Planning & Reasoning

Lorenzi Flavio
Mat. 1662963

History of First Order Logic

(intended meaning of functions and the initial absense of equality)

La Sapienza, Rome
2020

Contents

1.Introduction

The major goals of Artificial Intelligence are to produce computer programs that perform at high levels of competence in cognitive tasks and to understand and develop computational models of human intelligence. All these can be achieved with the help of knowledge representation and reasoning. Almost all Knowledge Representation (KR) languages are based, in some way, on *"formal logic"* like propositional logic, predicate logic and temporal logic, among others.

The Logic is the systematic study of valid rules of inferences, i.e. the relations that lead to the acceptance of one proposition (conclusion) on the basis of a set of other propositions (premises). It is a part of logic programming where it is used to represent knowledge while the inference to manipulate it.

Formal logic shares with English (more general, the spoken language) the advantage that it can express facts about the world. It shares with computer programming languages the advantage of being clear and never ambiguous.
The logic used to represent knowledge in logic programming is said *"clausal form"* and it is a subset of the First Order – predicate – Logic (**FOL**).
It is one of the most used powerful tool for knowledge representation and reasoning, that unlike Propositional Logic which deals with simple declarative propositions, it additionally covers predicates and quantifications, allowing reasoning about properties that are shared by many objects through the use of variables and also allowing more flexible and compact representation of the knowledge. So it is an extention of the Propositional Logic, considering whether things are true or false in a partial view of the world, called *domain*; in this way it is always well understood and able to represent all computational problems.

Note that for all this to happen, a KR system has to provide a fact management service. So the job of a KR system is to manage a knowledge base (KB) in some representation language (including both syntax and semantics) and to answer questions about what semantically follows from the current KB.

In this work we will see how and why the FOL was made for, going into details of its history up to deepen the positive and the negative aspects too, focusing through many examples on its theoretical part, explaining in detail the functions and their meaning.

In the end a brief experimental section is developed in which it is handled a very simple FOL solver, that given a certain number of queries (input), it returns as output the validity (True or False) of each sentence. This has been tried to work with something more practical, trying to transform some theoretical concept into concrete.

2. Crossing FOL history

Nowdays, first-order logic can seem an entirely natural object of study, and its discovery inevitable. It is semantically complete and it is adequate to the *"axiomatization of all ordinary mathematics"*. So it is not surprising that the FOL has long been considered as the best logic for investigations into the foundations of mathematics. Today it occupies the central place in modern textbooks of mathematical logic, with all other systems relegated to the sidelines.

Its history, however, is not straightforward, and is certainly not a matter of a sudden discovery by a single researcher. Its arising is bound up with technical discoveries, with differing conceptions of what constitutes logic, with different programs of mathematical research, and with philosophical and conceptual reflection. So the story is intricate and often contested by many mathematicians, philosophers and logicians influencing common thinking: thus for several decades history becomes a branching structure, with numerous researchers working in different traditions and only partially aware of one another's accomplishments.; this chapter will provide an overview of this.

2.1 Nineteenth century and first revolutionary ideas

The modern study of logic is commonly dated to **1847**, with the appearance of *George Boole*'s work *"Mathematical Analysis of Logic".* This work established that Aristotle's syllogistic logic can be translated into an algebraic calculus, whose symbols Boole interpreted as referring either to classes or to propositions. His system encompasses what is today called sentential (or *Boolean*) logic, but it is also capable of expressing rudimentary quantifications. For instance a proposition of the form *"All X are Y"* is represented by $xy = x$ with the multiplication being thought of either as an intersection of sets or logical conjunction. Boole's system, in modern terms, can be viewed as a *fragment of monadic first-order logic*. It is first-order because its notational resources cannot express a quantification that ranges over predicates. It is monadic (a predicate that can only refer to a single individual or to a single variable) because it has no notation for n-ary relations. And it is a fragment because it cannot express nested quantifications (e.g. "*for every man, there exists a woman who loves him*"). But all these are modern categories, not Boole's: his logical system has no symbols corresponding to the quantifiers; so even to call it a restricted system of quantificational logic is anachronistic and inexact.

So the next two principal augmentations to Boole's system that produced a recognizably modern logic were (1) the introduction, in addition to one-placed predicates (e.g. *"x is mortal"*), of many-placed relations *(*e.g. *"x is the brother of y"*; *"x lies between y and z"*); and (2) the introduction of a notation for universal and existential quantification.
Mainly three logicians working in the Boolean tradition carried out these steps, *Augustus De Morgan*, *Charles A. Pierce* and (more indipendently) *Gottlob Frege*.

