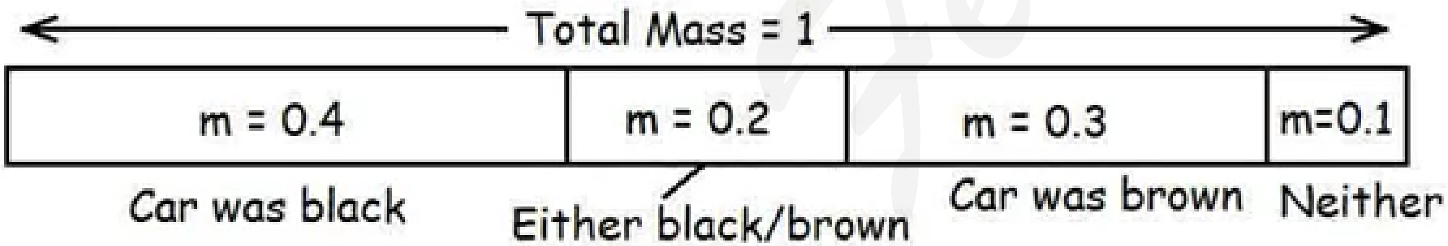
Dempster – Shafer Theory (DST)

Belief and Plausibility

Witness: "I'm fairly sure that the car was either brown or black. Probably black, though it could have been brown. I could be wrong though."



Belief: Bel(Black) = 0.4   Bel(Brown) = 0.3

Bel(not Black) = 0.3 + 0.1 = 0.4   Bel(not Brown) = 0.5

Plausibility: Pl(Black) = 0.4 + 0.2 = 0.6 = 1 - Bel(not Black)

Pl(Brown) = 0.2 + 0.3 = 0.5 = 1 - Bel(not Brown)

In general: $Pl(A) = 1 - Bel(\text{not } A)$

Belief in A is the sum of the mass values that form subsets of A.

The car was black:	0.25
The car was black with chrome trims:	0.05
The car was black with a powerful engine:	0.1
The car was brown:	0.3
The car was either black or brown:	0.2
The car was some other colour:	0.1

Subset of "black car"

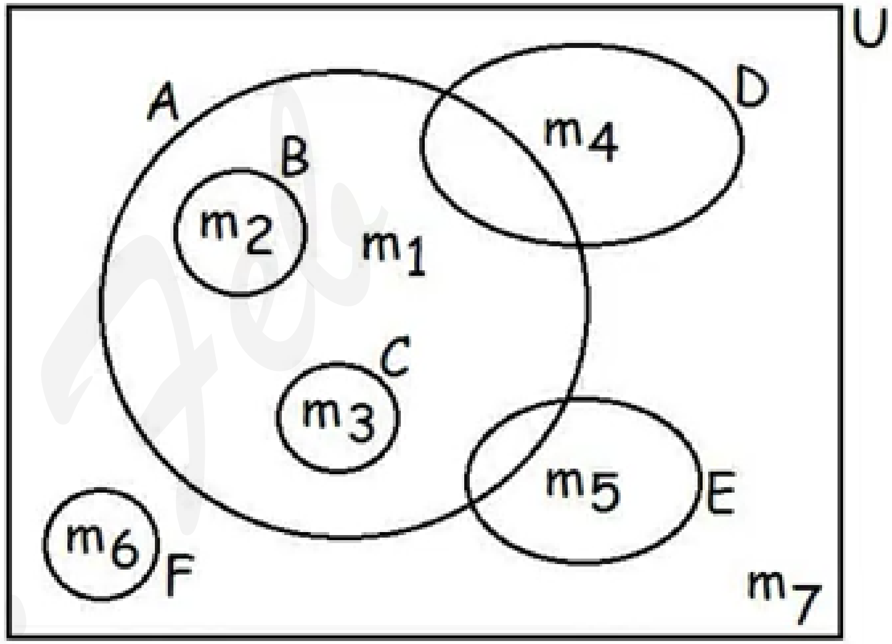
Compatible with black car (plausibility)

$$\mathrm{Bel}(A) = \sum_{X \subseteq A} m_X$$

$$= m_1 + m_2 + m_3$$

$$\mathrm{Pl}(A) = \sum_{X \cap A \neq \emptyset} m_X$$

$$= m_1 + m_2 + m_3 + m_4 + m_5$$



$\mathrm{Pl}(A) \geq \mathrm{Bel}(A)$

All masses add to 1.

Hypothesis	Mass	Belief	Plausibility
Null (neither real nor fake)	0	0	0
Real	0.2	0.2	0.5
Fake	0.5	0.5	0.8
Either (real or fake)	0.3	1.0	1.0

Example of Bayes theorem

Technicians regularly make repairs when breakdowns occur on an automated production line. Janak, who services 20% of the breakdowns, makes an incomplete repair 1 time in 20. Tarun, who services 60% of the breakdowns, makes an incomplete repair 1 time in 10. Gautham, who services 15% of the breakdowns, makes an incomplete repair 1 time in 10 and Prasad, who services 5% of the breakdowns, makes an incomplete repair 1 time in 20. For the next problem with the production line diagnosed as being due to an initial repair that was incomplete, what is the probability that this initial repair was made by Janak?

Repair

- P(Repaired by Janak) = 0.20
  - P(Incomplete|Repaired by Janak) = 1/20 = 0.05
- P(Repaired by Tarun) = 0.60
  - P(Incomplete|Repaired by Tarun) = 1/10 = 0.1
- P(Repaired by Gautham) = 0.15
  - P(Incomplete|Repaired by Gautham) = 1/10 = 0.1
- P(Repaired by Prasad) = 0.05
  - P(Incomplete|Repaired by Prasad) = 1/20 = 0.05

Reverse Probability

P(Janak | Incomplete) = ?

Solution:

P(Janak | Incomplete) = P(Repaired by Janak)*P(Incomplete | Repaired by Janak)/P(Incomplete)

P(Janak | Incomplete) = 0.20*0.05/[0.20*0.05 + 0.60*0.1 + 0.15*0.1 + 0.05*0.05]

Example

SpamAssassin works by having users train the system. It looks for patterns in the words in emails marked as spam by the user. For example, it may have learned that the word “free” appears in 20% of the emails marked as spam. Assuming 0.1% of non-spam mail includes the word “free” and 50% of all emails received by the user is spam, find the probability that a mail is a spam if the word “free” appears in it.

Data Given:

- $P(\text{Free} \mid \text{Spam}) = 0.20$
- $P(\text{Free} \mid \text{Non Spam}) = 0.001$
- $P(\text{Spam}) = 0.50 => P(\text{Non Spam}) = 0.50$
- $P(\text{Spam} \mid \text{Free}) = ?$

Data Given:

- $P(\text{Free} \mid \text{Spam}) = 0.20$
- $P(\text{Free} \mid \text{Non Spam}) = 0.001$
- $P(\text{Spam}) = 0.50 => P(\text{Non Spam}) = 0.50$
- $P(\text{Spam} \mid \text{Free}) = ?$

Data Given:

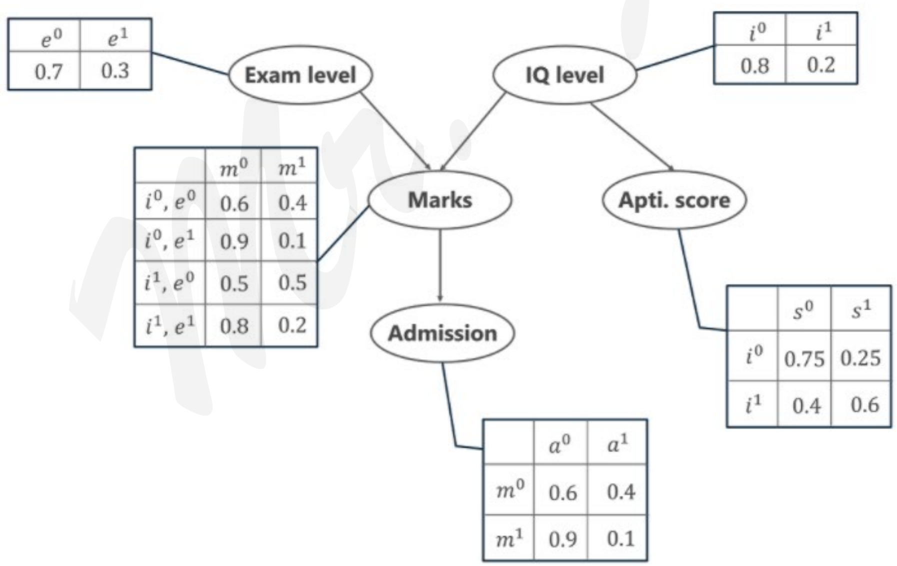
- $P(Free \mid Spam) = 0.20$
- $P(Free \mid Non\ Spam) = 0.001$
- $P(Spam) = 0.50 => P(Non\ Spam) = 0.50$
- $P(Spam \mid Free) = ?$

Using Bayes' Theorem:

- $P(Spam \mid Free) = P(Spam) * P(Free \mid Spam) / P(Free)$
- $P(Spam \mid Free) = 0.50 * 0.20 / (0.50 * 0.20 + 0.50 * 0.001)$
- $P(Spam \mid Free) = 0.995$

Exam Level (***e***)– This discrete variable denotes the difficulty of the exam and has two values (0 for easy and 1 for difficult)
IQ Level (***i***) – This represents the Intelligence Quotient level of the student and is also discrete in nature having two values (0 for low and 1 for high)

Additionally, the IQ level of the student also leads us to another variable, which is the Aptitude Score of the student (***s***). Now, with marks the student has scored, he can secure admission to a particular university. The probability distribution for getting admitted (***a***) to a university is also given below.



