UNIT –IV Knowledge Representation:

Introduction, Approaches to Knowledge representation, Knowledge representation using Semantic Network, Extended Semantic Network for KR, Knowledge representation using Frames Advance Knowledge Representation Techniques: Introduction, Conceptual Dependency Theory, Script Structure, CYC Theory, Case Grammars, Semantic Web

4.1 Knowledge Representation

- *Knowledge representation (KR)* is an important issue in both cognitive science and artificial intelligence.
  - In cognitive science, it is concerned with the way people store and process information and
  - In artificial intelligence (AI), main focus is to store knowledge so that programs can process it and achieve human intelligence.
- There are different ways of representing knowledge e.g.
  - predicate logic,
  - semantic networks,
  - extended semantic net,
  - frames,
  - conceptual dependency etc.
- In predicate logic, knowledge is represented in the form of rules and facts as is done in Prolog.

4.2. Semantic Network:

- Formalism for representing information about objects, people, concepts and specific relationship between them.
- The syntax of semantic net is simple. It is a network of labeled nodes and links.
  - It's a directed graph with nodes corresponding to concepts, facts, objects etc. and
  - arcs showing relation or association between two concepts.

- The commonly used links in semantic net are of the following types.
  - isa $\rightarrow$ subclass of entity (e.g., child hospital is subclass of hospital)
  - inst $\rightarrow$ particular instance of a class (e.g., India is an instance of country)

prop $\rightarrow$ property link (e.g., property of dog is ‘bark)

4.3. Representation of Knowledge in Sem Net

“Every human, animal and bird is living thing who breathe and eat. All birds can fly. All man and woman are humans who have two legs. Cat is an animal and has a fur. All animals have skin and can move. Giraffe is an animal who is tall and has long legs. Parrot is a bird and is green in color”.

4.3.1 Representation in Predicate Logic

- Every human, animal and bird is living thing who breathe and eat.

  $\forall X$ [human(X) $\rightarrow$ living(X)]

  $\forall X$ [animal(X) $\rightarrow$ living(X)]

  $\forall X$ [bird(X) $\rightarrow$ living(X)]

- All birds are animal and can fly.

  $\forall X$ [bird(X) $\wedge$ canfly(X)]

- Every man and woman are humans who have two legs.

  $\forall X$ [man(X) $\wedge$ haslegs(X)]

  $\forall X$ [woman(X) $\wedge$ haslegs(X)]

  $\forall X$ [human(X) $\wedge$ has(X, legs)]

- Cat is an animal and has a fur.

  animal(cat) $\wedge$ has(cat, fur)

- All animals have skin and can move.

  $\forall X$ [animal(X) $\rightarrow$ has(X, skin) $\wedge$ canmove(X)]

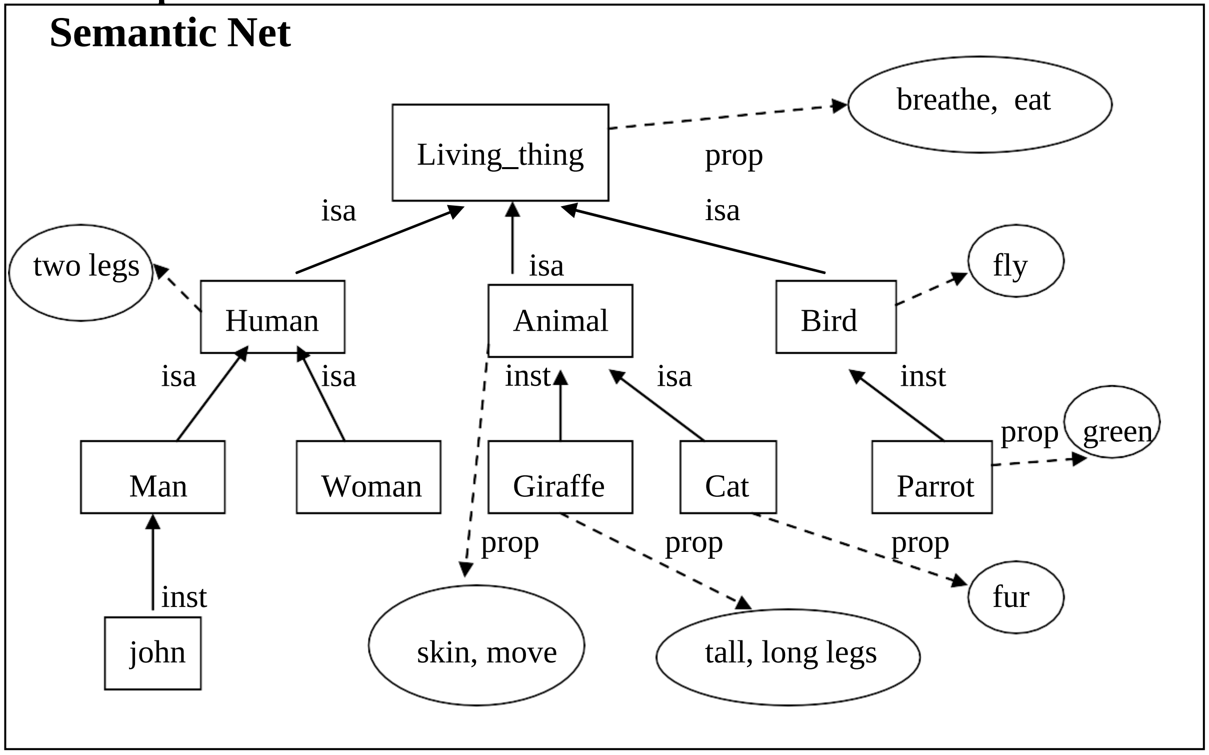
- Giraffe is an animal who is tall and has long legs.

  animal(giraffe) ∧ has(giraffe, long_legs) ∧ is(giraffe, tall)

- Parrot is a bird and is green in color.

  bird(parrot) ∧ has(parrot, green_colour)

4.3.2.Representation in Semantic Net



4.4.Inheritance

- Inheritance mechanism allows knowledge to be stored at the highest possible level of abstraction which reduces the size of knowledge base.
  - It facilitates inferencing of information associated with semantic nets.
  - It is a natural tool for representing taxonomically structured information and ensures that all the members and sub-concepts of a concept share common properties.
  - It also helps us to maintain the consistency of the knowledge base by adding new concepts and members of existing ones.

- Properties attached to a particular object (class) are to be inherited by all subclasses and members of that class.

