CEID

MSc on DATA DRIVEN COMPUTING AND DECISION MAKING (DDCDM)

KNOWLEDGE REPRESENTATION IN LOGIC

I. HATZILYGEROUDIS, PROFESSOR EMERITUS

LOGIC AND LOGICAL REASONING

The following knowledge is given

George works in Patras, while Pavlos in Aigio. George is Pluto's master and Paul is Bobby's. All live where they work. Everyone who has a master lives where his master lives.

We are asked to decide on

1. Whether Pluto lives in Patra
2. Who live in Aigio

The above require logical reasoning:

1. *Since Pluto has George as his master and George lives in Patra, so Pluto also lives in Patra.*
2. *Obviously Paul. But Bobby also lives there, since his master is Pavlos who lives in Aigio*.

LOGIC AND LOGICAL REASONING

The question is how this can be done automatically on the computer:

A language is needed to **represent the sentences** in the text **(First Order Logic):**

works-in(george, patras)
works-in(paul, rio)
master(george, pluto)
master(paul, boby)
(∀x)(∀y)(works-in(x, y) ⇒ lives-in(x, y))
(∀x)(∀y)(∀z)((master(x, y) ∧ lives-in(x, z)) ⇒ lives-in(y, z))

} Logical Formulas

and a **mechanism** for handling logical propositions and **making inferences.**

LOGIC AS KNOWLEDGE REPRESENATION LANGUAGE (1)

BASIC ELEMENTS OF LOGIC

- **Syntax**
- **Semantics or Model theory**
- **Proof theory**

LOGIC AS KNOWLEDGE REPRESENATION LANGUAGE (2)

MODEL THEORY

(Defining the meaning of sentences/formulas)

Interpretation: $I = < D , f_I >$

D : Set of primitive entities

f_I : Interpretation function

symbol $\xleftrightarrow{f_I}$ entity(ies)

Model of a sentence/formula: $I \models \varphi$ (φ is true based on I)

I <u>satisfies</u> φ or I is a <u>model</u> of φ

Model of a set of formulas S:

I model of S iff $\forall \varphi \in S$, I model of φ

LOGIC AS KNOWLEDGE REPRESENATION LANGUAGE (3)

Satisfiable formula:

iff ∃ I : I model of φ (I |= φ)

Satisfiable or consistent set of formulas:

iff ∃ I : ∀φ ∈ S, I model of φ

LOGIC AS KNOWLEDGE REPRESENATION LANGUAGE (4)

Logical implication

From a formula: **φ1 |= φ2** iff ∀I : I |= φ1 => I |= φ2

properties: reflexive, transistive

From a set of formulas: **S |= φ’** iff ∀φ ∈ S, ∀I: I |= φ => I |= φ’

Alternative terminology

valid consequence
logical consequence
semantic consequence

Logical equivalence

φ1 ≡ φ2 iff φ1 |= φ2 and φ2 |= φ1

LOGIC AS KNOWLEDGE REPRESENATION LANGUAGE (5)

PROOF THEORY
(Formula production)

Derivable formula: S |- φ
if φ ∈ S or is the result of application of an IR (inference rule) to formulas in S or derivable from S

(An inference rule takes one or more formulas and produces a new one)

Formulas in S → Premises or Axioms

Derivable from S → Conclusions or Theorems

LOGIC AS KNOWLEDGE REPRESENATION LANGUAGE (6)

Proof of formula φ from S

A sequence of formulas where φ is the last one and every other formula is either from S or has been derived from S.

Alternative terminology:

Deduction

Derivation

LOGIC AS KNOWLEDGE REPRESENATION LANGUAGE (7)

Sound process-Sound IRs

If every formula that can be derived from S is logically deducted from S:

$$S \vdash \varphi \quad \Rightarrow S \models \varphi$$

(It prevents the production of incorrect solutions)

Complete process-Complete IRs

If every formula that is logically deducted from S can be derived from S:

$$S \models \varphi \quad \Rightarrow S \vdash \varphi$$

(Prevents skipping solutions)

FIRST ORDER LOGIC-FOL (1)