In **1864** *A. De Morgan* pointed out that the Aristotelian syllogistic was incapable of handling such inferences as, *"If every man is an animal, then every head of a man is a head of an animal"*. He introduced a logic of relations, defined the converse and the contrary of a relation, and for relations like *"X is a lover of Y"* and *"Z is a servant of W"*, had explored such compositions of relations as *"X is the lover of a servant of Y"*. This work successfully expanded the Aristotelian syllogistic logic, but was also limited in several ways, operating only with binary relations and with clumsy notation (no separate sign for negation, nor for the Boolean propositional connectives).

So *C. A. Pierce* noticed these shortcomings, and in **1870** showed how to extend Boole's logic to cover the whole realm of formal logic, instead of being restricted to that simplest and least useful part of the subject, the logic of absolute terms which for Boole was the only formal logic known.
He studied the composition of relations with each other and with class-terms, and worked out the principal laws for the resulting abstract algebraic system, ultimately showing that the linear associative algebras studied by his father (Benjamin Peirce) could all be defined in terms of what he called *elementary relatives*. His 1870 system, although a large advance both on Boole and on De Morgan, remains notationally awkward and in retrospect it is clear that it needed the theory of quantification. But it was the first successful attempt to extend Boole's system into the logic of relations.
In his later writings (**1880-1885**), he proved to be far ahead of his time and to have a futuristic vision in many ways developing some insights and tricks; for instance he introduced a modern notation for what he was the first to call the **quantifier**. He viewed his quantifiers (for which he used the symbols Π and Σ). As a generalization of the Boolean connectives, with the universal quantifier Π being interpreted as a (possibly infinite) conjunction, so that Π P(x) is understood as *"a is P and b is P and c is P and …"*. Similarly the existential quantifier Σ is understood as a (possibly infinite) sum: *"a is P or b is P or c is P or …"*.
These flexible notations allowed him easily to express nested quantifications to any desired depth. Thus, in his notation, if l_{ij} represents *"i is a lover of j"*, ΣiΣj tells us that somebody loves somebody, whereas ΠiΣj tells us that everybody loves somebody.
(Note that the Σ and Π notations are of course intended, in a Boolean spirit, to emphasize the analogy to arithmetical sums and products).

In the same handful of years (**1879-1885**), a large and independent contribution came from the studies of *G. Frege*. They have their root in the work on the foundations of real analysis by such German mathematicians as *Riemann* and *Weierstrass*. From this tradition Frege took the idea of providing a rigorous foundation for mathematics and the central mathematical concepts of function and variable, who he employed in place of the Aristotelian concepts of predicate and subject. This latter step led him naturally to a logic of relations (since the functions considered in mathematics were multivariate); and his analysis of mathematical inference also led him to introduce a notation for quantificational logic.

Frege's logical system had several advantages over Peirce. His axiomatic presentation of a purely syntactic calculus was considerably more precise, and his analysis of the number concept went deeper. His system permitted both variables and functions to be quantified. This was a central component of his program for providing a logical foundation for arithmetic, since, in his logical system, identity, cardinal number, and mathematical induction were all defined via higher-order quantifications.
He distinguished between concepts of different order, so that if concept A falls under concept B, then B is of "second-order". In the more technical treatment in his *"Grundgesetze"* (**1893**) he considered third-order quantifications, though his actual derivation of arithmetic proceeded entirely within second-order logic.
Frege was thus one of the first logicians to recognize the importance of a **hierarchy of logical levels**. His discovery was virtually simultaneous with Peirce's, and arrived at entirely independently, in pursuit of different goals. Frege's discovery was to have the greater impact: it formed the basis for Russell's theory of types, fundamental for the emergence of the modern FOL.
Despite their huge contribution, obviously neither Frege nor Peirce can be credited with a modern understanding of the difference between first-order and higher-order logics, but all this work lay decades in the future.

Later some mathematicians such as *Peano* and *Schröder* worked on Pearce and Frege's papers developing new forms while maintaining a certain critique of previous works, introducing some innovations but actually never exceeding the discoveries of these two, managing to be even less clear than previous documents and publications.

