Ch: 9 Weak Slot and filler Structures

⇒ Slot & filler structures are the devices to support property inheritance along isa & instance links. (Knowledge Representation Technique)

* "is a" "instance" rel. fig in previous (Inheritance).

Slot and filler structures

- Weak slot & filler structure
  - Semantic nets
  - frames
- Strong slot & filler structure
  - Conceptual Dependency
  - Scripts

* **Weak slot & filler structures:-** are "knowledge-poor" or "weak" as very little importance is given to the specific knowledge.

Slot = Attribute
filler = Value of Attributes.

⇒ **Advantage of slot & filler structure:-**

1. Monotonic reasoning can be performed more effectively. ~~there with more~~
2. Makes it easy to describe properties of relations.
3. form of object oriented programming and has advantages such as modularity & easy of viewing by people.

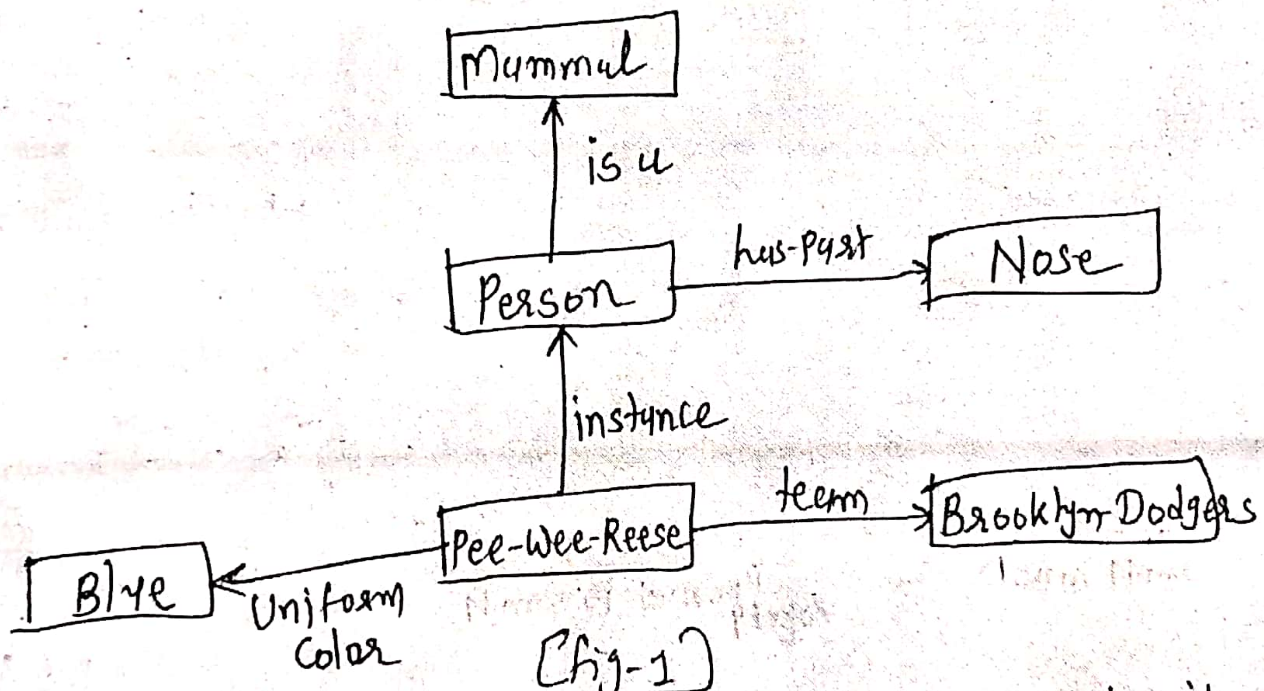
Scanned by CamScanner
Scanned by CamScanner

9.1 * Semantic Nets :-

↳ In Semantic Nets information is represented as : - set of nodes connected to each other by a set of labelled arcs.

- Nodes Represents : Various objects / values of the attributes of object.

- Arcs represent : Relationships among nodes.



[fig-1]

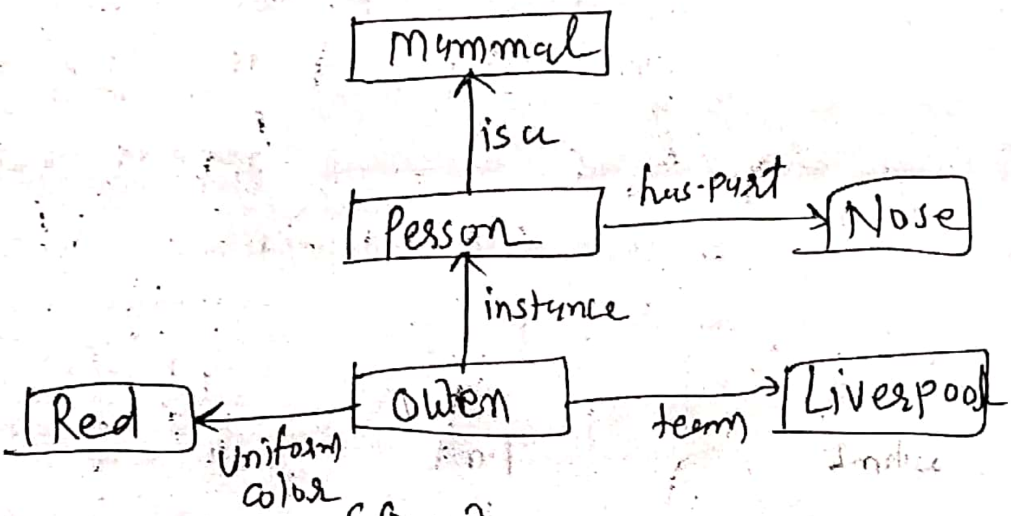
↳ In this network we could use inheritance to derive the additional info :

$\boxed{\text{has\_part (Pee-Wee-Reese, Nose)}}$

Intersection Search

↳ one way to find relationships among objects is to spread the activation (links) out from two nodes & find out where it meets.

