

MCA Part III

Paper- XXI

Topic: Structured Knowledge Representation

Prepared by: Dr. Kiran Pandey

School of Computer science

Email-Id: kiranpandey.nou@gmail.com

INTRODUCTION

Knowledge can be represented in different ways. The production rules, frames, scripts, semantic network are used in representation of knowledge. We will discuss some of them.

PRODUCTION RULES

A production system consists of four basic components:

- (i). **A set of rules:** It is of the form **COND(i) → ACT(i)** where **COND(i)** is the condition part and **ACT(i)** is the action part. The condition determines when a given rule is applied, and the action determines what happens when it is applied.
- (ii). **Knowledge databases:** It consist of one or more than one knowledge database that contain relevant information for the given problem. Some parts of the database may be permanent, while others may temporary and only exist during the solution of the current problem.

(iii). **A control strategy:** This determines the order in which the rules are applied to the database, and provides a way of resolving any conflicts that can arise when several rules match at once.

(iv). **A rule applier:** It is the computational system that implements the control strategy and applies the rules.

SEMANTIC NETS (ASSOCIATIVE NETWORK)

Network representation provide a means of structuring and exhibiting the structure in knowledge. A semantic network or a semantic net is a structure for representing knowledge as a pattern of interconnected nodes and arcs. It is also defined as a graphical representation of knowledge. The objects under consideration are presented as nodes and the relationships with another nodes give the arcs. In a semantic net, information is represented as a set of nodes connected to each other by a set of labels, which represent relationships among the nodes. A fragment of a typical semantic net is shown in Figure below:

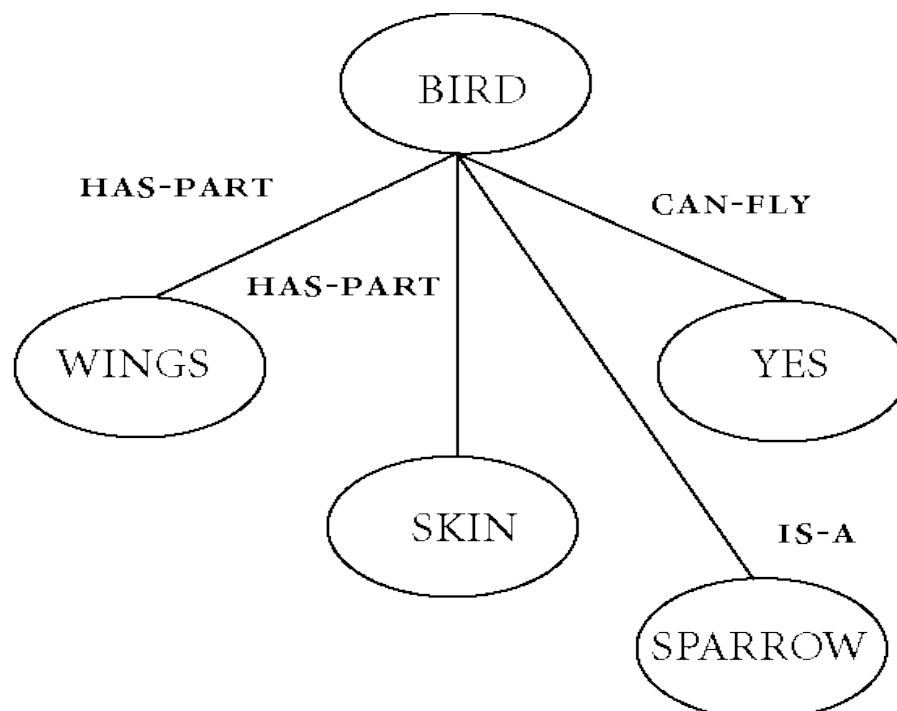


Figure 1: A semantic network

This network contains example of the **is-a** instance relations, as well as some other relationships such as **has-parts** and **can-fly**.

FRAMES

A frame structure is used to analyze new situations from scratch and then build a new knowledge structure to describe those situations. A frame in other words, describes a class of objects, such as **CHAIR** or **ROOM**. It consists of a collection of “slots” that describe aspects of the objects. Associated with each slot there can be a set of conditions that must be met by any filler for it. Each slot may also be filled with a default value, so that in the absence of specific information, things can be associated to be as they usually are. Procedural information may also be associated with particular slots. The AI systems exploit not one but many frames. Related frames can be grouped together to form a frame system.

