

Advanced Knowledge Representation Techniques

15.1 Introduction

We have discussed various knowledge representation formalisms in Chapter 7 and seen that knowledge can be represented in the declarative structures, viz. Predicate Logic, Production rules, Semantic Networks, Extended Semantic Nets and Frames. Basically, knowledge representation is an area of artificial intelligence whose fundamental goal is to represent knowledge in a manner that helps drawing conclusions from that knowledge. Generally, the knowledge representation system should not only hold knowledge but also set of basic inference rules. Inference rules are syntactic procedures or functions that operate on knowledge and derive new information. Syntactic information of rules is also used by the production system to decide which rule is applicable to be fired. Predicate logic and production rules are syntactic in nature. Syntactic-structure-based knowledge representation methods are weak methods but at the same time are adequate for general problem solving. Semantically oriented approaches for knowledge representation are quite useful to guide searches more efficiently. Semantic nets capture semantic relationship among concepts (entities) and are used with set of inference rules, which interpret the specific arcs showing relationship. The frame system is found to be more structured than semantic net and contains larger set of inference rules with the capability of integrating other knowledge structures (such as predicate rules, semantic nets, and production rules etc.) within frames in order to make it more powerful. The inheritance capability is much faster than resolution-based theorem proving in predicate logic. A hybrid structure can be created that combines the advantages of slot-and-filler structure of semantic net with the advantages of predicate logic. Extended semantic network (ESNet), CYC, and Ontology are some examples of such structures. The ESNet incorporates clausal rules in

semantic net; thus enhancing the capabilities of semantic nets using advantages of predicate logic. There is another system called KRYPTON (Brachman et al., 1985) in which knowledge is divided into two distinct repositories: *TBox* and *ABox*. Here *TBox* is a slot-and-filler structure containing concept information, whereas *ABox* contains logical assertions. The atomic predicates used in *ABox* assertions refer to concept defined in *TBox*.

These representations are inadequate for representing knowledge required for core Natural Language understanding applications. We require structures to hold deep sense and meaning of the sentences. For capturing indented semantics of the knowledge, there are various approaches, viz., Conceptual Dependency, Scripts, and Case Frames.

This chapter discusses Conceptual Dependency, Scripts, CYC, Case Frames, Semantic web, and Ontology frameworks. *Conceptual Dependency* is a semantic representation rather than a syntactic representation. It provides abstract structure with actions and the dependency relationship. Script structure is similar to frames except that slots in script keep information by using conceptual dependency primitives that are useful for reasoning about situations. The CYC uses both frames and logic to encode specific types of knowledge and inferences for *commonsense reasoning*. It provides the most built-in knowledge to manipulate specific kinds of knowledge structures. Ontology is again a structure holding detailed semantic knowledge and can be viewed as an extension of semantic net.

15.2 Conceptual Dependency Theory

Conceptual Dependency (CD) is a theory of deep semantics for concepts used for natural language understanding, which was developed by Roger Schank at Stanford University in 1969. It mainly focuses on the concept and meaning rather than on syntax or structure. Conceptual Dependency uses the basic hypothesis, that the ACTION is the basis of any proposition. All propositions that describe events are made up of conceptualizations that comprise an *actor*, an *action*, and a set of *cases* that depends on that action. The main important concept of the CD theory is that there must be a canonical form for meaning representations, i.e., different words and structures having same meaning should also have the same representation. The types of conceptual roles in this canonical form do not correspond to syntactic roles; rather the words are broken down into their conceptual parts and are then placed in a meaning representation with respect to their conceptual role instead of the syntactic role. It should be noted that the words do not appear in conceptualizations. Similarities and overlaps in meaning of words are handled by the concept of the PRIMITIVE ACTION. The primitives are not category names but rather elements that can be used in varied combinations to express the underlying meaning of a given word.

Conceptual Dependency representations use eleven primitive acts as the basis for most activities. These acts were designed by Schank by observing the structural similarities in the sentences

represented in actor-action-object framework (Schank R., 1969). A large set of states and a set of conceptual roles can be expressed using this minimal set of primitive actions.