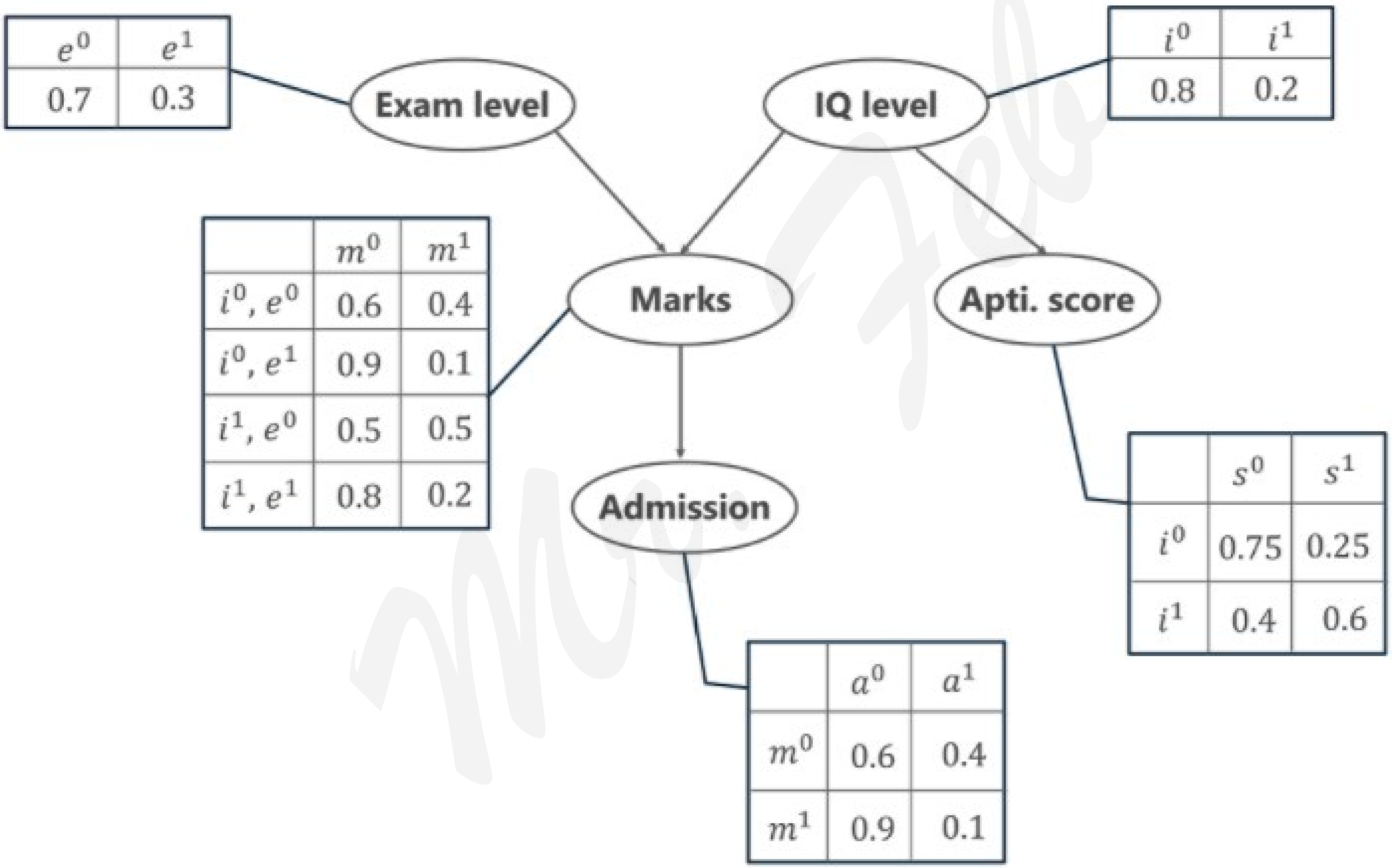
e^0	e^1
0.7	0.3

i^0	i^1
0.8	0.2

	m^0	m^1
i^0, e^0	0.6	0.4
i^0, e^1	0.9	0.1
i^1, e^0	0.5	0.5
i^1, e^1	0.8	0.2

	s^0	s^1
i^0	0.75	0.25
i^1	0.4	0.6

	a^0	a^1
m^0	0.6	0.4
m^1	0.9	0.1



e^0	e^1
0.7	0.3

i^0	i^1
0.8	0.2

	m^0	m^1
i^0, e^0	0.6	0.4
i^0, e^1	0.9	0.1
i^1, e^0	0.5	0.5
i^1, e^1	0.8	0.2

	s^0	s^1
i^0	0.75	0.25
i^1	0.4	0.6

	a^0	a^1
m^0	0.6	0.4
m^1	0.9	0.1

Exam Level (e)
IQ Level (i)
Aptitude Score (s)
Marks (m)
Admission (a)
These five variables are represented in the form of a Directed Acyclic Graph (DAG) in a Bayesian Network format with their Conditional Probability tables. Now, to calculate the Joint Probability Distribution of the 5 variables the formula is given by,
$P[a, m, i, e, s] = P(a \mid m) \cdot P(m \mid i, e) \cdot P(i) \cdot P(e) \cdot P(s \mid i)$

Calculate the probability that in spite of the exam level being difficult, the student having a low IQ level and a low Aptitude Score, manages to pass the exam and secure admission to the university

Calculate the probability that in spite of the exam level being difficult, the student having a low IQ level and a low Aptitude Score, manages to pass the exam and secure admission to the university

Calculate the probability that in spite of the exam level being difficult, the student having a low IQ level and a low Aptitude Score, manages to pass the exam and secure admission to the university

From the above word problem statement, the Joint Probability Distribution can be written as below,

P[a=1, m=1, i=0, e=1, s=0]

From the above Conditional Probability tables, the values for the given conditions are fed to the formula and is calculated as below.

P[a=1, m=1, i=0, e=0, s=0] = P(a=1 | m=1) . P(m=1 | i=0, e=1) . P(i=0) . P(e=1) . P(s=0 | i=0)

= 0.1 * 0.1 * 0.8 * 0.3 * 0.75

= **0.0018**

In another case, calculate the probability that the student has a High IQ level and Aptitude Score, the exam being easy yet fails to pass and does not secure admission to the university.

In another case, calculate the probability that the student has a High IQ level and Aptitude Score, the exam being easy yet fails to pass and does not secure admission to the university.

