Frames:

A frame is a collection of attributes (slots) and associated values that describe some entity in the world.

Frame System:

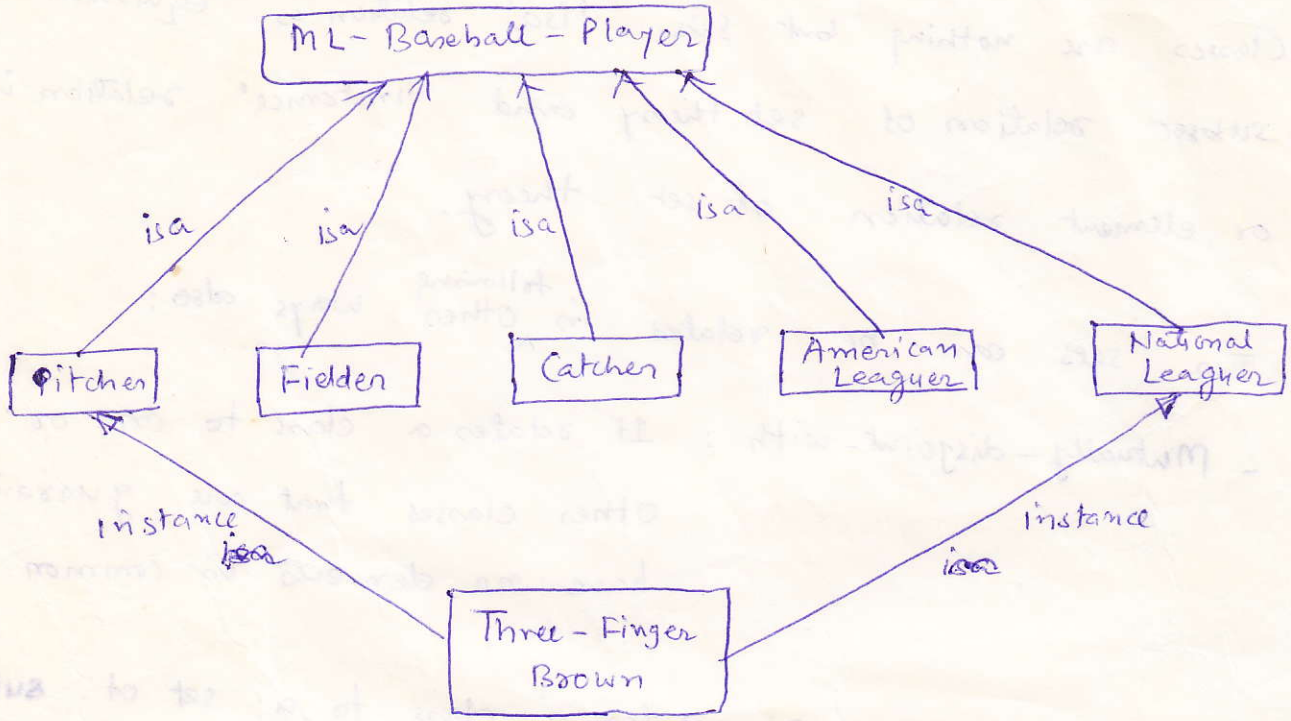
It's a collection of frames which are connected to each other by virtue of the fact that the value of an attribute of one frame may be another frame.

Ways Of Relating Classes to each Other :

- Classes are nothing but sets. 'isa' relation is equivalent to subset relation of set theory and 'instance' relation is m or element relation of set theory.
- Two sets can be related in following ^ Other ways also :
  - mutually-disjoint-with : It relates a class to one or mor other classes that are guaranteed have no elements in common with
  - is-covered-by : It relates a class to a set of subclas the union of which is equal to it.
- If a class is is-covered-by a set S of mutually disjoint c then S is called a partition of the class.

Example : Consider the class of major league baseball player Everyone is either a pitcher, a catcher or a fielde (No one is more than one of these). In addition, everyone plays in either the National League or the American League, but not both.

Representing Relationships among Classes :



ML - Baseball - Player

is-covered-by : 1) { Pitcher, Fielder, Catcher },

2) { American Leaguer, National Leaguer }

(1) and (2) are individually partition of ML - Baseball - Player class.

