The pragmatics in FOL

It specifies how well-formed expressions are to be used.

To be able to reason, concepts like "Dog", "Democratic Country" should be given an intended interpretation.

Logical consequence

Although the interpretation of the nonlogical symbols defines the semantic interpretation in FOL, still there are connections between sentences that do not depend on the meaning of those symbols.

Example: α, β sentences in FOL

let $\gamma = \neg(\beta \wedge \neg\alpha)$

if $\mathfrak{I}$ is an interpretation where α is true then γ is true under $\mathfrak{I}$ and its truth value does not depend on how we understand the nonlogical symbols in α and β.

We say that α logically entails γ, or γ is a logical consequence of α.

For a set of sentences S and a sentence α, we say that α is a logical consequence of S (or S logically entails α) iff for every interpretation $\mathfrak{I}$ with $\mathfrak{I} \models S$ then $\mathfrak{I} \models \alpha$.

Or equivalently, there is no interpretation $\mathfrak{I}$ where $\mathfrak{I} \models S \cup \{\neg\alpha\}$.

We write $S \models \alpha$

A sentence α is logically valid, which we write $\models \alpha$, if it is a logical consequence of the empty set (i.e. α is valid iff $\forall \mathfrak{I}$, $\mathfrak{I} \models \alpha$).

if $S = \{\alpha_1, \dots, \alpha_n\}$ finite and α is a sentence, then

$$S \models \alpha \text{ iff } [(\alpha_1 \wedge \dots \wedge \alpha_n) \supset \alpha] \text{ is valid.}$$

The logical entailment is the key of a knowledge-based system.

For example, if Fido is a dog then a reasoning system should be able to conclude that Fido is a mammal.

if a set of sentences S entails a sentence α, then α is true in every interpretation where S is true. Other sentences that are not entailed by S may or may not be true, but a Knowledge-based system must conclude that the entailed sentences are true.

if we have an interpretation $\mathfrak{J}$ where Dog(fido) is true, then the system can conclude that $\neg\neg$Dog(fido) and (Dog(fido) $\vee$ Happy(john)) are true. These conclusions are logically safe but this is not the kind of reasoning we would be interested in.
Something more useful would be if a system concludes from Dog(fido) that Mammal(fido).
We can find an interpretation where Dog(fido) is true and Mammal(fido) is false.

For example, $\mathfrak{J} = \langle \Delta, I \rangle$

$$\Delta = \{d\}$$
$$I[Dog] = \{d\}$$
$$I[P] = \{\ \} \quad \text{for every other predicate } P \neq Dog$$
$$I[f](d, \dots, d) = d$$

We have $\mathfrak{J} \models Dog(fido)$ but $\mathfrak{J} \not\models Mammal(fido)$.
So, there is no logical connection between the two sentences. To create it, we need to include in S a statement that connects the nonlogical symbols involved:

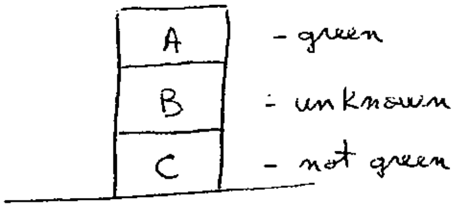
$$\forall x.\, Dog(x) \supset Mammal(x)$$

Thus, Mammal(x) becomes the logical consequence of Dog(x) and we rule out all the interpretations where the set of dogs is not included in the set of mammals.

"Truth in the intended interpretation"

Reasoning based on logical consequence allows only safe, logically guaranteed conclusions in a knowledge-based system.

Exercise



Is there a green block directly on top of a nongreen one?