2.2 Early 20th century: Russel and Löwenheim

Bertrand Russell's discovery in **1901** of the *"Russell Paradox"* led him within a few months, in a letter to Frege to propose a tentative version of the *theory of types*. The central idea he took from Frege's theory was of functions of the first, second, and higher orders. Russell viewed the universe as striated into levels or types. The first type comprises the individuals; the second type comprises the "first-order" propositions whose quantifiers range over the individuals of the first type; in general, the quantifiers in propositions of the $n + 1^{st}$ type range over propositions of the n^{th} type. Russell's system in fact comprises two distinct hierarchies: one to deal with the paradoxes of set theory and the other to deal with the semantical paradoxes. This dual structure, branching in two directions, gives his theory the name **ramified theory of types**. In order to be able to establish classical analysis, he was forced to adopt the axiom of reducibility, which provides that any function of level $n + 1$ is coextensive with a predicate of function of lower level. The system was immensely complicated; in time, at the hands of *Chwistek*, *Ramsey*, *Carnap*, *Tarski*, and *Church*, it was recognized that the hierarchy dealing with the semantical paradoxes could be pruned away, leaving the "*simple theory of types"*.
Russell thus possessed a notation for the two quantifiers, as well as a distinction between quantifications of the first and higher types. But this is not the same as

possessing a conception of first-order logic, conceived as a free-standing logical system. There were essentially two things blocking the way: its object of study was not multiple logical systems and he thought of its system as an interpreted system, stating the truths of logic, rather than as a formal calculus in the sense of *D. Hilbert* (explained later), which use their axiomatization as the starting-point for his own axiomatizations of various systems of logic; but until the distinction between logic and metalogic had been formulated, it did not naturally occur to anybody to pose the metalogical questions of completeness and consistency; and it was only once such notions became the focus of attention that the significance of first-order logic became apparent.

In **1915**, *Leopold Löwenheim* with his work marks (from certain points of view) the beginning of model theory. Löwenheim considered a class of what he called *"counting expressions"* whose quantifiers range only over the domain of objects in the universe, but not over relatives. In modern terminology, his *"counting expressions"* are formulas of first-order logic; but his terminology shows no influence either from Peirce's logic of *"first intention"*, or from Russell's theory of types.
The Löwenheim theorem was in time to be recognized as isolating a fundamental property of first-order logic. But the full implications of his result were not to become clear until later, with *Hilbert* and the metamathematical study of logical systems.

2.3 Hilbert, Gödel and the birth of the modern FOL

So the first very large step was taken by *David Hilbert* during his lecture course *"Prinzipien der Mathematik"*, in the winter semester of **1917-1918**.
He delivered his programmatic lecture *"Axiomatisches Denken"* in Zürich calling for an axiomatic treatment of logic along the lines he had earlier explored in his axiomatization of geometry, and explicitly proposing metalogical investigations; according to Hilbert notes:

"When we consider the matter more closely, we soon recognize that the question of the consistency for integers and for sets is not one that stands alone, but that it belongs to a vast domain of difficult epistemological questions which have a specifically mathematical tint: for example [...] the problem of the subsequent checkability of the results of a mathematical investigation [...]".

These lectures are a remarkable document and mark the birth of modern mathematical logic. Hilbert for the first time clearly distinguishes metalanguage from object language, and step-by-step presents a sequence of formal logical calculi of gradually increasing strength. Each calculus is carefully studied in its turn; its strengths and its weaknesses are identified and balanced, and the analysis of the weaknesses is used to prepare the transition to the next calculus. He begins with the propositional calculu**s**, then moves to monadic quantificational logic and then to the **function calculus**.
The *"function calculus"* is a system of (many-sorted) first-order logic, with variables for sentences as well as for relations. It is here, for the first time, that we encounter a precise, modern formulation of first-order logic, presented in its own right as an

axiomatic logical system, suitable for study using the new metalogical techniques; unfortunately it was still primarily introduced as an expository device and its importance was not yet clear. So until the 20s a full understanding of the significance of the Hilbert ideas was missing. The Hilbert school throughout the 1920s regarded first-order logic as a fragment of type theory, and made no argument for it as a uniquely favored system. Only later with the monograph *Hilbert & Ackermann* (**1928**) that Hilbert explicitly called attention to the completeness of first-order logic as an open question. That set the stage for the work of *Gödel*.

The crucial technical breakthroughs came in **1929** and **1931** with the publication, by Kurt Gödel, first, of the completeness theorem for first-order logic, and then of the incompleteness theorems. With these results (and many others) it finally became clear that there were important metalogical differences between first-order logic and higher-order logics. Perhaps most significantly, first-order logic is complete, and can be fully formalized (a sentence is derivable from the axioms just in case it holds in all models). First-order logic moreover satisfies both *compactness* and the *downward Löwenheim-Skolem property*; so it has a tractable model theory. Second-order logic does not.

Note that the *downward Löwenheim-Skolem property* establishes that: *"if N is a model of (infinite) cardinality κ and if λ is an infinite cardinal smaller than than κ, then N has a submodel of cardinality λ which satisfies exactly the same sentences as N itself does".*

The fact that set theory is today a FO theory is probably due to *Thoralf Skolem*. In **1923** Skolem presented the original FO axiomatization of *Zermelo* set theory. Although Skolem was critical of second-order set theory as the foundation of mathematics, he proved that his (descendant) theorem was valid in FOL.
So we can conclude that the *downward Löwenheim–Skolem theorem* is one of the two key properties, along with the *compactness theorem*, that are used in Lindström's theorem to characterize first-order logic. In general, as we already said, this theorem does not hold in stronger logics such as second-order logic.

