INTRODUCTION TO LOGICAL LANGUAGES (II)

LANGUAGE, LOGIC, AND COMPUTATION

Yanjing Wang, Department of Philosophy

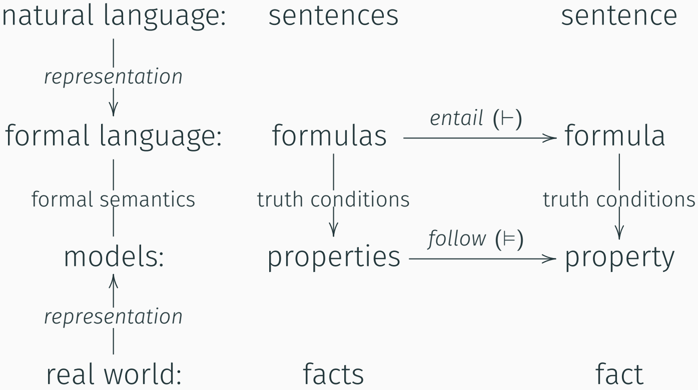
Nov. 6th, 2017

Recap

First-order logic

Recap

THE GENERAL PICTURE



PROPOSITIONAL LOGIC AND MODAL LOGIC: SYNTAX

Language of propositional logic:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \to \varphi)$$

Language of (propositional) modal logic:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \to \varphi) \mid \Box\varphi \mid \Diamond\varphi$$

where $p \in \mathsf{PV}$ (a given set of proposition letters)

First-order logic

FIRST-ORDER LANGUAGES

A *signature* (符号表) gives the *non-logical* symbols:

- variables (变元) $x, y \dots \in Var$
- constants (常元) $c, d \dots \in Con$
- predicate symbols (谓词符号) $P, Q \dots$ each has an arity
- function symbols (函数符号) $f, g \dots$ each has an arity

Given this signature the first-order language is defined as:

$$\begin{array}{lcl} \text{term (项) } t & ::= & x \mid c \mid f(\vec{t}) \quad \text{where } x \in Var, c \in Con \\ \text{formula } \varphi & ::= & t \equiv t \mid P\vec{t} \mid \neg\varphi \mid (\varphi \to \varphi) \mid \forall x \varphi \end{array}$$

We can define $\exists x$ as $\neg\forall x\neg$.

SYLLOGISMS CAN BE EXPRESSED

- All A are B: $\forall x(A(x) \to B(x))$
- Some A is B: $\exists x(A(x) \land B(x))$
- No A is B : $\forall x(A(x) \to \neg B(x))$
- Some A is not B: $\exists x(A(x) \land \neg B(x))$

What about swapping $\to$ and $\land$ in the above formulas:

- $\forall x(A(x) \land B(x))$
- $\exists x(A(x) \to B(x))$
- $\forall x(A(x) \land \neg B(x))$
- $\exists x(A(x) \to \neg B(x))$

MUCH MORE EXPRESSIVE: AN EXAMPLE LANGUAGE

FOL	symbols	intuition	NL
constants	a, b, c	names	Alice, Bill, Charles
1-place predicates	H, R	property	happy, rich
2-place predicates	L, O	relation	love, older than
unary function	f, m	function	father of, mother of

- Alice is rich but not happy: $R(a) \wedge \neg H(a)$
- Alice loves Bill but Bill doesn't love Alice: $L(a, b) \wedge \neg L(b, a)$.
- Bill's mother is older than Charles' father: $O(m(b), f(c))$.
- Not all the rich men are happy: $\neg\forall x(R(x) \to H(x))$,
- Alice is the one who loves Bill: $L(a, b) \wedge \forall x(L(x, b) \to x \equiv a)$
- Every happy person loves someone: $\forall x(H(x) \to \exists y(L(x, y)))$

UNDERSTANDING THE ABSTRACTION OF FO-LANGUAGE

A predicate is an *incomplete* expression.

non-logical symbols	NL constructions (roughly)
1-place predicates	be happy, hit the wall, cry, be a student ...
2-place predicates	love, be older than, see, graduate from
3-place predicates	give, tell, ...introduce...to..., ...kill...by ...
4-place predicates	...meet...on...at...