Ex:1 relation between Red and liverpool.



[fig-2]

Ex. 2

↳ In fig-1 "What is the connection between the Brooklyn Dodgers & blue?"

9.1.2 ⟹ Representing Nonbinary Predicates:

↳ Semantic nets are a natural way to represent relationships that would appear as ground instances of binary predicates in predicate logic.

for Ex:, Some of the arcs from fig. 9.1 could be represented in logic as:

Scanned by CamScanner

Scanned by CamScanner

isa (Person, Mammal)
instance (Pee-wee-Reese, Person)
team (Pee-wee-Reese, Brooklyn-Dodgers)
uniform-color (Pee-wee-Reese, Blue)

↳ But the knowledge expressed by predicates of other arities can also be expressed in Semantic nets.

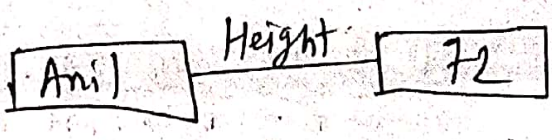
Ex:

Man(marcus)

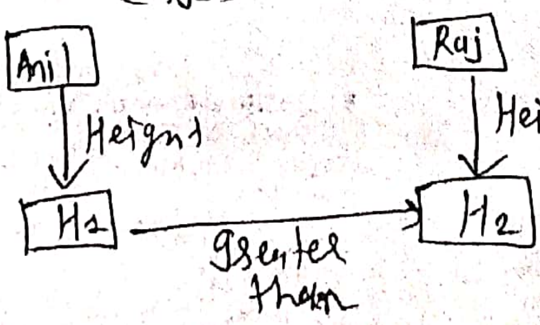
Could be rewritten as.

in semantic net } ⇒ instance(marcus, Man)

9.1.3 Making Some Important Distinctions:



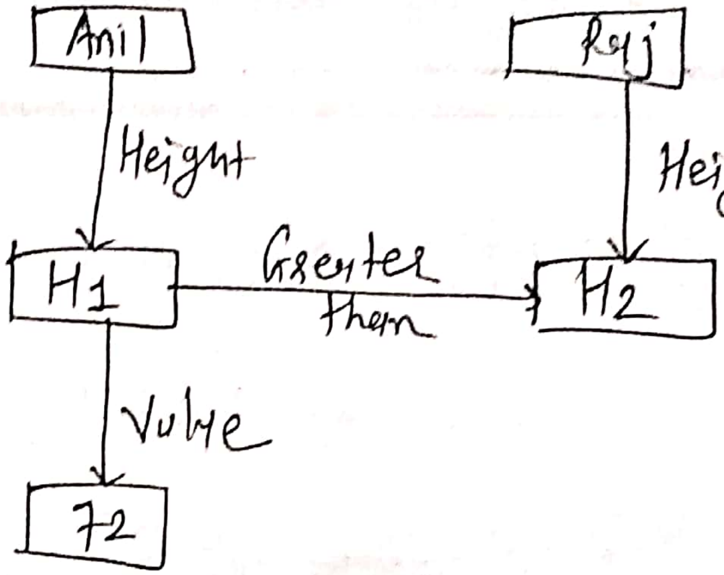
(fig3)-Semantic Network



(fig-4) Semantic Network

Scanned by CamScanner

Scanned by CamScanner



[fig-5- Representing distinction in semantic network]

↳ There should be a difference between the link that defined a new entity & one that relates two existing entities:

↳ Suppose we want to represent the fact like "Anil" is taller than "Raj."

↳ The nodes H_1 and H_2 represent Anil's and Raj's Height respectively.

→ Sometimes it is useful to introduce the arc value to make the distinction clear. [In fig-5]