Frames represent an object as a group of attributes. Each attributes in a particular frame is stored in a separate slot. For example, when a furniture salesman says “I have a nice chair that I want you to see”, the word ‘chair’ would immediately trigger in our minds a series of expectations. We would probably expect to see an object with four legs, a seat, a back and possibly (but not necessarily) two arms. We would expect it to have a particular size and serves a place to sit. In an AI system, a frame **CHAIR** might include knowledge organized as shown below:

Frame: TABLE

Parts: legs and top-platform

Number of legs: 4

Color : Wooden polish

Default: 0

Example: A frame for a book is given below:

Slots	Fillers
Publisher Name	
Title	
Author Name	
Print_Edition	
Year	
No. of pages	

The above example is a simple one. Some frames are of generic type. The frame given below is generic type:

Slot	Fillers
name	computer
description	Is_a machine
types	(super, mainframe, mini, desktop, laptop)
speed	fast
location	(office and home)
warranty	(yes, no)

CONCEPTUAL DEPENDENCIES AND SCRIPTS

The theory of Conceptual Dependency (**CD**) was developed by Schank in-between 1973 to 1975. This theory was developed to represent the meaning of Natural Language sentences. It helps in drawing inferences and 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

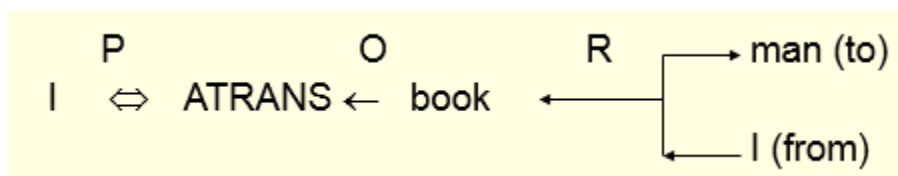
- **ATRANS** Transfer of an abstract relationship (i.e. give)

- **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

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. For further readings refer to the suggested readings

Rule 1: PP ⇔ 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:** Shyam ran

P

CD Rep: Shyam ⇔ PTRANS

Rule 2: ACT ← 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: Shyam pushed the bike

O

CD Rep: Shyam ⇔ PROPEL ← bike

Rule 3: PP ↔ PP

It describes the relationship between two PP's, one of which belongs to the set defined by the other.

Example: Shyam is doctor

CD Rep: Shyam ↔ 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: Shyam's dog

CD Rep: dog poss-by ← Shyam

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: Shyam is fat

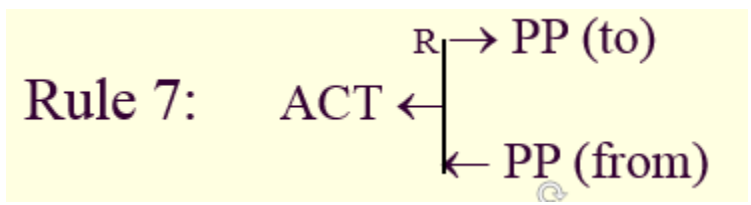
CD Rep: Shyam ↔ weight (> 80)

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.

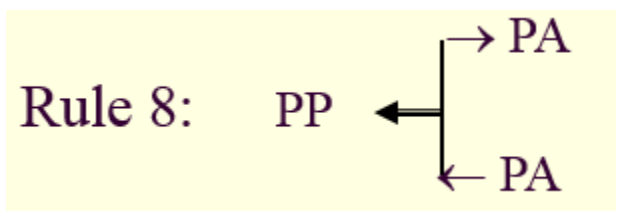
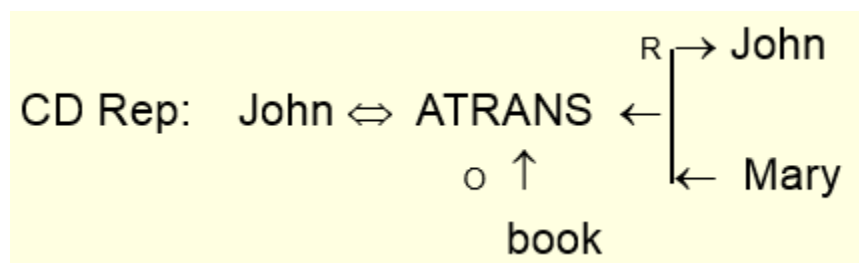
Example: Smart Shyam

CD Rep Shyam ← smart



It describes the relationship between an ACT and the source and the recipient of the ACT

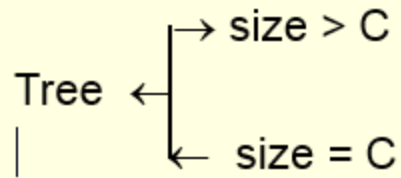
Example: Shyam took the book from Mira.



