SCRIPTS, PLANS, GOALS, AND UNDERSTANDING

R.C. Schank & R.P. Abelson

1977

(Chapters 1 - 3)

1 Introduction

1.1 What this book is about

This book reflects a convergence of interests at the intersection of psychology and artificial intelligence. What is the nature of knowledge and how is this knowledge used? These questions lie at the core of both psychology and artificial intelligence. The psychologist who studies 'knowledge systems' wants to know how concepts are structured in the human mind, how such concepts develop, and how they are used in understanding and behavior. The artificial intelligence researcher wants to know how to program a computer so that it can understand and interact with the outside world. The two orientations intersect when the psychologist and the computer scientist agree that the best way to approach the problem of building an intelligent machine is to emulate the human conceptual mechanisms that deal with language. There is no way to develop adequate computer 'understanding' without providing the computer with extensive knowledge of the particular world with which it must deal. Mechanistic approaches based on tight logical systems are inadequate when extended to real-world tasks. The real world is messy and often illogical. Therefore artificial intelligence (henceforth AI) has had to leave such approaches behind and become much more psychological (cf. Schank and Colby, 1973; Bobrow and Collins, 1975; Boden, 1976). At the same time, researchers in psychology have found it helpful to view people as 'information processors' actively trying to extract sense from the continual flow of information in the complicated world around them. Thus psychologists have become more interested in machine models of real-world knowledge systems. The name 'cognitive science' has been used to refer to this convergence of interests in psychology and artificial intelligence (Collins, 1976).

This working partnership in 'cognitive science' does not mean that psychologists and computer scientists are developing a single comprehensive theory in which people are no different from machines. Psychology and artificial intelligence have many points of difference in methods and goals. Intellectual history, like political history, is full of shifting alliances between different interest groups. We mention this because for many commentators, the blood quickens when computers and human beings are associated in any way. Strong claims for similarity (e.g., Newell and Simon, 1972) are countered by extravagant alarms (e.g., Weizenbaum, 1976). Enthusiasts and horrified skeptics rush to debate such questions as whether a computer could ever be in love. We are not interested in trying to get computers to have feelings (whatever that might turn out to mean philosophically), nor are we interested in pretending that feelings don't exist. We simply want to work on an important area of overlapping interest, namely a theory of knowledge systems. As it turns out, this overlap is substantial. For both people and machines, each in their own way, there is a serious problem in common of making sense out of what they hear, see, or are told about the world. The conceptual apparatus necessary to perform even a partial feat of understanding is formidable and fascinating. Our analysis of this apparatus is what this book is about.

1.2 Knowledge: Form and Content

A staggering amount of knowledge about the world is available to human beings individually and collectively. Before we set out on a theory of knowledge systems, we ought to ask ourselves: knowledge about what? We must be wary of the possibility that knowledge in one domain may be organized according to principles different from knowledge in another. Perhaps there is no single set of rules and relations for constructing all potential knowledge bases at will. A desire for generality and elegance might inspire a theorist to seek a 'universal' knowledge system. But if you try to imagine the simultaneous storage of knowledge about how to solve partial differential equations, how to smuggle marijuana from Mexico, how to outmaneuver your opponent in a squash game, how to prepare a legal brief, how to write song lyrics, and how to get fed when you are hungry, you will begin to glimpse the nature of the problems.

Procedures for intelligently applying past knowledge to new experience often seem to require common sense and practical rules of thumb in addition to, or instead of, formal analysis (Abelson, 1975). The prospects for the general theorist to cope with all the varied applications of common sense are especially dismal. Nevertheless, many artificial intelligence researchers take a generalist point of view. It is in the best tradition of mathematics (in which computer scientists are generally well trained) that great power is gained by separating form and content: the same system of equations may account for a great many apparently disparate phenomena. It is also a central tenet in computer science that generality is highly desirable. Turing's (1936) original principle of the general purpose machine has often been embraced as though the computer were (or soon would be) in practice a general purpose machine. The field of artificial intelligence is full of intellectual optimists who love powerful abstractions and who strive to develop all-embracing formalisms.

It is possible to be somewhat more pragmatic about knowledge, however. The five-year-old child learning to tie shoelaces need not in the process be learning anything whatsoever about mathematical topology. There is a range of psychological views on the nature of knowledge, and we shall say a little more about this in the next section. For now, we simply note that we will take a pragmatic view. We believe that the form of knowledge representation should not be separated too far from its content. When the content changes drastically, the form should change, too. The reader will encounter plenty of abstractions in this book, but each set of them will be pegged specifically to a particular type of real-world content. Where generalizing is possible, we will attempt to take advantage of it, but we will not try to force generality where it seems unnatural.

In order to adopt this attitude, we have set some boundaries on the type of knowledge we will to consider. Our focus will be upon the world of psychological and physical events occupying the mental life of ordinary individuals, which can be understood and expressed in ordinary language. Our knowledge systems will embody what has been called 'naive psychology' (Heider, 1958) – the common sense (though perhaps wrong) assumptions which people make about the motives and behavior of themselves and others – and also a kind of 'naive physics', or primitive intuition about physical reality, as is captured in Conceptual Dependency (CD) theory (Schank, 1972, 1975). This book goes well beyond CD theory, however. That theory provides a meaning representation for events. Here we are concerned with the intentional and contextual connections between events, especially as they occur in human purposive action sequences. This new stratum of conceptual entities we call the Knowledge Structure (KS) level. It deals with human intentions, dispositions, and relationships. While it is possible computers cannot actually experience such intentions and relationships, they can perfectly well be programmed to have some understanding of their occurrence and significance, thus functioning as smart observers. If our theory is apt, it will provide a model of the human observer of the human scene; it will also explain how to construct a computer observer of the human scene, and lead to the eventual building of a computer participant in the human world.