15.2.1 Conceptual Primitive Actions

Actions in natural languages are generally specified by the verbs. Most of the verbs (in English language) have been categorized under the following eleven primitive actions. Sometimes one verb may fall into two categories and in that case the most relevant in the context might be chosen at the time of interpreting sentences. For example, verbs *push*, *throw*, and *kick* require physical force to be applied to an object and may fall under PROPEL Act. But MOVE and GRASP primitives are more specific where the verb *kick* falls under MOVE and verb *throw* falls under GRASP. These primitives ACTIONS are listed as follows:

1. ATRANS
2. PTRANS
3. PROPEL
4. MOVE
5. GRASP
6. INGEST
7. MTRANS
8. MBUILD
9. EXPEL
10. SPEAK
11. ATTEND

Let us describe all the primitives proposed by Schank as follows:

ATRANS: Transfer of an abstract relationship such as possession, control, or ownership. It requires an actor, object, and recipient. For example,

- The verb '*give*' is represented by an act ATRANS which transfers an object from an actor to the recipient.
- The verb '*take*' is represented by an act ATRANS which transfers an object from someone to an actor.
- The verb '*purchase*' can be represented by two ATRANS acts, each causing the other. First, act ATRANS is used for transferring money from the first actor to the second actor, and, second, act ATRANS is used for transferring of an object from the second to the first actor.

PTRANS: Transfer of the physical location of an object (or actor) that requires an actor, object, and direction. For example,

- The verbs 'go' or 'walk' are type of act PTRANS where an actor is moving himself to some location.
- The verb 'fly' is represented by an act PTRANS where an actor is transferring himself to a location. It can use another act as its instrument such as airplane, etc.

PROPEL: Application of physical force to an object (e.g., push, pull). For example,

- The verb 'push' is a PROPEL act that pushes an object in a direction by an actor.
- The verb 'pull' is represented by an act PROPEL where an object is pulled in a direction (towards actor) by an actor.
- The verb 'throw' is represented by PROPEL act where an object is moving in a direction by an actor using MOVE action whose instrument is hand.

MOVE: Movement of a body part by its owner (e.g., kick, throw, hit). For example,

- 'kick' is represented by MOVE owner's foot to PTRANS an object in a direction by owner of foot.

GRASP: Grasping of an object by an actor (e.g., catch, clutch). For example,

- 'catch' is represented by GRASP of an object by an actor. The object was PROPEL before GRASP.

INGEST: Ingesting of an object by an animal (e.g., eat, drink, smoke). It requires an actor, object, and direction. For example,

- INGEST (eat or drink) a solid eatable or liquid into the body through the mouth.
- INGEST smoke from a cigarette into the mouth by INGEST air through the cigarette.
- INGEST (breathe) air into the body through the mouth.

MTRANS: Transfer of mental information between animals or within an animal itself (e.g., tell, read, speak, sing). In MTRANS action, the various sense organs serve as the origins. Recipient, object, and actor are required, although here these cases refer to the above mental locations as recipients and mental information as possible objects. For example,

- 'tell', 'speak', and 'sing' mean that an actor MTRANS information to another actor.
- 'read' MTRANS information from a book to memory (IM: intermediate memory where current context is kept) of actor by means of ATTEND eye to the book for reading purpose.
- 'remember' MTRANS information from memory (LTM: long term memory) to IM of the same actor.

MBUILD: The construction of new information from old information within an actor, i.e. to create or combine thoughts internally. It takes place within IM and receives its inputs from the CP (the conscious processor) and from IM. It transforms them into a new idea and places that idea in the CP. For example,

- 'decide' MBUILD new information out of old information.
- 'describe' MBUILD a description of a new state.
- 'imagine' MBUILD a new idea that is hypothetical.
- 'consider' MBUILD from a mentioned input into an unspecified output.
- 'answer' MBUILD a specific output to input given by other that was just received in the CP.

EXPEL: Expulsion of something from the body of an animal. For example,

- 'weep' EXPEL tears (discharge liquid) from eyes.
- 'cry' EXPEL tears and produce sound from mouth.
- 'sweat' EXPEL moisture out of the pores of the skin.

SPEAK: Producing sounds (e.g. say, tell, and sing). For example,

- 'say', 'tell', or 'sing' SPEAK words so as to communicate something to other by vocal cord of an actor.

ATTEND: Focusing of a sense organ toward a stimulus (e.g. listen, watch, see, and look)

All these primitives are grouped and classified in the broad categories as shown in Table 15.1.

Table 15.1 Grouping of Primitives

Category	Primitive Acts
State Change Primitives	ATRANS and PTRANS
Mental Action Primitives	MTRANS and MBUILD
Senses (often instruments for other Acts) such as eyes, ears, tongue, etc	ATTEND and SPEAK
Physical Action Primitives	INGEST, MOVE, GRASP, PROPEL, EXPEL

The primitive ACT's are so powerful that they can not only point out similarities between words such as the use of ATRANS for 'give', 'take', and 'buy' but also highlight their differences. In addition, CD structure also helps in disambiguating ambiguous words. We have described above only one sense of each verb.