Pitcher

isa : ML - Baseball - Player

mutually disjoint with : { Fielder, Catcher }

Catcher

isa : ML - Baseball - Player

mutually disjoint with : { Pitcher, Fielder }

Fielder

isa : ML - Baseball - Player

mutually disjoint with : { Pitcher, Catcher }

American Leaguer

isa : ML-Baseball-Player

mutually disjoint with : {National-Leaguer}

National Leaguer

isa : ML-Baseball-Player

mutually disjoint with : {American Leaguer}

Three-Finger-Brown

isa : Pitcher

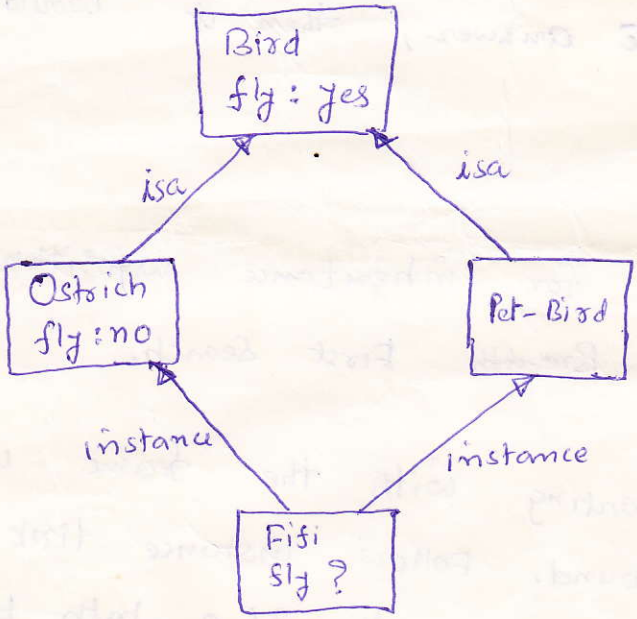
instance : National-Leaguer

instance :

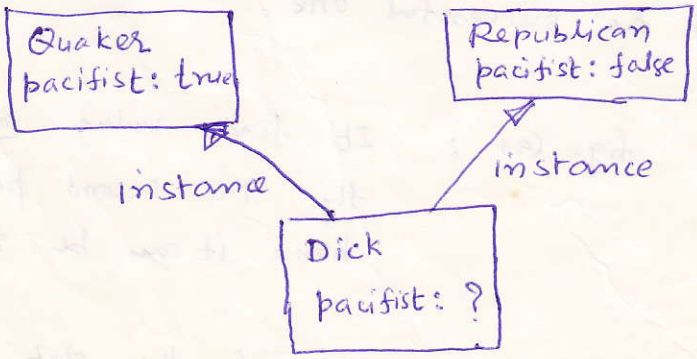
Inheritance Algo. Revisited (given in ch. 4)

"isa" hierarchy could be ~~any~~ arbitrary directed acyclic graph (DAG). Hierarchies that are not tre~~e~~ are called tangled hierarchies.

- Discussion about algorithm for inheriting values for single-valued slots in a tangled hierarchy.



(a)



(b)

Tangled Hierarchies

In Figure (a), we want to decide whether Fifi can fly. The correct answer is no. If we take Path (2), we get answer as "Fifi can fly", however if we take Path (1), we get answer as "Fifi can't". So, we need to think of an algorithm for traversing 'isa' and 'instance' hierarchy that guarantees that specific knowledge will always dominate more general facts.

In Figure (b), we would like to know - Is Dick pacifist? There are two answers, and they conflict with each other. If an algorithm finds one of the answers randomly, without looking for alternate answer, then it wouldn't notice ambiguity present.

So, possible basis for inheritance algorithm is path length while executing Breadth First Search.

Algo. A : DO BFS, starting with the frame whose slot value is to be found. Follow instance link and then follow 'isa' link upward. If a path produces a value, then it can be terminated, as can all paths once their length exceeds that of the successful one.

Result of this Algo. for fig. (a) : It finds value no for the slot fly. The second path has length 2, hence it can be terminated.

fig. (b) : Value for the slot pacifist is true from path 1.
Value for the slot pacifist is false from path 2. Both have same

Tangled Hierarchies

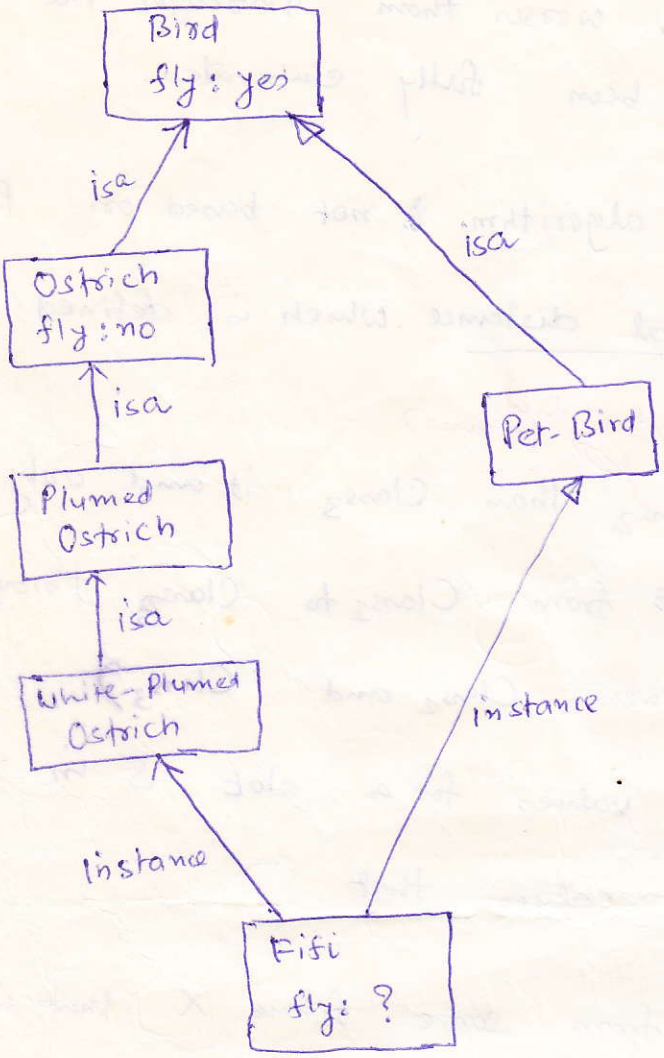


fig. (c)