At this point in the **1930s** several other strands of thinking about logic now coalesced. The intellectual situation was complex and the famous papers by *Carnap*, *von Neumann*, and *Heyting* at the 1931 Königsberg congress had identified the logicist, formalist, and intuitionist schools: their debates were to shape thinking about the foundations of mathematics for the next several decades.
By the end of the decade, a consensus had been reached that, for purposes of research in the foundations of mathematics, mathematical theories ought to be formulated in first-order terms. Classical first-order logic had become “standard”.

3. Focus on theory and its rules

The first order logic is born as extension of the propositional logic, that is unable to effectively express some general concepts such as *"if someone is alone, they are not with someone else"* or *"there is no child older than its parents"*; so it becomes limited and expressionless in complex environments. To express principles like these, we need a way to talk about objects and individuals, as well as their properties and the relationships between them. These are exactly what is provided by the more expressive logical framework of the first-order logic.
While propositional logic deals with simple declarative propositions, FOL additionally introduces *predicates* and *quantifications*.

The first order logic, like any other logic, has two main part. The *syntax* determines which finite sequences of symbols are legal expressions, while the *semantics* determine the meanings behind these expressions. It adopts the fundation of the propositional logic taking representation ideas from the natural language, assuming that the world contains *objects* (e.g. people, houses, numbers, …), *relations* (e.g. red, true, brother of, …) and *functions* (e.g. father of, best friend, …). So in this chapter we well focus on functions and their meanings, providing an overview of all syntax and semantics using numerous examples and an initial reference model (Fig. 1); after we will move on to talk about another important aspect for logic in general, the concept of *equality* and its introduction in the FOL.

3.1 Intended meaning of functions: syntax and semantics

This section is characterized by a *"step-by-step introduction"* of the various elements of the FOL language, explaining also their semantics as we go along.

Models and domain

The *models* of a logical language are the formal structures that constitute the possible worlds under consideration. Each model links the vocabulary of the logical sentences to elements of the possible world, so that the truth of any sentence can be determined. Thus, while models for propositional logic link proposition symbols to predefined truth values, FOL models are much more interesting. First, they have objects in them. The *domain* of a model is the set of objects or domain elements it contains. The domain is required to be nonempty every possible world must contain at least one object. Mathematically speaking, it doesn't matter what these objects are, all that matters is how many there are in each particular model.
In Figure 1 we have a model with five **objects**: Rick, his younger brother Jack; the left legs of Rick and Jack; and a hat. The objects in the model may be related in various ways. For instance, Rick and Jack are brothers. Formally speaking, a **relation** is just the set of tuples of objects that are related.

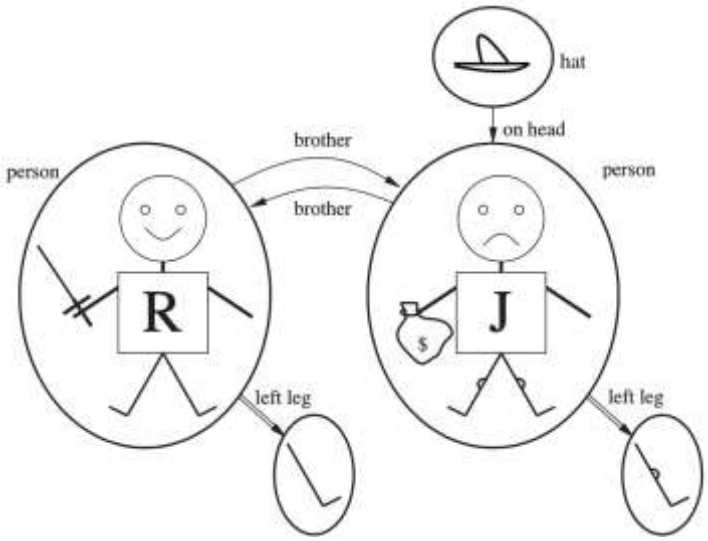

Figure 1: Model containing five objects, five relations and one unary function

For instance the tuple {⟨Rick, Jack ⟩, ⟨Jack, Rick⟩} will be the brotherood relation of the model.