4.4.1. Property Inheritance Algorithm

Input: Object, and property to be found from Semantic Net;

Output: Yes, if the object has the desired property else return false;

Procedure:

```
● Find an object in the semantic net;   Found = false;
● While {(object ≠ root) OR Found } DO
      { If there is a a property attribute attached with an object then
            { Found = true; Report 'Yes'}          else
            object=inst(object, class) OR isa(object, class)
      };
● If Found = False then report 'No';   Stop
```

Coding of Semantic Net in Prolog

Isa facts	Instance facts	Property facts
isa(living_thing, nil).	inst(john, man).	prop(breathe, living_thing).
isa(human, living_thing).	inst(giraffe, animal).	prop(eat, living_thing).
isa(animals, living_thing).	inst(parrot, bird)	prop(two_legs, human).
isa(birds, living_thing).		prop(skin, animal).
isa(man, human).		prop(move, animal).
isa(woman, human).		prop(fur, bird).
isa(cat, animal).		prop(tall, giraffe).
		prop(long_legs, giraffe).
		prop(tall, animal).
		prop(green, parrot).

4.4.2. Inheritance Rules in Prolog

Instance rules:

```
instance(X, Y)       :-    inst(X, Y).
instance (X, Y)      :-    inst(X, Z), subclass(Z,Y).
```

Subclass rules:

```
subclass(X, Y)       :-    isa(X, Y).
subclass(X, Y)       :-    isa(X, Z), subclass(Z, Y) .
```

Property rules:

```
property(X, Y)       :-    prop(X, Y).
property(X, Y)       :-    instance(Y,Z), property(X, Z).
property(X, Y)       :-    subclass(Y, Z), property(X, Z).
```

Queries

• Is john human?	?- instance(john, humans). Y
• Is parrot a living thing?	?- instance (parrot, living_thing). Y
• Is giraffe an aimal?	?- instance (giraffe, animal).Y
• Is woman subclassof living thing	?- subclass(woman, living_things). Y
• Does parrot fly?	?- property(fly, parrot). Y
• Does john breathe?	?- property (john, breathe). Y
• has parrot fur?	?- property(fur, parrot). N
• Does cat fly?	?- property(fly, cat). N

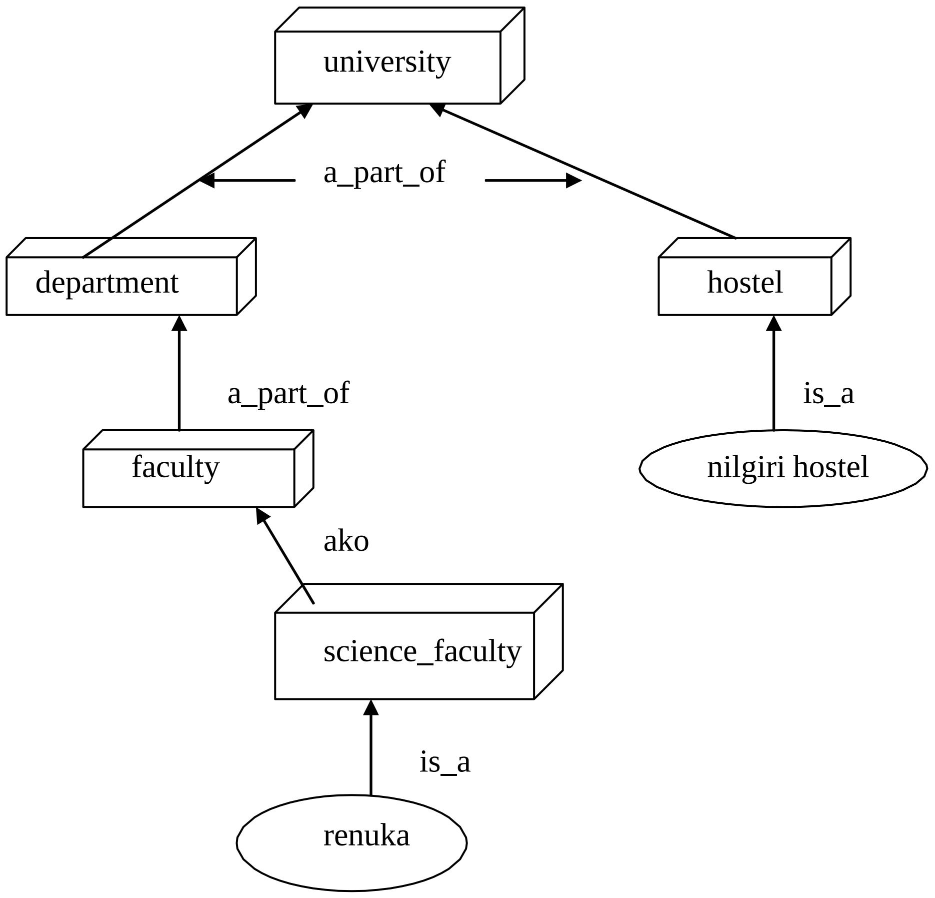
4.5.Knowledge Representation using Frames

- Frames are more structured form of packaging knowledge,
  - used for representing objects, concepts etc.
- Frames are organized into hierarchies or network of frames.
- Lower level frames can inherit information from upper level frames in network.
- Nodes are connected using links viz.,
  - **ako / subc** (links two class frames, one of which is subclass of other e.g., science_faculty class is **ako** of faculty class),
  - **is_a / inst** (connects a particular instance of a class frame e.g., Renuka **is_a** science_faculty)
  - **a_part_of** (connects two class frames one of which is contained in other e.g., faculty class **is_part_of** department class).