Often our emphasis will be on the nature of potential understanding of two or three sentences, story fragments, or longer stories. These provide a straightforward and helpful way to pose the major issues. Lurking beneath the surface, however, is an interest in the ingredients of personal belief systems about the world, which dispose people toward alternative social, religious, or political actions. One of us has a major interest in belief systems and ideologies (Abelson, 1973). This book is not directly addressed to that interest, but the concepts developed are a major part of that total effort.

What we will not present in this book is a general apparatus for attempting to represent any and all knowledge. We give no information retrieval methods of interest to library scientists. The reader with a passion for mathematics and/or logic will be disappointed. Likewise, anyone wondering, for example, whether we could get a computer to play squash or roll pasta dough should not wait with bated breath. The geometry of bouncing balls, the 'feel' of dough texture, and many other aspects of human activities involve knowledge falling outside of our present boundaries. This is because (among other reasons) visual and kinesthetic processes cannot readily be represented in verbal form. However, a great deal of the human scene can be represented verbally, and we have no lack of things to work on.

1.3 Traditional Points of View

We have mentioned that our task lies at the intersection of psychology (more specifically, cognitive psychology and cognitive social psychology) and artificial intelligence. Since we are concerned with verbally expressible knowledge, there is also an overlap with linguistics. When one tries to work in a disciplinary intersection, one inevitably comes into conflict with the traditional standards, habits, and orientations of the parent disciplines. This is especially true when the disciplines correspond to university departments, breeding suspicion of out-groups (cf. Campbell, 1969). Here we briefly sketch some of these conflicts, which we have resolved somewhat differently from others working at the same intersection.

Psychology is a heterogeneous discipline. The major subdivisions are developmental, clinical, cognitive and social psychology, and psychobiology. It is surprising to the non-psychologist but familiar to all but the youngest generation of psychologists that cognitive psychology is a relatively new branch of study. American experimental psychology was dominated for so long by behaviorism – roughly, from 1935 to 1960 – that the study of mental processes lay almost entirely dormant while other branches of psychology were developing rapidly. Since mental events could not be observed directly, there was scientific resistance toward relying on them to explain anything, whatever the scientist's common sense might tell him. Introspective evidence was not regarded as objectively trustworthy.

Since 1960, there has been an enormous surge of careful experimental work on mental phenomena. Skinner notwithstanding, human psychology could not seem to do without cognitive processes. Nevertheless, the methodological caution of the behaviorists was carried over into this resurgence. Acceptable scientific procedure called for quantitative response measurements such as accuracy of recall or choice reaction time when subjects were confronted with well-controlled stimulus tasks. In the verbal domain, stimulus control usually entailed repetitive trials on isolated verbal materials, deliberately avoiding meaningful connotations in the experimental situation. While recent experimental materials have not been as trivial as the old-fashioned nonsense syllables, neither have they been genuinely meaningful or even necessarily plausible. Experimental tasks are often unusual and/or unnatural in relation to tasks encountered daily by people in using language. For example, in a well-known experiment by Foss and Jenkins (1973), subjects listened to 48 sentences such as 'The farmer placed the straw beside the wagon', with instructions to press a key the instant they first heard the phoneme 'b'. In another well-known series of experiments by Anderson and Bower (1973), subjects heard 32 unrelated sentences such as 'In the park, the hippie kissed the debutante', 'In the bank, the tailor tackled the lawyer', etc., and an hour later were asked to recall as many of them as they could. The artificiality of tasks such as the latter led Spiro (1975) to remark tartly,

Why should a research subject integrate the to-be-remembered information with his or her other knowledge? The role the information will play in his or her life can be summarized as follows: take in the information, hold it for some period of time, give it back to the experimenter in as close to the original form as possible, and then forget it forever. The information cannot be perceived as anything but useless to the subject in his or her life (given the common employment of esoteric or clearly fictional topics as stimulus materials). The information, even when not clearly fictional, is probably not true. In any case, the subject knows that the relative truth of the information has nothing to do with the purpose of the experiment. (p.11)

In complaining about the lack of meaningful context in experiments such as these, it is no doubt unfair to present them out of their context. The experimenters had serious purposes, and the data were of some interest. But since our needs are for a set of interrelated constructs to explain the process of natural understanding of connected discourse, this style of experimentation is both too unnatural and too slow. There has been a gradual increase in research with connected discourse as stimulus material (e.g., Bransford and Johnson, 1972; Kintsch, 1974; Frederiksen, 1975; Thorndyke, 1977) but the field is still marked with a very cautious theoretical attitude. We are willing to theorize far in advance of the usual kind of experimental validation because we need a large theory whereas experimental validation comes by tiny bits and pieces. Our approach, in the artificial intelligence tradition, is discussed in Section 1.6.

If the research properties of experimental cognitive psychology are often unduly restrictive, the traditions in the field of linguistics are even more restrictive. Linguistics has concerned itself with the problem of how to map deep representations into surface representations (see Chomsky, 1965). After a long obsession with syntactically dominated deep representations, recent work in linguistics has oriented deep representations much more towards considerations of meaning (Lakoff, 1971; Clark, 1974). Despite this reorientation linguists have managed to miss the central problems.

Two fundamental problems stand out: How do people map natural language strings into a representation of their meaning? How do people encode thoughts into natural language strings? Because of a purported interest in the purely formal properties of language, linguists have consciously avoided both of these naturalistic problems. The second question seems, on the surface, to be closer to a linguist's heart. But linguists treat generation as a problem of determining whether a string is grammatical, i.e., whether it can be generated by the grammar they have set up. A grammar that generates natural language strings would be interesting and useful of course, if, and this is a big 'if', it started at the right place. Linguists tend to start their grammars at the node S (for sentence). People, on the other hand, start with an already well-formed idea (or the beginnings of an idea) that they want to express. Linguists thus wind up concerning themselves with considerations of semantics at the level of 'Can I say this string? Will it mean something'? People already know what they want to say and that it is meaningful.

