جامــــعة المـــستـقـبـل
AL MUSTAQBAL UNIVERSITY

كلية العلوم
قـــســـــــــم الانـــظـــمـــة الـــطـبـيـة الـــذكـــــيــة

Lecture: (8)

Knowledge Representation Resolution and Unification

Subject: Artificial Intelligence

Class: Third

Lecturer: Dr. Maytham N. Meqdad

 Study Year: 2024-2025

Knowledge Representation Resolution and Unification

1. Resolution

It is an algorithm for proving facts true or false by *virtue of contradiction*. If we want to prove a theorem X is true, we have to show that the negation of X is not true.

Resolution theorem:
For any three clauses p, q and r,

$$p \vee r,\ q \vee \neg r \ \Rightarrow\ p \vee q$$

Modus Pones

It can be summarized as "P implies Q; P is asserted to be true, so therefore Q must be true." The history of modus ponens goes back to antiquity. Modus ponens allows one to eliminate a conditional statement from a logical proof or argument (the antecedents) and thereby not carry these antecedents forward in an ever-lengthening string of symbols; **on other words, the rule of logic stating that if a conditional statement ("if p then q ") is accepted, and the antecedent (p) holds, then the consequent (q) may be inferred. Such as:**

If it is raining, I will meet you at the theater.
It is raining.
Therefore, I will meet you at the theater.

Modus ponens can be stated formally as:

$$\frac{P \to Q,\ P}{\therefore Q}$$

The argument form has two premises (hypothesis). The first premise is the "if– then" or conditional claim, namely that P implies Q. The second premise is that P, the antecedent of the conditional claim, is true. From these two premises it can be logically concluded that Q, the consequent of the conditional claim, must be true as well. Inartificial intelligence, modus ponens is often called forward chaining.

An example of an argument that fits the form *modus ponens*:

If today is Tuesday, then John will go to work.
Today is Tuesday.
Therefore, John will go to work.

Example

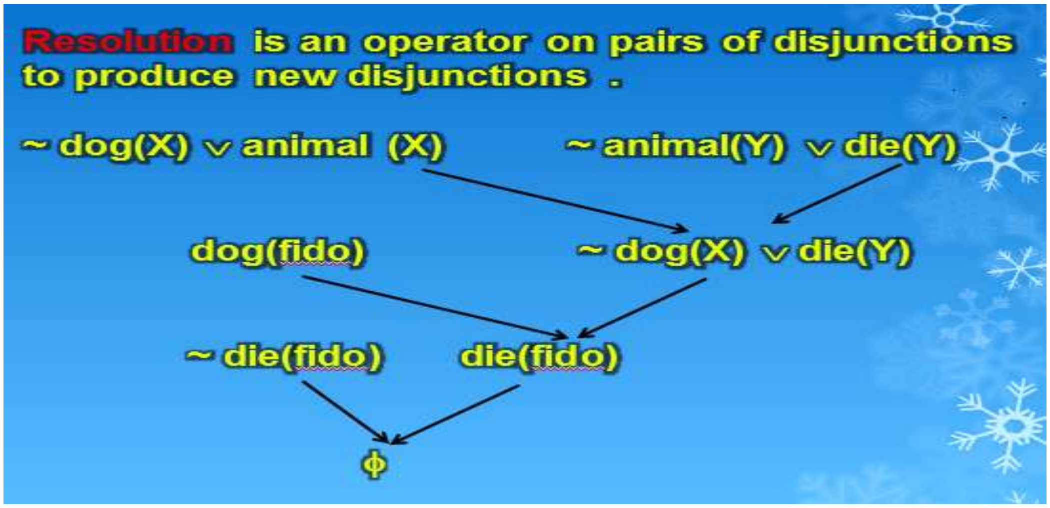
Prove that
"Fido will die" from the statements: die(Fido)

"Fido is a dog",
"all dogs are animals"

"all animals will die", with applying modus ponens

1- All dogs are animals: ∀X (dog(X) → animal (X))
2-Fido is a dog : dog (fido)
3-Modus Ponens and {fido / X} gives : animal(fido)
4- All animals will die : ∀Y (animal(Y) → die (Y))
5- Modus Ponens and {fido/Y} gives : die (fido).