Certain kinds of relationships are best considered as **functions**, in that a given object must be related to exactly one object in this way. For example, each person has one left leg, so the model has a unary *"left leg"* function that includes:

$$\langle \text{ Rick } \rangle \rightarrow \text{Rick's left leg} \quad \& \quad \langle \text{ Jack } \rangle \rightarrow \text{Jack's left leg}$$

Symbols

The basic syntactic elements of first-order logic are the symbols that stand for objects, relations, and functions.
Note that the logical connectives are (mentioned according with their precedence operator order from highest to lowest): ¬ for negation, = for equality,
$\wedge$ for conjunction, $\vee$ for disjunction, $\rightarrow$ for implication, $\leftrightarrow$ for biconditional.
The symbols are defined as *constants*, which stand for objects (e.g. Jack, X1, …); *predicate*s, which stand for relations (e.g Brother, OnHead, …); and *function* symbols, which stand for functions (e.g. LeftLeg, Mother, …). As with proposition symbols, the choice of names is entirely up to the user.
Every model must provide the information required to determine if any given sentence is true or false. Thus, in addition to its objects, relations, and functions, each model includes an **interpretation** that specifies exactly which objects, relations and functions are referred to by the constant, predicate, and function symbols. For instance *OnHead* symbol refers to the "on head" relation that holds between the hat and Jack. Obviously there are many other possible interpretations: even if the number of objects is restricted, the number of combinations can be very large.

Terms and Sentences

A term is a logical expression referred to an objects, so constant symbols are terms. Examples like *"x+y+z"* or *"square(x+y-z)"* are terms: intuitively, they name objects in the intended domain of discourse. So a **complex term** is formed by a function symbol followed by a parenthesized list of terms as arguments to the function symbol. Returning to our example in Figure 1, a possible term can be: *LeftLeg(Rick)*.

Once we have terms for referring to objects and predicate symbols for referring to relations, we can put them together to make **atomic sentences** that state facts, a predicate symbol optionally followed by a parenthesized list of terms (e.g. *Brother(Rick, Jack))*.
Note that we can use logical connectives to construct now more **complex sentences**, with the same syntax and semantics as in propositional calculus:
for example *Brother (Rick, Jack)* $\wedge$ *Brother (Jack, Rick)* means Rick is brother of Jack AND Jack is brother of Rick.

Quantifiers

Once we have a logic that allows objects, it is only natural to want to express properties of entire collections of objects, instead of enumerating the objects by name. Quantifiers let us do this, allowing us to make general assertions. First-order logic contains two standard quantifiers, called *universal* and *existential*.

The **universal quantifier** $\forall$ followed by a variable x is meant to represent the phrase "for every x." In other words, it asserts that every value of x has the property that follows it. To make an example we can write *"all men are people"* with:

$$\forall x \; \text{Man}(x) \Rightarrow \text{Person}(x)$$

that stands for: *"For all x, if x is a man, then x is a person"*

While universal quantification makes statements about "every" object, an **existential quantifier** $\exists$ can make a statement about "some" object in the universe without naming it. For example we can say: *"Jack has a hat on his head"*, so we write

$$\exists x \; \text{Hat}(x) \wedge \text{OnHead}(x, \text{Jack})$$

where $\exists x$ stands for: *"There exists an x such that…"*

What makes first-order logic powerful is the possibility to express more complex sentences using multiple quantifiers (**nested**), of different types but also of the same.

For instance the sentence *"There is someone who is hated by everyone"* is represented with $\exists y \; \forall x \; \text{Hates}(x, y)$ while the sentence *"Everybody hates somebody"* with $\forall x \; \exists y \; \text{Hates}(x, y)$. Just changing the quantifier the meaning of the sentence changes completely.

Note that the two quantifiers are actually intimately connected with each other, through negation. As serting that everyone dislikes cakes is the same as asserting there does not exist someone who likes them, and vice-versa:

$$\forall\ x\ \neg \text{Likes}\ (x, \text{Cakes}) \text{ is equivalent to } \neg \exists\ x\ \text{Likes}\ (x, \text{Cakes})$$

Brief introduction to the FOL usage

Once we have defined an expressive logical language, it is time to learn how to use it. Sentences are added to a **knowledge base** (KB) using TELL, exactly as in propositional logic. Such sentences are called *assertions*; for example, we can assert *"Rick is a person"* by writing: TELL(KB, Person(Rick)).
Then we can ask questions of the knowledge base using ASK. For example, ASK(KB, Person(Rick)) returns True. Questions asked in this way are called **queries** (or goals). Note that generally speaking, any query that is logically entailed by the knowledge base should be answered affirmatively.

In case of quantified queries, such as ASK(KB, $\exists$ x Person(x)) we want to know what value of x makes the sentence True, so we will need a different function, ASKVARS, which we call with ASKVARS(KB, Person(x)), which yields a stream of answers. In this case there will be more answers: for example {x/Jack} and {x/Rick}. Such an answer is called a **substitution** or binding list. So these are ways to represent simple domains (just some part of the world about which we wish to express some knowledge), but it must be said that it is possible to express much more complex representations; for reasons beyond the scope of this project they have not been more elaborated here.