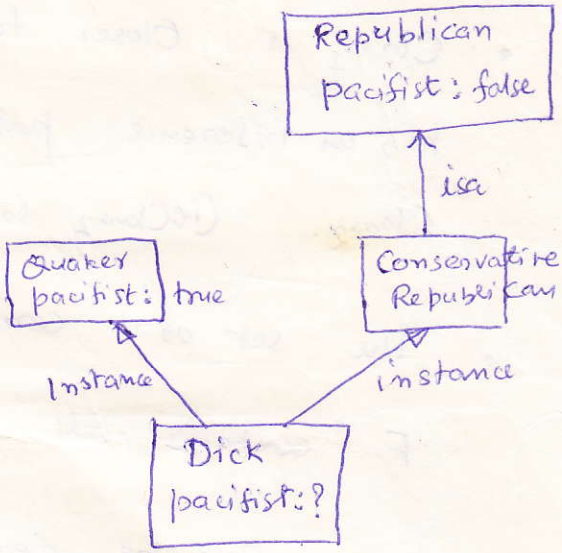


fig. (d)

Consider fig. (c) – Path through Pet-Bird is shorter, and because it is BFS, our algo. would reach Bird before it reaches Ostrich. So Fifi can fly, which is not true.

fig. (d) – From frame Quaker, it finds value for slot pacifist to be true. Now it can't consider other path, as it is lengthier. So, contradiction is not found.

Problem: path through the region which has been elaborated looks worser than through the region which have not been fully elaborated.

So, let us think of an algorithm not based on Path but based on inferential distance which is defined as

- $Class_1$ is closer (shorter inferential Distance) to $Class_2$ than $Class_3$ if and only if has an inference path from $Class_1$ to $Class_3$ through $Class_2$. (i.e. $Class_2$ is in between $Class_1$ and $Class_3$)
- The set of competing values for a slot S in a f
  F contains all those values that
  - Can be derived from some frame X that is ab
    F in the isa hierarchy.
  - Are not contradicted by some frame Y that a shorter inferential distance to F than X d

For fig.(a) we have two candidate classes from which to get answ
But Ostrich has shorter inferential distance to Fifi than Bi
does. So single answer no for the slot fly of Fifi fo

fig.(b) We get two answers and neither class is closer to Dick than other and hence contradiction is identifi

fig.(c) Class Ostrich has shorter inferential distance than cla
Bird and hence only answer is no.

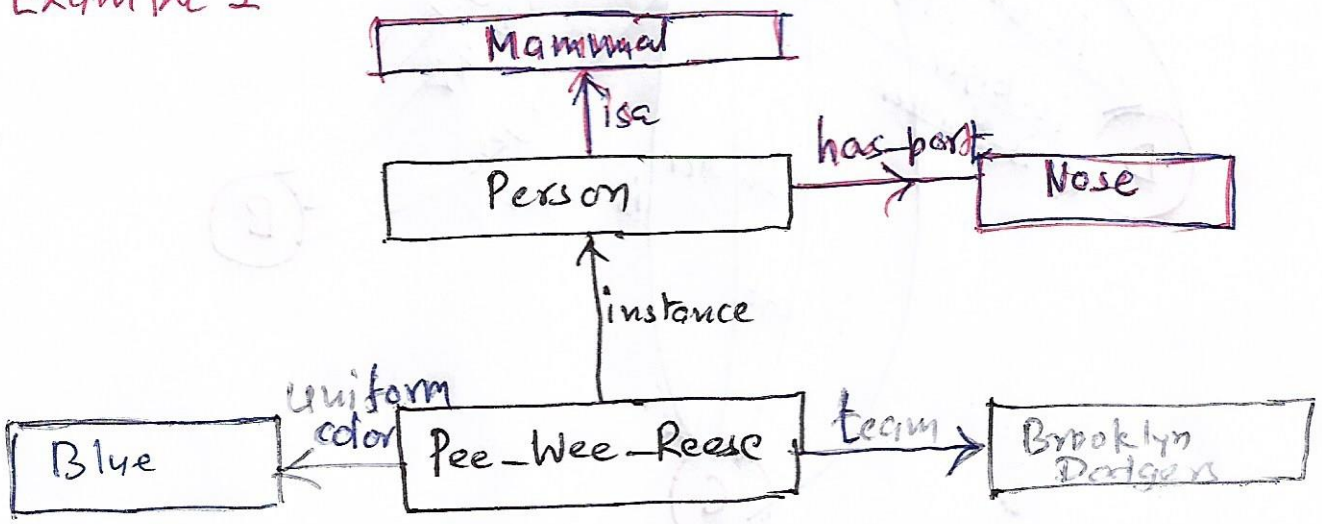
The 'Inferential Distance' is defined in such a manner that as long as 'Ostrich is a subclass of Bird', it will be closer to all its instances, no matter how many other classes are added to the system.

Algo. Property Inheritance - Refer book, ch. 9 Weak slot and filler structures.

Semantic Nets

- The main idea behind Semantic Nets is that the meaning of a concept comes from the ways it is connected to other concepts.
- In a Semantic Net, information is represented as a set of nodes connected to each other by a set of labelled arcs, which represent relationship among nodes.

