KNOWLEDGE USING PREDICATE LOGIC

Logic is a formal system in which the formulas or sentences have true or false values. Logics are of different types:

Propositional logic, Predicate logic, Temporal logic, Modal logic, Description logic etc;

They represent things and allow more or less efficient inference.

● Propositional Logic is the study of statements and their connectivity.

● Predicate Logic is the study of individuals and their properties. We need languages that allow us to describe properties (predicates) of objects, or a relationship among objects represented by the variables. Predicate logic satisfies the requirements of a language. Predicate logic is powerful enough for expression and reasoning. Predicate logic is built upon the ideas of propositional logic. Predicate Logic can represent Objects and Quantification. Theorem proving is semidecidable. Proposition is a declarative sentence whose value is either true or false

Basics of logic symbols:

"→" Material Implication

"┐ " Not

"v" Or

"^" and

"∀" for all

"Ǝ" There exists

Representing simple facts in Logic

● Proposition logic is simple where a decision procedure exists.

● Represent Real-World facts as logical propositions, written as well – formed formulas in propositional logic as shown in figure below:

Example: Conclusion

● The fact “It is raining“ the fact it is not sunny

● Limitations of propositional logic are overridden.

If we want to represent the fact stated by the classical sentence.

It is raining RAINING

Its is sunny SUNNY

It is windy WINDY

If it is raining, then it is not sunny RAINING → ⌐ SUNNY

Socrates is a man We could write: SOCRATESMAN

But if we also wanted to represent Plato is a man We would have to write something such as: PLATOMAN Which would be a totally separate assertion, and we would not be able to draw any conclusions about similarities between Socrates and Plato. It would be much better to represent these facts as:

MAN (SOCRATES) MAN (PLATO)

Since, now the structure of the representation reflects the structure of the knowledge itself. But to do that, we need to be able to use predicates applied to arguments. We are in even more difficulty if we try to represent the equally classic sentence All men are mortal. We could represent this as: MORTALMAN To capture the relationship between any individual being a man and that individual being a mortal. To do this, we need variable and quantification unless willing to write separate statements about the mortality of every known man.

Predicate Logic

This logic is one way of representing knowledge because it permits representations of things that cannot reasonably be represented in prepositional logic.

● If we use logical statements as a way of representing knowledge, than we have good way of reasoning with that knowledge.

● Determining the validity of a propositional logic is easy but may be computationally hard.

● Predicate logic provides a good way of reasoning with knowledge.

● It provides a way of deducing new statements from old ones.

● It does not possess a decision procedure, even an exponential one. There exist procedures that will find a proof of a proposed theorem if indeed it is a theorem. First order predicate logic is semi decidable. Such a procedure is to use the rules of inference to generate theorems form the axioms in some fashion, testing each to see if it is the one for which a proof is sought.

Examples of Predicate Logic

Let's now explore the use of predicate logic as a way of representing knowledge by looking at a specific example. Consider the following set of sentences:

1 . Marcus was a man.

2. Marcus was a Pompeian.

3. All Pompeians were Romans.

4. Caesar was a ruler.

5. All Romans were either loyal to Caesar or hated him.

6. Everyone is loyal to someone.

7. People only try to assassinate rulers they are not loyal to.

8. Marcus tried to assassinate Caesar.

The facts described by these sentences can be represented as a set of wff's in predicate logic as follows:

1. Marcus was a man

Man (Marcus)

Although this representation fails to represent the notion of past tense (which is clear in the English sentence), it captures the critical fact of Marcus being a man. Whether this omission is acceptable or not depends on the use to which we intend to put the knowledge.

2. Marcus was a Pompeian

Pompeian (Marcus)

3. All Pompeians were Romans

$\forall x$: Pompeian (x) $\rightarrow$ Roman(x)

4.Caesar was a ruler

Ruler (Caesar)

Since many people share the same name, the fact that proper names are often not references to unique individuals, overlooked here. Occasionally deciding which of several people of the same name is being referred to in a particular statement may require a somewhat more amount of knowledge and logic

5. All Romans were either loyal to Caesar or hated him

$\forall x$: Roman(x)$\rightarrow$ loyalto(x, Caesar) V hate(x, Caesar)

In English, the word "or" sometimes means the logical inclusive-or and sometimes means the logical exclusive-or (XOR). Here we have used the inclusive interpretation. Soome people will argue, however, That this English sentence is really stating an exclusive-or. To express that, we would have to write:

$\forall x$: Roman(x) $\rightarrow$ [(loyalto (x, Caesar) V hate(x, Caesar)) V $\neg$ (loyalto(x, Caesar) ^ hate(x, Caesar))]

6. Everyone is loyal to someone

$\forall x$:$\rightarrow\exists y$: loyalto (x,y)

The scope of quantifiers is a major problem that arises when trying to convert English sentences into logical statements. Does this sentence say, as we have assumed in writing the logical formula above, that for each person there exists someone to whom he or she is loyal, possibly a different someone for everyone? Or does it say that there is someone to whom everyone is loyal?

