P\}$ is a **proposition**.
- A = is an **alphabet** of atomic propositions
- FR is a **set of formation rules**

Formation rules – BNF

$< P >$ ::= <atomic proposition> |

$\neg < P >$ |

$< P > \wedge < P >$ |

$< P > \vee < P >$ |

$< P > \supset < P >$ |

$< P > \equiv < P >$ |

$< P > \oplus < P >$

<atomic proposition> ::= $P_1 \ldots P_n \in \{P\}$

LoP – The Logic of Propositions

- Introduction
- **NL to LoP + Modeling mistakes**
- Informal EG to LoP

Modeling mistakes - And (1)

We express conjunction with many words other than "and", including "but," "moreover," "however, "although", and "even though".

For example: "I enjoyed the holiday, even though it rained a lot" can be translated into the facts "I enjoyed the holiday" and "It rained a lot".

Sometimes "and" joins adjectives.

For example: "The leech was long and wet and slimy." This can be paraphrased as "The leech was long, and the leech was wet, and the leech was slimy.

Modeling mistakes - And (2)

Sometimes "and" does not join whole propositions into a compound proposition. Sometimes it simply joins nouns. This cannot be paraphrased. In these cases, the "and" is expressed inside the propositional variable, and not as logical connective.

For example: "Bert and Ernie are brothers". This cannot be paraphrased. "Bert is a brother and Ernie is a brother", for that does not assert that they are brothers to each other.

Modeling mistakes - Inclusive vs. Exclusive disjunction

The natural, but longwinded, way to express exclusive disjunction is

$(\neg p \vee q) \wedge (p \vee \neg q)$.

The way to say they have different truth values is to deny their equivalence:

$\neg (p \equiv q)$.

For example: When a menu says "cream or sugar", it uses an inclusive "or", because you may take one, the other, or both. But when it says "coffee or tea", it uses an exclusive "or", because you are not invited to take both.

Modeling mistakes - Implication

$p \supset q$ translates a wide variety of English expressions, for example, "if p, then q", "if p, q", "p implies q", "p entails q", "p therefore q", "p hence q", "q if p", "q provided p", "q follows from p", "p is the sufficient condition of q", and "q is the necessary condition of p". The least intuitive is " p only if q" (to be understood from $\neg q \supset \neg p$).

For example the following all translate to $p \supset q$:

- If Mario goes to the party, (then) I'll go too.
- I'll go to the party if/provided that Mario comes too.
- I'll go to the party only if Mario goes.
- Mario going to the party is the sufficient condition of me going to the party.
- Me going to the party is necessary condition of Mario going to the party.
- The decrease in white blood cells implies the antibiotic is working.

Modeling mistakes - Even If

"p even if q" means "p whether or not q" or "p regardless of q". Therefore one perfectly acceptable translation of it is simply "p". If you want to spell out the claim of "regardlessness", then you could write "$p \wedge (q \vee \neg q)$".

For example:

- I'll go to the party even if Mario doesn't go.
- I'll go to the party whether or not Mario goes.
- I'll go to the party regardless of whether Mario comes or not

Modeling mistakes - Unless

Sometimes "unless" should be translated as inclusive disjunction (∨), and sometimes as exclusive disjunction (⊕).

For example (inclusive disjunction): "I'll go to the party unless I get another offer" means that I'll go if nothing else comes along, namely an exclusive disjunction. In many contexts it also means that I might go anyway; the second offer might be worse. So I'll go or I'll get another offer or both.

For example (exclusive disjunction): Consider by contrast, "I'll go to the party unless Rufus is there". In many contexts this means that if I learn Rufus is going, then I'll change my mind and not go. So either I'll go or Rufus will go but not both.

Modeling mistakes - Necessary and Sufficient conditions

We say that p is a *sufficient condition* of q when p's truth guarantees q's truth. By contrast, q is a *necessary condition* of p when q's falsehood guarantees p's falsehood.

In the ordinary material implication, $p \supset q$, the antecedent p is a *sufficient condition* of the consequent q, and the consequent q is a *necessary condition* of the antecedent p.

Notice that $p \supset q$ if and only if $\neg q \supset \neg p$.

For example: "If Socks is a cat, then Socks is a mammal". Being a cat is a sufficient condition of being a mammal. Being a mammal is a necessary condition of being a cat."