Clause form:
~ dog(X) ∨ animal (X)
dog(fido)
~ animal(Y) ∨ die(Y)
~ die(fido)



2. <u>Unification</u>

Unification is a technique for taking two sentences in predicate logic and finding a substitution that makes them look the same.
- A variable can be replaced by a constant.
- A variable can be replaced by another variable.
- A variable can be replaced with a predicate, as long as the predicate does not contain that variable.

Unification Conditions:
1- The two predicate names must be the same.
2-The two predicates must have same no. of arguments.

Example:

Given the following set of predicates,
1.hates(X , Y)
2.hates(John, Football).
3. hates(Adam, Spinach).

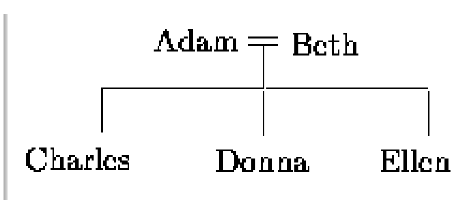
Unify 1 and 2: S={john/X, football/ Y}
Unify 1 and 3: S={adam/X, spanich/Y}

If we introduce more complex unifications:

4. Like(X, season(Y))
5. Like(George, season(summer)).
6. Like(Z, summer)

Unify 4 and 6: S={Z/X, summer/Y}
Unify 4 and 5: S={george/X, summer/Y}

3-Frames
A frame is a collection of attributes which defines the state of an object and its relationship to other frames (objects).
Frames are called Slot-and-Filler data representations.
Slots are the data values,
Fillers are attached procedures.
Frames are often linked into a hierarchy to represent has-part and isa relationships.



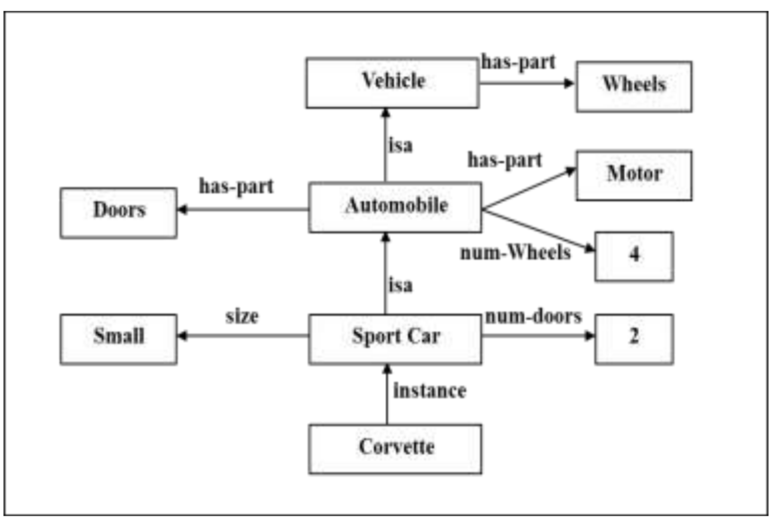
Adam:
sex: Male
spouse: Beth
child: (Charles Donna Ellen)

Frame problem

The problem is that if we copy the *complete frame* for each step in the sequence, then we may quickly use up the *computer memory*, as we duplicate the same unchanged knowledge over and over. and it is not always obvious which attributes in the frame should change.
Solution is to specify which parts must match for a condition to be true, and to only change those slots or attributes.

4-Semantic Nets

Semantic nets are used to define the meaning of a concept by its relationships to other concepts. A graph data structure is used, with nodes used to hold concepts, and links with natural language labels used to show the relationships. Frames and semantic nets are very closely related to predicate logic .



5-Probabilities

a- Unconditional probabilities:
Represent the chance that something will happen.
Example: we can look in a weather almanac and see that, on average, it has rained 10 days in March in London over the last hundred years. So the probability that it will rain on any given day in March is roughly 33 percent.

b- Conditional probability:

Expressed as P(H | E), that is read as the probability of hypothesis H given that we have observed evidence E.
Example: suppose we know that a big storm is blowing in from South Babil, and it will reach Baghdad tomorrow. Given that knowledge, we may say there is an 80 percent chance of rain.