7. People only try to assassinate rulers they are not loyal to

$\forall x : \forall y : person(x) \wedge ruler(y) \wedge tryassassinate(x,y) \rightarrow \neg loyalto(x,y)$

Like the previous one this sentence too is ambiguous which may lead to more than one conclusion. The usage of "try to assassinate" as a single predicate gives us a fairly simple representation with which we can reason about trying to assassinate. But there might be connections as try to assassinate and not actually assassinate could not be made easily.

8. Marcus tried to assassinate Caesar

tryassasinate (Marcus,Caesar) If we want to use these statements to answer the question: "Was Marcus loyal to Caesar?" Using 7 and 8, prove that Marcus was not loyal to Caesar.

To produce a Formal proof, reasoning backward from the desired goal:

¬loyalto(Marcus, Caesar)

To prove the goal

● Use the rules of inference to transform it into another goal that can be transformed and so on, until there are no unsatisfied goals remaining.

● This process requires the search of an AND-OR graph when there is an alternative way of satisfying individual goals. We show only a single path for simpler way.

● The below figure shows an attempt to produce a proof of the goal by reducing the set of necessary but as yet unattained goals to the empty set. If attempt fails, there is no way to satisfy the goal person with the available statements. Problem: Marcus was a man, need to add the representation of another fact to our system.

9. All men are people

$\forall x : man(x) \rightarrow person(x)$

Now we can satisfy the last goal and produce a proof that Marcus was not loyal to Caesar.

-loyalto (Marcus, Caesar)

↑ (7,substitution)

person (Marcus) ^ ruler (Caesra) ^ tryassassinate (Marcus, Caesar)

↑ (4)

person (Marcus) tryassassinate (Marcus, Caesar)

↑ (8)

person (Marcus)

Computable functions and predicates

Some of the computational predicates like Less than, Greater than used in knowledge representation. Simple facts were expresses as combinations of individual predicates, such a: Tryassassinate (Marcus, Caesar) If Number of facts is not large or if the facts are unstructured then there is little alternative.

If we want to express simple facts, like greater than and less than relations:

gt(1,0)

lt(0,1)

gt(2,1)

lt(1,2)

gt(3,2)

lt(2,3)

There is infinite representation of each of these fact. But if we consider the finite number of them that can be represented, using a single machine word per number, extremely inefficient to store explicitly large a set of statements. Instead, we compute each one as we need it.

Functions and predicate use

Consider the following set of facts Again involving Marcus:

1. Marcus was a man

Man (Marcus) Again we ignore the issue of tense.

2. Marcus was a pompeian

Pompeian (Marcus)

3. Marcus was born in 40 A.D

Born (Marcus, 40) For simplicity, we will not represent A.D. explicitly, just as we normally omit it in everyday discussions. If we ever need to represent dates B.C., then we will have to decide on a way to do that such as by using negative numbers.

4. All men are mortal

$\forall x$: men(x)→ mortal(x)

5. All Pompeians died when the volcano erupted in 79 A.D

erupted(volcano,79) $\forall x$:[pompeian(x)→died(x, 79)]

This sentence clearly asserts the two facts represented above. It may also assert another that we have not shown, namely that the eruption of the volcano caused the death of the Pompeians. People often assume causality between concurrent events if such causality seems possible.

6. No mortal lives longer than150 years

$\forall x$: $\forall t1$: $\forall t2$: mortal(x) ∧ born(x,t1) ∧gt(t2-t1,150)→ dead(x,t2)

There are several ways that the content of this sentence could be expressed.

For example, we could introduce a function age and assert that its value is never greater than 150.

7. It is Now 1991

Now=1991

Here we exploit the idea of equal quantities that can be substituted for each other. Now suppose we want to answer the question “Is Marcus alive?” A quick glance through the statements we have suggests that there may be two ways of deducing an answer. Either we can show that Marcus is dead because he was killed by the volcano or we can show that he must be dead because he would otherwise be more than 150 years old, which we know is not possible.

As soon as we attempt to follow either of those paths rigorously, however, we discover, just as we did in the last example, that we need some additional knowledge. For example, our statements talk about dying, but they say nothing that relates to being alive, which is what the question is asking. So we add the following facts:

8. Alive means not dead

∀x: ∀t: [alive(x,t) → ¬ dead(x,t)] Ù[¬ dead(x,t)→alive(x,t)]

This is not strictly correct, since ¬ dead implies alive only for animate objects. This is an example of the fact that rarely do two expressions have truly identical meanings in all circumstances.

9. If someone dies then he is dead at all later times

∀x: ∀t1: ∀t2: died(x,t1) ^gt(t2,t1)→ dead(x,t2) This representation says that one is dead in all years after the one in which one died. It ignores the question of whether one is dead in the year in which one died.