Example 1



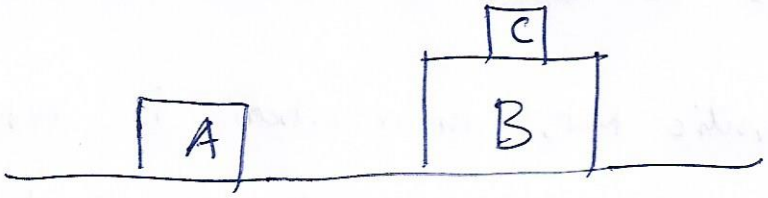
Relations : isa, instance (General Relations)

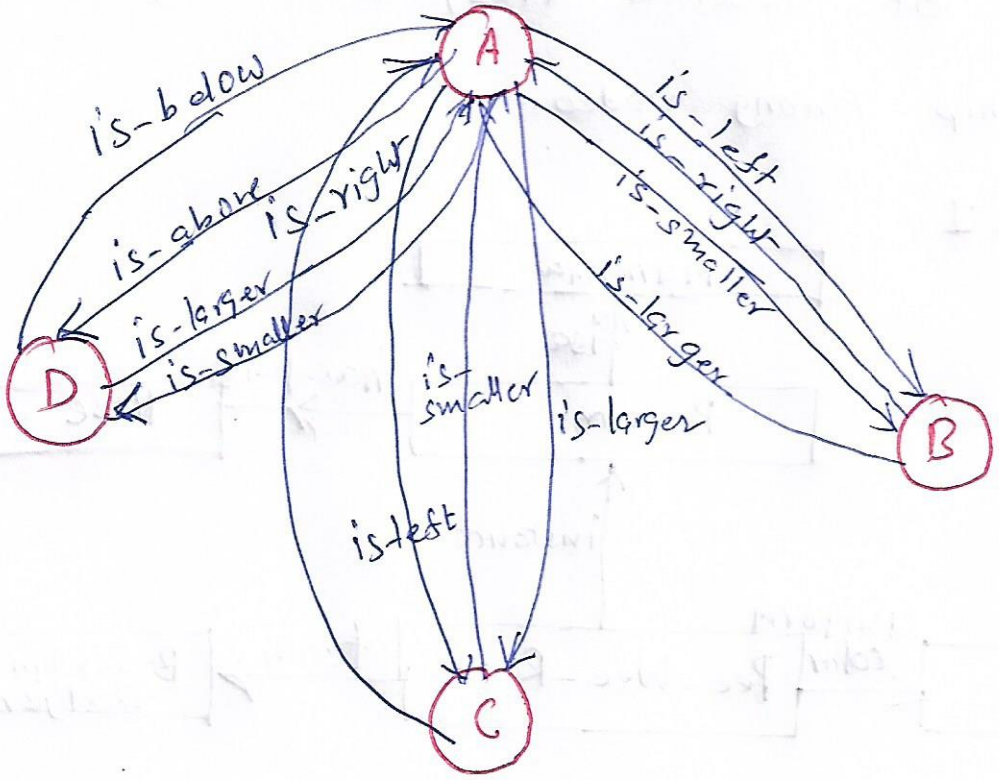
uniform-color, has-part, team
(Domain Specific Relations)

Additional relation can be derived through inheritance. eg. has-part (Pee_Wee_Reese, Nose)

Example 2

Semantic Net Representation Of "Block World" system





Here, samples relationship is given among A-B, A-C, A-D.

We need to construct arcs betⁿ B-C, B-D and C-D

- A Semantic Net is a knowledge representation technique
- Mathematically, it can be defined as Labeled Directed Graph.
- Semantic Nets are also called Associative Nets / Propositional Net.
- Semantic Nets allow Multiple Inheritance. So, an object can belong to more than one category and a category can be a subset of more than one category.

Advantages

- Semantic Nets have the ability to represent default values for categories. This may be overridden by specific values.
- Meaning is transparently conveyed.
- Simple and easy to understand
- Can be easily translated into PROLOG

Disadvantages

- No standard definition for link names.

Representation of relationship (ares) of Semantic Nets in Logic

Binary Predicates

isa (person, mammal)

instance (Pee Wee Reese, Person)

team (Pee Wee Reese, Brooklyn-Dodgers)

- Predicates with non binary arity can also be represented in Semantic Nets, using general purpose predicates such as "instance".

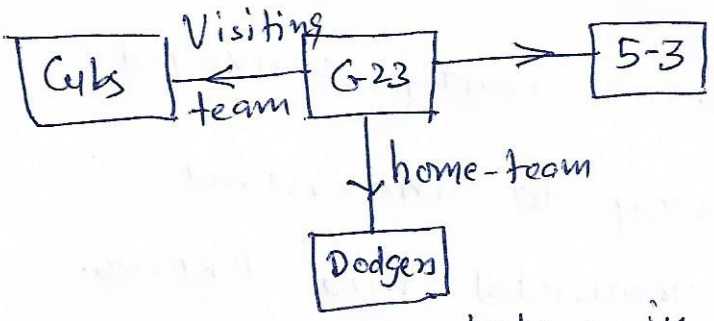
Examples

1. man (Marcus) ← a unary predicate

may be represented as instance (Marcus, man)