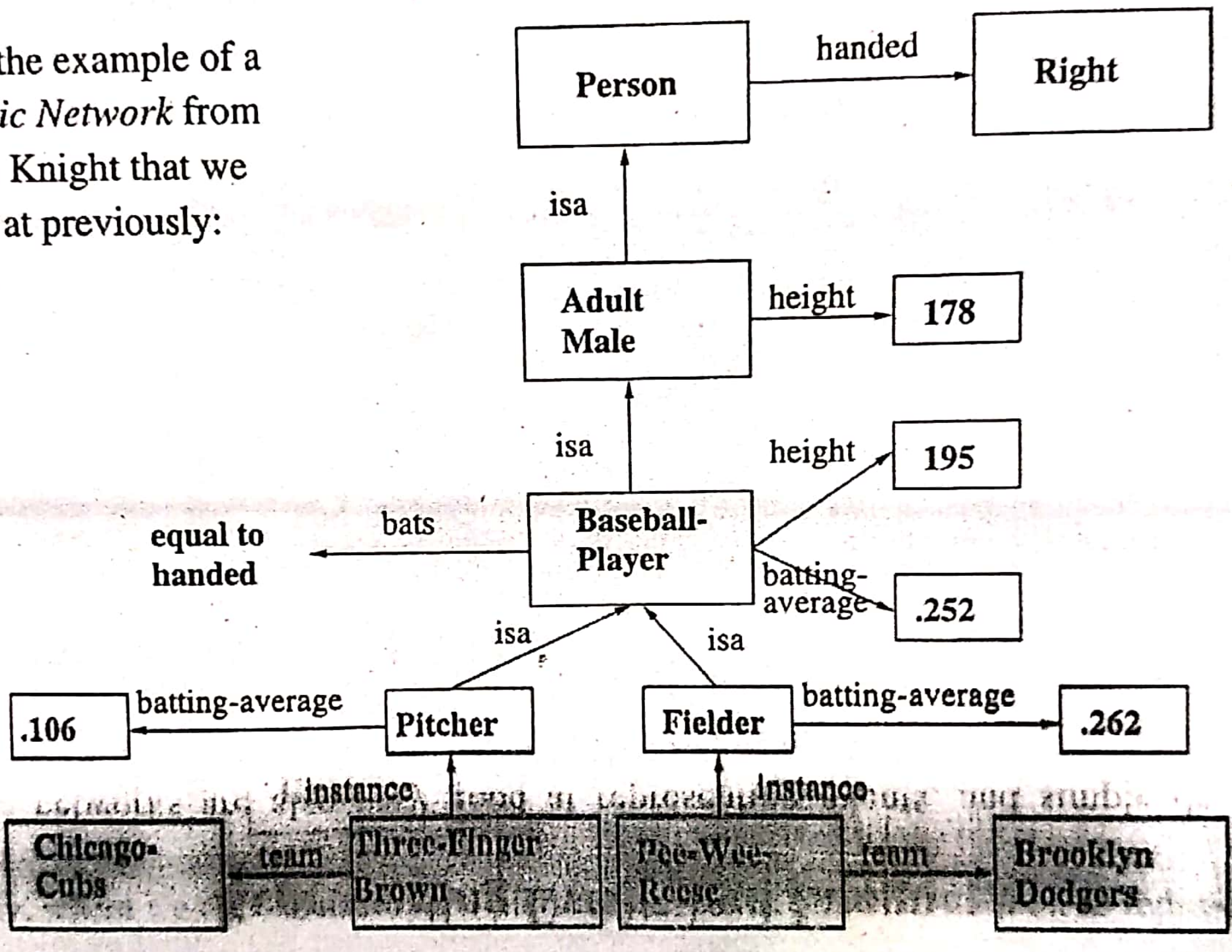
↳ Some arcs ~~such~~ such as height define New entities. ~~and~~ whereas some arcs like greater than and value describes the relation among existing entities.

Scanned by CamScanner

Scanned by CamScanner

A Typical Mixed-Type Semantic Network

Here's the example of a *Semantic Network* from Rich & Knight that we looked at previously:



Scanned by CamScanner

(8)

Components of a Semantic Network

We can define a *Semantic Network* by specifying its fundamental components:

Lexical part
- nodes – denoting objects
- links – denoting relations between objects
- labels – denoting particular objects and relations

Structural part
- the links and nodes form directed graphs
- the labels are placed on the links and nodes

Semantic part
- meanings are associated with the link and node labels (the details will depend on the application domain)

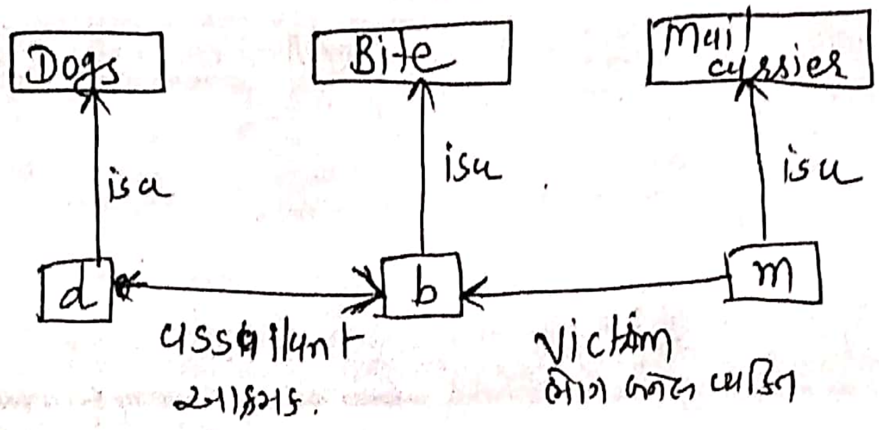
Procedural part
- constructors allow creation of new links and nodes
- destructors allow the deletion of links and nodes
- writers allow the creation and alteration of labels
- readers can extract answers to questions

Clearly we are left with plenty of flexibility in creating these representations.

Scanned by CamScanner

2.4 Partitioned Semantic Nets

↳ Suppose we want to represent simple quantified expressions in semantic nets. One way to do this is to partition the semantic net into a hierarchical set of spaces, each of which corresponds to the scope of one or more variables.