Exercise 1 - NL to LoP

Assuming A = "Angelo comes to the Party",B = "Bruno comes to the Party",
C = "Carlo comes to the Party", D = "David comes to the Party",

David or Bruno come to the party	$D \vee B$
Bruno will come to the party	B
Bruno will come to the party, unless David is there	$D \oplus B$
Carlo comes to the party, therefore David comes too	$C \supset D$
Either David or Bruno come to the party	$(D \wedge \neg B) \vee (B \wedge \neg D)$
Neither Carlo nor David will come to the party	$\neg C \wedge \neg D$

Exercise 2 - NL to LoP

Assuming A = “Angelo comes to the Party”, B = “Bruno comes to the Party”, C = “Carlo comes to the Party”, D = “David comes to the Party”,

“If David comes to the party then Bruno and Carlo come too” $D \supset (B \wedge C)$

“Angelo will come to the party, provided that Bruno comes and Carlo does not” $(B \wedge \neg C) \supset A$

“Carlo comes to the party given that David doesn't come, but, if David comes, then Bruno doesn't come” $(\neg D \supset C) \wedge (D \supset \neg B)$

“Carlo comes to the party only in case Angelo and Bruno do not come” $(\neg A \wedge \neg B) \equiv C$

“A sufficient condition for Angelo coming to the party, is that Bruno and Carlo aren't coming” $(\neg B \wedge \neg C) \supset A$

“A necessary and sufficient condition for Angelo coming to the party, is that Bruno and Carlo aren't coming” $(\neg B \wedge \neg C) \equiv A$

LoP – The Logic of Propositions

- Introduction
- NL to LoP + Modeling mistakes
- **Informal EG to LoP**

Tell – Model building

Intuition (Model building). The model building is performed in three steps

- **(Step 1):** Define the LoP reference model, that is, the set of LoDE assertions which describe the facts which are true in the model
- **(Step 2):** Define the LoP language, that is, the set of atomic propositions and logical connectives which are used to judge what is true / false in the model
- **(Step 3):** Define the LoP theory, that is, the set of (atomic and complex) propositions which constrain what is the case in the model by:
  - (1) specifying the negative knowledge,
  - (2) completing the partial information encoded by the model, and
  - (3) putting further constraints on what is the case via complex propositions.

Tell – Model building (step 1)

Intuition (Define the LoP Reference Model). The first step is articulated in five phases:

- **(Phase 1a)** Define the set of LoE assertions of the EG
- **(Phase 1b)** Define the set of LoD language definitions
- **(Phase 1c)** Define the set of LoD knowledge descriptions
- **(Phase 1d)** Perform the LoD unfolding
- **(Phase 1e)** Perform the LoDe expansion

Observation (Define the LoP reference model). Any of the first three steps is optional. Step 1d and Step 1e are performed only when needed. The key observation is that LoP propositions can be built by espressing judgements on all three LoDE components: ground facts about entities, facts about defined etypes, facts about language concepts.

Tell – Model building (step 2)

Intuition (Define the LoP Language). The second step is articulated in three phases:

- **(Phase 2a)** Select which LoDE assertions are going to be judged
- **(Phase 2b)** Select a uniform method for encoding a LoDE assertion a into a LoP assertion a'^{+}, a'^{-}. This in turn is composed of two steps
  - (1) How to encode a structured formula into an atomic formula, e.g., from *HasFriend(Stefania#1,Paolo#1)* to *HF-S.P*
  - (2) which of the possible positive or negative encodings a'^{+}, a'^{-} select and how to encode them in the proposition name, e.g., from *HF-S.P to HF-S.P0* and *HF-S.P1*
- **(Phase 2c)** Select the logical connectives, not necessarily used to write complex propositions

Intuition (Phase 2b). There is a std encoding which performs a 1-to-1 mapping (see later).