Formalization in FOL

a, b, c the names of the blocks

G unary predicate symbol for "green"

O binary predicate symbol for "on"

$$S = \{ O(a,b),\ O(b,c),\ G(a),\ \neg G(c) \}$$

the sentence α is $\exists x \exists y .\ G(x) \wedge \neg G(y) \wedge O(x,y)$

we want to prove that $S \models \alpha$

Let $\mathfrak{I}$ a logical model for S $\quad \mathfrak{I} \models S$

1. Suppose that $\left.\begin{array}{l} \mathfrak{I} \models G(b) \\ \neg G(c),\ O(b,c) \in S \end{array}\right\} \Rightarrow \mathfrak{I} \models G(b) \wedge \neg G(c) \wedge O(b,c$

$\Rightarrow \mathfrak{I} \models \exists x \exists y .\ G(x) \wedge \neg G(y) \wedge O(x,y)$

2. Suppose that $\left.\begin{array}{l} \mathfrak{I} \models \neg G(b) \\ G(a),\ O(a,b) \in S \end{array}\right\} \Rightarrow \mathfrak{I} \models G(a) \wedge \neg G(b) \wedge O(a,$

$\Rightarrow \mathfrak{I} \models \exists x \exists y .\ G(x) \wedge \neg G(y) \wedge O(x,y)$

Thus, α is a logical consequence of S.

There is no automated procedure in FOL to decide in all cases whether a sentence is entailed or not from others.

A reasoning process is
- <u>logically sound</u> if whenever it produces α, then α is guaranteed to be a logical consequence
- <u>logically complete</u> if it is guaranteed to produce α whenever α is entailed

The barber's paradox (formulated by Bertrand Russell)

- Anyone who does not shave himself must be shaved by the barber
- Whomever the barber shaves, must not shave himself

Show that no barber can fulfill these requirements.

$$\forall x.\ Person(x) \wedge (\neg Shave(x,x) \supset Shave(barber, x))$$

$$\forall x.\ Person(x) \wedge (Shave(barber, x) \supset \neg Shave(x,x))$$

$$\mathfrak{I} \models \forall x.\ Person(x) \wedge (\neg Shave(x,x) \supset Shave(barber, x))$$

$$\mathfrak{I} \models Person(barber) \wedge (\neg Shave(barber, barber) \supset Shave(barber, barber))$$

$$\mathfrak{I} \models \neg Shave(barber, barber) \supset Shave(barber, barber)$$

$$\mathfrak{I} \models Shave(barber, barber) \supset \neg Shave(barber, barber)$$

$$\alpha \supset \beta \text{ is } \neg\alpha \vee \beta$$

$$\mathfrak{I} \models Shave(barber, barber)$$

$$\mathfrak{I} \models \neg Shave(barber, barber)$$

Expressing knowledge – creating a knowledge-base

Knowledge engineering – is the first step when creating a knowledge base – and it means deciding on the representatio language followed by determining the kinds of objects important to the agent, the properties those objects have and the relationships among them.

Vocabulary

We start by identifying the essential entities in the agent's world:

- constant symbols
  - persons: johnSmith
  - institutions: government
  - places: centralStation
- description of the basic types of objects:
  - Person(x), Country(x), Restaurant(x)
- the set of attributes of objects:
  - Rich, Nice, Smart
- express relationships:
  - DaughterOf (ana, mary)
  - MarriedTo (ana, ion)
- functions:
  - bestFriendOf (john)
  - firstChildOf (ana, john)

Basic Facts

They are represented by atomic sentences and negations of atomic sentences ($P(t_1, \dots, t_n)$ and $t_1 = t_2$)

Man(john), Rich(mary), WorksFor (john, george)

$\neg$ HappilyMarried(john)

bestFriendOf(john) = george

$y \neq$ ana

Complex Facts

Connectors are used to express various beliefs

$$\forall y\ [\ Rich(y) \wedge Man(y) \supset Loves(y, mary)]$$

$$\forall y\ [\ Woman(y) \wedge y \neq jane \supset Loves(y, john)]$$

We can express general facts

$$\forall x\ \forall y\ [\ Loves(x, y) \supset \neg\ Blackmails(x, y)]$$

or incomplete knowledge

$$Loves(jane, john) \vee Loves(jane, jim)$$

$$\exists x\ [\ Adult(x) \wedge Blackmails(x, john)]$$

Relationships among predicates

if john is Man then Women(john) should be false.

if MarriedTo(ana, john) is true then MarriedTo(john, ana) should be true.