The formula for the JPD is given by

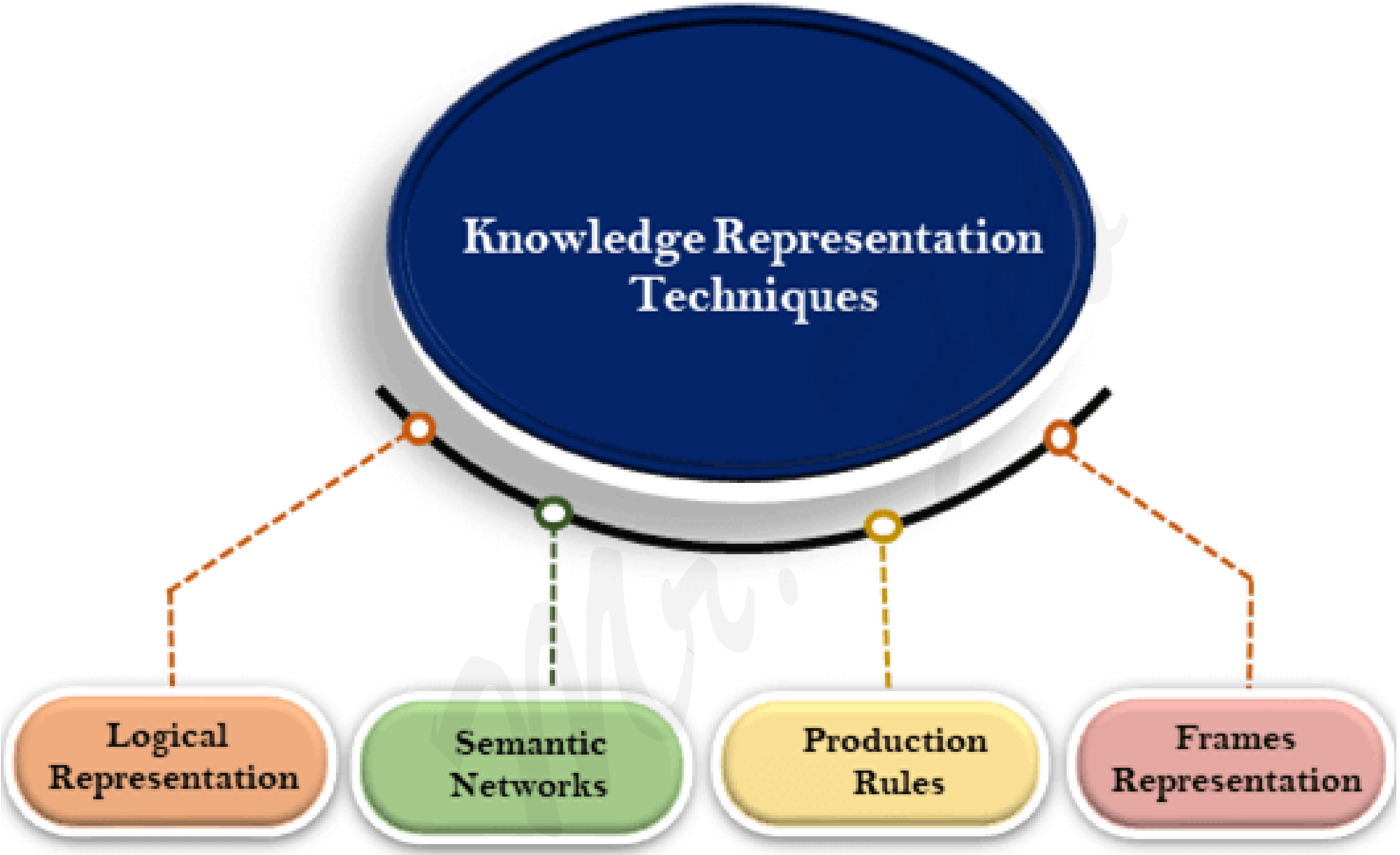
P[a=0, m=0, i=1, e=0, s=1]

Thus,

P[a=0, m=0, i=1, e=0, s=1]= P(a=0 | m=0) . P(m=0 | i=1, e=0) . P(i=1) . P(e=0) . P(s=1 | i=1)

= 0.6 * 0.5 * 0.2 * 0.7 * 0.6

= **0.0252**


Knowledge Representation Techniques
Logical Representation
Semantic Networks
Production Rules
Frames Representation

1. Logical Representation

Logical representation is a language with some concrete rules which deals with propositions and has no ambiguity in representation. Logical representation means drawing a conclusion based on various conditions. This representation lays down some important communication rules. It consists of precisely defined syntax and semantics which supports the sound inference. Each sentence can be translated into logics using syntax and semantics.

Syntax:

- Syntaxes are the rules which decide how we can construct legal sentences in the logic.
- It determines which symbol we can use in knowledge representation.
- How to write those symbols.

Semantics:

- Semantics are the rules by which we can interpret the sentence in the logic.
- Semantic also involves assigning a meaning to each sentence.

Logical representation can be categorised into mainly two logics:

1. Propositional Logics
2. Predicate logics

Semantic Network Representation

Semantic networks are alternative of predicate logic for knowledge representation. In Semantic networks, we can represent our knowledge in the form of graphical networks.

This network consists of nodes representing *objects and arcs* which describe the relationship between those objects. Semantic networks can categorize the object in different forms and can also link those objects.

Semantic networks are easy to understand and can be easily extended.

Semantic Net

- Form of knowledge representation
- Predicate logic alternative
- Labelled directed graph
- Components:
  - Nodes – object or concept
  - Links – relation between nodes.

Tom is a cat.

Tom caught a bird.

Tom is owned by John.

Tom is ginger in colour.

Cats like cream.

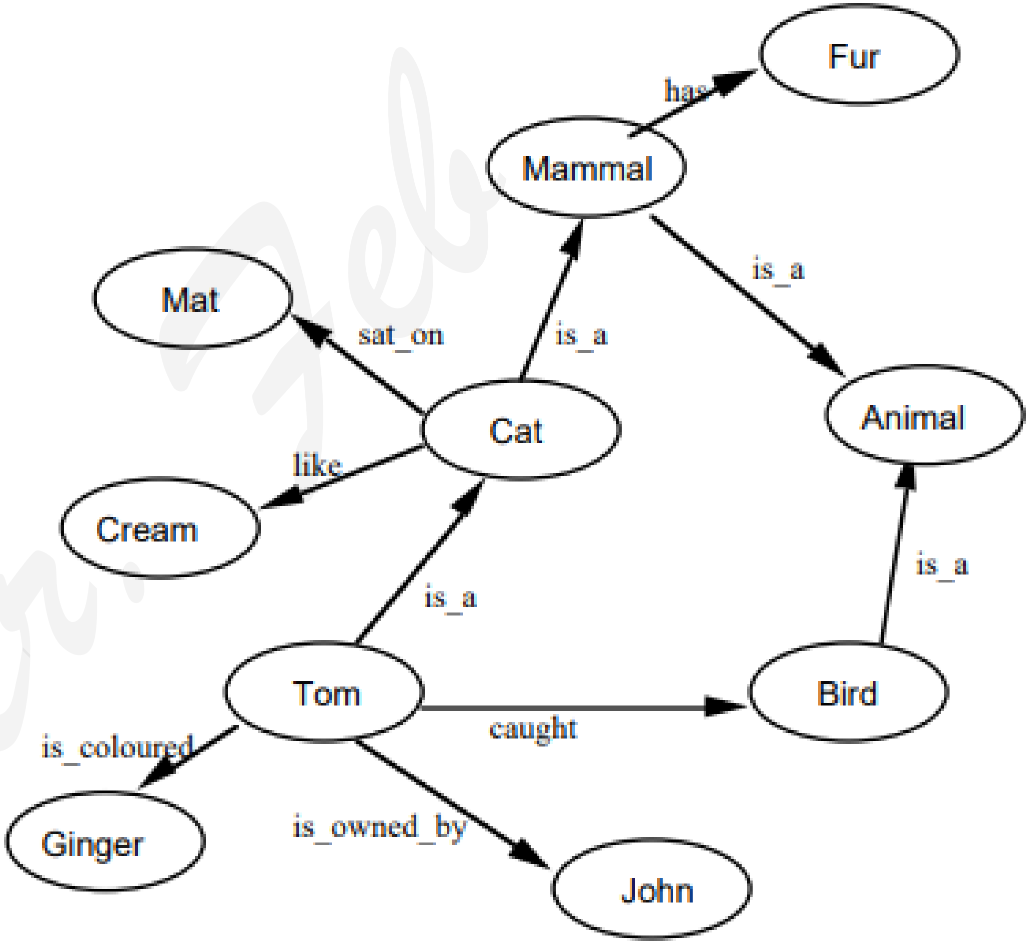
The cat sat on the mat.