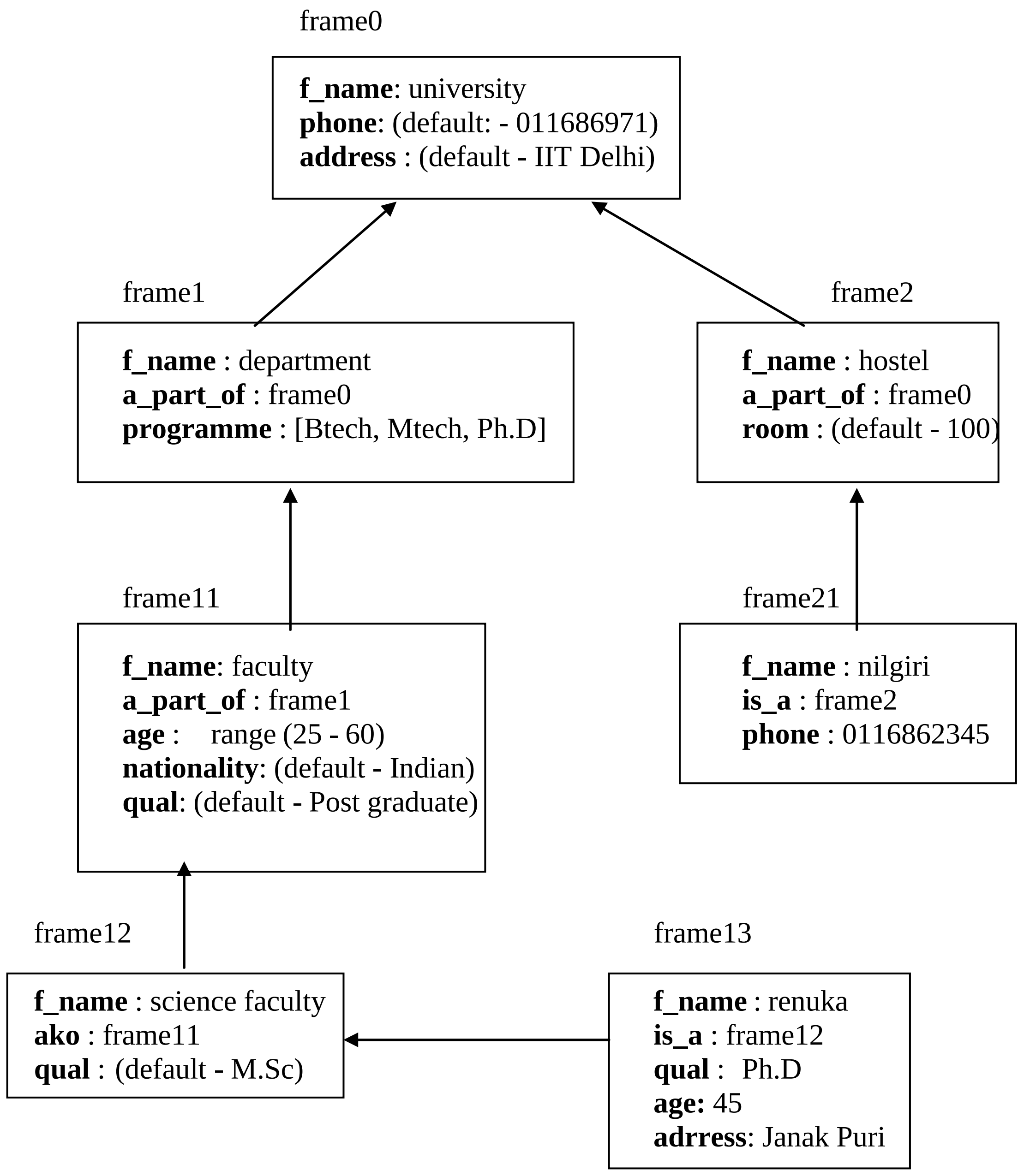
Property link of semantic net is replaced by SLOT fields.

- A frame may have any number of slots needed for describing object. e.g.,
  - faculty frame may have name, age, address, qualification etc as slot names.
- Each frame includes two basic elements : slots and facets.
  - Each slot may contain one or more **facets** (called fillers) which may take many forms such as:
    - **value** (value of the slot),
    - **default** (default value of the slot),
    - **range** (indicates the range of integer or enumerated values, a slot can have),
    - **demons** (procedural attachments such as if_needed, if_deleted, if_added etc.) and
    - **other** (may contain rules, other frames, semantic net or any type of other information).

4.5.1. Frame Network – Example



4.5.2. Detailed Representation of Frame Network



4.5.3. Description of Frames

- Each frame represents either a class or an instance.
- Class frame represents a general concept whereas instance frame represents a specific occurrence of the class instance.

- Class frame generally have default values which can be redefined at lower levels.
- If class frame has actual value facet then decedent frames can not modify that value.
- Value remains unchanged for subclasses and instances.

4.5.4. Inheritance in Frames

- Suppose we want to know nationality or phone of an instance-frame frame13 of renuka.
- These informations are not given in this frame.
- Search will start from frame13 in upward direction till we get our answer or have reached root frame.
- The frames can be easily represented in prolog by choosing predicate name as frame with two arguments.
- First argument is the name of the frame and second argument is a list of slot - facet pair.

4.5.5. Coding of frames in Prolog

```
frame(university, [phone (default, 011686971),
        address (default, IIT Delhi)]).

frame(deaprtment, [a_part_of (university), programme ([Btech, Mtech, Ph.d]))]).

frame(hostel, [a_part_of (university), room(default, 100)]).

frame(faculty, [a_part_of (department), age(range,25,60), nationality(default,
indian), qual(default, postgraduate)]).

frame(nilgiri, [is_a (hostel), phone(011686234)]).

frame(science_faculty, [ako (faculty),qual(default, M.Sc.)]).

frame(renuka, [is_a (science_faculty), qual(Ph.D.), age(45), address(janakpuri)]).
```

4.5.6. Inheritance Program in Prolog

```
find(X, Y) :- frame(X, Z), search(Z, Y), !.
find(X, Y) :- frame(X, [is_a(Z),_]), find(Z, Y), !.

find(X, Y) :- frame(X, [ako(Z), _]), find(Z, Y), !.

find(X, Y) :- frame(X, [a_part_of(Z), _]), find(Z, Y).
```

- Predicate **search** will basically retrieve the list of slots-facet pair and will try to match Y for slot.
- If match is found then its facet value is retrieved otherwise process is continued till we reach to root frame

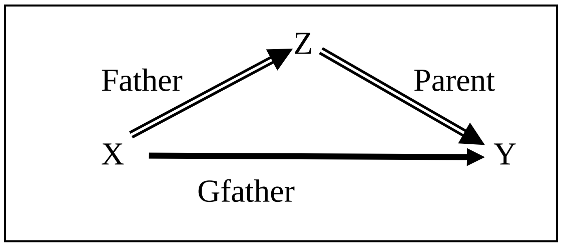
4.6. Extended Semantic Network

- In conventional Sem Net, clausal form of logic can not be expressed.
- Extended Semantic Network (ESNet) combines the advantages of both logic and semantic network.
- In the ESNet, terms are represented by nodes similar to Sem Net.
- Binary predicate symbols in clausal logic are represented by labels on arcs of ESNet.
  - An *atom* of the form "Love(john, mary)" is an arc labeled as 'Love' with its two end nodes representing 'john' and 'mary'.
- *Conclusions* and *conditions* in clausal form are represented by different kinds of arcs.
  - Conditions are drawn with two lines ⇐ and conclusions are drawn with one heavy line ⬅ .

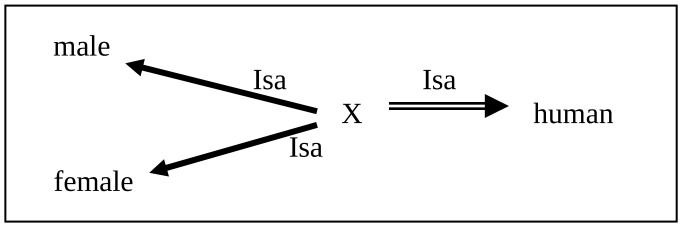
Example:

- Represent 'grandfather' definition

  Gfather(X, Y) ← Father(X, Z), Parent(Z, Y) in ESNet.