But a KB does not generate by itself such inferences. We need to provide a set of facts about the terminology we are using.

- Disjointness - the assertion of one implies the negation of the other

$$\forall x\ [\ Man(x) \supset \neg\ Woman(x)]$$

- Subtypes - $\forall x\ [\ Surgeon(x) \supset Doctor(x)]$

- Exhaustiveness - two or more subtypes completely account for a supertype

$$\forall x\ [\ Adult(x) \supset (Man(x) \vee Women(x))]$$

- Symmetry - $\forall x, y\ [\ MarriedTo(x, y) \supset MarriedTo(y, x)]$

- Type restrictions - defining the meaning of a predicate requires arguments of certain types

$$\forall x, y\ [\ MarriedTo(x, y) \supset Person(x) \wedge Person(y)]$$

- Full definitions: predicates that are completely defined by a logical combination of other predicates

$$\forall x \, [\, RichMan(x) \equiv Rich(x) \wedge Man(x) \,]$$

Entailments

It means to derive implicit conclusions from explicit knowledge in KB.

Example 1

$$KB \begin{cases} 1. & Rich(john) \\ 2. & Man(john) \\ 3. & \forall y \, [\, Rich(y) \wedge Man(y) \supset Loves(y, jane) \,] \\ 4. & john = ceoOf(insuranceCompany) \\ 5. & Company(insuranceCompany) \end{cases}$$

Question: is there a company whose CEO loves Jane?

in FOL $\quad \exists x \, [\, Company(x) \wedge Loves(ceoOf(x), jane) \,]$

Let $\mathcal{I}$ an interpretation that is a logical model for KB.

$$\left.\begin{array}{l} \mathcal{I} \text{ satisfies } 1, 2, 3 \text{ from KB} \Rightarrow \mathcal{I} \models Loves(john, jane) \\ \qquad\qquad\qquad\qquad\qquad\qquad john = ceoOf(insuranceCompany) \end{array}\right\} \Rightarrow$$

$$\left.\begin{array}{l} \mathcal{I} \models Loves(ceoOf(insuranceCompany), jane) \\ \qquad\qquad Company(insuranceCompany) \end{array}\right\} \Rightarrow$$

$$\mathcal{I} \models Company(insuranceCompany) \wedge Loves(ceoOf(insuranceCompany), jane)$$

$$\Rightarrow \mathcal{I} \models \exists x \, [\, Company(x) \wedge Loves(ceoOf(x), jane) \,] \qquad \square$$

<u>Obs</u>. $KB \models (\alpha \supset \beta)$ iff $\forall \mathcal{I}$, $\mathcal{I} \models KB$ then $\mathcal{I} \models \neg\alpha \vee \beta$

a) if $\mathcal{I} \models \neg\alpha$ then $\mathcal{I} \models \neg\alpha \vee \beta$

b) if $\mathcal{I} \models \alpha$ then $\mathcal{I} \models (\alpha \supset \beta)$ iff $\mathcal{I} \models \beta$

so if $\mathcal{I} \models \alpha$ then $KB \models (\alpha \supset \beta)$ iff $KB \cup \{\alpha\} \models \beta$

Example 2

$$\text{KB}\left[\begin{array}{l} 1.\ \exists x\,[\,Adult(x) \wedge Blackmails(x, john)\,] \\ 2.\ \forall x\,[\,Adult(x) \supset (Man(x) \vee Women(x))\,] \\ 3\ \ Loves(john, jane) \\ 4.\ \forall y\,[\,(Women(y) \wedge y \neq jane) \supset Loves(y, john)\,] \\ 5.\ \forall x \forall y\,[\,Loves(x,y) \supset \neg Blackmails(x,y)\,] \end{array}\right.$$