For example, *take* in *take a medicine or juice* would be INGEST rather than ATRANS. Here the knowledge about how words fit together in English and some general knowledge about the physical world are used to make such distinctions. The correct conceptual structure to an input sentence can be assigned by asking series of questions about the nature of object around it: Whether an object is something to be ingested or can be ATRANS or is really an action such as *take a blow*? In this case, it is neither ATRANS or INGEST but a causal structure.

Similarly, *smoke in the jungle is smoking* (even though not a usual usage) would not have INGEST but should have description of a state involving *smoke* and *heat*.

15.2.2 Conceptual Category

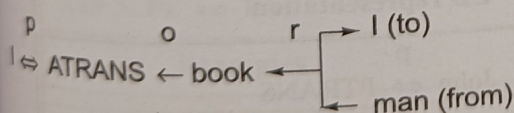
CD provides specific set of building blocks from which representations can be made rather than only a structure in which knowledge can be represented. Building blocks are the set of allowable dependencies among the conceptualizations for different events. There are four primitive conceptual categories from which dependency structures can be built.

• ACT	Actions {one of the CD primitives representing verb}
• PP	Objects {picture producers or noun/pronoun}
• AA	Modifiers of actions {action aiders or adverb}
• PA	Modifiers of PP's {picture aiders or adjective}

The relationships between concepts are called dependencies. The main conceptualization of a clause is a two-way dependency between a PP (the actor) and an action. It is important to note that actions are broken down into sequence of primitive ACT's.

A set of rules describe the syntax of the conceptual level, and these rules specify which type of concepts can depend on which other type, as well as the different kinds of dependency relationships between concepts. Specific concepts depending on other concepts based on the particular meaning of these concepts is determined by the semantics of the conceptual level. There exists a dictionary of ACT's which specifies different meanings with its conceptual structure for each verb.

Example: The CD representation for sentences such as "I took a book from the man", "The man gave me a book", and "The book was given by man to me" having the same intended meaning is given as follows:



Here we notice that some special notations and symbols are used. These will be explained as we move ahead, but the conventions used throughout this chapter are as follows:

- Arrows indicate directions of dependency.
- Double arrow indicates two-way links between actor and action.
- Following characters are used to represent different concepts
 - o – for the object case relation
 - r – for the recipient case relation
 - d – destination

Conceptual relations, at a higher level, indicate dependencies between conceptualizations, annotated with conceptual tenses such as past, future, and conditional. Other types of conceptual relations are the time and location of a conceptualization. So the conceptualizations representing events can be modified in various ways to supply information normally indicated in languages by the tense, mood, or aspect of a verb form. The set of conceptual tenses includes

- p – Past
- f – Future
- t – Transition
- ts – Start transition
- tf – Finished transition
- k – Continuing
- c – Conditional

15.2.3 Rules for Conceptualization Blocks in CD

Dependency structures are themselves conceptualization and therefore can serve as components for larger dependency structures. The dependencies among conceptualization correspond to semantic relations among the underlying concepts. Here different types of arrows are used to represent different underlying semantics. Following is the list of the most important allowable structures or the rules:

Rule 1: Rule representing relationship between an actor and the event caused by him or her is called a two-way dependency, as neither actor nor event can be considered primary. The letter p in the dependency link indicates that the event occurred in the past.

Rule 1: PP $\Leftrightarrow$ ACT	
Examples	CD representation
John ran	p John $\Leftrightarrow$ PTRANS
Mary cried	p Mary $\Leftrightarrow$ EXPEL

Rule 2: Rule representing relationship between an ACT and the PP (object) of the ACT is shown by the direction of an arrow toward the ACT, since the context of the specific ACT determines the meaning of the object relation.

Rule 2: ACT $\leftarrow$ PP	
Examples	CD representation
John gave a book to Mary	John $\Leftrightarrow$ PTRANS $\leftarrow$ book
Mary kicked the ball	Mary $\Leftrightarrow$ MOVE $\leftarrow$ ball

Rule 3: This rule shows relationship both ways between two PPs. One PP belongs to the set defined by the other PP. For example, John belongs to a set of doctors in the sentence "John is a doctor".