The quantifiers may not be explicit in NL sentences:

- A man is an animal: $\forall x(M(x) \to A(x))$
- Logicians who are insane are good philosophers: $\forall x(L(x) \land I(x) \to G(x))$
- John did not see a person: $\forall x(P(x) \to \neg S(j, x))$

WHAT DO THESE FORMULAS SAY?

- $\exists x(i \equiv x) \vee \neg\exists x(i \equiv x)$
- $Think(i) \to \exists x(i \equiv x)$
- $\forall x(Country(x) \to \neg\exists y \forall t(Future(t) \to Friend(x, y, t)))$
- $\forall x \forall y(Family(x) \wedge Family(y) \to (((H(x) \wedge H(y) \to Sim(x, y) \wedge (\neg H(x) \wedge \neg H(y) \to \neg Sim(x, y))$

See Hamkins' collection.

SCOPE, FREE AND BOUND VARIABLES

In $\forall x\varphi$, φ is in the scope (辖域) of $\forall x$. In a formula, a variable may occur *free* or *bound*. An occurrence of a variable x is free in a formula if it is not in the scope of any quantifier of x, otherwise it is bound.

For example, in $\forall xP(x, f(y)) \wedge \exists yQ(x, y)$ the second occurrence of x and the first occurrence of y are free. We often say x is a *free variable* in φ if there is some free occurrence of x in φ.

Intuitively, the above formula can also be "relettered" into $\forall zP(z, f(y)) \wedge \exists uQ(x, u)$ (约束变元易字).

- open formulas (开公式): there is some free variable.
- closed formulas or sentences (闭公式): no variable is free.

DONKEY SENTENCE

Every farmer who owns a donkey beats it.

- $\forall x\,(\text{FARMER}(x) \wedge \exists y\,(\text{DONKEY}(y) \wedge \text{OWNS}(x,y)) \rightarrow \text{BEAT}(x,y))$?
- $\forall x\,\exists y\,(\text{FARMER}(x) \wedge \text{DONKEY}(y) \wedge \text{OWNS}(x,y) \rightarrow \text{BEAT}(x,y))$?
- $\forall x\,\forall y\,((\text{FARMER}(x) \wedge \text{DONKEY}(y) \wedge \text{OWNS}(x,y)) \rightarrow \text{BEAT}(x,y))$

FO-ARITHMETIC LANGUAGE WITH SIGNATURE $(0, s, +, \times)$:

- 0 is a constant symbol,
- S is a *successor function* symbol, to represnet natural numbers, e.g., $S0$ denotes 1.
- $+$ is a binary function symbol for “addition”, $\times$ for “multiplication”

AXIOMS OF PEANO ARITHMETIC

What can we say? (universal quantification $\forall x$, $\forall y$ omitted)

- 0 is not a successor of any number: $0 \not\equiv Sx$
- If the successors of x, y are the same then x and y are the same: $Sx \equiv Sy \to x \equiv y$
- Definition of addition: $0 + x \equiv x$, $x + Sy \equiv S(x + y)$
- Definition of multiplication: $0 \times x \equiv 0$, $x \times Sy \equiv (x \times y) + x$.
- Axiom schema for induction: $(\varphi(0) \wedge (\forall x(\varphi(x) \to \varphi(Sx)))) \to \forall x \varphi(x)$

Defining natural numbers by their properties.

FIRST-ORDER STRUCTURES AND MODELS

Given a first-order language, a structure is $\mathfrak{A} = \langle D, I \rangle$:

- D is an non-empty set (the *domain*, also denoted as $|\mathfrak{A}|$)
- I is an interpretation function for non-logical symbols:
    - to each constant c gives an element in D ($c^{\mathfrak{A}}$)
    - to each n-ary P gives an n-ary relation over D ($P^{\mathfrak{A}}$)
    - to each n-ary f gives an n-ary function over D ($f^{\mathfrak{A}}$)

A model $\mathfrak{M}$ is a structure plus an assignment $\langle \mathfrak{A}, \sigma \rangle$, where the assignment σ tells which element in D each x refers to.