Question: If no man blackmails John, then is he blackmailed by someone he loves?

in FOL $\forall x\,[\,Man(x) \supset \neg Blackmails(x, john)\,] \supset$

$$\exists y\,[\,Loves(john, y) \wedge Blackmails(y, john)\,]$$

Let $\mathfrak{I} \models KB$ and $\mathfrak{I} \models \forall x\,[\,Man(x) \supset \neg Blackmails(x, john)\,]$

We want to show that $\mathfrak{I} \models \exists y\,[\,Loves(john, y) \wedge Blackmails(y, john)\,]$

$$\left.\begin{array}{l} 1.\ \exists x\,[\,Adult(x) \wedge Blackmails(x, john)\,] \\ 2.\ \forall x\,[\,Adult(x) \supset (Man(x) \vee Women(x))\,] \\ \quad \forall x\,[\,Man(x) \supset \neg Blackmails(x, john)\,] \end{array}\right\} \Longrightarrow$$

6. $\mathfrak{I} \models \exists x\,[\,Women(x) \wedge Blackmails(x, john)\,]$

$$\left.\begin{array}{l} 4.\ \forall y\,[\,(Women(y) \wedge y \neq jane) \supset Loves(y, john)\,] \\ 5.\ \forall x \forall y\,[\,Loves(x,y) \supset \neg Blackmails(x,y)\,] \end{array}\right\} \Longrightarrow$$

7. $\mathfrak{I} \models \forall y\,[\,(Women(y) \wedge y \neq jane) \supset \neg Blackmails(y, john)$

$$\left.\begin{array}{l} 6, 7 \Longrightarrow \mathfrak{I} \models Blackmails(jane, john) \\ \qquad\quad 3.\ \mathfrak{I} \models Loves(john, jane) \end{array}\right\} \Longrightarrow$$

$\mathfrak{I} \models Loves(john, jane) \wedge Blackmails(jane, john) \Longrightarrow$

$\mathfrak{I} \models \exists y\,[\,Loves(john, y) \wedge Blackmails(y, john)\,]$

$\square$

Abstract individuals

In FOL we represent facts in a domain, but we can get a greater flexibility in representation if we map objects onto predicates and functions.

Reification means to transform a sentence into an object, by creating new abstract individuals.

For instance, we can say that John purchases a bike:

Purchases (john, bike)

Purchases (john, bike, oct11)

Purchases (john, bike, oct11, 1000RON) ...

The arity of "Purchases" depends on the level of the details that we want to express.

We will consider "purchase" to be an abstract individual called, for instance, p17. We can now describe this purchase using predicates and functions:

$$Purchase(p17) \wedge agent(p17) = john \wedge object(p17) = bike \wedge$$
$$amount(p17) = 1000RON \wedge time(p17) = 16$$

Instead of $time(p17) = 16$ we can say $time(p17) = t19 \wedge$
$hour(t19) = 16 \wedge minute(t19) = 23$

The advantage now is that the arities of the predicate and function symbols are determined in advance.

Other types of facts

- Statistical and probabilistic facts

  Half of the companies are profitable.

  Most of the students work.

  There are 10% chances that tomorrow will be sunny.

- Default and prototypical facts – usually true, unless stated otherwise

  Cars have four wheels.
  Companies do not allow employees that work together to be married.

- Intentional facts

  John believes that Henry blackmails him.
  Jane does not want Jim to know that she loves him.

Exercise

KB:
- Tony, Mike and John belong to the Alpine Club.
- Every member of the Alpine Club who is not a skier is a mountain climber.
- Mountain climbers do not like rain and anyone who does not like snow is not a skier.
- Mike dislikes whatever Tony likes and likes whatever Tony dislikes.
- Tony likes rain and snow

a) Represent the above sentences in FOL, using a consistent vocabulary (which you must define)

b) Prove that the sentences logically entail that there is a member of the Alpine Club who is a mountain climber but not a skier.