Two remedies for this linguistic notion of semantics come to mind. For the generation problem the obvious solution is to start the process earlier. How do people get thoughts to express? Linguists explicitly consign this question to other disciplines; yet it is an important part of the generation process, and one which when treated as a linguistic question completely changes the process under investigation. The other remedy is to apply such semantic considerations as 'Does this string mean something?' to the problem of understanding what someone else has said. Questions of how strings can be meaningfully interpreted belong to the domain of understanding, not generation, where Chomsky (1965, 1971) has repeatedly put them. (Actually Chomsky would deny that he works on generation. Transformationalists prefer to think of themselves as working on an abstract formalism with no process notions present at all.)

Linguists have almost totally ignored the question of how human understanding works. Since human understanding is dependent on the ability to decode language this seems odd at best. Some 'computational linguists', (e.g., Friedman, 1969 and Kay, 1973) have attacked the problem. However, they have followed linguistic tradition and consequently have maintained one of the fundamental flaws of linguistics in their work. They have divided the problem into linguistic and non-linguistic parts, a division that holds up no better for understanding than it does for generation.

Artificial intelligence has a somewhat more congenial recent history. The field is relatively new, and its early efforts were predominantly oriented toward getting computers to solve logical and mathematical problems (e.g., Newell, Shaw and Simon, 1957; Minsky, 1961; Feigenbaum and Feldman, 1963; Nilsson, 1971), and to play games such as checkers (Samuel, 1963) and chess (Bernstein et al, 1958; Newell, Shaw and Simon, 1958) intelligently. Early efforts to have computers deal with natural language were marked either by drastic failure (as in the case of mechanical translation from one language to another) or drastic oversimplification in the admissible vocabulary (Green et al, 1961) and grammar (Abelson, 1963; Colby and Gilbert, 1964), or by programming tricks producing smooth locutions which made the computer seem smarter than it actually was (Weizenbaum, 1966).

It has nevertheless been consistently regarded as important that computers deal well with natural language. In practical terms, such a development would mean that anyone could interact with a computer without learning a programming language or some special code to communicate about a special problem, whether it be library or consumer information, travel and ticket reservations, suggestions about home repairs, crop protection, first aid, etc. Computerized teaching programs would not have to be restricted to giving multiple-choice tests of the student's knowledge, but could interpret and respond intelligently to free-form answers and questions from the student. None of these high-sounding things are possible, of course, unless the computer really 'understands' the input. And that is the theoretical significance of these practical questions – to solve them requires no less than articulating the detailed nature of 'understanding'. If we understood how a human understands, then we might know how to make a computer understand, and vice versa.

In the last several years there have been two clusters of developments in artificial intelligence which are miles ahead of previous efforts. First, there is a new generation of programs for 'parsing' sentences (in English and other languages) – that is, for deciding the proper features (such as what part of speech) to assign to each word in a sentence. The approaches of Woods (1970), Winograd (1972), Riesbeck (1975) and Marcus (1975) differ in the relative priority they give to syntactic or semantic features in parsing, but all agree that semantic features are considerably more important than linguists had generally been willing to acknowledge. Second, there has been increasing recognition that context is of overwhelming importance in the interpretation of text. Implicit real-world knowledge is very often applied by the understander, and this knowledge can be very highly structured. The appropriate ingredients for extracting the meaning of a sentence, therefore, are often nowhere to be found within the sentence.

There are several famous illustrations of this latter point. Collins and Quillian's (1972) is:

1 The policeman held up his hand and stopped the car.

Somehow in understanding this sentence we effortlessly create a driver who steps on a brake in response to seeing the policeman's hand. None of these intermediate links are mentioned in sentence (1). Another example, (from Abelson, 1969) is:

2 I went to three drugstores this morning.

Very innocently, the concept that the person must not have found what he wanted in the first two drugstores is implied, otherwise why would he have gone to three? This kind of implicit inference is very common – and of course can be wrong, but it is intrinsic to natural understanding that useful, fallible presumptions creep in.

Perhaps the simplest example of implicit inferences can be seen in a simple sentence such as (from Schank, 1972):

3 I like apples.

The speaker is talking about 'eating' but this is not explicitly mentioned. And why should it be? The speaker, unless he is deliberately trying to fool his listener, knows that the listener knows what action is being implicitly referenced. These examples were constructed with a point in mind, but are not really unusual. In all of them, and in many, many other examples to be found in this book, more is at issue than 'semantics'. It is 'pragmatics', the way things usually work – not how they might conceivably work – which most often impels the reader toward an interpretation. The reader brings a large repertoire of knowledge structures to the understanding task. Elsewhere these structures have been called 'frames' (Minsky, 1975) and 'schemata' (Rumelhart, 1976). Rumelhart puts the matter very well when he says, 'The process of understanding a passage consists in finding a schema which will account for it.'