- Represent clausal rule **“Male(X), Female(X) ← Human(X)”** using binary representation as “Isa(X, male), Isa(X, female) ← Isa(X, human)” and subsequently in ESNet as follows:

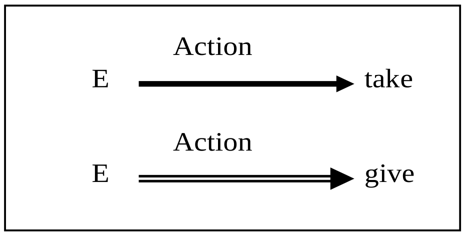


4.6.1.Inference Rules in ESNet

- Inference rules are embedded in the representation itself.
- The inference that “for every action of giving, there is an action of taking” in clausal logic written as

  “Action(E, take) ← Action(E, give)”.

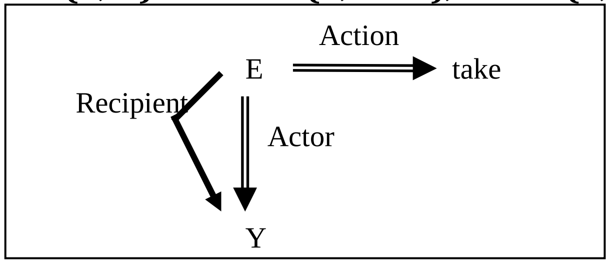
ESNet



- The inference rule such as “an actor of taking action is also the recipient of the action” can be easily represented in clausal logic as:
  - Here E is a variable representing an event where an action of taking is happening).

Recipient(E, Y) ← Acton(E, take), Actor (E, Y)

ESNet



Example 2:

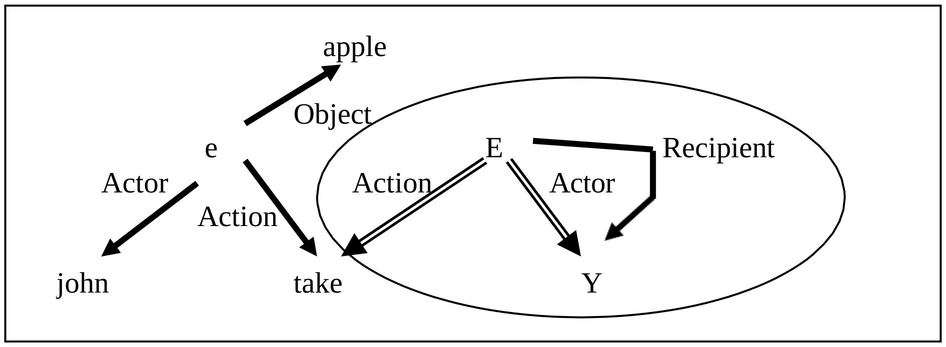
- Represent the following clauses of Logic in ESNet.

  Recipient(E, Y) ← Acton(E, take), Actor (E, Y)

  Object (e, apple).

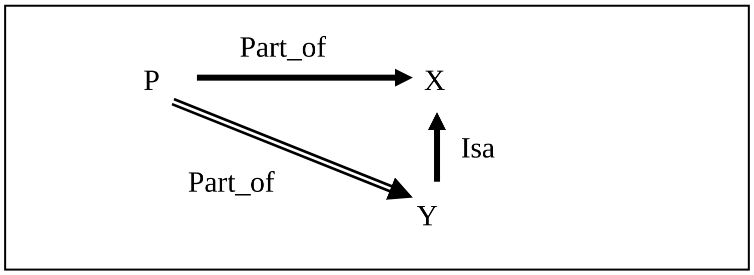
  Action(e, take).

  Actor (e, john) .



4.6.2.Contradiction

- The contradiction in the ESNet arises if we have the following situation.



4.6.3. Deduction in ESNet

- Both of the following inference mechanisms are available in ESNet.

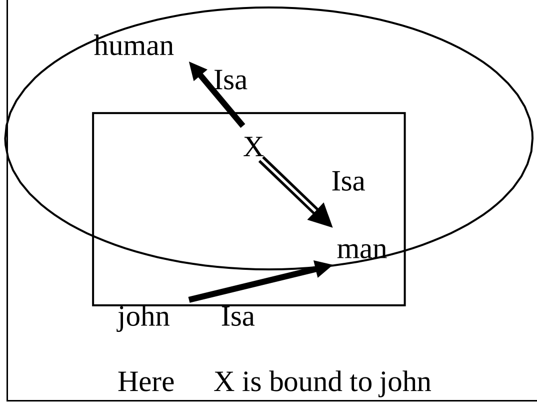
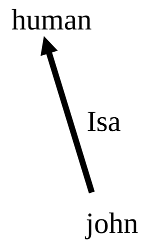
– Forward reasoning inference (uses bottom up approach)

  - *Bottom Up Inferencing:* Given an ESNet, apply the following reduction (resolution) using modus ponen rule of logic ({A ← B, B} then A).

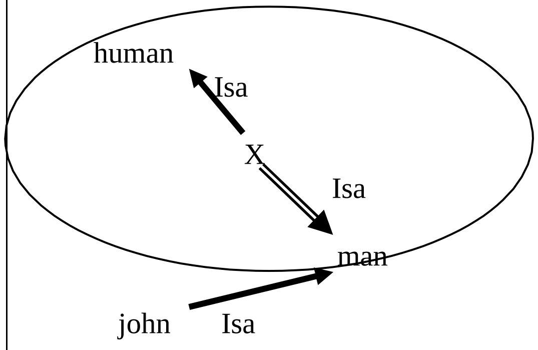
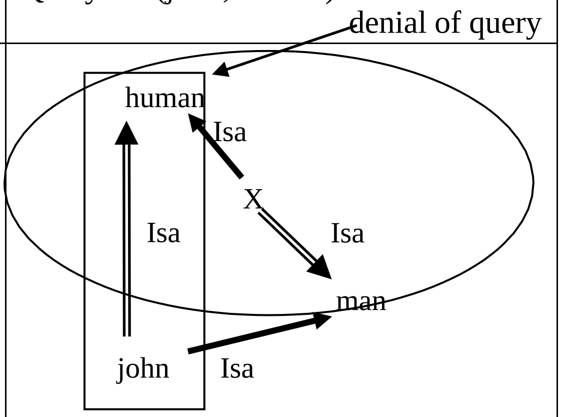
– Backward reasoning inference (uses top down approach).

