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.

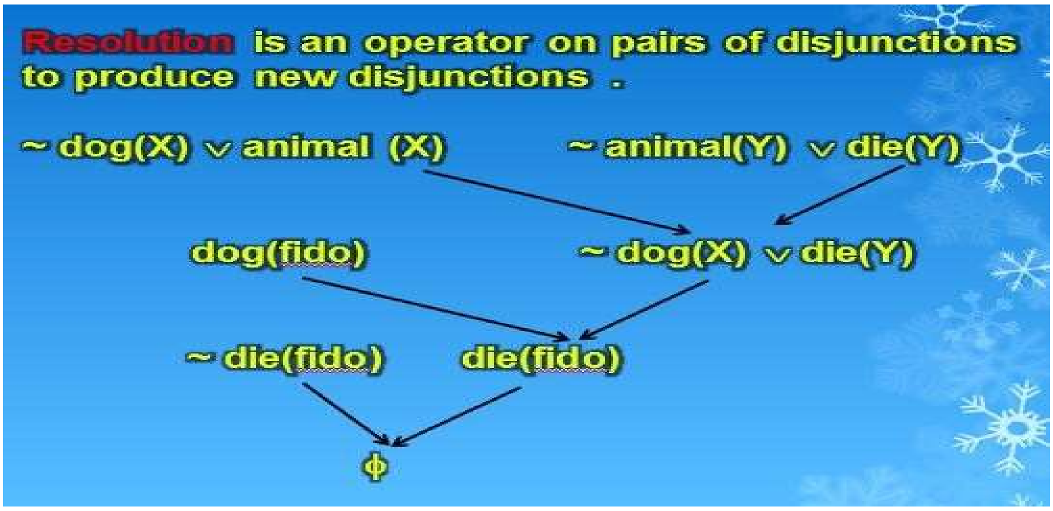
Resolution theorem:
For any three clauses p, q and r,
$p \vee r,\ q \vee \neg r \quad \Rightarrow \quad p \vee q$

Example

Prove that
"Fido will die" from the statements:
"Fido is a dog",
"all dogs are animals"
"all animals will die", with applying modus ponens

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

Clause form :
$\sim dog(X) \vee animal\ (X)$
$dog(fido)$
$\sim animal(Y) \vee die(Y)$
$\sim die(fido)$



Resolution is an operator on pairs of disjunctions to produce new disjunctions .

~ dog(X) ∨ animal (X)

~ animal(Y) ∨ die(Y)

dog(fido)

~ dog(X) ∨ die(Y)

~ die(fido)

die(fido)

Φ

2. Unification

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).

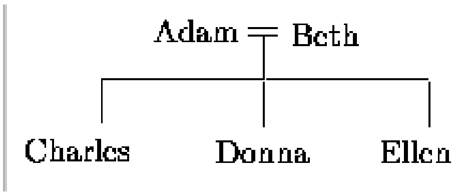
Unify 1 and 2: S={john/X, footbay/Y}
Unify 1 and 3: S={adam/X, spanich/Y}
If we introduce more complex unifications:

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

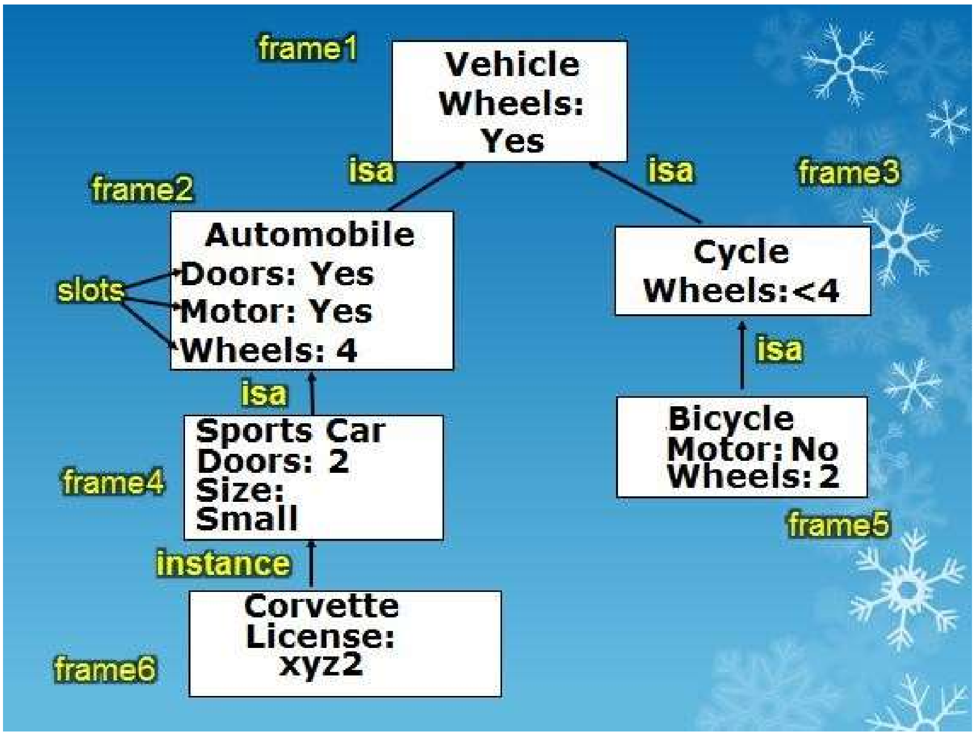
A frame is a collection of attributes which defines the state of an object and its relationship to other frames (objects).
Frames are called Slot-and-Filler data representations.
Slots are the data values,
Fillers are attached procedures.
Frames are often linked into a hierarchy to represent has-part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



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

A frame has two different types of name:

1-True name (tname), (frame-27)
2-Number of public names (pnames): Public names are stored as values in the name slot of the frame.



From the above figure:

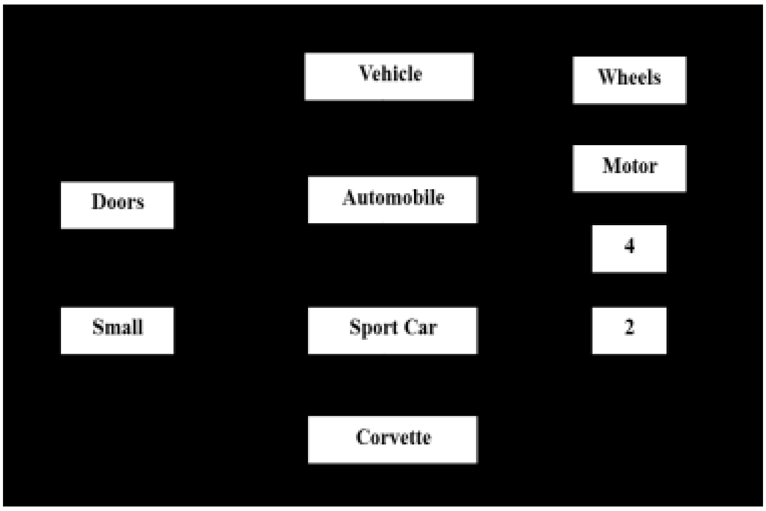
- Each frame has a set of slots.
- The automobile frame has three slots.
- The automobile has doors, a motor, and four wheels, It is a subset of vehicles.
- A sports car is a subset or type of automobile. It has two doors and is small.
- Corvette is an instance of a sports car, and each instance has a unique license number.

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.