[fig. 6]

⇒ To see how this works, consider ~~first~~ fig 6. This net corresponds to the statement

"The dog bit the mail carrier"

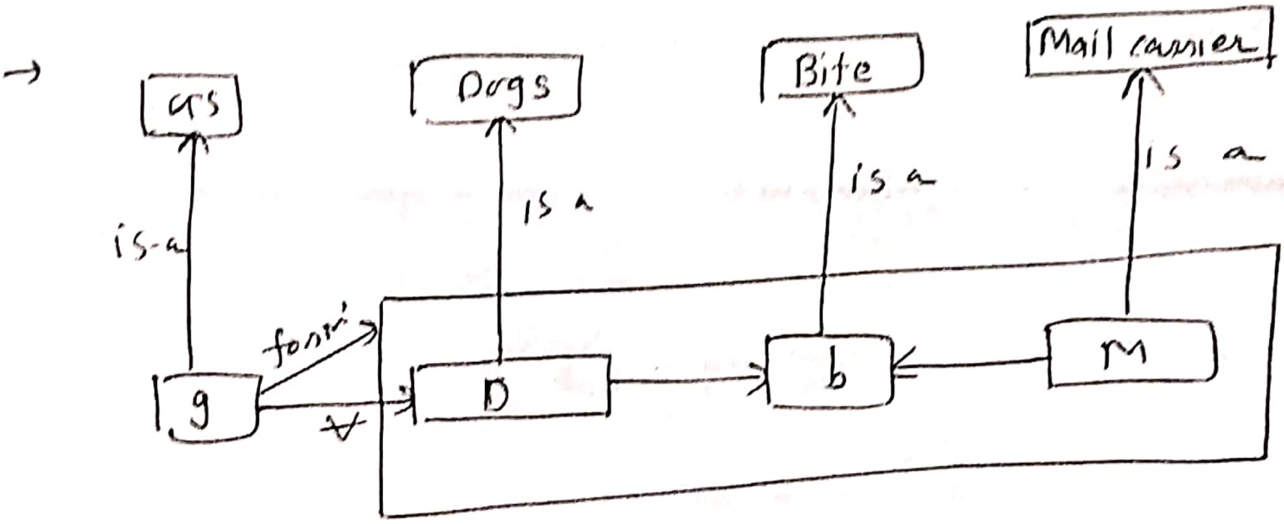
↳ The nodes Dogs, Bite and Mail-carrier represent the classes of dogs, bitings and mailcarriers respectively.

↳ While the nodes d, b and m represent a particular dog, a particular biting and a particular mail-carrier. This fact can easily be represented

Scanned by CamScanner
Scanned by CamScanner

⇒ But now suppose that we want to represent the fact
"Every dog has bitten a mail carrier"
or in logic:

2) Every dog has bitten a mail carrier.



↳ To Represent this fact, it is necessary to encode the scope of the universally quantified variable x.

↳ This can be done using partitioning as shown in fig-7.

↳ Node g is an instance of the special class GS of general statements about the world.

↳ Every element of GS has at least two attributes

(1) a form, which states the relation that is being asserted

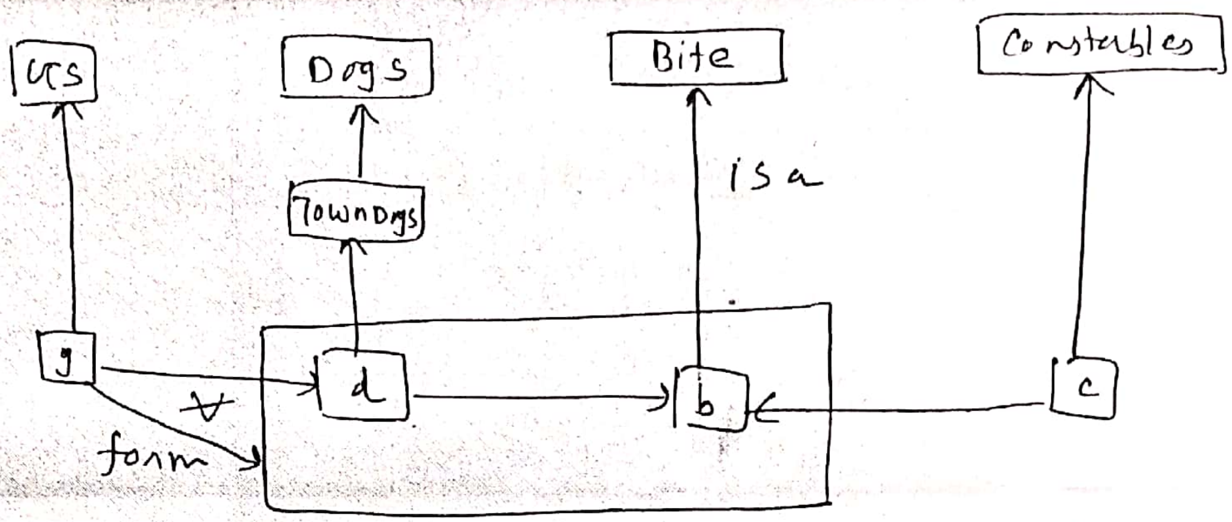
(2) one or more ∀ connections

Scanned by CamScanner

Scanned by CamScanner

To see how partitioning makes variable quantification explicit, consider next the similar sentence:

3) Every Dog in town has bitten the Constables



↳ The representation of this sentence is shown in fig-8.

→ In this net, the node c representing the victim lies outside the form of the general statement.