Rule 3: PP $\leftrightarrow$ PP	
Examples	CD representation
John is a doctor	John $\leftrightarrow$ doctor
Mary is a teacher	Mary $\leftrightarrow$ teacher

Rule 4: It shows a relationship between two PPs. One of the PP provides a particular kind of information about the other PP. The most common types of information to be encoded in this ways are possession (shown as 'poss-by') and location (shown as 'loc'). The direction of the arrow is again toward the concept being described.

Rule 4: PP $\leftarrow$ PP	
Examples	CD representation
John's car	poss-by car $\leftarrow$ John
Mary is in Delhi	loc Delhi $\leftarrow$ Mary

Rule 5: It shows a relationship between a PP and a PA that is asserted to describe it. In this case, PA represents the states of PP such as height, weight, health, etc. on numeric scales and has both ways arrows.

Rule 5: PP $\leftrightarrow$ PA	
Examples	CD representation
John is fat	John $\leftrightarrow$ weight (> 50)
Mary is poor	Mary $\leftrightarrow$ income (< 2000)

Rule 6: It shows a relationship between a PP and an attribute that already has been predicated of it. The direction of the arrow is toward the PP being described.

Rule 6: PP $\leftrightarrow$ PA	
Examples	CD representation
Smart John	John $\leftarrow$ smart
Charming Mary	Mary $\leftarrow$ charming

Rule 7: It shows a relationship between an ACT and its physical source and destination locations of ACT. Here 'd' indicates the source and destination case relation, as this representation is also used for recipient case relation later.

Rule 7: ACT $\xleftarrow{d}$ $\begin{cases} \text{PP (destination)} \\ \text{PP (source)} \end{cases}$	
Examples	CD representation
John went to school from home	$\text{John} \xleftrightarrow{p} \text{PTRANS} \xleftarrow{d} \begin{cases} \text{School} \\ \text{Home} \end{cases}$
Mary traveled from Delhi to Bombay by train	$\text{Mary} \xleftrightarrow{p} \text{PTRANS} \xleftarrow{d} \begin{cases} \text{Bombay} \\ \text{Delhi} \end{cases}$ $v \uparrow$ train

Here we can use 'v' for vehicle which can be single object or full conceptualization.

Rule 8: It shows a relationship between an ACT and its source and the recipient of ACT. The letter 'r' indicates source and recipient case relation.

Rule 8: ACT $\xleftarrow{r}$ $\begin{cases} \text{PP (to)} \\ \text{PP (from)} \end{cases}$	
Examples	CD representation
John gave a book to Mary	$\text{John} \xleftrightarrow{p} \text{ATRANS} \xleftarrow{o} \text{book} \xleftarrow{r} \begin{cases} \text{Mary} \\ \text{John} \end{cases}$
Mary told story to John	$\text{Mary} \xleftrightarrow{p} \text{MTRANS} \xleftarrow{o} \text{story} \xleftarrow{r} \begin{cases} \text{John} \\ \text{Mary} \end{cases}$

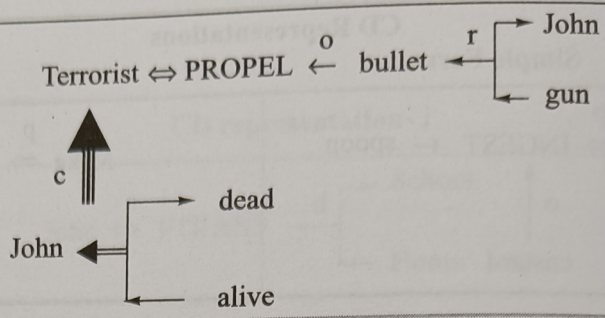
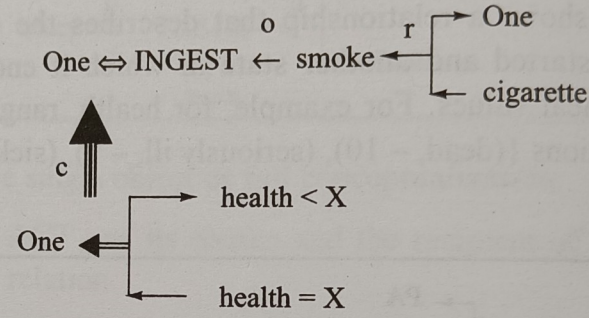
Rule 9: It shows a relationship between an ACT and the instrument using which it is performed. In the simplest form, instrument can be represented by just a single physical object or must be expanded to a full conceptualization. Here DO represents some act done by actor using instrument. The letter 'i' represents instrument.