Top Down Inferencing: Prove a conclusion from a given ESNet by adding the denial of the conclusion to the network and show that the resulting set of clauses in the network is inconsistent

Example: Bottom Up Inferencing

Given set of clauses	**Inferencing**
Isa(X, human) ← Isa(X, man) Isa(john, man).	Isa(john, human)
 human Isa X Isa man john Isa Here X is bound to john	

Example: Top Down Inferencing

Given set of clauses	**Prove conclusion**
Isa(X, human) ← Isa(X, man) Isa(john, man).	Query: Isa(john, human)
	 denial of query human Isa Isa X Isa man john Isa

4.7.Advance Knowledge Representation Techniques:

4.7.1.Conceptual Dependency (CD)

- CD theory was developed by Schank in 1973 to 1975 to represent the meaning of NL sentences.
    - It helps in drawing inferences
    - It is independent of the language
- CD representation of a sentence is not built using words in the sentence rather built using conceptual primitives which give the intended meanings of words.
- CD provides **structures** and specific **set of primitives** from which representation can be built.

Primitive Acts of CD theory

● ATRANS	Transfer of an abstract relationship (i.e. give)
● PTRANS	Transfer of the physical location of an object (e.g., go)
● PROPEL	Application of physical force to an object (e.g. push)
● MOVE	Movement of a body part by its owner (e.g. kick)
● GRASP	Grasping of an object by an action (e.g. throw)
● INGEST	Ingesting of an object by an animal (e.g. eat)
● EXPEL	Expulsion of something from the body of an animal (e.g. cry)
● MTRANS	Transfer of mental information (e.g. tell)
● MBUILD	Building new information out of old (e.g decide)
● SPEAK	Producing of sounds (e.g. say)
● ATTEND	Focusing of a sense organ toward a stimulus (e.g. listen)

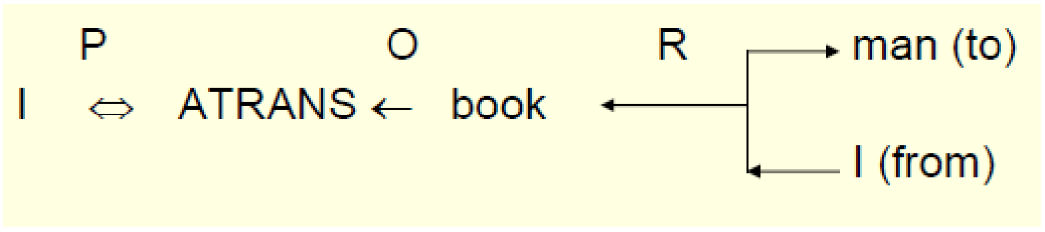
Conceptual category

- There are four conceptual categories

— ACT	Actions {one of the CD primitives}
— PP	Objects {picture producers}
— AA	Modifiers of actions {action aiders}
— PA	Modifiers of PP’s {picture aiders}

Example:

I gave a book to the man. CD representation is as follows:



- It should be noted that this representation is same for different saying with same meaning. For example
  - I gave the man a book,
  - The man got book from me,
  - The book was given to man by me etc.

Few conventions

- Arrows indicate directions of dependency
- Double arrow indicates two way link between actor and action.

O – for the object case relation

R – for the recipient case relation

P – for past tense

D - destination

Some of Conceptualizations of CD

- Dependency structures are themselves conceptualization and can serve as components of larger dependency structures.
- The dependencies among conceptualization correspond to semantic relations among the underlying concepts.
- We will list the most important ones allowed by CD.

Remaining can be seen from the book.

Rule 1: **PP $\Leftrightarrow$ ACT**

- It describes the relationship between an actor and the event he or she causes.
  - This is a two-way dependency, since neither actor nor event can be considered primary.

– The letter P in the dependency link indicates past tense.

- Example: John ran

CD Rep: $\text{John} \overset{P}{\Leftrightarrow} \text{PTRANS}$

Rule 2: **ACT $\leftarrow$ PP**

- It describes the relationship between a ACT and a PP (object) of ACT.
  - The direction of the arrow is toward the ACT since the context of the specific ACT determines the meaning of the object relation.
  - Example: John pushed the bike

CD Rep: John $\Leftrightarrow$ PROPEL $\leftarrow$ bike

Rule 3: **PP $\leftrightarrow$ PP**

- It describes the relationship between two PP's, one of which belongs to the set defined by the other.
- Example: John is doctor

CD Rep: John $\leftrightarrow$ doctor

Rule 4: **PP ← PP**

- It describes the relationship between two PP's, one of which provides a particular kind of information about the other.
  - The three most common types of information to be provided in this way are possession (shown as POSS-BY), location (shown as LOC), and physical containment (shown as CONT).
  - The direction of the arrow is again toward the concept being described.
  - Example: John's dog

CD Rep $\text{dog} \xleftarrow{\text{poss-by}} \text{John}$

Rule 5: **PP ⇔ PA**

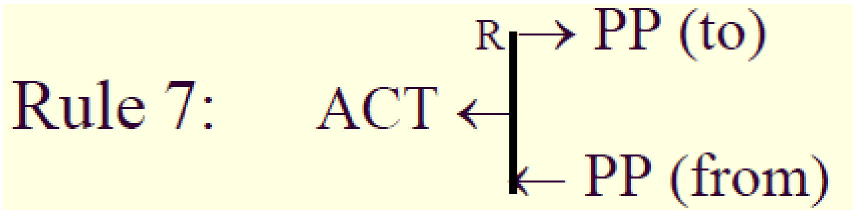
- It describes the relationship between a PP and a PA that is asserted to describe it.
  - PA represents states of PP such as height, health etc.
- Example: John is fat

  CD Rep John ⇔ weight (> 80)

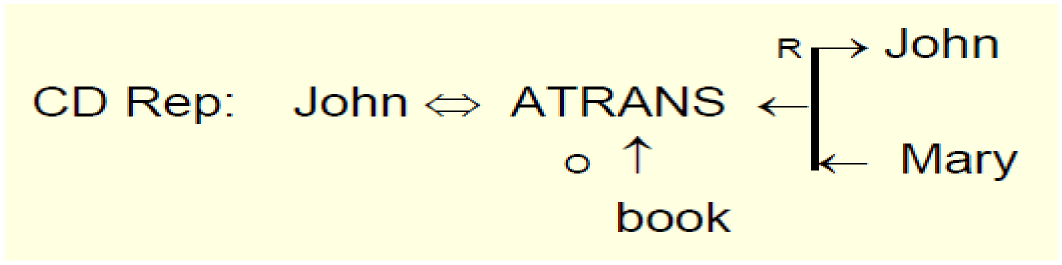
Rule 6: **PP ← PA**

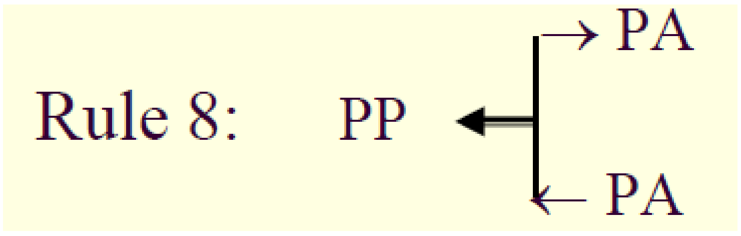
- It describes the relationship between a PP and an attribute that already has been predicated of it.
  - Direction is towards PP being described.
  - Example: Smart John