⇒ figure 9 shows how yet another similar sentence:

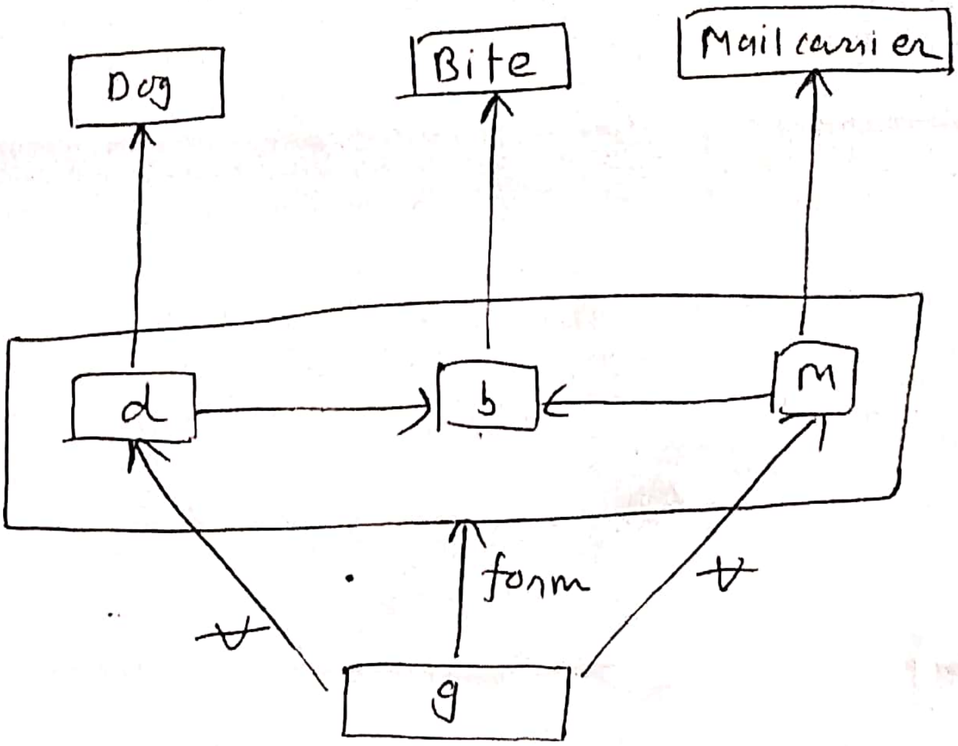
"Every dog has bitten every mail carrier"

↳ In this case g has two ∀ links, one pointing to d, which represents any dog and one pointing to m, representing any mail carriers.

Scanned by CamScanner

Scanned by CamScanner

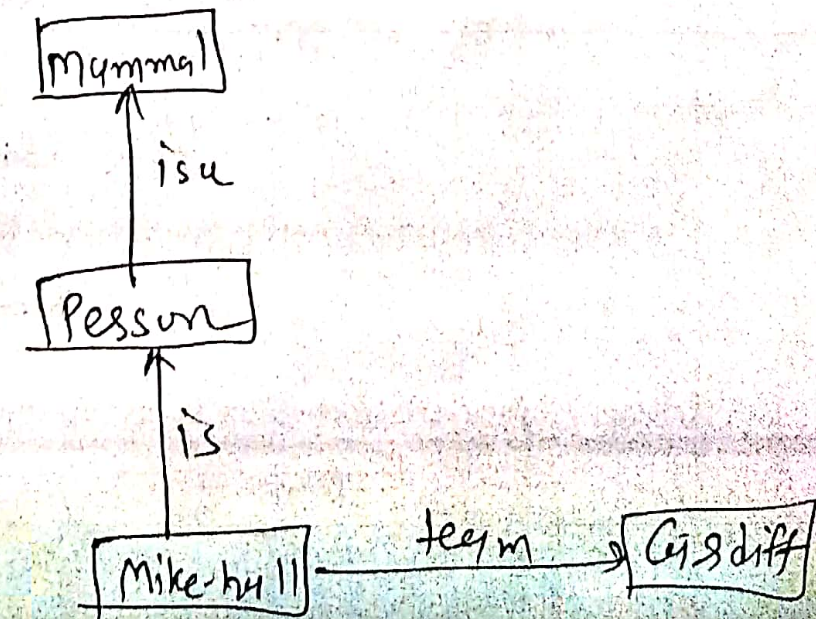
4) Every Dog has bitten every mail carrier.



Example ⇒ Represent following sentences using Semantic networks.

isa (Person, mammal), instance (mike-Hall, Person), team (mike-Hall, Cardiff) all in one graph