SEMANTICS

Given a model $\mathfrak{M} = \langle \mathfrak{A}, \sigma \rangle$, we know the meaning of each variable, and σ can be extended to all the terms:

- $\sigma(c) = c^{\mathfrak{A}}$
- $\sigma(f(t_1 \cdots t_n)) = f^{\mathfrak{A}}(\sigma(t_1), \dots \sigma(t_n))$.

The semantics is based on models $\mathfrak{M} = \langle \mathfrak{A}, \sigma \rangle$:

$$
\begin{array}{rcl}
\mathfrak{A}, \sigma \vDash t \equiv t' & \Leftrightarrow & \sigma(t) = \sigma(t') \\
\mathfrak{A}, \sigma \vDash P(t_1 \cdots t_n) & \Leftrightarrow & (\sigma(t_1), \cdots, \sigma(t_n)) \in P^{\mathfrak{A}} \\
\mathfrak{A}, \sigma \vDash \neg\varphi & \Leftrightarrow & \mathfrak{A}, \sigma \nvDash \varphi \\
\mathfrak{A}, \sigma \vDash \varphi \to \psi & \Leftrightarrow & \text{If } \mathfrak{A}, \sigma \vDash \varphi \text{ then } \mathfrak{A}, \sigma \vDash \psi \\
\mathfrak{A}, \sigma \vDash \forall x \varphi & \Leftrightarrow & \text{for all } a \in |\mathfrak{A}| \ : \mathfrak{A}, \sigma[x \mapsto a] \vDash \varphi
\end{array}
$$

$\sigma[x \mapsto a]$ changes the assignment of x to a but keep everything else as in σ.

EXAMPLE MODEL

Let $D = \{Alice, Bill, Charles\}$ and I be defined as follows:

non-logical symbols	Interpretation
constant a, b, c	Alice, Bill, Charles
one-place predicate H, R	$\{Charles, Bill\}, \{Alice, Charles\}$
two-place predicate L	$\{\langle Alice, Bill\rangle, \langle Charles, Alice\rangle, \langle Bill, Bill\rangle\}$

Check the truth value of following formulas:

- $L(a, b) \wedge \neg L(b, a)$.
- $\neg\forall x(R(x) \to H(x))$, equiv. $\exists x(R(x) \wedge \neg H(x))$.
- $L(a, b) \wedge \forall x(L(x, b) \to x \equiv a)$ (false)
- $\forall x(H(x) \to \exists y(L(x, y)))$ (equivalent to $\forall x \exists y(H(x) \to L(x, y))$?)

A SOUND AND COMPLETE HILBERT PROOF SYSTEM

- Axioms and rules for propositional logic
- $\varphi \to \forall x\varphi$ (x is not in φ)
- $\forall x\varphi \to \varphi[t/x]$ ($[t/x]$ is "admissible")
- $\forall x(\varphi \to \psi) \to (\forall x\varphi \to \forall x\psi)$
- Axioms for $\equiv$:
    - $t \equiv t$
    - $\vec{t} \equiv \vec{s} \to f(\vec{t}) \equiv f(\vec{s})$ where $\vec{t} = t_1, \ldots, t_n$.
    - $\vec{t} \equiv \vec{s} \to (P\vec{t} \to P\vec{x})$
- From $\vdash \varphi$ infer $\vdash \forall x\varphi$.

First-order theory: first-order logic+extra axioms, e.g., axioms of Peano arithmetic.

AXIOMATIC SET THEORY ZEMELO-FRANKLE + AXIOM OF CHOICE

Consider the FO language with a single binary predicate $\in$: .

Below is the foundation of almost all the mathematics.