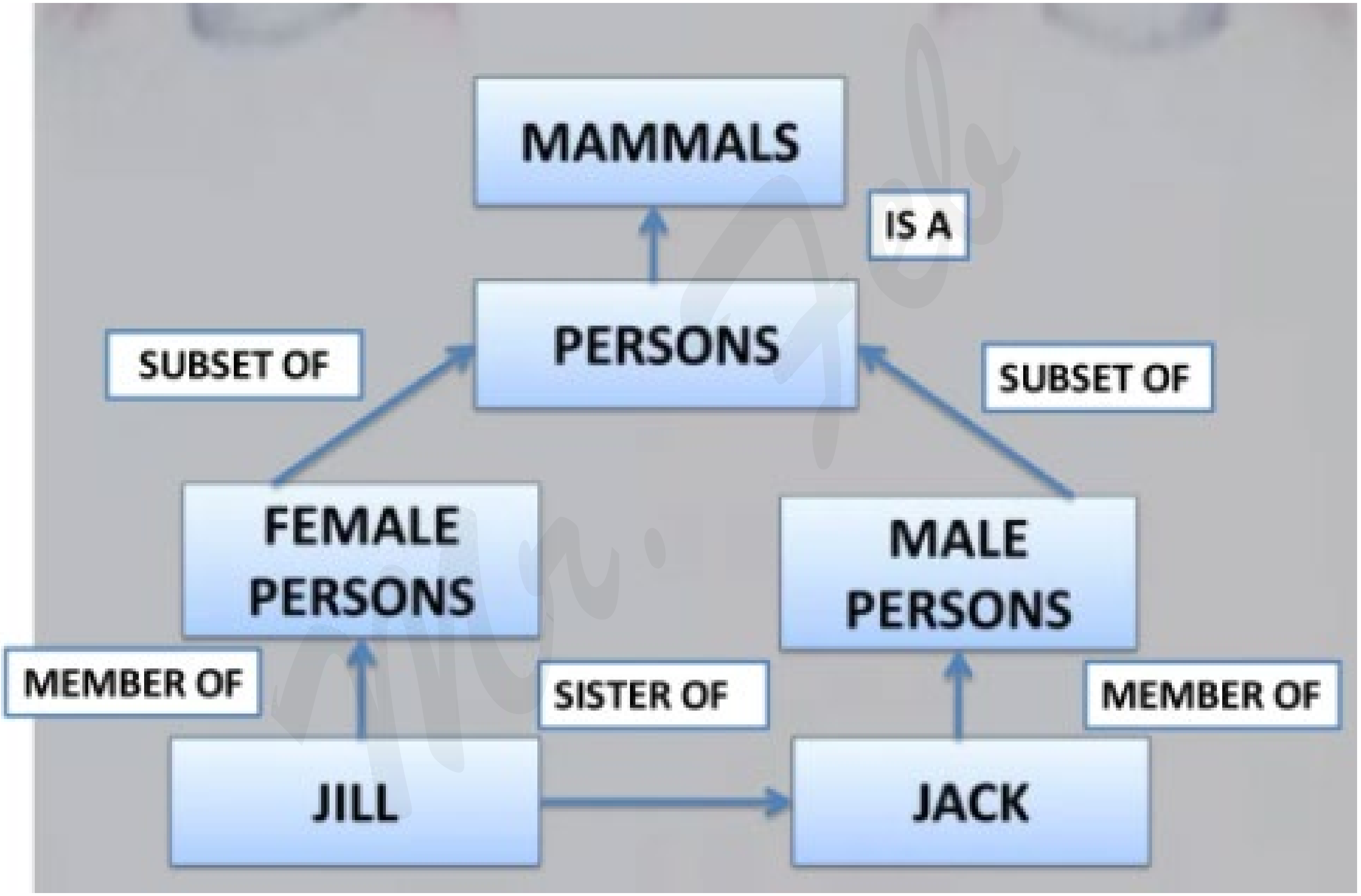
A cat is a mammal.

A bird is an animal.

All mammals are animals.

Mammals have fur.




MAMMALS
IS A
PERSONS
SUBSET OF
SUBSET OF
FEMALE PERSONS
MALE PERSONS
MEMBER OF
SISTER OF
MEMBER OF
JILL
JACK

Intersection Search

One of the early ways that semantic nets were used was to find relationships among objects by spreading activation out from each of two nodes and seeing where the activation met. This process is called *intersection search* [Quillian, 1968]. Using this process, it is possible to use the network of Fig. 9.1 to answer questions such as "What is the connection between the Brooklyn Dodgers and blue?"

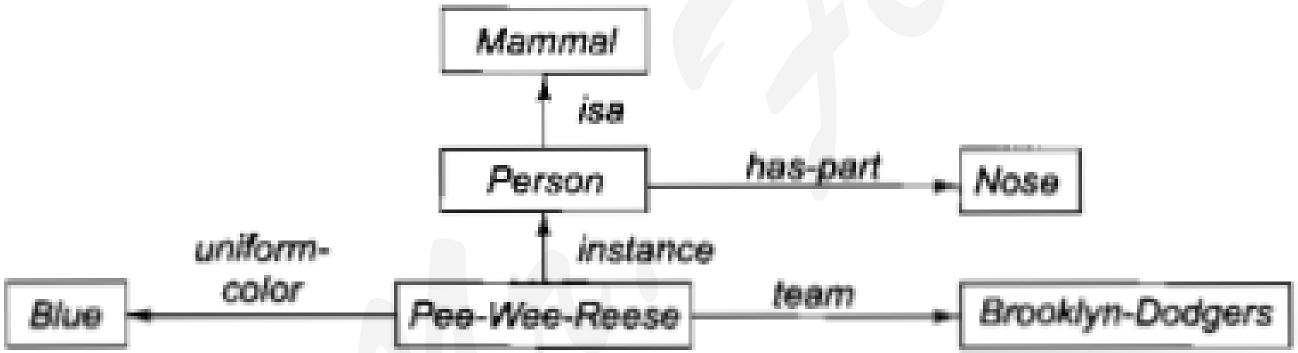
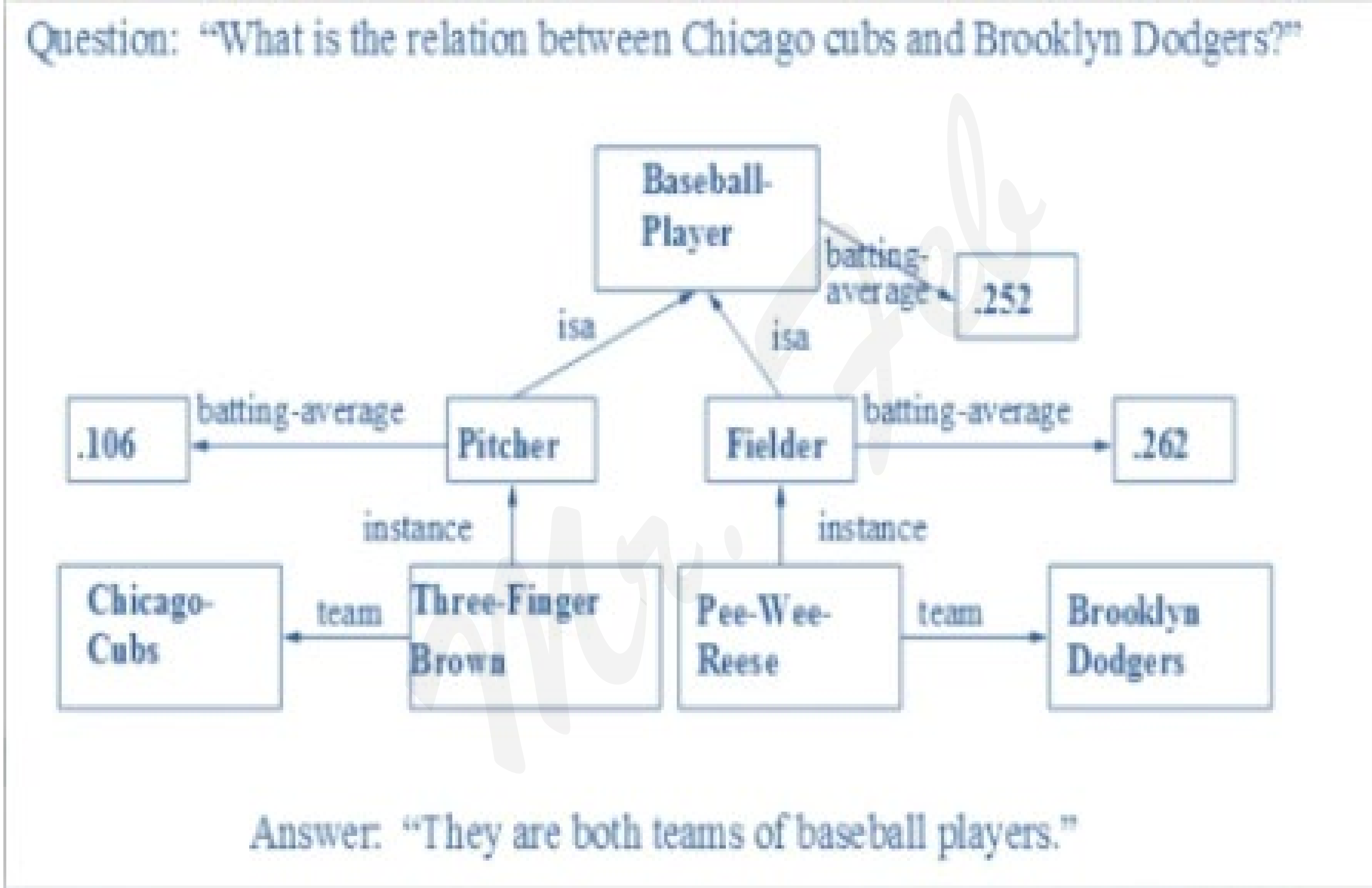