Interestingly, the idea of the schema in the interpretation of human events has a long tradition in social psychology. American social psychology had its roots in Gestalt psychology, and therefore did not succumb to the excesses of behaviorism the way human experimental psychology did. The phenomenology of mental life maintained a central role, largely through the towering influence of Kurt Lewin in the 1940's. Lewin (1936) wrote about human goal strivings in terms of internal images people had of their 'life spaces'. Since then a long succession of social psychologists have appealed to structured ideational kernels of the way people supposed the world to be organized: Heider's (1946, 1958) 'balance principle' and 'naive psychology'; Festinger's (1957) 'cognitive dissonance theory'; Abelson and Rosenberg's (1958) 'psycho-logic'; Kelley's (1967) and Jones and Davis' (1966) 'attribution theory', and many more. The terminology of the 'schema' is very much active in the 1970's (cf. Kelley, 1971; Tesser, 1977), even in areas well beyond social psychology (Rumelhart, 1975; Bobrow and Norman, 1975; Rumelhart and Ortony, 1976). The second author's orientations in the present book can be traced back to earlier excursions into 'hot cognition' (Abelson, 1963), 'individual belief systems' (Abelson and Carroll, 1965), and 'implicational molecules' (Abelson and Reich, 1969).

There is a very long theoretical stride, however, from the idea that highly structured knowledge dominates the understanding process, to the specification of the details of the most appropriate structures. It does not take one very far to say that schemas are important: one must know the content of the schemas. To be eclectic here is to say nothing. If one falls back on the abstract position that only form is important, that the human mind is capable of developing knowledge structures of infinitely varied content, then one sacrifices the essence of the structure concept, namely the strong expectations which make reality understandable. In other words, a knowledge structure theory must make a commitment to particular content schemas.

The commitment to particular content is a policy we follow consistently throughout the book. Whether we are talking of scripts, plans, goals, themes, etc., we try whenever feasible to lay out the particulars of members of these conceptual categories. This is the same policy as was followed by the first author in developing Conceptual Dependency theory (Schank, 1972) to describe individual actions.

There has been much debate over whether the conceptual primitives of CD theory are the 'right' primitives, and some criticism that the theory is ad hoc. For many purposes, however, the important criterion is whether the theory is useful. Further, we would argue that any theory proposed as a replacement will have to come to grips with the same content issues as CD theory, and will more than likely end up with much the same primitives (as did Norman and Rumelhart (1975) for example). Indeed, the systematic linguistic exploration by Jackendoff (1976) of candidates for primitives seems to point in this direction.

We anticipate that there will be similar debate about the primitives we will propose in this book for higher-level knowledge structures. We will not be dogmatic about particular primitives, however, knowing that revisions in our scheme will no doubt be necessary as psychological validations and unanticipated theoretical considerations come along.

1.4 Conceptual Dependency Theory

In order to understand what follows in this book it is helpful to have a rudimentary exposure to Conceptual Dependency Theory. The theory has been described at length elsewhere (see especially Schank, 1975); we need not go into it in much detail here.

Conceptual Dependency (henceforth CD) is a theory of the representation of the meaning of sentences. The basic axiom of the theory is:

A For any two sentences that are identical in meaning, regardless of language, there should be only one representation.

The above axiom has an important corollary that derives from it.

B Any information in a sentence that is implicit must be made explicit in the representation of the meaning of that sentence.

These two rules have forced us to look for one economical form for representing meaning. In doing so, we have invented the initial framework:

C The meaning propositions underlying language are called conceptualizations. A conceptualization can be active or stative.

D An active conceptualization has the form:
Actor Action Object Direction (Instrument)

E A stative conceptualization has the form:
Object (is in) State (with Value)

The form that we postulate for conceptualizations has led us to the principle of primitive actions. That is, because a conceptualization is defined as an actor doing something to an object in a direction, we have had to determine just what an actor can do. Clearly, Principle A forces us to look closely at actions that seem similar to see if we can extract the essence of their similarity. Principle B forces us to make explicit whatever differences there might be between two actions and to express them accordingly. For example, two verbs in a language may share a similar primitive element (as 'give' and 'take' share the primitive element **TRANSFER of POSSESSION**) but also have differences. The best representation for our purposes for a given verb then, will be the primitive element it shares with other verbs, plus the explicitly stated concepts that make it unique. As it happens, these explicitly stated concepts also turn out to share similar elements with other verbs, so that often a verb is represented as a particular combination of primitive actions and states none of which are unique to that verb, but whose combination is entirely unique. (Many verbs are represented entirely by states with no primitive act used at all.)

The primitive acts of Conceptual Dependency are:

ATRANS The transfer of an abstract relationship such as possession, ownership or control. Thus, one sense of 'give' is: **ATRANS** something to someone else; a sense of 'take' is: **ATRANS** something to oneself. 'Buy' is made up of two conceptualizations that cause each other, one an **ATRANS** of money, the other an **ATRANS** of the object being bought.

PTRANS The transfer of the physical location of an object. Thus, 'go' is **PTRANS** oneself to a place; 'put' is **PTRANS** of an object to a place.

PROPEL The application of a physical force to an object. **PROPEL** is used whenever any force is applied regardless of whether a movement (**PTRANS**) took place. In English, 'push', 'pull', 'throw', 'kick', have **PROPEL** as part of them. 'John pushed the table to the wall' is a **PROPEL** that causes a **PTRANS**. 'John threw the ball' is **PROPEL** that involves an ending of a **GRASP** ACT at the same time. Often words that do not necessarily mean **PROPEL** can probably infer **PROPEL**. Thus, 'break' means to **DO** something that causes a change in physical state of a specific sort (where **DO** indicates an unknown ACT). Most of the time the ACT that fills in the **DO** is **PROPEL** although this is certainly not necessarily the case.

MOVE The movement of a body part of an animal by that animal. **MOVE** is nearly always the ACT in an instrumental conceptualization for other ACTs. That is, in order to throw, it is necessary to **MOVE** one's arm. Likewise **MOVE** foot is the instrument of 'kick' and **MOVE** hand is often the instrument of the verb 'hand'. **MOVE** is less frequently used noninstrumentally, but 'kiss', 'raise your hand', 'scratch' are examples.