SYNTAX

VOCABULARY

- Constants $\{C_i\}$: Each C_i represents an element of D.
- Logical constants : {T, F}
- Variables $\{v_i\}$: Each v_i represents a subset of D.
- Predicates $\{P_i^n\}$: $D^n \rightarrow \{T, F\}$
- Functions $\{f_i^n\}$: $D^n \rightarrow D$

FIRST ORDER LOGIC-FOL (2)

- Logical connectives: $\neg$ (not), $\vee$ (or), $\wedge$ (and), $\Rightarrow$ (implies), $\Leftrightarrow$ (equivalent)
- Quantifiers: $\forall$ (universal)

  $\exists$ (existential)

FIRST ORDER LOGIC-FOL (3)

SYNTAX RULES

- Atomic expression or atom: $P^n (t_1, t_2, \ldots, t_n)$
- Term:
  i. constant, ii. variable,
  iii. $f^n (t_1, t_2, \ldots, t_n)$, where t_i term.
- WFF (Well Formed Formulas):
  1. atom
  2. $\neg F$, $(F \vee G)$, $(F \wedge G)$, $(F \Rightarrow G)$, $(F \Leftrightarrow G)$
  where F, G WFFs
  3. $(\forall x) F$, $(\exists x) F$, where x free variable and F WFF.

FIRST ORDER LOGIC-FOL (4)

EXAMPLES OF WFF FORMULAS

$(\forall x)\ (\exists y)\ GREATER(x, y)$
$(\forall x)\ ((Q(x) \wedge P(y)) \Rightarrow R(x)$
$(\forall x)\ (P(x) \Rightarrow (\exists y)\ Q(x, y))$
$(\forall x)\ (\neg\ (\exists y)\ on(x, y) \Rightarrow clear(x))$

Scope of quantifier
- The expression it is applied to
- Everything at its right

Open formula
- Includes free variables

Closed formula
- Does not include free variables (wffs are closed)

FIRST ORDER LOGIC-FOL (5)

NORMAL FORMS OF WWFs

Conjunctive Normal Form-CNF

$(F \vee G) \wedge (\neg F \vee G) \wedge \ldots.$

Disjunctive Normal Form-DNF

$(F \wedge G) \vee (\neg F \wedge G) \vee \ldots.$

Prenex Normal Form-PNF

$(Q_1\, x_1)\, (Q_2\, x_2) \ldots (Q_n\, x_n)\, (F)$

FIRST ORDER LOGIC-FOL (6)

SEMANTICS

Interpretation function

1. $f_I(c_i) = d_i \in D$

2. $f_I(v_i) = \{d_1, d_2, \ldots, d_n\} \subseteq D$

3. $f_I(f_i^{\,n}) = \{ \langle\langle d_{11}, d_{12}, \ldots, d_{1n} \rangle\; d_1 \rangle,$

$\langle\langle d_{21}, d_{22}, \ldots, d_{2n} \rangle\; d_2 \rangle,$

$\ldots$

$\langle\langle d_{m1}, d_{m2}, \ldots, d_{mn} \rangle\; d_m \rangle\}$

where $\forall j, \quad (d_{j1}, d_{j2}, \ldots, d_{jn}) = d_j \in D \;\; (D^n \rightarrow D)$

FIRST ORDER LOGIC-FOL (7)

4. $f_I(P_i^n) = \{ < d_{11} , d_{12} , \ldots, d_{1n} >,$

$< d_{21} , d_{22} , \ldots, d_{2n} >,$

$\ldots$

$< d_{m1} , d_{m2} , \ldots, d_{mn} >\}$

where $\forall j, < d_{j1} , d_{j2} , \ldots, d_{jn} > \subseteq D \quad (D^n \rightarrow \{T, F\})$

FIRST ORDER LOGIC-FOL (8)