2. Score (Cubs, Dodgers, 5-3)

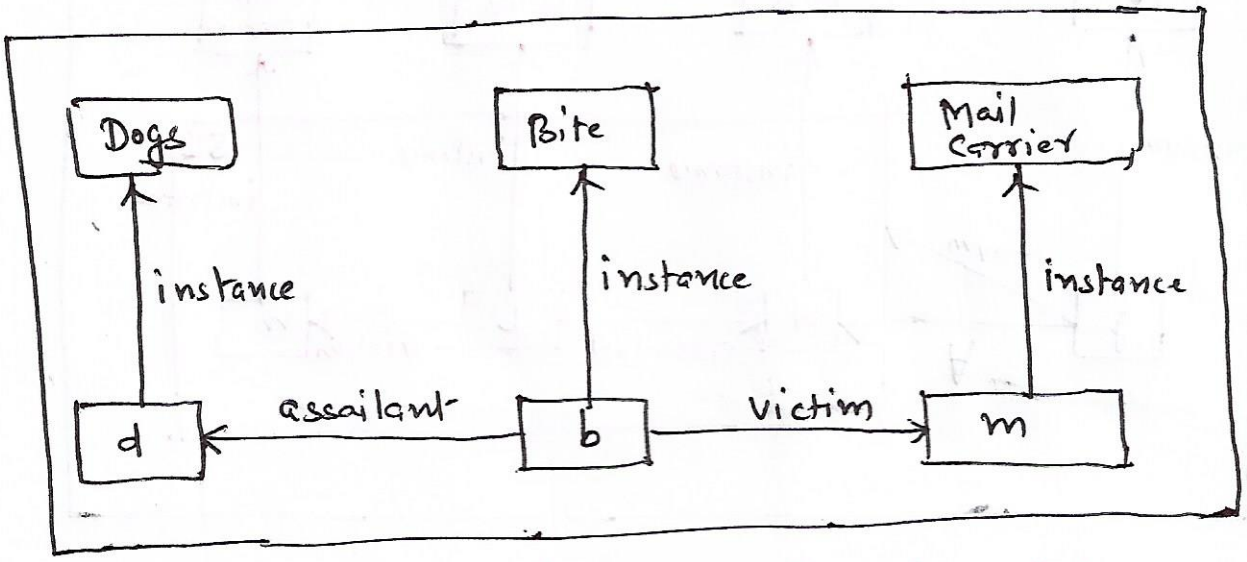
Could be represented as:



3. Sentence Representation in a Semantic Net

consider the statements

* The dog bit the mail carrier.

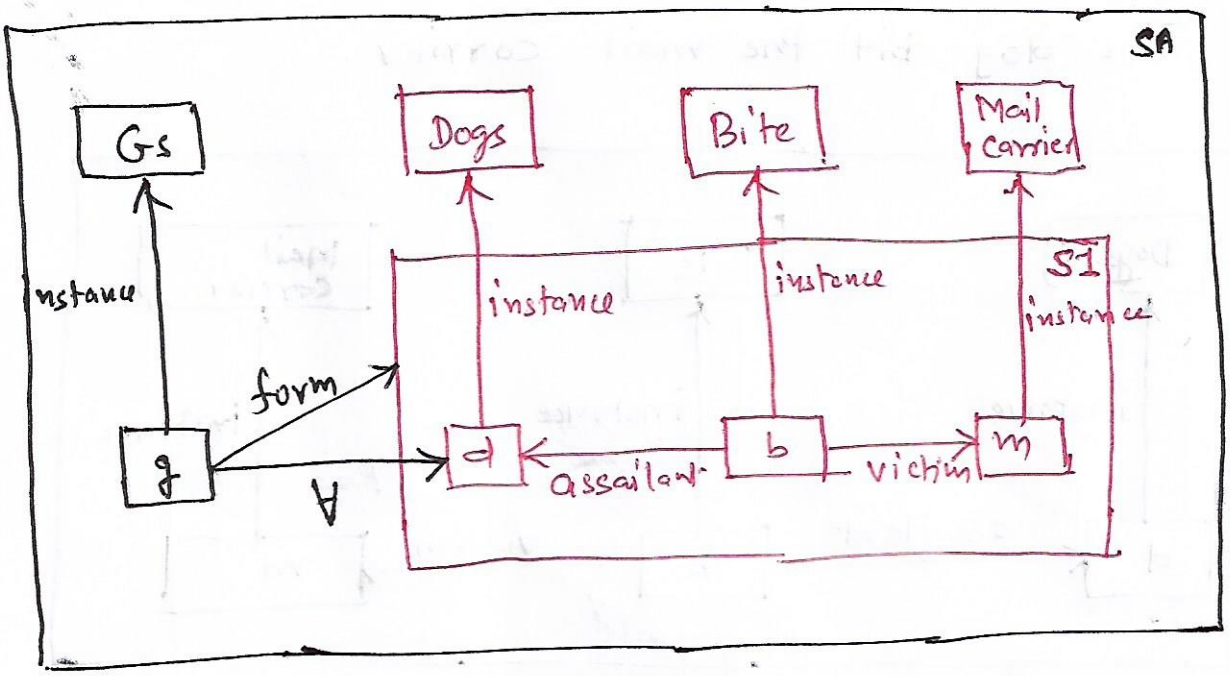


No partition of Semantic Nets

Here, Dogs, Bite and Mail-Carrier are classes of dogs, bitings and mail-carriers respectively. While, nodes d, b and m represent a particular dog, a particular biting and a particular mail carrier.