GRASP The grasping of an object by an actor. The verbs 'hold', 'grab', 'let go', and 'throw' involve **GRASP** or the ending of a **GRASP**.

INGEST The taking in of an object by an animal to the inside of that animal. Most commonly the semantics for the objects of **INGEST** (that is, what is usually **INGEST**ed) are food, liquid, and gas. Thus, 'eat', 'drink', 'smoke', 'breathe', are common examples in **INGEST**.

EXPEL The expulsion of an object from the body of an animal into the physical world. Whatever is **EXPEL**ed is very likely to have been previously **INGEST**ed. Words for excretion and secretion are described by **EXPEL**, among them, 'sweat', 'spit', and 'cry'.

MTRANS The transfer of mental information between animals or within an animal. We partition memory into two pieces: The **CP** (conscious processor where something is thought of), and the **LTM** (long term memory where things are stored). The various sense organs can also serve as the originators of an **MTRANS**. Thus, 'tell' is **MTRANS** between people, 'see' is **MTRANS** from eyes to **CP**, 'remember' is **MTRANS** from **LTM** to **CP**, 'forget' is the inability to do that, 'learn' is the **MTRANS**ing of new information to **LTM**.

MBUILD The construction by an animal of new information from old information. Thus, 'decide', 'conclude', 'imagine', 'consider', are common examples of **MBUILD**.

SPEAK The actions of producing sounds. Many objects can **SPEAK**, but human ones usually are **SPEAK**ing as an instrument of **MTRANS**ing. The words 'say', 'play music', 'purr', 'scream' involve **SPEAK**.

ATTEND The action of attending or focusing a sense organ towards a stimulus. **ATTEND** ear is 'listen', **ATTEND** eye is 'see' and so on. **ATTEND** is nearly always referred to in English as the instrument of **MTRANS**. Thus, in Conceptual Dependency, 'see' is treated as **MTRANS** to **CP** from eye by instrument of **ATTEND** eye to object.

Some set of primitive ACTs is essential for representing meanings, especially if sentences that have the same meaning are going to be be represented in only one way. The ACTs presented here are not category names for verbs. They are the elements of action. An analogous situation is the formation of compounds from the elements in chemistry.

The use of such primitives severely reduces the inference problem (see Schank, 1975), since inference rules need only be written once for any ACT rather than many times for each verb that references that ACT. For example, one rule is that if you **MTRANS** something to your LTM, then it is present there (i.e., you know it). This is true whether the verb of **MTRANS**ing was 'see', 'hear', 'inform', 'memorize', or whatever. The inference comes from the ACT rather than the verb.

Conceptualizations that are attribute-value statements make use of a large number of SCALES. These scales run between boundaries which by convention are labeled -10 to 10. Scales are useful for indicating changes in state. Some of the scales we use, with their boundaries and some steps in between, are indicated below. In current applications of Conceptual Dependency Theory, it has not been necessary to undertake a serious quantitative scaling of relative points along the -10 to 10 continuum. At present, the occasional numerical references are only used suggestively.

HEALTH (dead, diseased, under the weather, tolerable, in the pink)

ANTICIPATION (terrified, nervous, hoping, confident)

MENTAL STATE (broken, depressed, all right, happy, ecstatic)

PHYSICAL STATE (end of existence, damaged, OK, perfect)

AWARENESS (dead, unconscious, asleep, awake, keen)

The symbol ⇑ denotes causality. Some example sentences and their representations are:

John killed Mary.

John **DO**

⇑

Mary **HEALTH**(-10)

John kicked Mary.

John **PROPEL** foot **to** Mary

⇑

foot(John) **BE PHYSICAL CONTACT**(Mary)

John told Mary that Bill was happy.

John **MTRANS**(Bill **BE MENT.ST**(5)) **to** Mary

John read a book.

John **MTRANS**(Information) **to LTM**(John) **from** book
inst(John **ATTEND** eyes **to** book)

In the original development of Conceptual Dependency theory, we spent most of our effort on representation of verbs and states. The bulk of Chapter 3 is one answer to the question of how to represent nouns. How does one represent a restaurant? Is it 'a place where people eat'? Or 'a place you go to eat where someone serves you and you pay'? How far do you go in such a representation? Scripts, although invented to handle a different but related problem, form the basis of the answer to the representation of certain complex nouns as well. How to represent concrete nouns is discussed briefly when we deal with memory in the next section.

Other researchers in artificial intelligence have much discussed the primitive actions that we have developed. Many of them seem to adopt one or more of them for their purposes, while usually rejecting either the rest of the set or the principle that it is necessary to represent sentences at the level of primitive actions each and every time. The most often heard suggestion is that one should only 'break down words into primitives when necessary'.

When is it necessary to break down a sentence into its minimal meaning units? The answer is simple enough: only when you need to exploit the 'meaning' itself. It is not necessary for word association tasks, for microworlds where there is little or no ambiguity or overlap in meaning, or for simple retrieval tasks where the meaning of the elements dealt with is not needed.

If you need to know the meaning of what you are dealing with, then it is necessary to look at the elements that make up that meaning. The only argument to this can be an argument based upon when you break down a sentence, not if you break down a sentence.

The 'when' question seems clear enough to us, although others differ with our position. Since memory ideally stores information in only one way, any pattern matching that needs to be done against information stored in memory requires a canonical form for the information. That is, information in memory must be stored in something like the primitive terms of Conceptual Dependency, and likewise the inference processes that are part of memory must be in those terms.