3.2 Introduction of the Equality

In symbolic logic, we use the expression *a=b* to express the fact that *'a'* and *'b'* are *"equal"* or *"identical"*. The equality symbol is meant to model what we mean when we say, for example, *"Marco's sister is the cheerleader",* or *"2 + 2 = 4"*. We are asserting that two different descriptions refer to the same object. Because the notion of identity can be applied to virtually any domain of objects, it is viewed as falling under the province of logic. Also *Gottlob Frege* in his study on the *philosophy of language* in the distant 1880 had introduced his own thought about *"identity statements"*: he belived that every statement of this type all have the form *a=b* where *a* and *b* are either names or descriptions that denote individuals, and it is true if and only if the object *a* just is (identical to) the object *b*.

So talking about **equality** raises messy philosophical questions. Am I the same person I was three days ago? Are the two copies of Huckleberry Finn sitting on my shelf the same book, or two different books? Using symbolic logic to model identity presupposes that we have in mind a certain way of carving up and interpreting the

world. We assume that our terms refer to distinct entities, and writing x=y asserts that the two expressions refer to the same thing. Axiomatically, we assume that equality satisfies the three properties of reflexivity, symmetry and transitivity.
However these properties are not enough to characterize equality. If two expressions denote the same thing, then we should be able to substitute one for any other in any expression. It is convenient to adopt the following convention: if r is any term, we may write $r(x)$ to indicate that the variable x may occur in r. Then, if *'a'* is another term, we can thereafter write r(a) to denote the result of replacing a for x in r.r
The substitution rule for terms thus reads as follows: if a=b then r(a)=r(b).
Note that using equality, we can define even more quantifiers.
We can express *"there are at most two elements x such that $A(x)$ holds"* as $\forall x \forall y \forall z(A(x) \wedge A(y) \wedge A(z) \rightarrow x=y \vee y=z \vee x=z)$. This states that if we have three elements a for which $A(a)$ holds, then two of them must be equal.

So it is now clear that first-order logic includes one more way to make atomic sentences, other than using a predicate and terms as described earlier. We can use the equality symbol to signify that two terms refer to the same object. For example referring again to our "case of the two brothers" (Fig. 1), we have that:

Brother(Jack) = Rick

says that the object referred to the brother of Jack and the object referred to Rick are the same. Because an interpretation fixes the referent of any term, determining the truth of an equality sentence is simply a matter of seeing that the referents of the two terms are the same object.
So the equality symbol can be used to state facts about a given function, as we just did, but it can also be used with negation to insist that two terms are not the same object. To say that Rick has at least two brothers, we would write:

$\exists$ x,y Brother(x,Rick) $\wedge$ Brother(y,Rick) $\wedge \neg$(x=y)

Here the sentence *"* $\exists x,y$ *Brother(x,* Rick *)* $\wedge$ *Brother(y,* Rick *)"* does not have the intended meaning. In particular, it is true in the model of Figure 1, where Rick has only one brother. To see this, we must consider the extended interpretation in which both x and y are assigned to Jack. The addition of $\neg(x = y)$ rules out such models.
Note: the notation $x \neq y$ is sometimes used as an abbreviation for $\neg(x = y)$.

We can conclude that the introduction of the equality symbol introduces in turn a very important concept in first order logic, that of *uniqueness*, a condition wherein someone or something is unlike anything else in comparison. Let's see a new example. The sentence *"There is a king"* can be easily written as $\exists$ x King(x); conversely the sentence *"There is only one king"* needs to be expressed in this way:

$\exists$ x (King(x) $\wedge$ $\forall$ y (King(y) $\rightarrow$ y=x))

4. Limits and additional aspects

Although first-order logic is well known to be extremely powerful and it is sufficient for formalizing much of mathematics, and is commonly used in computer science and other fields, it has certain limitations. These are about its **expressiveness** and limitations of the **fragments of natural languages** that it can describe; furthermore it cannot in general define the transitive closure of a relation.
For instance, first-order logic is undecidable, meaning a sound, complete and terminating decision algorithm for provability is impossible. This has led to the study of interesting decidable fragments, such as C_2: a first-order logic with two variables and the counting quantifiers $\exists^{\geq n}$ and $\exists^{\leq n}$.