Fig. 9.1 *A Semantic Network*

Examples of Intersection Search



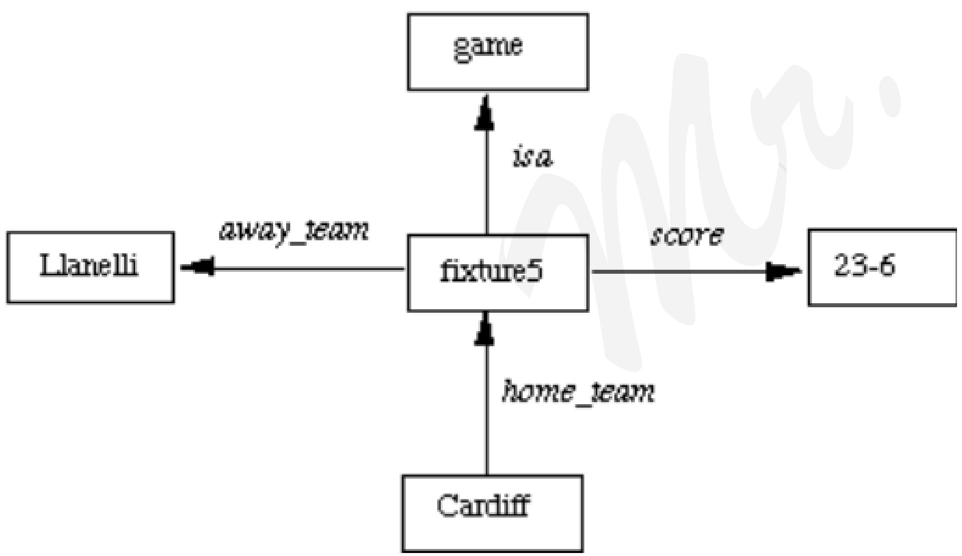
Representing non-binary predicates

- Semantic nets are a natural way to represent relationships that would appear as ground instances of binary predicates in logic
- For example :
- Is a(baseball player, pitcher)
- Is a(baseball player, fielder)
- Instance(three-finger brown ,pitcher)
- Instance(pee-wee Reese ,fielder)
- Team(three-finger brown , Chicago cubs)
- Team(pee-wee Reese ,Brooklyn dodgers)

We know that conventional predicates such as *lecturer(dave)* can be written as *instance (dave, lecturer)* Recall that *isa* and *instance* represent inheritance and are popular in many knowledge representation schemes. But we have a problem: *How we can have more than 2 place predicates in semantic nets? E.g. score(Cardiff, Llanelli, 23-6)*

Solution:

- Create new nodes to represent new objects either contained or alluded to in the knowledge, *game* and *fixture* in the current example.
- Relate information to nodes and fill up slots.



Representing non-binary predicates

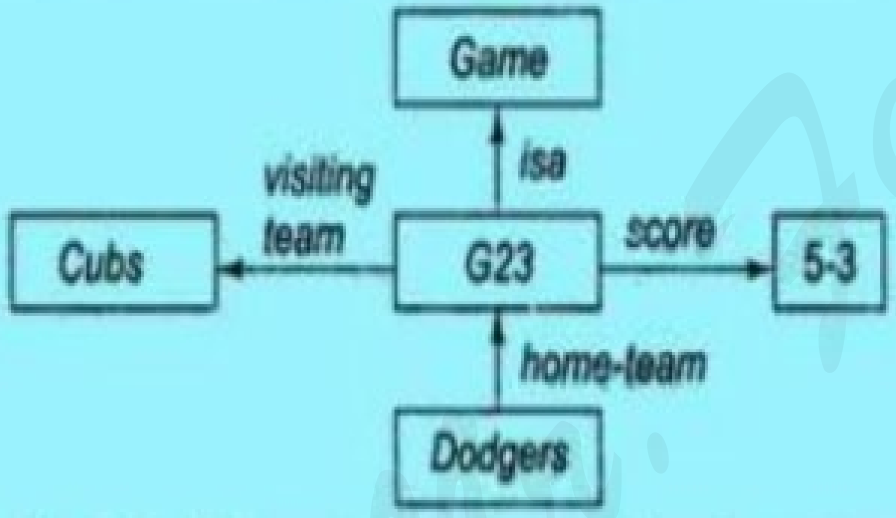


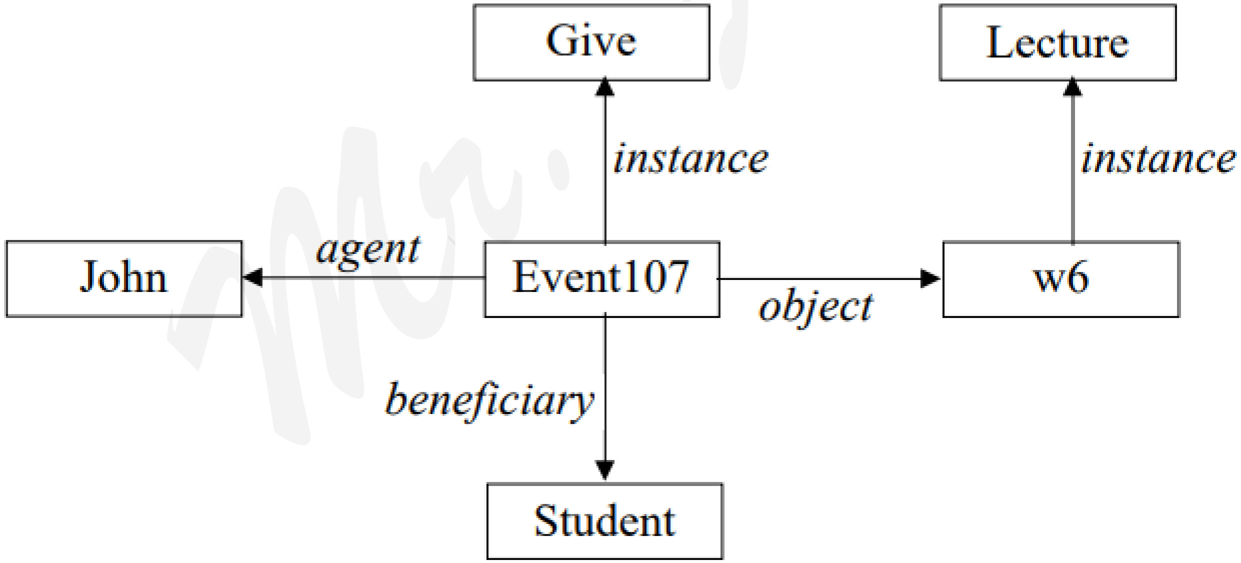
Fig. 9.2 *A Semantic Net for an n-Place Predicate*

Score(cubs,dodgers,5-3)

Representing Events and Language

Semantic networks are also very good at representing events, and simple declarative sentences, by basing them round an "event node". For example:

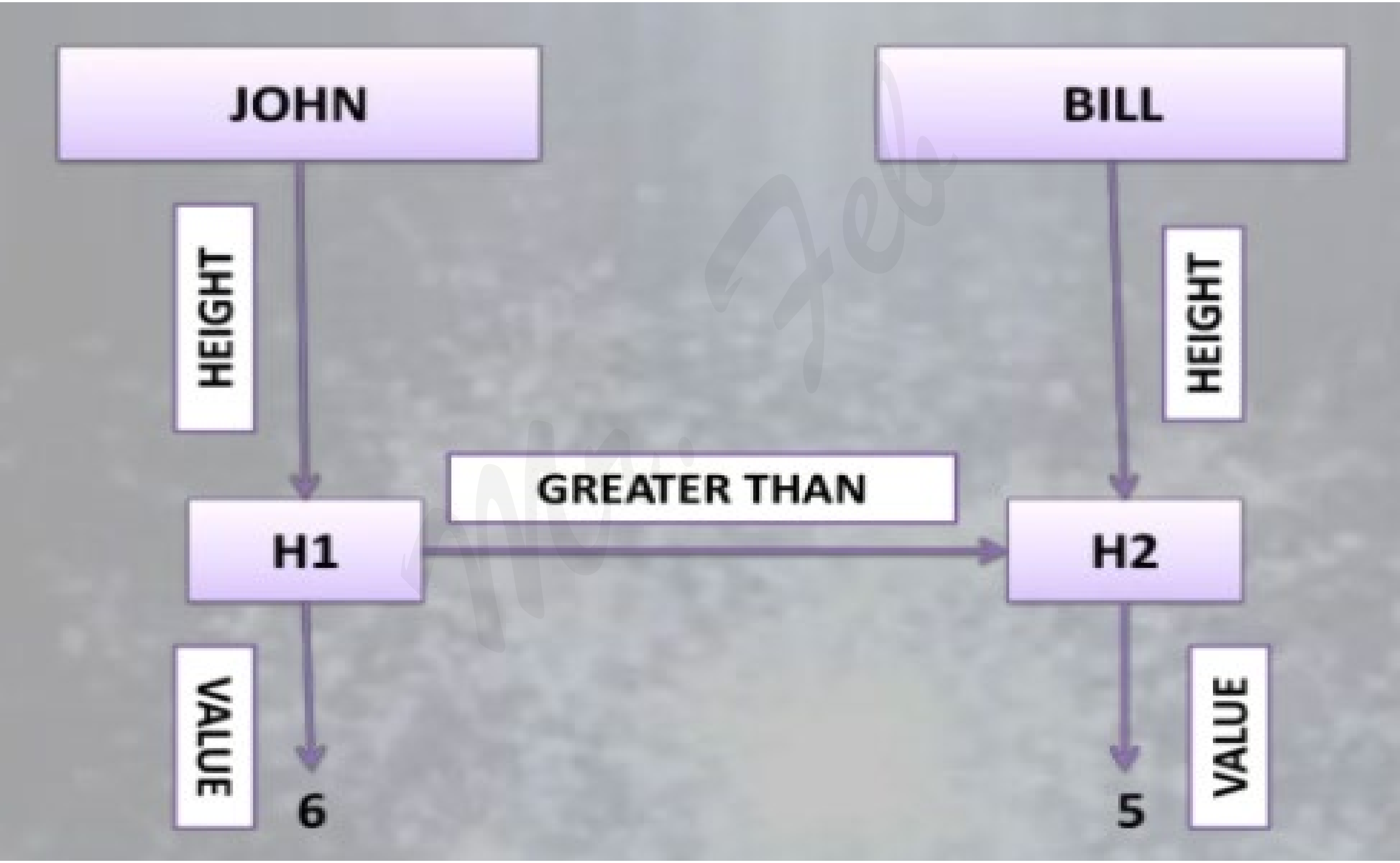
"John gave lecture w6 to his students"



In fact, several of the earliest semantic networks were English-understanding programs.