CD Rep John ← smart

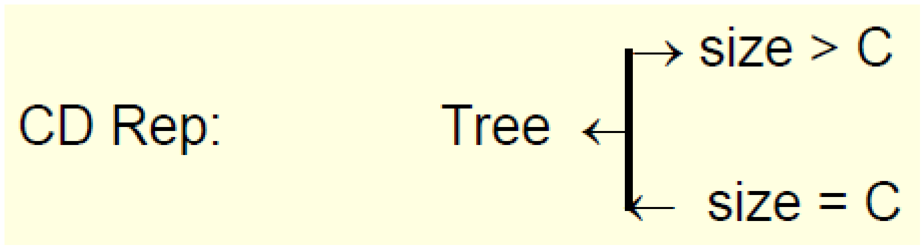


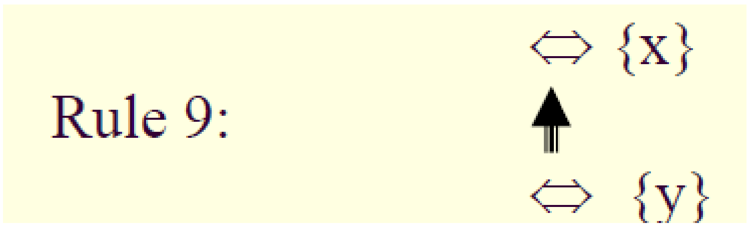
- It describes the relationship between an ACT and the source and the recipient of the ACT
  - Example: John took the book from Mary





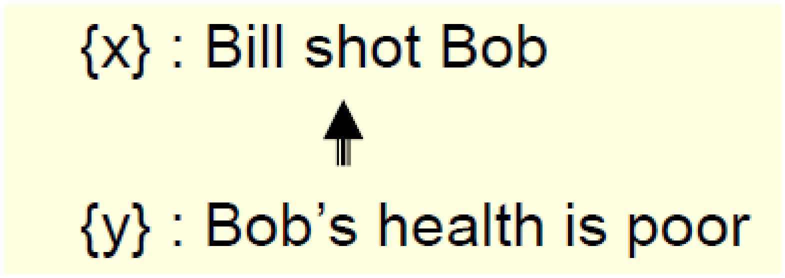
- It describes the relationship that describes the change in state.
  - Example: Tree grows





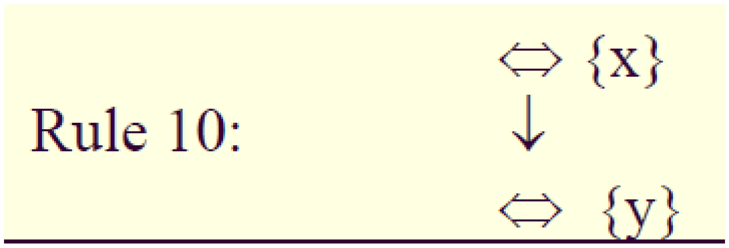
- It describes the relationship between one conceptualization and another that causes it.
  - Here {x} is causes {y} i.e., if x then y

  Example: Bill shot Bob


{x} : Bill shot Bob

$$\Uparrow$$

{y} : Bob's health is poor


Rule 10:

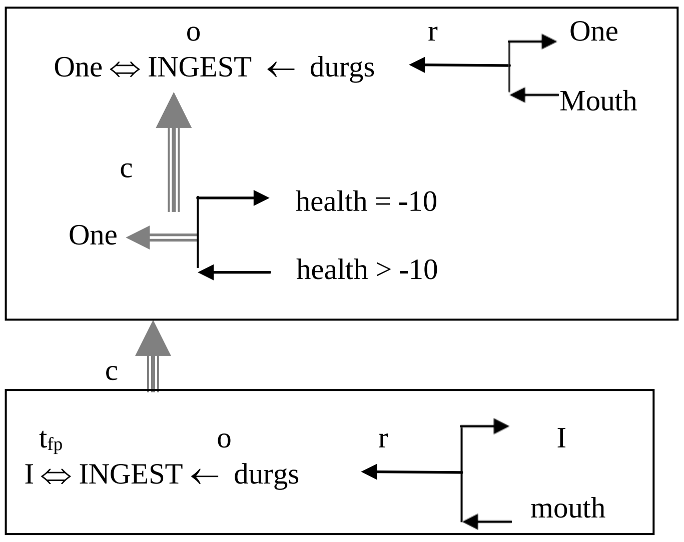
$$\begin{tikzcd} \Leftrightarrow \{x\} \arrow[d] \\ \Leftrightarrow \{y\} \end{tikzcd}$$

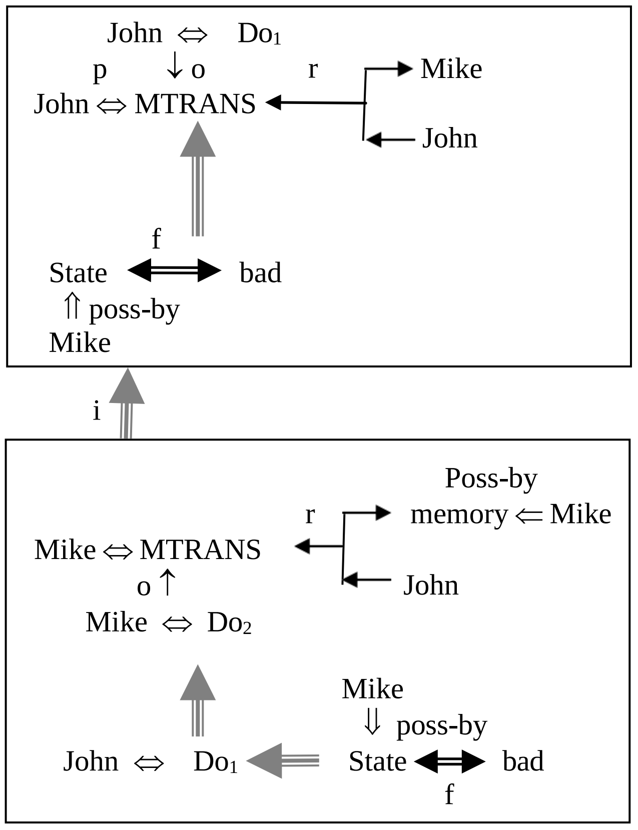
- It describes the relationship between one conceptualization with another that is happening at the time of the first.
  - Here {y} is happening while {x} is in progress.
  - Example: While going home I saw a snake

$$\begin{tikzcd} \text{I am going home} \arrow[d] \\ \text{I saw a snake} \end{tikzcd}$$