* Every dog has bitten a Mail Carrier

$$\forall x : dog(x) \longrightarrow \exists y : Mail\_Carrier(y) \wedge Bite(x, y)$$



Gs : General Statement

Every elements of Gs has atleast two attributes: a form, which states the relation that is being asserted, and one or more ∀ connections, one for each of the universally quantified variable.

Every dog in town has bitten the constable

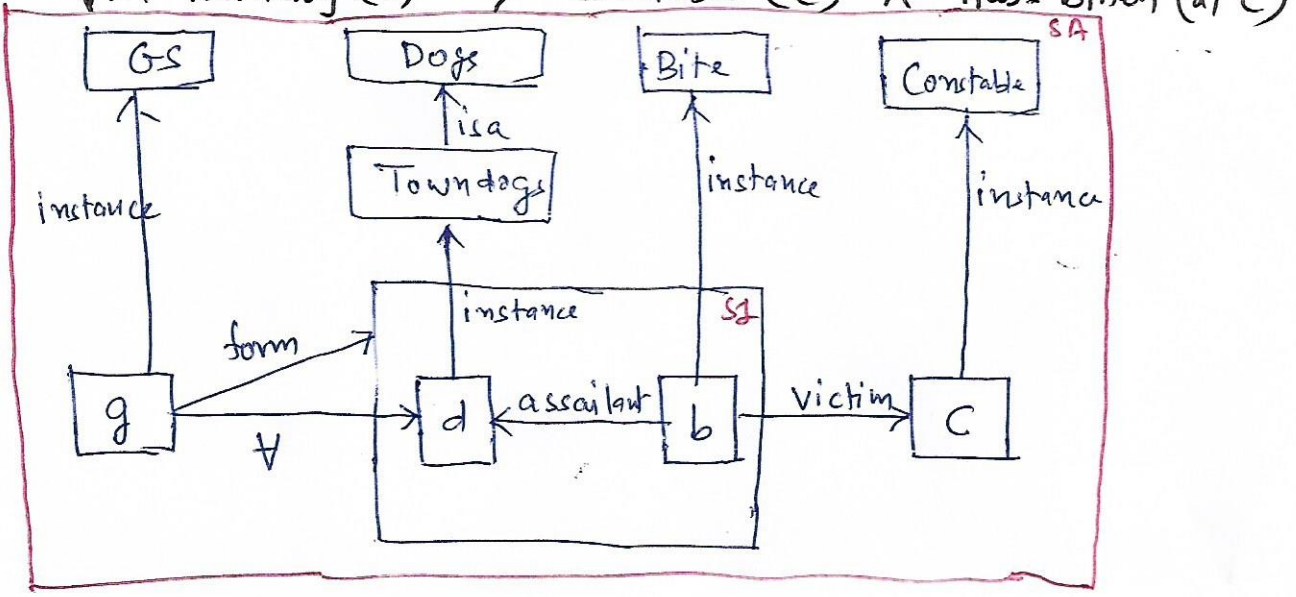
{ In this sentence, we are talking of all dogs of all towns, whereas in the previous one is applicable to all dogs — both from towns, villages etc.
}

FOPL - Sentence I

$$\forall x: \text{town}(x) \rightarrow ((\forall y: \text{dog}(y) \wedge \text{lives}(y, x)) \rightarrow \text{constable}(c) \wedge \text{has\_bitten}(y, c))$$