Semantics rules

1. If $\varphi \equiv P^n (t_1, t_2, \ldots, t_n)$ then

   $I \models \varphi$ iff $< t_1 , t_2 , \ldots, t_n > \in f_I(P^n)$

2. If $\varphi \equiv \neg F$ then $I \models \varphi$ iff $I \not\models F$

3. If $\varphi \equiv (F \vee G)$ then $I \models \varphi$ iff $I \models F$ ή $I \models G$

4. If $\varphi \equiv (F \wedge G)$ then $I \models \varphi$ iff $I \models F$ και $I \models G$

5. If $\varphi \equiv (F \Rightarrow G)$ then $I \models \varphi$ iff $I \not\models F$ ή $I \models G$

6. If $\varphi \equiv (\forall x)$ F then $I \models \varphi$ iff $\forall x \in D \Rightarrow I \models F$

7. If $\varphi \equiv (\exists x)$ F then $I \models \varphi$ iff $\exists x \in D \Rightarrow I \models F$

FIRST ORDER LOGIC-FOL (9)

Example

A simple FOL langauge:
* Three constants: a, b, c
* Predicates : P^1, Q^1, R^2

<u>An interpretation</u>

D = {mary, john, george}

f: f(a) = mary
f(b) = john
f(c) = george
f(P) = {mary}
f(Q) = {john, george}
f(R) = {< mary, john >, < john, mary >}

<u>Predicate notions</u>
P $\equiv$ woman, Q $\equiv$ man
R $\equiv$ has-married

FIRST ORDER LOGIC-FOL (10)

INTERPRETATION OF FORMULAS

1. P(a) → T
2. R(a, b) → T
3. P(c) $\Rightarrow$ R(b, c) → T
4. ($\exists$x) P(x) → T
5. ($\forall$x) ($\forall$y) (P(x) $\wedge$ Q(y)) $\Rightarrow$ R(y, x) → F