Making some important distinctions

- By defining the relationship the complexity of the relation can also be easily represented in semantic nets .
- For example : tom weight is 60 kg.



JOHN
BILL
HEIGHT
HEIGHT
GREATER THAN
H1
H2
VALUE
VALUE
6
5

PARTITIONED SEMANTIC NETS

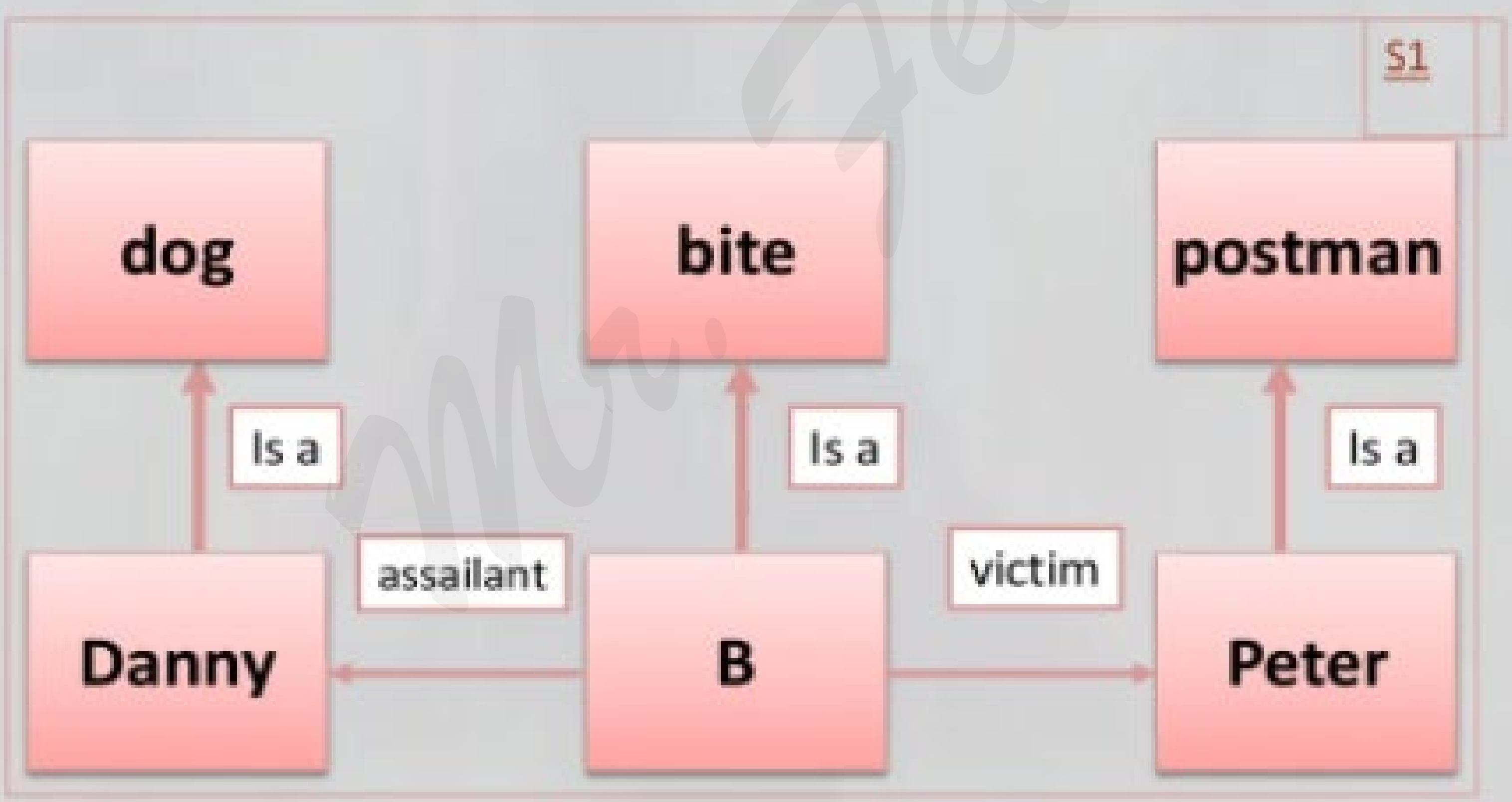
- Hendrix developed the **partitioned semantic network** to represent the difference between the description of an individual object or process and the description of a set of objects. The set description involves **quantification**.
- Hendrix partitioned a semantic network whereby a semantic network, loosely speaking, can be **divided** into one or more networks for the description of an individual.
- The central idea of partitioning is to allow groups, nodes and arcs to be bundled together into units called **spaces** – fundamental entities in partitioned networks, on the same level as nodes and arcs

PARTITIONED SEMANTIC NETS

- Suppose that we wish to make a specific statement about a dog, Danny, who has bitten a postman, Peter:
  - " Danny the dog bit Peter the postman"

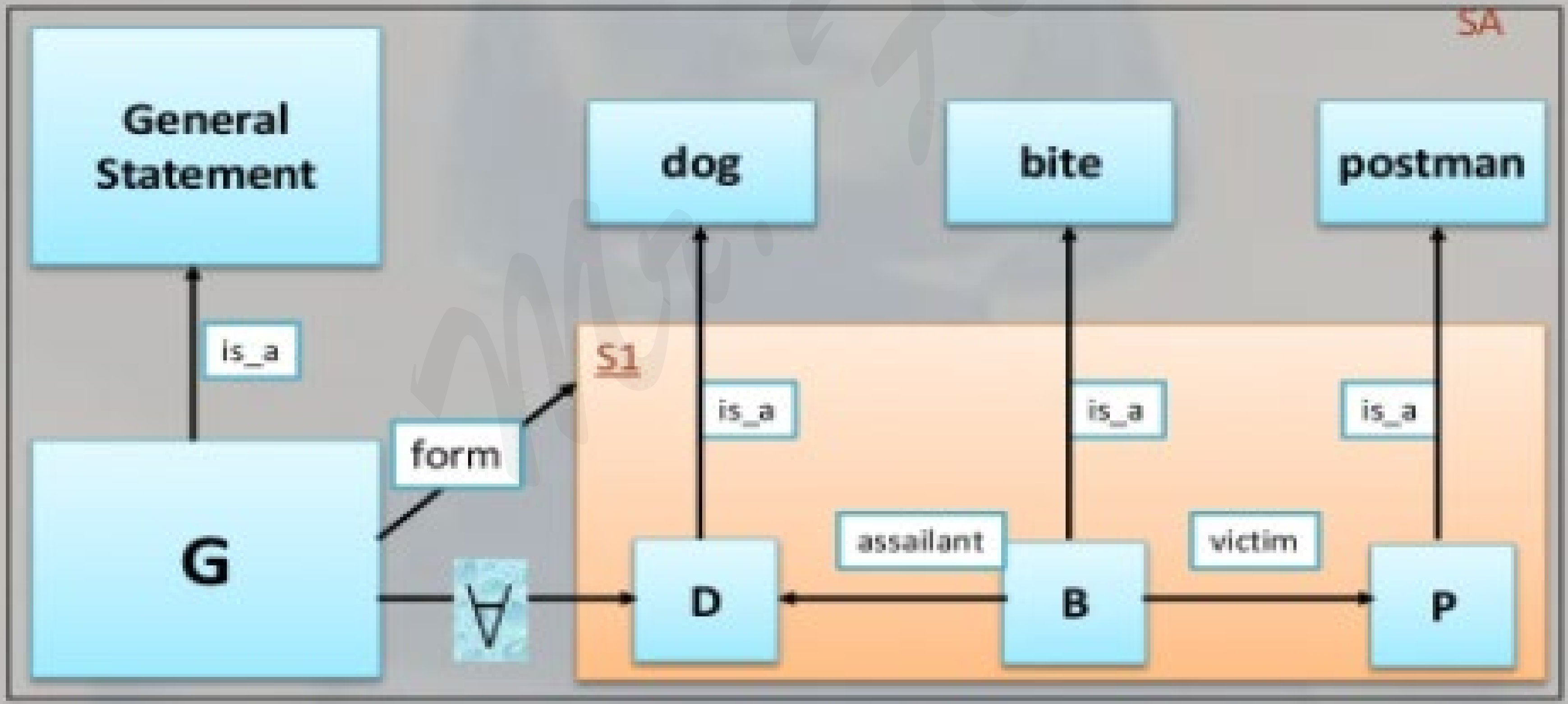
- Hendrix's Partitioned network would express this statement as an ordinary semantic network:

PARTITIONED SEMANTIC NETS



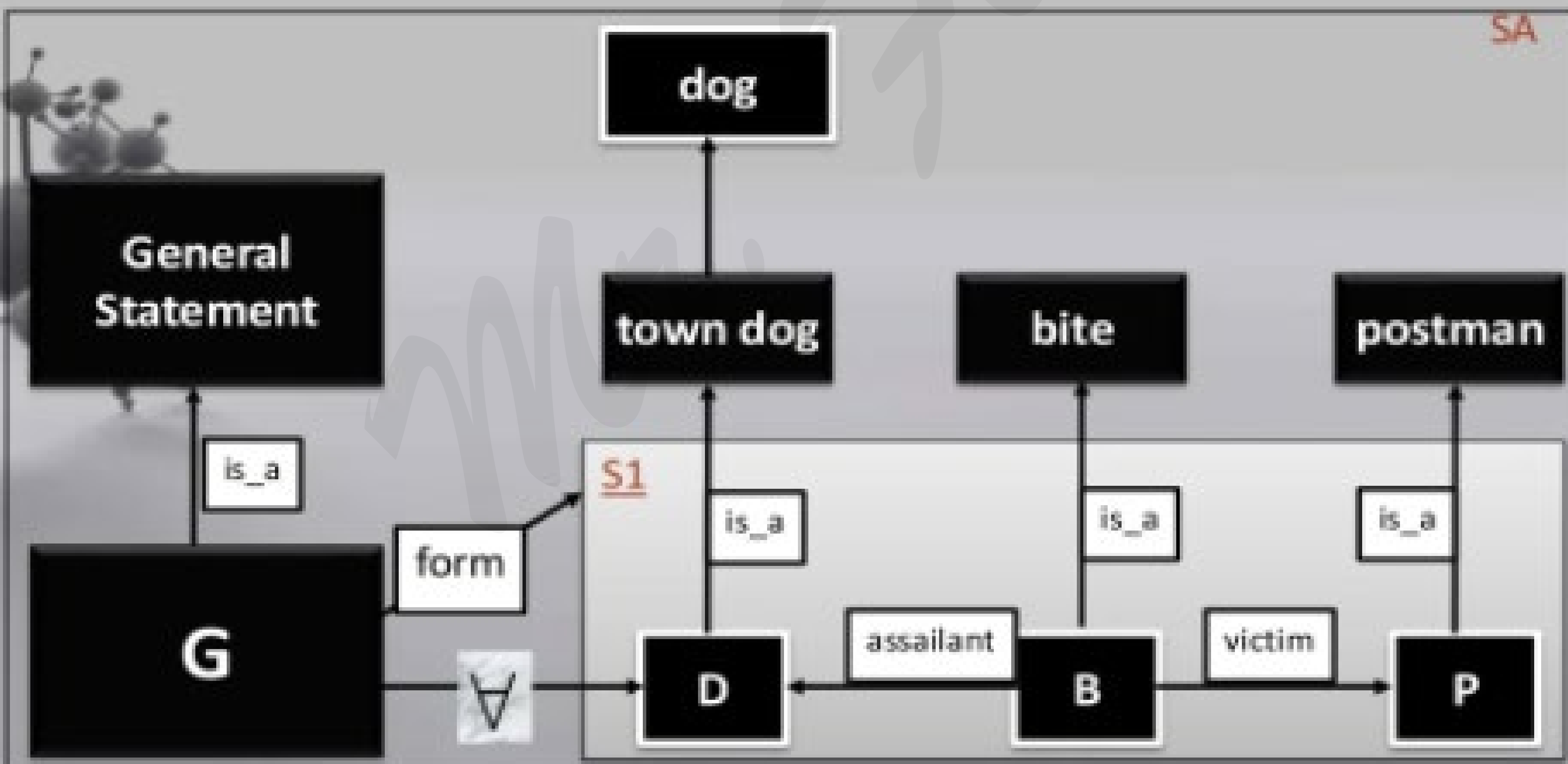
PARTITIONED SEMANTIC NETS

"Every dog has bitten a postman"



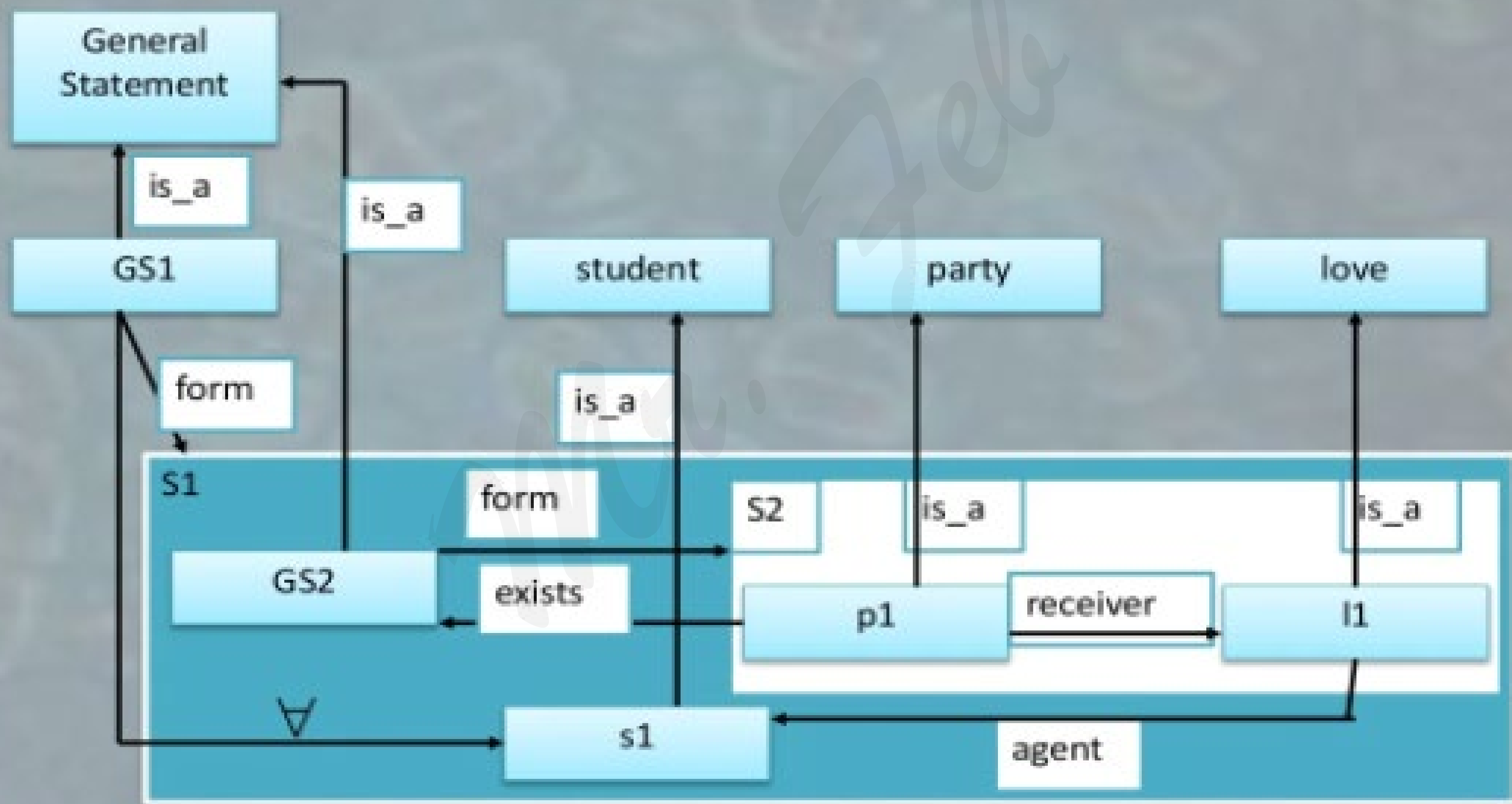
PARTITIONED SEMANTIC NETS

- "Every dog in town has bitten the postman"



"John believes that pizza is tasty"

"Every student loves to party"



advantages

- Easy to visualise and understand.
- The knowledge engineer can arbitrarily defined the relationships.
- Related knowledge is easily
  categorised.
- Efficient in space requirements.
- Related knowledge is
  easily clustered.

FRAMES

- Frames can also be regarded as an extension to Semantic nets.
- Indeed it is not clear where the distinction between a semantic net and a frame ends.
- Semantic nets initially were used to represent labelled connections between objects. As tasks became more complex the representation needs to be more structured. The more structured the system is, it becomes more beneficial to use frames.
- A frame is a collection of attributes or slots and associated values that describe some real world entity.