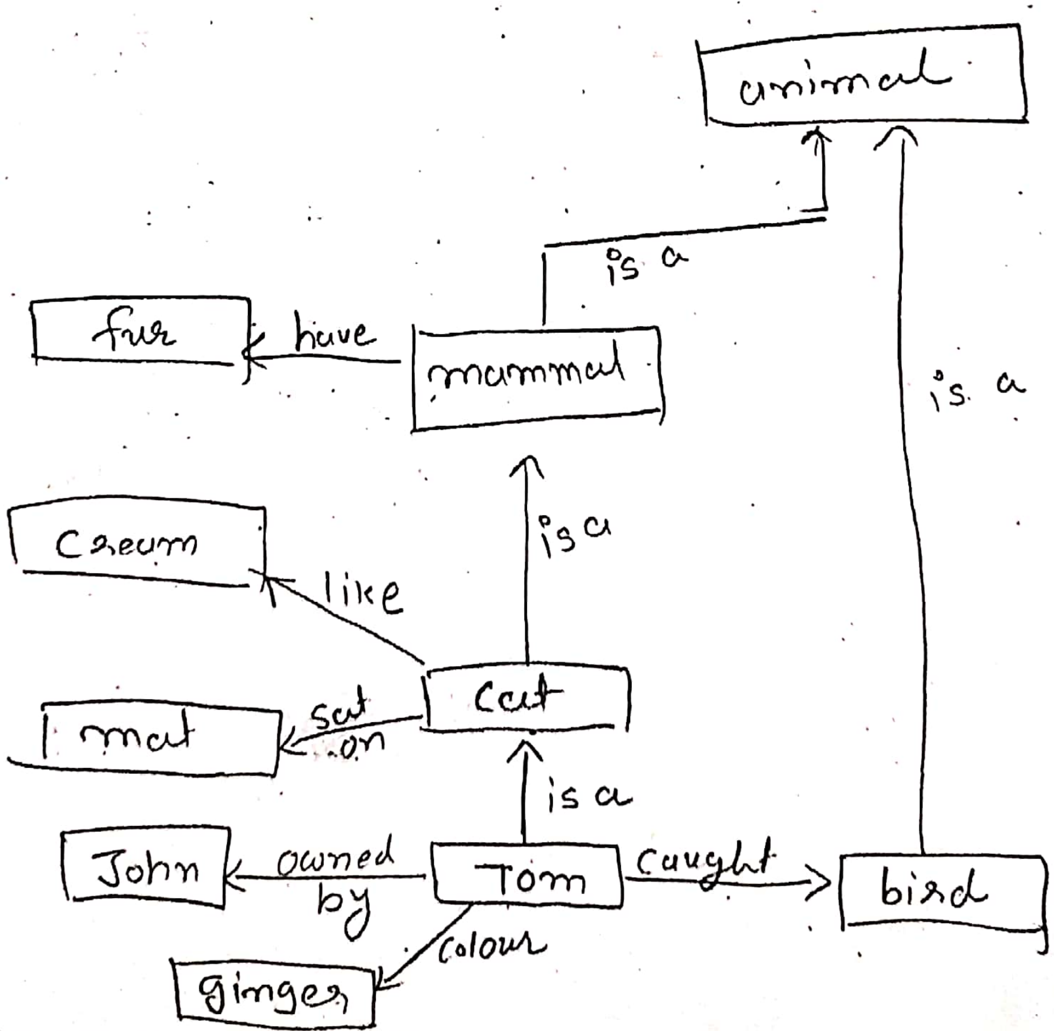
→



Scanned by CamScanner

Que. Draw semantic net of following statements.

- Tom is a cat
- Tom caught a bird
- Tom is owned by John
- Tom is ginger in color
- Cats like cream
- The cat sat on the mat
- A cat is a mammal
- A bird is an animal
- All mammals are Animals
- Mammals have fur