Extensionality	$\forall x \forall y (x \equiv y \leftrightarrow \forall z (z \in x \leftrightarrow z \in y))$
Paring	$\forall x \forall y \exists z \forall w (w \in z \leftrightarrow w \equiv x \vee w \equiv y)$
Union	$\forall x \exists y \forall z (z \in y \leftrightarrow \exists w (w \in x \wedge z \in w))$
Empty set	$\exists x \forall y (\neg y \in x)$
Infinity	$\exists x (\emptyset \in x \wedge \forall y (y \in X \rightarrow y \cup \{y\} \in x))$
Power set	$\forall x \exists y \forall z (z \in y \leftrightarrow \forall w (w \in z \rightarrow w \in x))$
Replacement	$\forall x \in y \exists ! z R(x, z) \rightarrow \exists w \forall v (v \in w \leftrightarrow \exists x (x \in y \wedge R(x, v))$
Regularity	$\forall x (\neg x \equiv \emptyset \rightarrow \exists y \in x (x \cap y \equiv \emptyset))$
Choice	$\forall x ((\neg \emptyset \in x \wedge \forall y, z \in x (\neg y \equiv z \rightarrow y \cap z \equiv \emptyset))$
	$\rightarrow \exists y \forall z \in x \exists ! w \in z (z \in y)))$

$\exists !$ (there exists one and only one...), $\cup$, $\cap$, $\emptyset$, $\{y\}$ can be defined.

DEFINITION OF NATURAL NUMBERS IN SETS

John von Neumann's definition:

- $0 := \emptyset$
- $S(X) = X \cup \{X\}$

We can then build all the natural numbers in the following fashion:

- $1 := \emptyset \cup \{\emptyset\} = \{\emptyset\}$
- $2 := 1 \cup \{1\} = \{\emptyset\} \cup \{\{\emptyset\}\} = \{\emptyset, \{\emptyset\}\} = \{0, 1\}$
- $3 := 2 \cup \{2\} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} = \{0, 1, 2\}$
- ...

WE WANT TO REASON AUTOMATICALLY! HERE IS AN EXAMPLE

周迅的前男友窦鹏是窦唯的堂弟；窦唯是王菲的前老公；周迅的前男友宋宁是高原的表弟；高原是窦唯的前任老婆；周迅的前男友李亚鹏是王菲的现任老公；周迅的前男友朴树的音乐制作人是张亚东；张亚东是王菲的前老公窦唯的妹妹窦颖的前老公，也是王菲的音乐制作人；张亚东是李亚鹏前女友瞿颖的现男友。请问下列说法不正确的是:

- 王菲周迅是情敌关系;
- 瞿颖王菲是情敌关系;
- 窦颖周迅是情敌关系;
- 瞿颖周迅是情敌关系.

BTW 怎么写周迅的某前男友是王菲的某前老公的堂弟?

APPLICATIONS

- Knowledge representation and reasoning.
- Automate mathematical proofs.

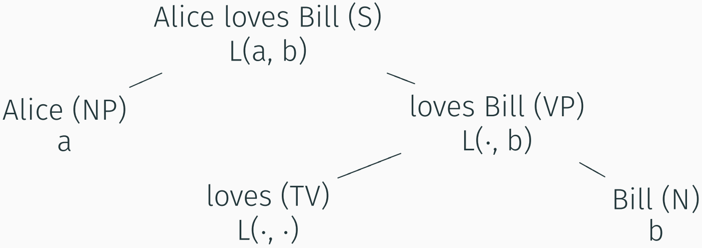
HOW TO REASON AUTOMATICALLY?

- Is there a finitary mechanic method (algorithm) to tell whether an arbitrary first-order formula is valid?
- *Entscheidungsproblem* proposed by Hilbert.
- What is this finitary method?
- Computation models
    - Turing machine by Turing.
    - λ-calculus by Church.
    - Church-Turing thesis.

(Un)fortunately, the answer is negative: there is no such a method to decide whether a FO-formula is valid in general.

BACK TO NL: TROUBLES WITH FIRST-ORDER LANGUAGE

- How to represent the parts of a natural language sentence?
- How to go from meaning of lexicon and the syntactic structure step by step to the logical form?



NEXT

λ-calculus: build the logical form by functions and functional applications.

LOJBAN

A more "natural" language inspired by logic Lojban:
`https://mw.lojban.org/papri/Lojban`

To test the (in)famous Sapir-Whorf hypothesis: language determines/influence the thought.