Components of a Frame

- 3 components of a frame
  - frame name
  - attributes (slots)
  - values (fillers: list of values, range, string, etc.)

Features of Frame Representations

- Frames can support values more naturally than semantic networks.
- Frames can be easily implemented using object-oriented programming techniques.
- Demons allow for arbitrary functions to be embedded in a representation.
- But a price is paid in terms of efficiency, generality, and modularity !
- Inheritance can be easily controlled.

Example : Frame

Slot	Filters
Title	Artificial Intelligence
Branch	Computer Science
Author	Ritch and Knight
Edition	Third Edition
Year	2020
Page	210

Simple Frame Example

Slot Name	Filler
name	Astérix
height	small
weight	low
profession	warrior
armor	helmet
intelligence	very high
marital status	presumed single

Example : Frame Network

Title	Hotel Room
Superclass	Room
Location	Hotel
Contains	Hotel Chair Hotel Phone Hotel Bed

Example : Frame Network

Title	Hotel Room
Superclass	Room
Location	Hotel
Contains	Hotel Chair Hotel Phone Hotel Bed

Title	Hotel Chair
Superclass	Chair
Height	20-40 cm
Legs	4
Use	Sitting

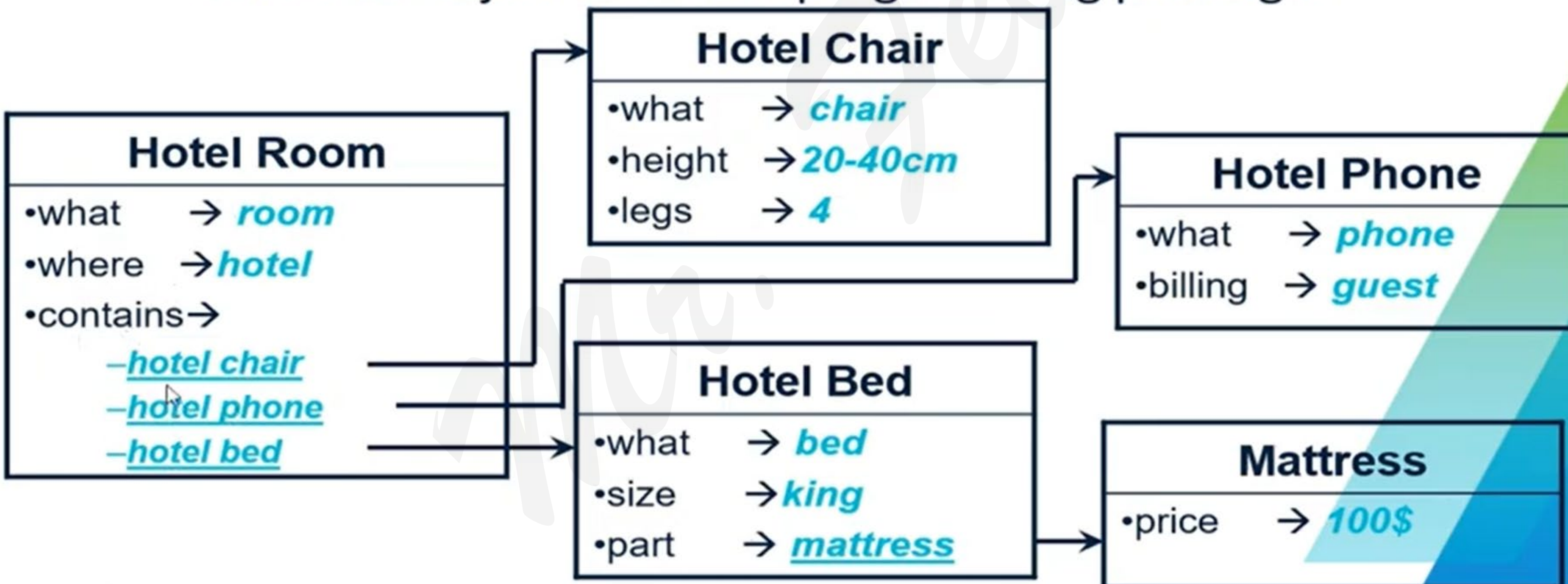
Title	Hotel Phone
Superclass	Phone
Use	Calling Room Service
Billing	Through Room

Title	Hotel Bed
Superclass	Bed
Use	Sleeping
Size	King
Part	Mattress

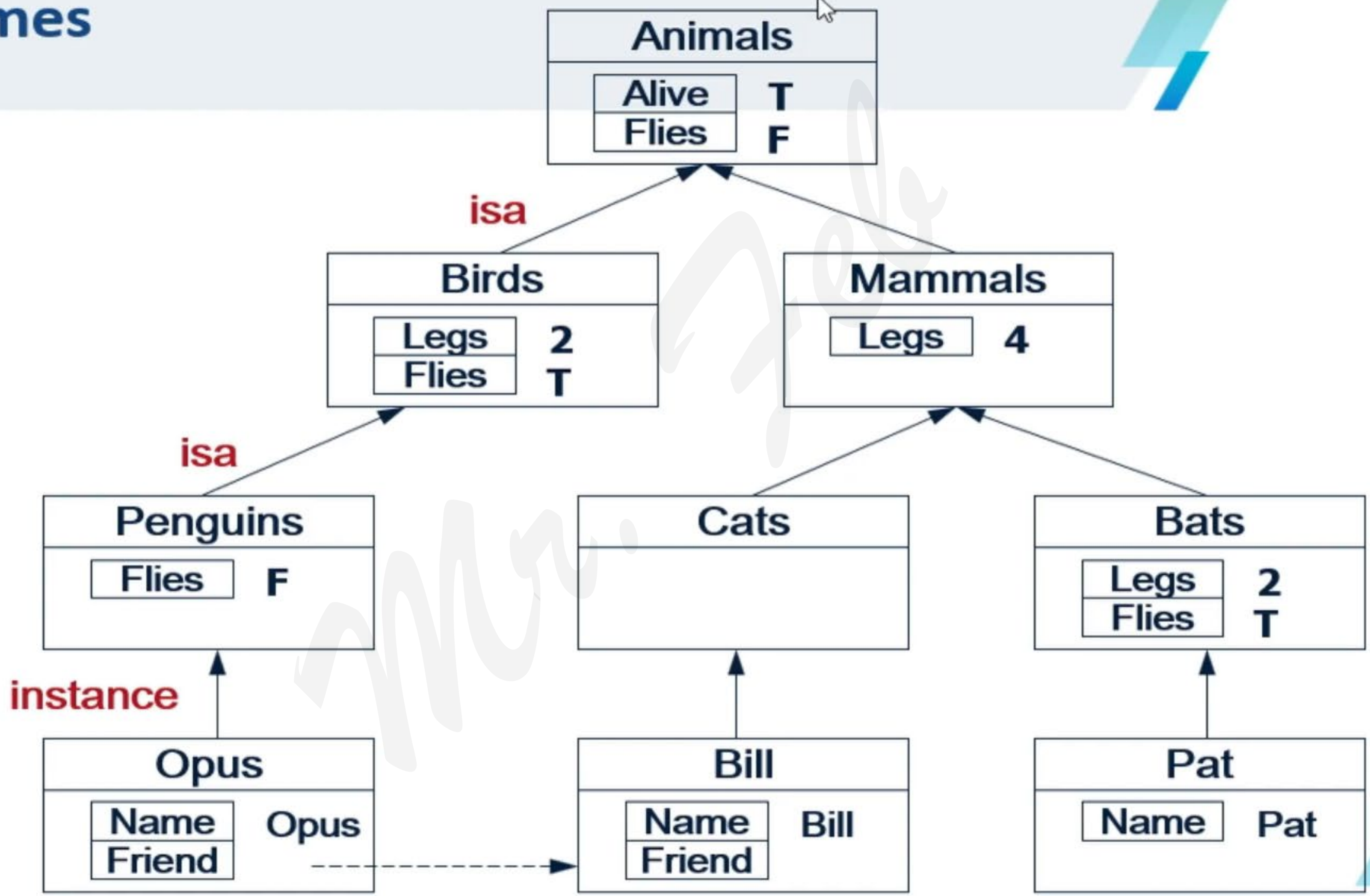
Title	Mattress
Superclass	Cushion
Firmness	Firm

Inheritance

- Similar to Object-Oriented programming paradigm



Frames


Animals
Alive T
Flies F
isa
Birds
Legs 2
Flies T
Mammals
Legs 4
isa
Penguins
Flies F
Cats
Bats
Legs 2
Flies T
instance
Opus
Name Opus
Friend
Bill
Name Bill
Friend
Pat
Name Pat

Features of Frame Representation

- More natural support of values then semantic nets (each slots has constraints describing legal values that a slot can take)
- Can be easily implemented using object-oriented programming techniques
- Inheritance is easily controlled