Should the breakdown into primitives occur after parsing ('when necessary'), or during parsing (assuming it is always necessary)? A good parser should exploit the meaning of a sentence. In understanding it seems doubtful that people first do a syntactic analysis without recourse to meaning and than look at the meaning. People understand as they go. Our parser (Riesbeck, 1975) has been quite successful using predictions that it generates based upon the kinds of meanings that it expects. Since it is hard to find a case when such breakdown is not necessary (in a real and complex system), we see little choice but to 'break down the words' every time.

One exception to this has occurred as a result of this book. In Chapters 4-7, various Knowledge Structures are introduced as an adjunct to Conceptual Dependency. We are beginning to find that it is sometimes better to parse directly into our Knowledge Structure representation rather than going by way of Conceptual Dependency. Thus, for example, the word 'want', which seemed primitive enough but was not so treated in Conceptual Dependency, is primitive in Knowledge Structures. It is reasonable with such words to go directly to where we want to be, thus bypassing Conceptual Dependency. This is, in fact, a complaint sometimes made about our work, namely that at the highest memory levels it will be necessary to reorganize information at places other than the primitive actions and thus we will have to 'unbreak down' again. The Knowledge Structure representation that we develop should answer this complaint.

1.5 Memory

Before we get into the substance of this book, it is worthwhile to introduce one more issue, namely memory. For a long time, the problems of natural language processing seemed to be separate from the problems of memory. Recently, Quillian (1968), Anderson and Bower (1973), Rieger (1975), Norman and Rumelhart (1975), and others have made it quite clear that memory and language are inextricably bound together. However, while the importance of dealing with memory has been generally agreed upon, the form that memory takes is still at issue. This book is, in a sense, entirely about memory. We are arguing here for certain theoretical entities that must form the basis of human memory organization.

The form of memory organization upon which our arguments are based is the notion of episodic memory. An episodic view of memory claims that memory is organized around personal experiences or episodes rather than around abstract semantic categories. If memory is organized around personal experiences then one of the principal components of memory must be a procedure for recognizing repeated or similar sequences. When a standard repeated sequence is recognized, it is helpful in 'filling in the blanks' in understanding. Furthermore much of the language generation behavior of people can be explained in this stereotyped way.

Other proposals for memory organization have stressed the more scholastic notion of semantic memory. Briefly, semantic memory is a memory for words that is organized in a hierarchical fashion using class membership as the basic link. For example, 'canary' is linked to 'bird' and 'bird' to 'animal' in a hierarchical tree.

We can see at once that such an organization will not work for verbs, nor for nouns that are abstract nor for nouns that do not submit easily to standard categories (such as 'teletype'). Even if other semantic links besides class membership are used, such an organization implies that propositions are stored by linking them to the words with which they are expressed. This is not possible in the conceptual, non-word-oriented system that we have described. We could overcome this difficulty by organizing concepts in networks but the complexity of the possible combination of elemental concepts makes this extremely cumbersome. There are other difficulties as well.

An episodic memory, on the other hand, is organized around propositions linked together by their occurrence in the same event or time span. Objects are most commonly defined by their place in a sequence of propositions describing the events associated with an object for an individual. A trip is stored in memory as a sequence of the conceptualizations describing what happened on the trip. Some of the conceptualizations will be marked as salient and some will have been forgotten altogether.

Nominal concepts (concrete nouns) fit in this view with a two-part definition. The first and primary part is a functional definition that attempts to generalize the salient events over particular episodes in which the noun has occurred. The complete functional definition of a given noun lists all distinguishable occurrences of that noun present in memory. The second part is a physical description of one particular member of the class that is being defined.

For a 'spoon,' for example, the definition in memory lists the general usage for a spoon first (e.g., a thing that you **PTRANS** into mushy or liquid objects in order to **PTRANS** that object to your mouth so as to **INGEST** it). All interesting specific instances would also be stored (including, for example, 'The time I was camping and washed my spoon in the sand.'). Last, we would have a physical description of a particular spoon (most likely the kind that you have at home). The over-all organization of memory is a sequence of episodes organized roughly along the time line of one's life. If we ask a man, 'Who was your girlfriend in 1968?' and ask him to report his strategy for the answer, his reply is roughly: 'First I thought about where I was and what I was doing in 1968. Then I remembered who I used to go out with then.' In other words, it really isn't possible to answer such a question by a direct look-up. Lists of 'past girlfriends' do not exist in memory. Such a list must be constructed. The process by which that list is constructed is a search through episodes organized around times and locations in memory.

Of course if we ask someone not about past girlfriends but about past history learned from books, say, 'Who ruled England in 1668?', then the memory search mechanism might not turn out to be episodic. The respondent might conceivably remember when he learned this fact, but it is more likely that such scholastic memories could get divorced from their episodic origins and become organized more 'semantically', as in Collins' (1976) model of book knowledge. The whole question of episodic vs. semantic memory is controversial (cf. Loftus and Loftus, 1976), and our clear preference for the episodic mode is partly a function of the non-scholastic character of the knowledge we are interested in.

Some episodes are reminiscent of others. As an economy measure in the storage of episodes, when enough of them are alike they are remembered in terms of a standardized generalized episode which we will call a script. Thus, rather than list the details of what happened in a restaurant for each visit to a restaurant, memory simply lists a pointer (link) to what we call the restaurant script and stores the items in this particular episode that were significantly different from the standard script as the only items specifically in the description of that episode. This economy of storage has a side effect of poor memory for detail. But such a side effect, we shall argue, is the price of having people able to remember anything at all. Script-based memory is what will enable computers to understand without having their memories filled up so much that search time is horrendously long.