4.7.2. Generation of CD representations:

Sentences	CD Representations
Jenny cried	$\begin{tikzcd} \text{Jenny} \arrow[r, Leftrightarrow, "p"] & \text{EXPEL} & \text{tears} \arrow[l, "o"'] & {} \arrow[l, "d"'] \arrow[r] & ? \\ & & & & \text{eyes} \arrow[ul] \\ & & & & \text{Jenny} \arrow[u, Rightarrow, "\text{poss-by}"] \end{tikzcd}$
Mike went to India	$\begin{tikzcd} \text{Mike} \arrow[r, Leftrightarrow, "p"] & \text{PTRANS} & {} \arrow[l, "d"'] \arrow[r] & \text{India} \\ & & & \text{? (source is unknown)} \arrow[ul] \end{tikzcd}$
Mary read a novel	$\begin{tikzcd} \text{Mary} \arrow[r, Leftrightarrow, "p"] & \text{MTRANS} & \text{info} \arrow[l, "o"'] & {} \arrow[l, "d"'] \arrow[r] & \text{CP(Mary)} \\ & & & & \text{novel} \arrow[ul] \\ & \text{i (instrument)} \arrow[uu] \end{tikzcd}$ $\boxed{\begin{tikzcd} \text{Mary} \arrow[r, Leftrightarrow, "p"] & \text{ATTEND} & \text{eyes} \arrow[l, "o"'] & {} \arrow[l, "d"'] \arrow[r] & \text{novel} \\ & & & & ? \arrow[ul] \end{tikzcd}}$

Sentence	CD Representation
Since drugs can kill, I stopped.	 o r One One ⇔ INGEST ← durgs Mouth c health = -10 One health > -10 c tfp o r I I ⇔ INGEST ← durgs mouth

Sentence	CD Representation
John warned Mike with dire consequence.	 John ⇔ Do1 p ↓ o r Mike John ⇔ MTRANS John f State bad ⇑ poss-by Mike i Poss-by r memory ⇐ Mike Mike ⇔ MTRANS o ↑ John Mike ⇔ Do2 Mike ⇓ poss-by John ⇔ Do1 State bad f

4.7.3. Inferences Associated with Primitive Act

- General inferences are stored with each primitive Act thus reducing the number of inferences that need to be stored explicitly with each concept.
- For example, from a sentence "John killed Mike", we can infer that "Mike is dead".
- Let us take another example of primitive Act INGEST.
- The following inferences can be associated with it.
  - The object ingested is no longer available in its original form.
  - If object is eatable, then the actor has less hunger.
  - If object is toxic, then the actor's heath is bad.
  - The physical position of object has changed. So PTRANS is inferred.
- Example: The verbs {give, take, steal, donate} involve a transfer of ownership of an object.
  - If any of them occurs, then inferences about who now has the object and who once had the object may be important.
  - In a CD representation, these possible inferences can be stated once and associated with the primitive ACT "ATRANS".
- ■ Consider another sentence "Bill threatened John with a broken nose"
  - Sentence interpretation is that Bill informed John that he (Bill) will do something to break john's nose.
  - Bill did (said) so in order that John will believe that if he (john) does some other thing (different from what Bill wanted) then Bill will break John's nose.

Problems with CD Representation:

- It is difficult to
  - construct original sentence from its corresponding CD representation.

  - CD representation can be used as a general model for knowledge representation, because this theory is based on representation of events as well as all the information related to events.

- Rules are to be carefully designed for each primitive action in order to obtain semantically correct interpretation.
- Many verbs may fall under different primitive ACTs, and it becomes difficult to find correct primitive in the given context.
- The CD representation becomes complex requiring lot of storage for many simple actions.
- For example, the sentence “John bet Mike that Indian cricket team will win incoming world cup” will require huge CD structure.

4.7.4. Conceptual Parsing

- Conceptual parsing is required for generating CD representation from source sentences in natural language.
- The main steps involved in CD parsing are as follows:
  - Syntactic processor extracts main verb and noun along with syntactic category of the verb (transitive or intransitive) from the sentence.
  - Conceptual processor then makes use of verb–ACT dictionary.
  - Once the correct entry from dictionary is chosen, CD processor analyses the rest of sentence looking for arguments for empty slots of the verb.
  - CD processor examines possible interpretation in a well-defined order.

Example:

- **Case1**: Handling of ‘with PP’ phrase by CD processor and formulating strategies to disambiguate the meanings.
  - **Type1**: John broke the door with **hammer** non animate
  - **Type2:** John broke the door with **Mike** animate

- **Rule 1:** If PP in 'with PP' phrase is non-animate and CD Act requires instrument then the sentence is of Type1, where PP (hammer) is resolved to instrument.
- **Rule 2:** If PP in 'with PP' phrase is animate and CD Act requires instrument then the sentence is of Type2, where PP (Mike) is resolved as co-actor.
- **Case2:** If PPs in both the sentences are non-animate, then they have to be resolved using semantic lexicon.
  - **Type3:** John went to the garden *with flowers*
  - **Type4:** John went to the garden *with bag*
- In Type3, non-animate noun 'flowers' is part of garden, whereas in Type4, non-animate 'bag' is some object not related to garden.
- Such association of word senses could be found in Word-Net and then disambiguation is possible.
- Here noun 'bag' is treated as possession by John.
- Case 3: If PPs in the sentences are animate, then they have to be resolved using semantic lexicon and context.
- Consider the following examples.
  - **Type5:** John went to the *garden with Mike*
  - **Type6:** John went to the *garden with butterflies*
  - **Type7:** John went to the *garden with dog*
- In these sentences, Mike, butterflies and dog are animate PPs and can be resolved as follows:

Mike is easily resolved to co-actor of John as both are human and have similar characteristics

- Word-Net can be used to check if butterfly and garden has some common sense.
- Dog is still ambiguous.

- It may be treated as possession of actor or may be a part of garden as animals many wonder in garden.
- Such situations can be further resolved by considering the context of sentences.
- We can use semantic lexicon dictionary to resolve some of the ambiguities.

4.8.Script Structure:

- Scripts were introduced by Schank and Abelson introduced in 1977 that used CD framework.
- The scripts are useful in describing certain stereotyped situations such as going to theater
- It consists of set of slots containing default values along with some information about the type of values similar to frames.
- It differs from FS as the values of the slots in scripts must be ordered and have more specialized roles.
- In real world situations, we see that event tends to occur in known patterns because of clausal relationship to the occurrence of events

4.8.1. Script Components:

- Each script contains the following main components.
  - ***Entry Conditions***: Must be satisfied before events in the script can occur.
  - ***Results:*** Conditions that will be true after events in script occur.
  - ***Props:*** Slots representing objects involved in the events.
  - ***Roles:*** Persons involved in the events.
  - ***Track:*** Specific variation on more general pattern in the script. Different tracks may share many components of the same script but not all.
  - ***Scenes:*** The sequence of *events* that occur. Events are represented in conceptual dependency form.