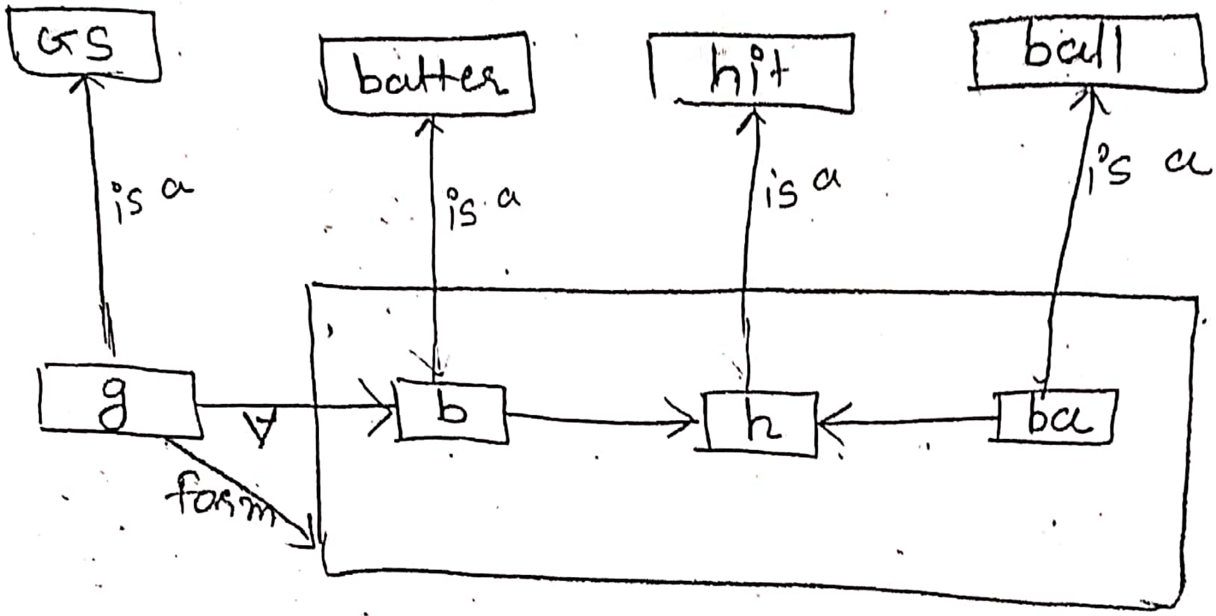


Scanned by CamScanner

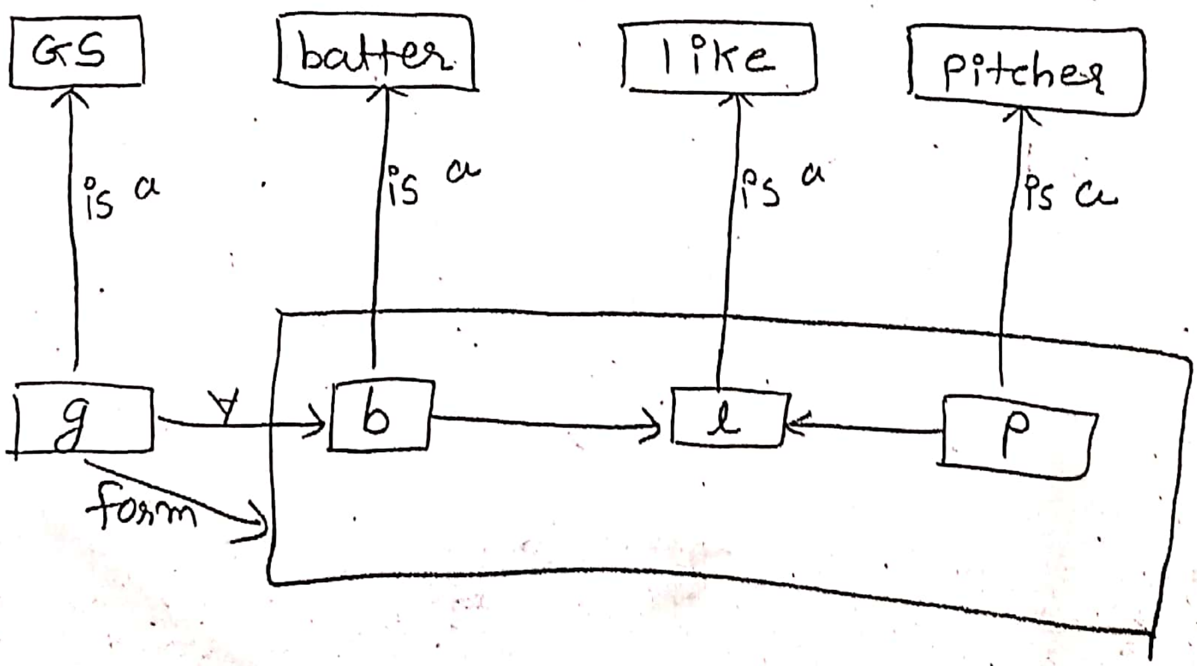
Que

Construct partitioned semantic net for the following!