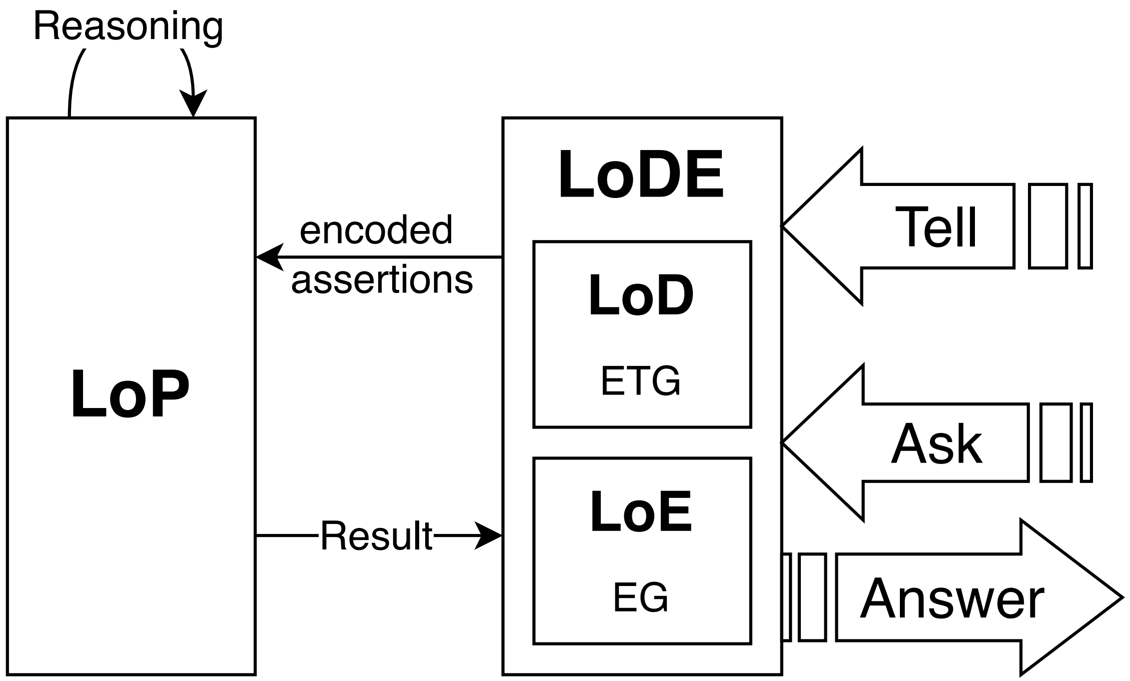
Tell – Model building (step 3)

Intuition (Define the LoP Theory). The second step is articulated in three phases:

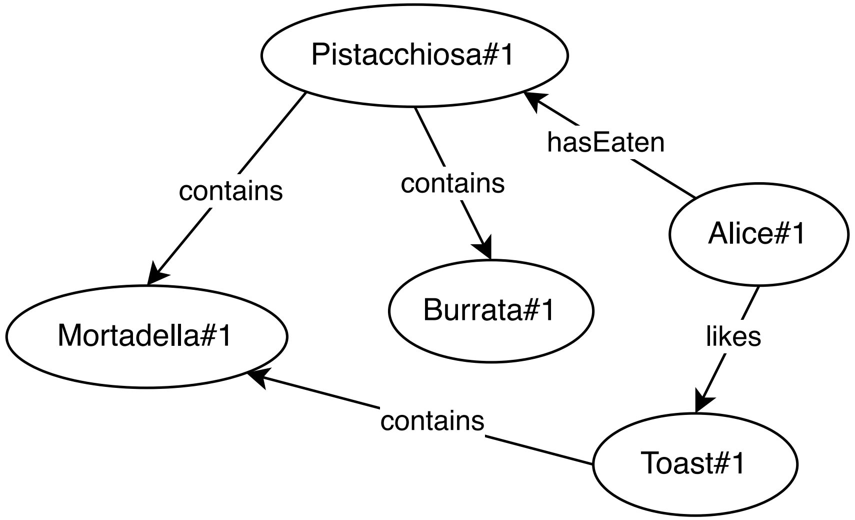
- **(Phase 3a)** Select the LoDE assertions which are going to be judged. This usually turns out to be a set of atomic or conjunctions of atomic propositions
- **(Phase 3b)** Select the negative knowledge, implicitly encoded in the LoDE theory, to be made explicit in the LoP theory. This usually turns out to be a set of negations, or disjointness or implication axioms.
- **(Phase 3c)** Select the partial knowledge, implicitly encoded in the LoDE theory, to be made explicit in the LoP theory. This usually turns out to be a set of disjunction axioms.

Observation (Define the LoP theory). Usually, not all the implicit negative and partial knowledge of a LoD theory is made explicit in a LoP theory, in particular when it takes, implicitly or explictly, the form of disjunctions. The reason being that the complexity of reasoning grows exponentially with the number of disjunctions.

Using LoP


Reasoning
LoP
encoded
assertions
Result
LoDE
LoD
ETG
LoE
EG
Tell
Ask
Answer

Exercise 3 - From Informal EG to LoP



- "Pizza and Sandwich are two distinct types of Dish"
- "Ingredients are not a Dish"
- "Pistacchiosa#1 contains only one between Mortadella#1 and Burrata#1"
- "If Alice#1 likes Toast#1 then she likes Mortadella#1"
- "If Alice#1 likes Mortadella#1 then Pistacchiosa#1 contains Mortadella#1 and if Alice#1 likes Burrata#1 then Pistacchiosa#1 contains Burrata#1"
- "Alice has not eaten Toast#1"

Q1: Is Burrata a dish?
Q2: Has Alice eaten Toast#1?
Q3: Does Alice#1 likes Burrata#1?