Expressiveness

As we shown in chapter 1, the Löwenheim–Skolem theorem shows that if a first-order theory has any infinite model, then it has infinite models of every cardinality. In particular, no first-order theory with an infinite model can be categorical. Thus there is no first-order theory whose only model has the set of natural numbers as its domain, or whose only model has the set of real numbers as its domain. Many extensions of first-order logic, including infinitary logics and higher-order logics, are more expressive in the sense that they do permit categorical axiomatizations of the natural numbers or real numbers. This expressiveness comes at a metalogical cost, however: by Lindström's theorem, the compactness theorem and the downward Löwenheim–Skolem theorem cannot hold in any logic stronger than first-order.

Formalizing natural languages

First-order logic is able to formalize many simple quantifier constructions in natural language, such as "every person who lives in Rome, lives in Italy". But there are many more complicated features of natural language that cannot be expressed anyway. So any logical system which is appropriate as an instrument for the analysis of natural language needs a much richer structure than first-order predicate logic.
For example: *"If Mario is self-satisfied, then there is at least one thing he has in common with Giovanni"* requires a quantifier over predicates, which cannot be implemented in single-sorted first-order logic, $Zj \rightarrow \exists X(Xj \wedge Xp)$.
If we have: *"Jumbo is a small elephant"* cannot be analysed as $Sj \wedge Ej$; predicate adjectives are not the same kind of thing as second-order predicates such as colour.
FOL also bad at handling default information, leading to inconsistency, like in the following example (that is unsatisfiable):

$$\forall x \; bird(X) \rightarrow flies(x)$$
$$bird(tweety), bird(opus), not\ flies(opus)$$

Another shortcoming of first-order logic is that it is not possible to express the concept of *"path"*: so some other logics are used in this case, to formalize certain concepts, like temporal logic (based on quantifcation over time) and linear logic.

5. Brief experimental part

The aim of this section is to transform some theoretical concepts in something practical and concrete, going to work with a very **simple FOL solver**. All this was possible thanks to a short code implemented in Python.

Given an *input file* of the form:

<NQ = # OF QUERIES> <QUERY 1> ... <QUERY NQ>

<NS = # OF GIVEN SENTENCES IN THE KB> <SENTENCE 1> ... <SENTENCE NS>

where: each query will be a single literal of the form Predicate(Constant) or ~Predicate(Constant), variables are all single lowercase letters, all predicates and constants are case-sensitive alphabetical strings that begin with an uppercase letter and then each predicate takes at least one argument.

The code takes the input file and for each query, determine if that query can be inferred from the knowledge base or not, one query per line:

<ANSWER 1>
...
<ANSWER NQ>

where each answer should be either TRUE if you can prove that the corresponding query sentence is true given the knowledge base, or FALSE if you cannot.

So the output file will be of the form:

TRUE
TRUE
FALSE
...

Note that some other considerations were made for this work too: all variables are assumed to be universally quantified. There is no existential quantifier and Skolem functions (or Skolem constants) are not used. There will be no parentheses in the sentences, other than around arguments of predicates. The knowledge base will be considered always consistent. So there are no contracting rules or facts in the knowledge base. In this very simple version we will work only with NOT (~) and OR (|) operators: in this case their priorities are respected (negation has higher priority than disjunction).

The python code (attached with this paper) is taken from a previous work and optimized for our purpose; it is very simple and just some main methods are presents,

in particular: *resolution* method deals with the analysis of the Knowledge Base by learning each rule and scrolling through it following each visit order, learning every property in this way (thanks to checking methods like *isAlreadyVisted* or *isRepeated*); then we have *parse_input* and *write_output* methods: they will deal with the reading of the input file and its initial analysis, and the final writing of the output file using the *inference* method for appending 'True' and/or 'False' strings in that file.

Now let's move to our experiment. The input file is:

```
5
~Brother(Alan,Jack)
Brother(Rick,Jack)
Brother(Wayne,Rick)
Friend(Jack,Alan)
~Friend(Jack,Wayne)
8
Windsor(Jack)
Windsor(Rick)
Snow(Alan)
Rooney(Wayne)
~Windsor(x) | ~Windsor(y) | Brother(x,y)
~Snow(x) | ~Windsor(y) | ~Brother(x,y)
~Rooney(x) | ~Windsor(y) | ~Brother(x,y)
~Windsor(z) | ~Snow(y) | ~Rooney(x) | ~Friend(y,z) | Friend(x,z)
```

where it's defined a relationship of brotherhood and friendship for three different family groups (Windsor, Snow, Rooney) and four people (Jack, Rick, Alan, Wayne): for instance the two Windsors are brothers while Snows and Rooneys are friends.

So by analizing the five queries, we will obtain as output:

```
TRUE
TRUE
FALSE
FALSE
FALSE
```

that is consistent with what we defined into the Knowledge base.