Rule 9: ACT $\xleftarrow{i}$ detailed conceptualization for instrument	
Example: Mike ate custard with spoon	
CD Representations	
Simple Form	Detailed Form
$ \begin{array}{c} p \qquad i \\ \text{Mike} \leftrightarrow \text{INGEST} \leftarrow \text{spoon} \\ \uparrow \\ o \\ \text{custard} \end{array} $	$ \begin{array}{c} p \qquad i \qquad \text{Mary} \\ \text{Mike} \leftrightarrow \text{INGEST} \leftarrow \uparrow \\ \uparrow \qquad \text{DO} \leftarrow o \\ \text{custard} \qquad \text{spoon} \end{array} $

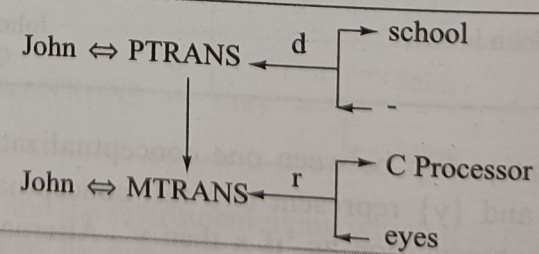
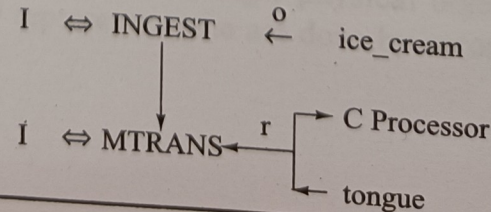
Rule 10: It shows a relationship that describes the change in state between PP and a state in which it started and another state in which it ended. Here states of object are described using numerical values. For example, for health, range could be - 10 to 10 describing various health conditions {(dead, - 10), (seriously ill, - 9), (sick, [- 1, - 8]), (alright, 0), (fine, 5), (perfect health, 10)}.

Rule 10: PP $\xleftrightarrow{\quad}$ PA	
Examples	CD Representation
The trees grow	$ \begin{array}{c} \text{Trees} \xleftrightarrow{\quad} \begin{cases} \text{size} > C \\ \text{size} = C \end{cases} \end{array} $
John is sick	$ \begin{array}{c} \text{John} \xleftrightarrow{\quad} \begin{cases} \text{health} (< \text{normal}) \\ \text{health} = \text{normal} \end{cases} \end{array} $

Rule 11: Relationship between one conceptualization and other conceptualization that causes it. Here {x} and {y} represent two full conceptualizations where {y} is caused by {x}. It is a conditional representation as 'if x then y'. Alternatively, we can say that 'y' is a consequence of 'x'.

Rule 11: $\Leftrightarrow \{x\}$  $\Leftrightarrow \{y\}$ Here $\Leftrightarrow$ represents conceptualization of x and y.	
Examples	CD Representation
Terrorist shot dead John	
Smoking is harmful for health	

Rule 12: It shows a relationship between one conceptualization with another that is happening at the time of the first. Here event 'y' is happening while event 'x' simultaneously.

Rule 12: $\Leftrightarrow \{x\}$ $\downarrow$ $\Leftrightarrow \{y\}$	
Examples	CD Representation
While going school John saw a snake	
I enjoyed while eating ice-cream	

Rule 13: It shows a relationship between conceptualization and the place at which it happened.

Rule 13: PP ⇓ ⇔	
Examples	CD Representation
John saw a snake in the house	<pre> graph TD House --> John John <--> MTRANS MTRANS -- o --> snake MTRANS -- r --> CProcessor[C Processor] MTRANS -- r --> eyes </pre>
I heard noise in the school	<pre> graph TD School --> I I <--> MTRANS MTRANS -- o --> noise MTRANS -- r --> CProcessor[C Processor] MTRANS -- r --> ears </pre>

These rules of CD representation help in drawing inferences and are independent of the language, as it is built using conceptual primitives which give the intended meanings of the words. There should be only unique representation of the meaning conveyed by two or more sentences. Sentences are represented as a series of diagrams depicting actions using both abstract and real physical situations. The actions are built up from a set of primitive acts which can be modified by tense. CD provides a structure that contains node representing information using a specific set of primitive acts. It is basically an Interlingua representation in which sentences in different languages having same meaning will have same CD representation.

15.2.4 Generation of CD representation

The following table shows CD representations of English sentences having simple but deep intended meanings.

CD Representations for Simple Sentences:

Table 15.2 CD Representations

Examples	CD Representation
Jenny cried	
Mike went to India	
Mary read a novel	
John purchased house from Mike	
Mary annoyed John	
Mike killed Mary	

- It is easier to describe inference rules that are represented only once for each primitive action.