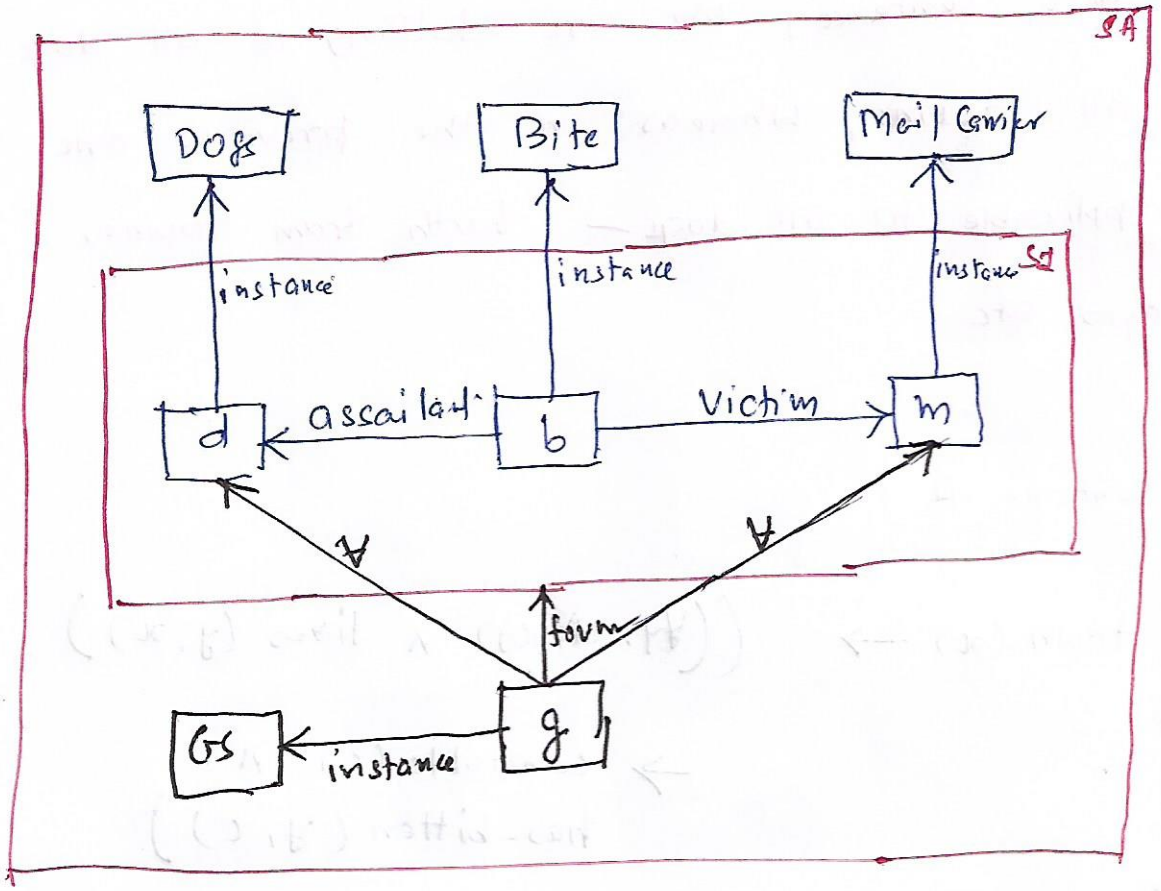
FOPL - Sentence II

$$\forall x: \text{towndog}(x) \rightarrow \text{constable}(c) \wedge \text{has\_bitten}(x, c)$$



- Here, c lies outside the form of General Statement. Therefore, not existentially quantified. Here, it stands for a particular constable.

- Every dog has bitten every mail carrier



A wff can be:

(i) Valid (Tautology): It is true under all interpretations.

eg. $P \vee \neg P$

Above formula is always true irrespective of the interpretation. Hence it is Tautology.

(ii) Invalid: Formula is False under atleast one interpretation

(iii) Contradiction: " " " " all interpretations.

(iv) Satisfiable: Formula is true under atleast one interpretation

(v) Not Satisfiable; " " False under all interpretations.

Valid → Satisfiable

¬ Satisfiable → Invalid

¬ satisfiable ↔ Contradiction

An interpretation is a model of wff it is true under that interpretation.

If it is false, then it is a counter-model for that wff.

- For Tautologies, all interpretations are <u>models</u>.
- A formula is Valid if its negation is a Contradiction (Used in Resolution)

Examples

1. $\forall x : P(x)$

$P(x)$ is relation x is odd.

$D = \{0, 1, 2, \ldots\}$

The formule is false under above interpretation.

(∵ All numbers are not odd)

2 a. $\forall x : \exists y : P(x, y)$

b. $\exists y : \forall x : P(x, y)$

(i) $D = \{0, 1, 2\}$

and $P(x, y)$ is $x \leq y$

2a. True [∵ $x = 0$ $y = 0$ or 1 or 2

$x = 1$ $y = 1$ or 2

$x = 2$ $y = 2$]

2b. True [∵ $y = 2 \geq 0, 1, 2$]

(ii) $D = \{\ldots, -2, -1, 0, 1, 2, 3, \ldots\}$

$P(x, y)$ is $x \leq y$

2a. True

2b. False

3. $\forall x : P(x, f(x))$

Interpretations

(i) Domain $D = \{0, 1, 2, \ldots\}$

Function : $f(x) = x + 1$

Predication : $P(x, y)$ is True iff $x \leq y$

True ($\because$ y is always $x+1$)

(ii) Domain $D = \{a, f(a), f(f(a)), \ldots\}$

$f^1(a)$ $f^2(a)$ $f^3(a)$ etc. when f^4 is applied to it

Function : $\begin{cases} a = f^0(a) \ldots \text{added afterwards} \\ f(a) = f^1(a) \\ f(f^n(a)) = f^{n+1}(a) \end{cases}$ $\forall x : P(x, f(x))$

Predicate $P(x, y)$ is True iff

$x = f^n(a)$, $y = f^m(a)$ and

$n \leq m$

True.

4. $(P(a) \wedge (\forall x : \underbrace{P(x) \rightarrow P(S(x))}_{I})) \rightarrow \forall x : P(x)$

Interpretations

(i) $D = \{0, 1, 2, \ldots\}$

$a = \phi$

$S \rightarrow$ Successor Function $S(x) = x + 1$

$P \rightarrow$ Any property

$\forall x$: $P(x)$ is False for at least one x when we look for True → False. At that time we have True → False is Part I.

Therefore, $P(x) \rightarrow P(S(x))$ is not true for all x.

∴ False → * is True.

(LHS → RHS should be True → False for it to be False. This is not possible for any property, because $p(0)$ must be True if we want to look for True → False)

Infact, this is the Principle of Mathematical Induction. (with given interpretation, i.e. successor function. property can be any property.)

(ii) $P(x)$ is True if x is odd

So, $P(0)$ is False.

LHS of '→' is False.