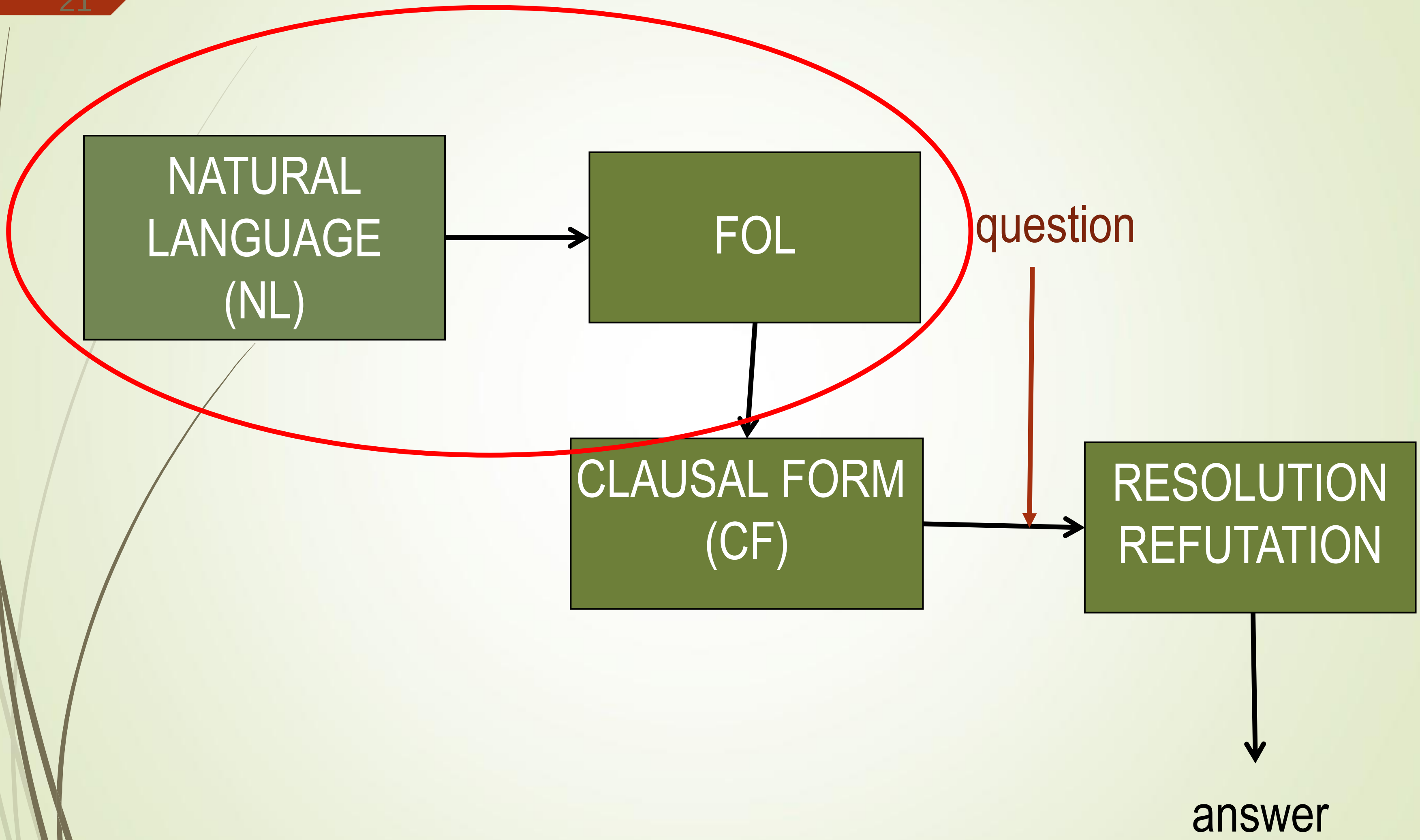
CHANGE OF INTERPRETATION FUNCTION

f(b) = george, f(c) = john

Then

2. R(a, b) → F

LOGIC AND AUTOMATED REASONING



CONVERSION FROM NL TO FOL (1)

Hints-Process:

1. Identify predicates (verbs, nouns, adjectival determiners)
2. Identify functions (verbs or predicates that indicate a 1-1 relationship)
3. Specify arguments of predicates or functions (number, type: constant, variable, function type, and symbols)
4. Specify quantifiers of the variables
5. Formation of atomic expressions (atoms)
6. Combination of atomic expressions-formula formation

CONVERSION FROM NL TO FOL (2)

EXAMPLES

«All humans are mortal»

$(\forall x)$ human(x) $\Rightarrow$ mortal(x)

«All live where they work»

$(\forall x)$ $(\exists y)$ εργάζεται(x, y) $\Rightarrow$ μένει(x, y)

$(\forall x)$ μένει(x, τόπος_εργασίας_του(x))

Order of quantifiers matters:

$(\forall x)$ $(\exists y)$ μένει(x, y) means different from

$(\exists y)$ $(\forall x)$ μένει(x, y)

ΜΕΤΑΤΡΟΠΗ ΦΓ ΣΕ ΚΛΠΤ (2)

Wrong use of $\wedge$

«All sports funs like soccer»

$(\forall x)$ sports-fun(x) $\wedge$ likes(x, soccer)

$(\forall x)$ sports-fun(x) $\Rightarrow$ likes(x, soccer)

Equivalence: $A \Rightarrow (B \Rightarrow C) \equiv (A \wedge B) \Rightarrow C$

"All humans eat some food"

$(\forall x)\ (\exists y)$ (human(x)$\wedge$ food(y) $\Rightarrow$ eats(x, y))

$(\forall x)\ (\exists y)$ (human(x) $\Rightarrow$ (food(y) $\Rightarrow$ eats(x, y)))

USE OF EQUALITY

- The use of the '=' operator is sometimes necessary to express a NL sentence into FOL. This operator is usually used with infix notation: x=y.
- There are special ways and rules for handling equality and corresponding logics for deriving/proving propositions.
- However, it can also be handled within the framework of FOL by simply adding some axioms regarding equality:
  - $\forall x\ (x=x)$
  - $\forall x \forall y\ (x=y \Rightarrow y=x)$
  - $\forall x \forall y \forall z\ (x=y \wedge y=z \Rightarrow x=z)$

USE OF EQUALITY

"Every student loves some student"

$\forall x \, (\text{student}(x) \Rightarrow \exists y \, (\text{student}(y) \wedge \text{loves}(x,y)))$

"Every student loves some other student"

$\forall x \, (\text{student}(x) \Rightarrow \exists y \, (\text{student}(y) \wedge \neg (x = y) \wedge \text{loves}(x,y)))$.