1.6 The Methodology of AI

Although the work we describe in this book is intended to lead towards the eventual computer understanding of natural language, it is not necessary to have much familiarity with computers in order to understand what we are saying. This fact is, or ought to be, true of nearly all clearly written work in artificial intelligence (AI). The computer is used in AI research as an omnipotent, but very dull and plodding, god. Under this view, it sometimes seems unnecessary to actually write the program that embodies the theory. One only need show the process in convincing enough detail. This is what one imagines. However, the reality is somewhat different – researchers actually do write programs.

Whenever an AI researcher feels he understands the process he is theorizing about in enough detail, he then begins to program it to find out where he was incomplete or wrong. It is the rare researcher who can detail a theory, program it, and have the program work right away. The time between the completion of the theory and the completion of the program that embodies the theory is usually extremely long. In modelling such complex processes as comprehension of language, there are more things to keep track of than a human trying to be conscious of each variable can manage. Understanding at such a level of complexity is a relatively subconscious process in everyday life.

What AI has to contribute to psychology is exactly this experience with modelling processes. An AI researcher asks what the input is and what the output is for every subprocess he deals with. In asking these questions he recognizes, at the very least, the nature and number of the subprocesses that must make up the entire process he wishes to model.

An analogy can be seen in asking directions from one place to another. If, while in New York, one asks how to get to Coney Island, and is told to take the 'N' train to the last stop, these directions will be adequate only if this improperly specified algorithm can be filled out with a great deal of knowledge about how to walk, pay for subways, get in the train and so on. We call this information a 'script' (detailed in Chapter 3). The point here is that a computer that does not have any prior information would make no sense of this response. It must be given information about: parsing sentences, finding meanings, filling in substeps in a plan, recognizing trains, paying for subways, walking, and so on. An AI researcher is forced to specify each and every detail in a theory that accounts for the ability to understand such a simple response to a question. We cannot be satisfied by an answer such as 'well, you use your knowledge about subways.' To put it on a computer we must know what form that knowledge takes. How are subways represented in memory? What pieces are where? How are they accessed? When and why are they accessed? What happens before they are accessed? What happens afterwards?

It is the asking and answering of such questions, and then the testing of those answers on a computer, that constitutes AI research. We use those methods in this book and thus our arguments lose some of the traditional psychological flavor of theories. That is, we are not oriented toward finding out which pieces of our theory are quantifiable and testable in isolation. We feel that such questions can wait. First we need to know if we have a viable theory. Viable here means: Will it work on a computer? Can we properly specify each subpart?

Our attitude may be somewhat unsettling to psychologists accustomed to piecemeal experimental testing of theoretical propositions. To them we urge the same patience in judging our work that they use in tolerating the very slow accumulation of knowledge in the experimental tradition. The AI style of research is much more disciplined than it may look. Although running computer programs provides the ultimate test, there are also strong criteria of intuitive necessity and internal consistency in theory formulation. Throughout this book we will often appeal to examples where the human understander readily makes inferences not explicitly available in the text of the example, or perceives something as odd or ridiculous. We persistently ask: what does an understander have to know in order to fill in missing inferences or perceive oddities? Addressing this question leads us to an inductive, rather than experimental, style of psychology, and we feel that this is the more appropriate style for this stage of development in our problem area.

2 Causal Chains

2.1 Understanding Text

Early work in natural language understanding (e.g., Woods, 1970, Winograd, 1972, Schank, 1972) was concerned almost exclusively with individual sentences. Recently there has been a shift of attention towards whole texts (e.g.,Wilks 1973, Rumelhart 1975, Schank 1975). On the surface this shift seems simple enough. If one can understand individual sentences, then to understand a text, all one need do is treat it as a set of individual sentences and apply the same methods. Our work would be much simplified if this were the case, but sadly it is not. The meaning of a text is more than the sum of the meanings of the individual sentences that comprise it.

People, in speaking and writing, consistently leave out information that they feel can easily be inferred by the listener or reader. They try to be concise and therein begins the root of the problem.

In wondering how we might go about representing the meaning of a text we came up against the problem of what connections existed implicitly in a text. One of the things that people seem to leave out when they talk are the connectors of the text. For example consider paragraphs (1) and (2):

1 John came over yesterday. Boy, was he mad.

2 John came over yesterday. When Mary saw John she almost died laughing. Boy, was he mad.

In (1) we are expected to figure out that the relationship between the first sentence and the second is that the second was in some way related to the reason for the first. In (2), the second sentence has two parts in it. The second part is implicitly stated to be the result of the first. Similarly, the third sentence is implicitly the result of the second part of the second sentence.

The connectivity here is provided by causality relationships of various types. If the nature of the connectivity of text is causal, then what we need are rules for determining causal relationships between events.

One of the first problems in this search is determining just when causality is present. Although the use of causal words tends to point out that some causality relationship is present, often the stated causality is misleading. For example, consider sentence (3):

3 John cried because Mary said she loved Bill.

Sentence (3) is a meaningful, well constructed English sentence. Yet, it is literally quite silly. Certainly John didn't cry because of the event of Mary speaking. What 'speaking' does cause is 'thinking', which can cause 'sadness' which can be a reason for 'crying'. Since people don't really misunderstand sentences such as (3), there is little reason for speakers to worry about their imprecision. However, in designing a theory of understanding, there is a great deal to worry about. Sometimes when people say 'event X caused event Y' they mean it and sometimes they do not. In order to understand we must be able to fill in the gaps left implicit by a speaker. When we hear someone say 'X caused Y', and we know that X could not have caused Y, we react accordingly. We check to make sure the causalities we hear are correct. To model understanding it is necessary to find out precisely what the rules are for this.

The same problem exists with respect to sentences that claim a given event caused a given state. For example, compare the following two sentences:

4 Joe burned his hand because he touched the stove.

5 Joe burned his hand because he forgot the stove was on.

People have no trouble recognizing that (4) means what it says causally and (5) does not. 'Forgetting' does not cause 'burning'. Readers of sentences such as (5) understand that they are called upon to supply the 'real' causality themselves by making an inference. We might expect that they would infer (4) upon reading (5).