(1) Every batter hit a ball

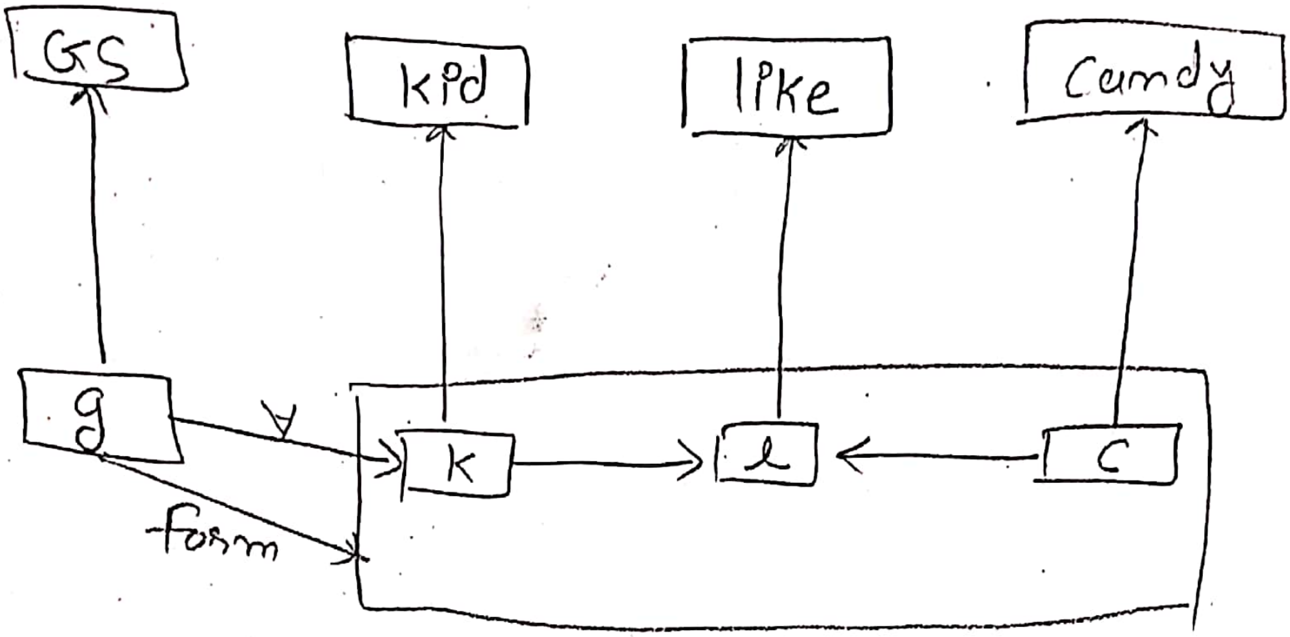


(2) All the batter like a the pitches

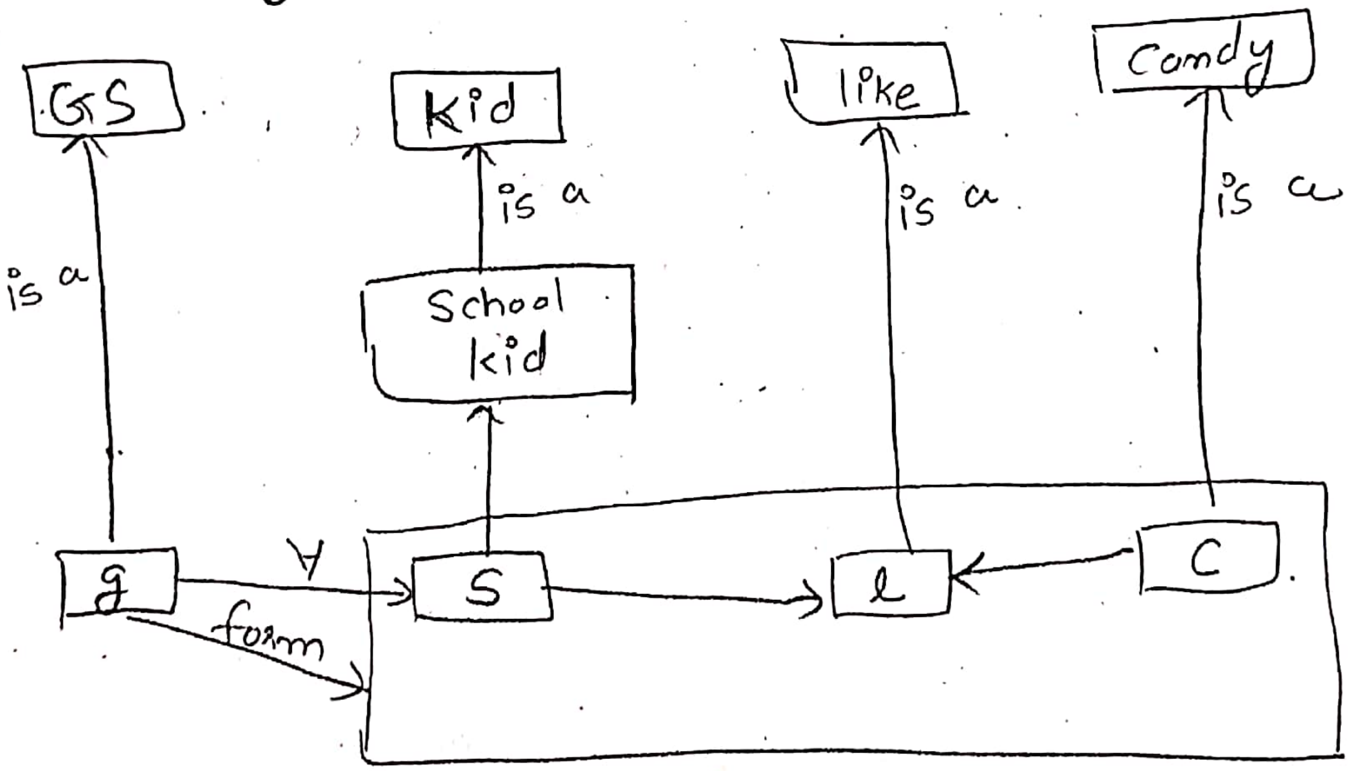


Scanned by CamScanner

3) Every kid likes candy



(4) Every School going kid likes candy



Scanned by CamScanner

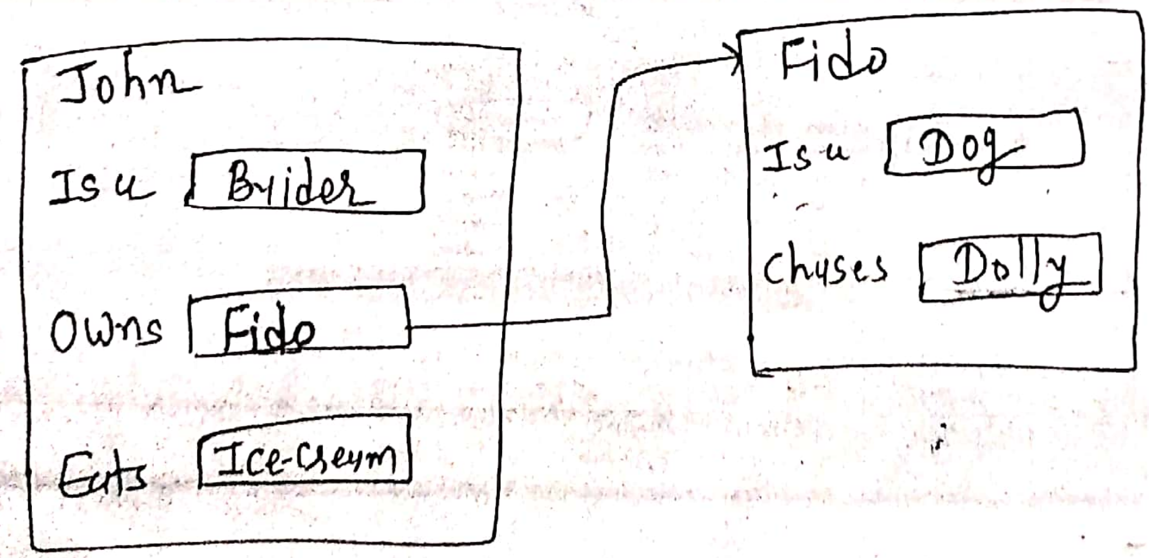
frames

→ frame Consist of number of frames which are Connected by links/edges like Semantic net.

→ for example Consider the following statements:

- John is builder
- John owns fido
- John eats Ice-cream
- Fido is a dog.
- fido chases Dolly.

↳ We Can represent such knowledge using frame Structure, in which John is a name of frame, is a name of slot (Attribute) and builder Value of slot. It is described in below fig



[fig: Frames]

Scanned by CamScanner