False → '*' is True

∴ True

(iii) $D = \{0, 1, 2, \ldots\}$

$a = \phi$

$S(x) = x + 1$

$P \longmapsto x = \phi$

True

Here, $P(\phi)$ is True (∵ Predicate is $x = \phi$)

∵ $\forall x$: $P(x) \rightarrow P(S(x))$ is False.

(∵ $P(0) \rightarrow P(1)$ is True → False which is False)

∴ LHS is False

and False → '*' is True

∴ Wff is True under the given interpretation.

Interpretation (iv)

Domain

$D_1 = \{O_1$

Interpretation (iv)

$D_1 = \{0, 1, 2, \ldots\}$

$D_2 = \{0', 1', 2', \ldots\}$ $\qquad D = D_1 \cup D_2$

$a \mapsto \emptyset$

$$S(x) = \begin{cases} x+1 & \text{if } x \in D_1 \\ (x+1)' & \text{if } x \in D_2 \end{cases}$$

$$P(x) = \begin{cases} \text{True} & \text{for any } x \in D_1 \\ \text{False} & \text{for any } x \in D_2 \end{cases}$$

LHS

Case I $\quad P(0)$ is True

$P(1), P(2), P(3) \ldots$ are also True.

as $P(x)$ is True for any $x \in D_1$.

$\therefore P(S(x)) = P(x+1)$ is True for $x \in D_1$

$\therefore T \to T$ is True for $x \in D_1$ (1)

Case II $\quad$ Similarly for elements of D_2, we have

$F \to F$ as $P(0'), P(1') \ldots$ etc are False.

But $F \to F$ is True

$\therefore$ LHS is either $T \to T$ or $F \to F$ which is True.

But RHS is False. $\therefore$ True $\to$ False is <u>False</u>

Thus, above interpretation is a Counter-Model for given Wff.

Are the following models or Counter-models for given Wff

1. $P(a) \wedge (\forall x: P(x) \rightarrow P(S(x)))$

(i) Domain $D = \{0, 1\}$

$a \mapsto \emptyset$

$I^{\mathcal{U}}$: $S(0) = 1$, $S(1) = 0$

Predicates : $P(0)$ & $P(1)$ are False

Counter-model as $P(0)$ is False.

(ii) Domain D = Rational Numbers

$a \longmapsto 0$

$I^{\mathcal{U}}$: $S(x) = x + 1$

$P(I)$ iff I is an integer

Sol^{n}/ $P(0)$ is True.

$\forall x: P(x) \rightarrow P(S(x))$ will be

either True $\rightarrow$ True

or False $\rightarrow$ False

$\therefore$ True.

$\therefore$ Model

List of Valid Formula

1. $\forall x: P(x) \rightarrow \exists x: P(x)$

2. $\forall x: P(x) \rightarrow P(a)$

3. $\exists y: \forall x: P(x,y) \rightarrow \forall x: \exists y: P(x,y)$

4. $\exists x: P(x,x) \rightarrow \exists x: \exists y: P(x,y)$

5. $(\forall x: P(x) \vee \forall x: Q(x))$

$\rightarrow \forall x: P(x) \vee Q(x)$

6. $\exists x: (P(x) \wedge Q(x)) \rightarrow$

$\exists x: P(x) \wedge \exists x: Q(x)$

7. $(\exists x: P(x) \rightarrow \forall x: Q(x))$

$\rightarrow \forall x: (P(x) \rightarrow Q(x))$

{ Here, scope of each quantifier is not the entire expression but it is in that part.

eg. $\exists y: \forall x: P(x,y) \rightarrow \forall x: \exists y: P(x,y)$

x & y before and after Implication are different.

Example of Invalid Formula

Just Reverse Expression on Left & Right in wff.

eg. 1. $\exists x : P(x) \rightarrow \forall x : P(x)$

2 $P(a) \rightarrow \forall x : P(x)$

etc.

Invalidity is shown by Counter-Example

(1) $D = \{0, 1\}$

$P(1)$ is True

$P(0)$ is False.

$\exists x : P(x)$ is True for $x = 1$

But, $\forall x : P(x)$ is False.

∴ True → False is False.

Interpretation is a Counter-Model

∴ Formula is Invalid,

(2) $D = \{0, 1\}$

$a = \phi$

$P(\phi)$ is True

$P(1)$ is False

∴ $P(a)$ is True

$\forall x : P(x)$ is False

True → False is False.

(7) Counter-Model

$D = \{2, 4, 6, 8\}$

$P(x)$ is x divisible by 4

$Q_2(x)$ is x divisible by 2.

(3) $D = \{0, 1\}$

$P(x, y)$ is True iff $x = y$

$P(0,0)$ is True

$P(1,1)$ is True

$\forall x: \exists y: P(x,y)$ is True

But, $\exists y: \forall x: P(x,y)$ is False.

(∵ Take $y=0$, Then if $x=0$, $P(0,0)$ is True
$x=1$, $P(1,0)$ is False
$\forall x: P(x,y)$ is False.

$\Rightarrow \exists y: \forall x: P(x,y)$ is False.

Thus, True $\rightarrow$ False is False.

∴ Counter-Model.

(4) Let $D = \{0, 1\}$

$P(x,y)$ is True iff $x \neq y$

$P(0,0)$ is False
$P(0,1)$ is True ⌙ LHS is True
$P(1,0)$ is True ⌙
$P(1,1)$ is False

True → False is False.
∴ Counter-Model.