Exercise 3 - From Informal EG to LoP

Inferred LoE assertions

Person(Alice#1)

Pizza(Pistacchiosa#1)

Sandwich(Toast#1)

Ingredient(Mortadella#1)

Ingredient(Burrata#1)

contains(Toast#1, Mortadella#1)

contains(Pistacchiosa#1, Mortadella#1)

hasEaten(Alice#1, Pistacchiosa#1)

likes(Alice#1, Toast#1)

contains(Pistacchiosa#1, Burrata#1)

Inferred LoD assertions

Pizza ⊑ Dish

Sandwich ⊑ Dish

Pizza ⊥ Sandwich

Ingredient ⊥ Dish

Expanded assertions

Dish(Pistacchiosa#1)

Dish(Toast#1)

¬Dish(Mortadella#1)

¬Dish(Burrata#1)

¬Sandwich(Pistacchiosa#1)

¬Pizza(Toast#1)

Q1: Is Burrata a dish? -> No
Q2: Has Alice eaten Toast#1? -> Not known
Q3: Does Alice#1 likes Burrata#1? -> Not known

Exercise 3 - From Informal EG to LoP

Encoding of the selected assertions

Person(Alice#1)	PrsA	Dish(Pistacchiosa#1)	DshP
Pizza(Pistacchiosa#1)	PzzP	Dish(Toast#1)	DshT
Sandwich(Toast#1)	SndT	¬Dish(Mortadella#1)	¬ DshM
Ingredient(Mortadella#1)	IngM	¬Dish(Burrata#1)	¬ DshB
Ingredient(Burrata#1)	IngB	¬Sandwich(Pistacchiosa#1)	¬ SndP
contains(Toast#1, Mortadella#1)	Tc.M	¬Pizza(Toast#1)	¬ PzzT
contains(Pistacchiosa#1, Mortadella#1)	Pc.M		
hasEaten(Alice#1, Pistacchiosa#1)	Ahe.P		
likes(Alice#1, Toast#1)	Alk.T		
contains(Pistacchiosa#1, Burrata#1)	Pc.B		

Exercise 3 - From Informal EG to LoP

New assertions inferred by prior knowledge

- "If Alice#1 likes Toast#1 then she likes Mortadella#1"

  - likes(Alice#1, Mortadella#1) AlkM

- "If Alice#1 likes Mortadella#1 then Pistacchiosa#1 contains Mortadella#1 and if Alice#1 likes Burrata#1 then Pistacchiosa#1 contains Burrata#1"

  - likes(Alice#1, Burrata#1) AlkB

- "Alice has not eaten Toast#1"

  - hasEaten(Alice#1, Toast#1) Ahe.T

Exercise 3 - From Informal EG to LoP

Adding negative knowledge and constraints from prior knowledge

- "Ingredients are not a Dish"
  $\neg$ DshB
- "Pistacchiosa#1 contains only one between Mortadella#1 and Burrata#1"
  Pc.M $\oplus$ Pc.B
- "If Alice#1 likes Toast#1 then she likes Mortadella#1"
  Alk.T $\rightarrow$ Alk.M
- "If Alice#1 likes Mortadella#1 then Pistacchiosa#1 contains Mortadella#1 and if Alice likes Burrata then Pistacchiosa contains Burrata"
  (Alk.M $\rightarrow$ Pc.M) $\wedge$ (Alk.B $\rightarrow$ Pc.B)
- "Alice has not eaten Toast#1"
  $\neg$Ahe.T

Exercise 3 - From Informal EG to LoP

Building the LoP theory

$\neg$ DshB
$\neg$Ahe.T
Alk.T
Pc.M $\oplus$ Pc.B
Alk.T $\rightarrow$ Alk.M
Alk.M $\rightarrow$ Pc.M
Alk.B $\rightarrow$ Pc.B

Q1: Is Burrata a dish? No

Q2: Has Alice eaten Toast#1? No

Q3: Does Alice#1 likes Burrata#1? No

Exercise 3 - From Informal EG to LoP

Alk.T	Pc.M ⊕ Pc.B	Alk.T → Alk.M	Alk.M → Pc.M	Alk.B → Pc.B	Pc.M	Pc.B	Alk.M	Alk.B
T	T	T	T	T	T	F	T	F

Given Alk.T and Alk.T → Alk.M, we know that Alk.M is true, and given Alk.M → Pc.M we know that also Pc.M is true. Because we know Pc.M ⊕ Pc.B, then we know that Pc.B is false, and with Alk.B → Pc.B we know that also Alk.B is false.

LoP- The Logic of Propositions Informal-to-Formal (T2MP)