For example, inference rule with ATRANS (transfer of ownership of an object) can be used to know who has an object after the action and who had it earlier, and also what is the object involved in this transfer. Similarly, inference rule with INGEST can be used to know what has been ingested by whom.

- Implicit information is deduced and stored as a part of CD representation for future use. It is achieved by applying set of inference rules associated with specific information. Once such knowledge is generated, it can be repeatedly used without applying rules again and again.
- CD representation is good a model for representing simple events. In CD representation all knowledge in the event is decomposed into very low level primitives.
- In CD representation, unspecified elements of the representation of one piece of information can be used as a focus for the understanding of later events as they are encountered.

Schank et al, using CD representation theory, developed a number of programs, such as MARGIE (Meaning Analysis, Response Generation and Inference on English) as a model for Natural Language Understanding, and SAM and PAM (Script/Plan Applier Mechanism) to understand stories.

But there are problems with CD representation as well.

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

15.3 Script Structure

Schank and Abelson introduced the concepts of scripts in 1977 that were build upon CD framework to handle story understanding by organizing episodes in a sequence or a chain of situations which could be anticipated using personal experiences rather than semantic categories. Scripts are structures representing procedural knowledge by describing a set of stereotyped events in a particular situation or context which could be expected to be followed from one to another event. It consists of set of slots which may contain default values along with some information about the type of values they might contain. It looks similar to frames, except that the values of the slots must be ordered and have more specialized roles. Script structure allows individual to make inferences

needed for understanding by filling in missing information. In real world situations, we see that event tends to occur in known patterns because of clausal relationship to the occurrence of events. So scripts are useful structure in such situations. A number of computer programs have been developed to demonstrate the theory. Schank applied his theoretical framework to story telling and the development of intelligent tutors. The classic example of Schank's theory is the restaurant script.*

Each script contains the following main components.

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

The scripts are useful in describing certain stereotyped situations such as going to theater or cinema hall, going to market, robbery in the bank, etc. To understand this concept better, let us consider another example of developing a script for going to theater to see a play. This might involve the following scenes:

- Going to theater.
- Buying ticket.
- Going inside hall and sitting on a seat
- Watching play
- Exiting from theater

Various components of script for going to theater are described as follows:

Entry Conditions:

- P wants to see a play
- P has money

Results:

- P saw play
- P has less money

* Source: <http://tip.psychology.org/schank.html>

Props:

- Ticket; Seat; Play

Roles:

- Person
- Ticket distributor
- Ticket checker

entire script is encoded in the Table 15.8 where we have considered five scenes.

Table 15.8 Complete Script for Event—Going to Theater

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
	Scene 5: Exiting • P PTRANS P out of Hall and theater

a particular script is to be applied, 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 should be held.

For example, given “John enjoyed the play in theater”, a script “Play in Theater” suggested above is invoked and 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.

If events follow a known track, we can use scripts to answer implicit or explicit questions. Different tracks may allow different outcomes of scripts (for example, the play was cancelled because of unseen situations, could not see play as it was full and no tickets were available, came out of the hall as bomb exploded, etc).

The main advantages of script structures are that they are capable of predicting implicit events and that a single coherent interpretation may be build up from a collection of observations. The disadvantage of script structures is that they are more specific and less general than frames. Furthermore, it is not suitable to represent all kinds of knowledge, because scripts are inflexible. To deal with inflexibility, smaller modules called memory organization packets (MOP) can be combined in a way that is appropriate for the situation.

15.4 CYC Theory

The CYC is a theory designed for describing the world knowledge (commonsense knowledge) to be useful in AI applications and more specifically in natural language understanding similar to CD theory. The CYC is more comprehensive, whereas CD is more specific theory for representing events. CYC was conceived by Lenat and Guha for capturing commonsense knowledge from the project to capture a large knowledge from 100 randomly selected articles in the EnCYClopedia Britannica (Lenat and Guha, 1990). CYC name came from this project. The CYC structure contains representations of events, objects, attitudes, space, time, motion, etc. and tends to be huge in structure. It is required to store special implicit knowledge or commonsense. So CYC is particularly concerned with the issues of scale. But the question is that why we build such large knowledge bases (KBs). Some of the reasons for large KBs are as follows:

Brittleness: Specialized knowledge bases are *brittle*. It is hard to encode new situations and there is degradation in the performance. Commonsense-based knowledge bases should have a firmer foundation.