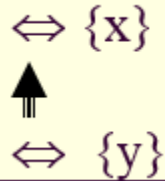
- It describes the relationship that describes the change in state.

Example: Tree grows

CD Rep:



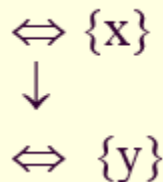
Rule 9:



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

Example: Bill shot Bob
 $\{x\}$: Bill shot Bob
 $\{y\}$: Bob's health is poor

Rule 10:



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

I am going home



I saw a snake

Problems with CD Representation

- (i) 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.
- (ii) Rules are to be carefully designed for each primitive action in order to obtain semantically correct interpretation.
- (iii) Many verbs may fall under different primitive ACTs, and it becomes difficult to find correct primitive in the given context.
- (iv) The CD representation becomes complex requiring lot of storage for many simple actions.

For example, the sentence “Shyam bet Ram that Indian cricket team will win incoming world cup” will require huge CD structure.

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:** Shyam broke the door with **hammer** non animate
 - **Type2:** Shyam broke the door with **Mira** 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 (Mira) 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:** Shyam went to the garden with flowers
 - **Type4:** Shyam 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 Shyam.

- 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:** Shyam went to the garden with Ram
 - **Type5:** Shyam went to the garden with butterflies
 - **Type7:** Shyam went to the garden with dog

In these sentences, Ram, butterflies and dog are animate PPs and can be resolved as follows:

- Ram is easily resolved to co-actor of Shyam 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.

SCRIPTS

The concept of scripts was given by Schank and Abelson in 1977. It used Conceptual Dependency framework. A script is also a knowledge representation scheme and is similar to a frame discussed earlier, but instead of describing an object, the script describes a sequence of events. Like the frame, the script portrays a stereotyped situation. Unlike the frame, it is usually presented in a particular context. To describe a sequence of events, the script uses a series of slots containing information about the people, objects, and actions that are

involved in the events. Script representation is interrelated with case-based reasoning. It consists of set of slots containing default values along with some information about the type of values similar to frames. In real world situations, we see that event tends to occur in known patterns because of clausal relationship to the occurrence of events.

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
<p>Track: Play in Theater</p> <p>Props:</p> <ul style="list-style-type: none"> • Tickets • Seat • Play <p>Roles:</p> <ul style="list-style-type: none"> • Person (who wants to see a play) – P • Ticket distributor – TD • Ticket checker – TC <p>Entry Conditions:</p> <ul style="list-style-type: none"> • P wants to see a play • P has a money <p>Results:</p> <ul style="list-style-type: none"> • P saw a play • P has less money • P is happy (optional if he liked the play) 	<p>Scene 1: Going to theater</p> <ul style="list-style-type: none"> • P PTRANS P into theater • P ATTEND eyes to ticket counter
	<p>Scene 2: Buying ticket</p> <ul style="list-style-type: none"> • P PTRANS P to ticket counter • P MTRANS (need a ticket) to TD • TD ATRANS ticket to P
	<p>Scene 3: Going inside hall of theater and sitting on a seat</p> <ul style="list-style-type: none"> • 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
	<p>Scene 4: Watching a play</p> <ul style="list-style-type: none"> • P ATTEND eyes on play • P MBUILD (good moments) from play
	<p>Scene5: Exiting</p> <ul style="list-style-type: none"> • 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 “Shyam 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.
- (i) Did Shyam go to theater?
 - (ii) Did he buy ticket?
 - (iii) Did he have money?

- If we have a sentence like “Shyam went to theater to pick his daughter”, then invoking this script will lead to many wrong answers.
 - (i) 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 of Script

- Capable of predicting implicit events
- Single coherent interpretation may be build up from a collection of observations.

Disadvantages of Scripts

- More specific (inflexible) and less general than frames.
- Not suitable to represent all kinds of knowledge.

QUESTIONS

1. Explain and distinguish between the following: -
 - a. Associative Network Structure
 - b. Frame Structure
2. What are the main difference between scripts and frame structure?
3. Write short note on the following:-
 - a. Exception & Defaults
 - b. Semantic Net
4. Explain the concept of conceptual dependency with examples.
5. Write a script for a scene at the booking counter of a cinema hall.