Script : Play in theater	**Various Scenes**
Track: Play in Theater **Props:** • Tickets • Seat • Play **Roles:** • Person (who wants to see a play) – P • Ticket distributor – TD • Ticket checker – TC **Entry Conditions:** • P wants to see a play • P has a money **Results:** • P saw a play • P has less money • P is happy (optional if he liked the play)	***Scene 1: Going to theater*** • P PTRANS P into theater • P ATTEND eyes to ticket counter ***Scene 2: Buying ticket*** • P PTRANS P to ticket counter • P MTRANS (need a ticket) to TD • TD ATRANS ticket to P ***Scene 3: Going inside hall of theater and sitting on a seat*** • P PTRANS P into Hall of theater • TC ATTEND eyes on ticket POSS_by P • TC MTRANS (showed seat) to P • P PTRANS P to seat • P MOVES P to sitting position ***Scene 4: Watching a play*** • P ATTEND eyes on play • P MBUILD (good moments) from play ***Scene5: Exiting*** • P PTRANS P out of Hall and theater

Script Invocation:

- It must be activated based on its significance.
- If the topic is important, then the script should be opened.
- If a topic is just mentioned, then a pointer to that script could be held.
- For example, given “John enjoyed the play in theater”, a script “Play in Theater” suggested above is invoked.
- All implicit questions can be answered correctly.
- Here the significance of this script is high.
  - Did john go to theater?

    - Did he buy ticket?
    - Did he have money?
- If we have a sentence like “John went to theater to pick his daughter”, then invoking this script will lead to many wrong answers.
    - Here significance of the script theater is less.

Getting significance from the story is not straightforward. However, some heuristics can be applied to get the value.

Advantages / Disadvantages of Script

- Advantages
    - Capable of predicting implicit events
    - Single coherent interpretation may be build up from a collection of observations.
- Disadvantage
    - More specific (inflexible) and less general than frames.
    - Not suitable to represent all kinds of knowledge.
- To deal with inflexibility, smaller modules called memory organization packets (MOP) can be combined in a way that is appropriate for the situation.

4.9.CYC

What is CYC?

- An ambitious attempt to form a very large knowledge base aimed at capturing commonsense reasoning.
- Initial goals to capture knowledge from a hundred randomly selected.
- Both Implicit and Explicit knowledge encoded.

Example: Suppose we read that *Wellington learned of Napoleon's death*

Then we (humans) can conclude *Napoleon never new that Wellington had died*.

How do we do this?

We require special implicit knowledge or commonsense such as:

- We only die once.
- You stay dead.
- You cannot learn of anything when dead.
- Time cannot go backwards.

Why build large knowledge bases:

Brittleness

-- Specialised knowledge bases are *brittle*. Hard to encode new situations and non-graceful degradation in performance. Commonsense based knowledge bases should have a firmer foundation.

Form and Content

-- Knowledge representation may not be suitable for AI. Commonsense strategies could point out where difficulties in content may affect the form.

Shared Knowledge

-- Should allow greater communication among systems with common bases and assumptions.

How is CYC coded?

- By hand.
- Special CYCL language:
  - LISP like.
  - Frame based
  - Multiple inheritance
  - Slots are fully fledged objects.
  - Generalised inheritance -- any link not just *isa* and *instance*.

4.10.CASE GRAMMAR

- **Case grammar** is a system of linguistic analysis, focusing on the link between the valence, or number of subjects, objects, etc., of a verb and the grammatical context it requires. The system was created by the American linguist Charles **J.** Fillmore in (1968), in the context of Transformational Grammar. This theory analyzes the surface syntactic structure of sentences by studying the combination of deep cases (i.e. semantic roles) -- Agent, Object, Benefactor, Location or Instrument—which are required by a specific verb. For instance, the verb "give" in English requires an Agent (A) and Object (O), and a Beneficiary (B); e.g. "Jones (A) gave money (O) to the school (B).
- According to Fillmore, each verb selects a certain number of deep cases which form its **case frame**. Thus, a case frame describes important

aspects of semantic valency, of verbs, adjectives and nouns. Case frames are subject to certain constraints, such as that a deep case can occur only once per sentence. Some of the cases are obligatory and others are optional. Obligatory cases may not be deleted, at the risk of producing ungrammatical sentences. For example, *Mary gave the apples* is ungrammatical in this sense.

- A fundamental hypothesis of case grammar is that grammatical functions, such as subject or object, are determined by the deep, semantic valence of the verb, which finds its syntactic correlate in such grammatical categories as Subject and Object, and in grammatical cases such as Nominative, Accusative, etc. Fillmore (1968) puts forwards the following hierarchy for a universal subject selection rule:
- Agent < Instrumental < Objective
- That means that if the case frame of a verb contains an agent, this one is realized as the subject of an active sentence; otherwise, the deep case following the agent in the hierarchy (i.e. Instrumental) is promoted to subject.
- The influence of case grammar on contemporary linguistics has been significant, to the extent that numerous linguistic theories incorporate deep roles in one or other form, such as the so-called Thematic structure in Government and Binding theory. It has also inspired the development of frame-based representations in AI research

4.11.SEMATIC WEB

The Semantic Web is an extension of the World Wide Web through standards set by the World Wide Web Consortium (W3C). The goal of the Semantic Web is to make Internet data machine-readable. To enable the encoding of semantics with the data, technologies such as Resource Description Framework (RDF) and Web Ontology Language (OWL) are used. These technologies are used to formally represent metadata. For example, ontology can describe concepts, relationships between entities, and categories of things. These embedded semantics offer significant advantages such as reasoning over data and operating with heterogeneous data sources.

These standards promote common data formats and exchange protocols on the Web, fundamentally the RDF. According to the W3C, "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries." The Semantic Web is therefore regarded as an integrator across different content and information applications and systems.

The term was coined by Tim Berners-Lee for a web of data (or data web) that can be processed by machines—that is, one in which much of the meaning is machine-readable. While its critics have questioned its feasibility, proponents argue that applications in library and information science, industry, biology and human sciences research have already proven the validity of the original concept.

In addition to the classic “Web of documents” W3C is helping to build a technology stack to support a “Web of data,” the sort of data you find in databases. The ultimate goal of the Web of data is to enable computers to do more useful work and to develop systems that can support trusted interactions over the network. The term “Semantic Web” refers to W3C’s vision of the Web of linked data. Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data. Linked data are empowered by technologies such as RDF, SPARQL, OWL, and SKOS.