This is a very trivial procedure for verifying the identity of certain queries, simple and a bit limited, if you wanted to deal with more complicated and large environments. But it is certainly a good basis to start from to generate a possible more effective FOL solver.

6. Conclusion and final discussion

Logic, in general, is a method of codifying language into a form that can be analysed using mathematical techniques, with the main focus of this analysis being the development of standards for reasoning.

In this work we have seen how the rise of the FOL was almost inevitable over time and how this has become quite important in the mathematical field and beyond.
In first-order logic, we analyse language as consisting of *names for objects* (often represented as a, b, c, etc.), *predicates*, which say something about particular objects. Predicates are often represented with capital letters like F or P, and combine with object-names to form a proposition; then we seen *connectives*, which combine propositions together, like conjunction, disjunction and negation (also considered a connective, even though it operates on a single proposition); finally we seen quantifiers, which allow for generalizations. Remember that the term *first-order* means that this logic only uses quantifiers to generalize over objects, and never over predicates.

We have seen how over time FOL has been gradually identified and discovered, thanks to a long work that lasted for more than a century of continuous research and study by a long group of mathematicians, scientists, philosophers and many schools of thought, which finally led in the *30s* to a clear separation of it from the higher order logics becoming one of the most used in this field.

Then it is also mentioned why and how it is not perfect anyway, highlighting its many limitations.
Here a question arises spontaneously, *"why do we use first-order logic?"*
Well, it has enough expressive resources to formalize most of what we say in contexts like mathematics, where formalization is important. It's also adequate for most of our everyday conversation.
The other big advantage of first-order logic is that it has a sound and complete proof theory, which means that we can use the formal tools to discover interesting results rather than just rephrasing what we've already said. If we go to systems with more expressive power (such as second-order logic) we lose this ability. The explanation for that has to do with *Godel*'s incompleteness theorems. Second-order logic actually has enough expressive power to define the natural numbers without assuming any mathematical axioms, and thus the incompleteness result bars it from having a complete proof theory.

David Hilbert said: *"Who of us would not be glad to lift the veil behind which the future lies hidden; to cast a glance at the next advances of our science and at the secrets of its development during future centuries?"*. This quote makes us understand what was at the basis of one of the minds that contributed to the birth of FOL, a continuous desire for research towards the future and innovation.

References

[1] https://en.wikipedia.org/wiki/First-order_logic

[2] https://leanprover.github.io/logic_and_proof/first_order_logic.html

[3] https://www.javatpoint.com/first-order-logic-in-artificial-intelligence

[4] https://plato.stanford.edu/entries/logic-firstorder-emergence/

[5] https://philosophy.stackexchange.com/questions/2617/how-did-first-order-logic-come-to-be-the-dominant-formal-logic

[6] http://www.doc.ic.ac.uk/~cclw05/topics1/history.html

[7] https://www.researchgate.net/publication/308046638_Application_of_First-Order_Logic_in_Knowledge_Based_Systems

[8] https://www.britannica.com/topic/history-of-logic

[9] https://philosophy.stackexchange.com/questions/10459/how-important-are-equality-functions-and-constants-for-first-order-logic

[10] http://www.lacl.fr/fmadelaine/Download/PositiveEqualityFreeTOCL.pdf

[11] https://rjlipton.wordpress.com/2010/01/17/a-limit-of-first-order-logic/

[12] https://mathworld.wolfram.com/First-OrderLogic.html

[13] "*Artificial Intelligence: A Modern Approach*", *Stuart Russel & Peter Norvig*

[14] https://math.stackexchange.com/questions/877211/example-of-first-order-logic-without-equality

[15] http://www.opentextbooks.org.hk/ditatopic/9631

[16] https://www.youtube.com/watch?v=PebraKYRzLs

[17] https://www.youtube.com/watch?v=Vg8lNba62wQ

[18] https://en.wikipedia.org/wiki/Logic

[19] https://en.wikipedia.org/wiki/Löwenheim–Skolem_theorem

[20] https://www.youtube.com/watch?v=zRsN8V6cisg

[21] https://github.com/smitp94/First-Order-Logic

[22] https://github.com/m0hamed/fol-solver

[23] https://en.wikipedia.org/wiki/Uniqueness

[24] http://homepage.cs.uiowa.edu/~tinelli/classes/295/Spring05/notes/equality1.pdf

[25] https://www.umsu.de/trees/

[26] https://www.encyclopedia.com/humanities/encyclopedias-almanacs-transcripts-and-maps/first-order-logic

[27] https://www.quora.com/What-is-first-order-logic-and-how-what-do-we-use-it-for

[28] http://www.phil.gu.se/posters/first-order-logic.pdf

[29] https://plato.stanford.edu/entries/frege/#FreConLog