Now, a simple theory would claim that this problem is simply resolved. Sentence (5) has a mental event causing a physical state and since mental events do not ordinarily cause physical states, we must infer an intermediate action that will fix it up. This is correct as far as it goes but there are other types of cases. Consider the following sentences:

6 John's leg was broken because Mary tripped him.

7 John leg was broken because Mary knocked over a pile of bricks.

In sentence (6) we are happy with the causality because we know that 'tripping someone' can cause a broken leg. However in (7) there is no such direct connection. Here again, we must infer the real causative event. The 'knocking over' must have propelled the bricks into contact with John's leg. That contact caused the broken leg. As with all inferences, this causality may be incorrect. What is important is that understanders can create such connections when the need arises. The rules that work for this example will be given in the next section.

Thus, statements of causality cannot be taken at face value. If we hear that X caused Y, we must ask if X could cause Y directly and if it cannot we must figure out the intermediate events. This is the principle of causal chaining.

2.2 Causal Types

A very simple causal syntax exists in natural thought, a syntax that can be violated in natural language expression. In the reconstruction of the thought that underlies the utterance, the causal syntax must be rigidly obeyed. To do this, a fairly complex causal semantics, or world knowledge store, must be exploited.

In the physical world, the causal syntax is as follows:

CS1 Actions can result in state changes.

CS2 States can enable actions.

In Conceptual Dependency, CS1 is denoted:

ACT
←r
STATE

---

and CS2 is represented:

STATE
←E
ACT

The meat of these rules is in their real world application. That is, not any action can result in any state, and not any state can enable any action. Thus, for every primitive action there is associated with it the set of states that it can affect as well as the set of states that are necessary in order to effect it. Since there are only eleven primitive actions this delimitation of world knowledge is easily accomplished.

To see how it all works we return again to sentence (7). In sentence (7) there is one action given – Mary **PROPEL** Mary; and one state change given – leg(John) **BE PHYS.ST**(-). For **PROPEL**, we have a list of the states that can result from **PROPEL** as well as the set of conditions under which those states can occur. The applicable rules here are:

A **PROPEL** can result in **PHYSICAL CONTACT** between the object of the **PROPEL** and any objects in the location specified in the Directive case.

B **PROPEL** results in **PHYS. ST** (-) if one of the objects in the **PROPEL** is human, if it results in **PHYSICAL CONTACT** (**PHYSCONT**) and if the force of the **PROPEL** is great.

In application, these rules when applied to Mary **PROPEL** Mary **to** bricks yield that Mary is in **PHYSCONT** with bricks. That is, according to the causal rules, Mary could be damaged but not John's leg. It is therefore necessary to hypothesize an action that would have resulted in a **PHYS. ST**.(-) for John's leg. Using the above rules backwards, we get the hypothetical event:

Something$_1$ **PROPEL** something$_2$ **to** leg(John)

This hypothetical event must be derived from some real event as a possible inference in order to make sense of the stated causation. This can be done if it is known that:

If a moveable object is put in **PHYS. CONT** with the object of a **PROPEL** then it can become the object of the new **PROPEL**, where the actor of the new **PROPEL** is the same as object of the original **PROPEL**.

We can then hypothesize as an inference:

Mary **PROPEL** bricks **to** leg(John)

Using the original rules for **PROPEL** given above, this hypothetical event is causally correct in the sense that the known event could cause it, and the known state could result from it.

The causal chaining inference rules given here are an extremely important part of the understanding process. People, and therefore computers, must have rules like those given above, or they would not be able to understand.

One problem in building causal chaining mechanisms is recognizing when causality considerations occur. Causality need not be expressed directly in English. There exists a class of verbs in English for example, that have implicit causal connections. Thus in the analysis of sentence (8):

8 John prevented Mary from leaving the room by hitting her.

there is an implicit causality. 'Prevent' means, 'to do something that causes something else not to happen'. According to our causal syntax, the action done resulted in a state that disabled an intended action.

So, one thing we have here is a modification of rule CS2 above which we call rule CS3:

CS3 States can disable actions.

CS3 is represented as:

STATE
↑dE
ACTION

The causally correct analysis here then is:

PROPEL(hitting) results in

STATE disable **PTRANS**(leaving the room)

This kind of analysis is important because it makes possible the inference of what is really going on. If we ask the question that derives from the above chain, we get: what **STATE** resulting from a **PROPEL** can disable a **PTRANS**? Many possibilities exist here, for example, unconsciousness.

Another possibility here is that the **STATE** resulted in a **MENTAL STATE** which caused the intended **PTRANS** to no longer be intended. For example, we can imagine Mary being frightened and deciding not to leave the room due to the possibility of being further harmed. We thus have a fourth rule:

CS4 States (or acts) can initiate mental states.

CS4 is represented as:

STATE (or **ACT**)
↑I
MENT.ST

A fifth rule occurs here too. In the example above, fear was a reason for a decision. The general rule is

CS5 Mental states can be reasons for actions.

CS5 is represented as:

MENT.ST
↑R
ACTION

Using these causal rules it is possible to do three things: First, we can decide what is and is not a causal chain. When we encounter a 'because' type word, we can try to connect together causally the two clauses. If we cannot make the connection because of a causal syntax violation, we postulate some set of unknown states and actions that would correctly complete the chain. These empty conceptualization holders become the primary candidates for inferences.

Second, we can now analyze correctly the conceptual representation of words such as 'prevent', 'help', 'allow', such that their dictionary definition contains implicit causal chains that demand inferences to be made.

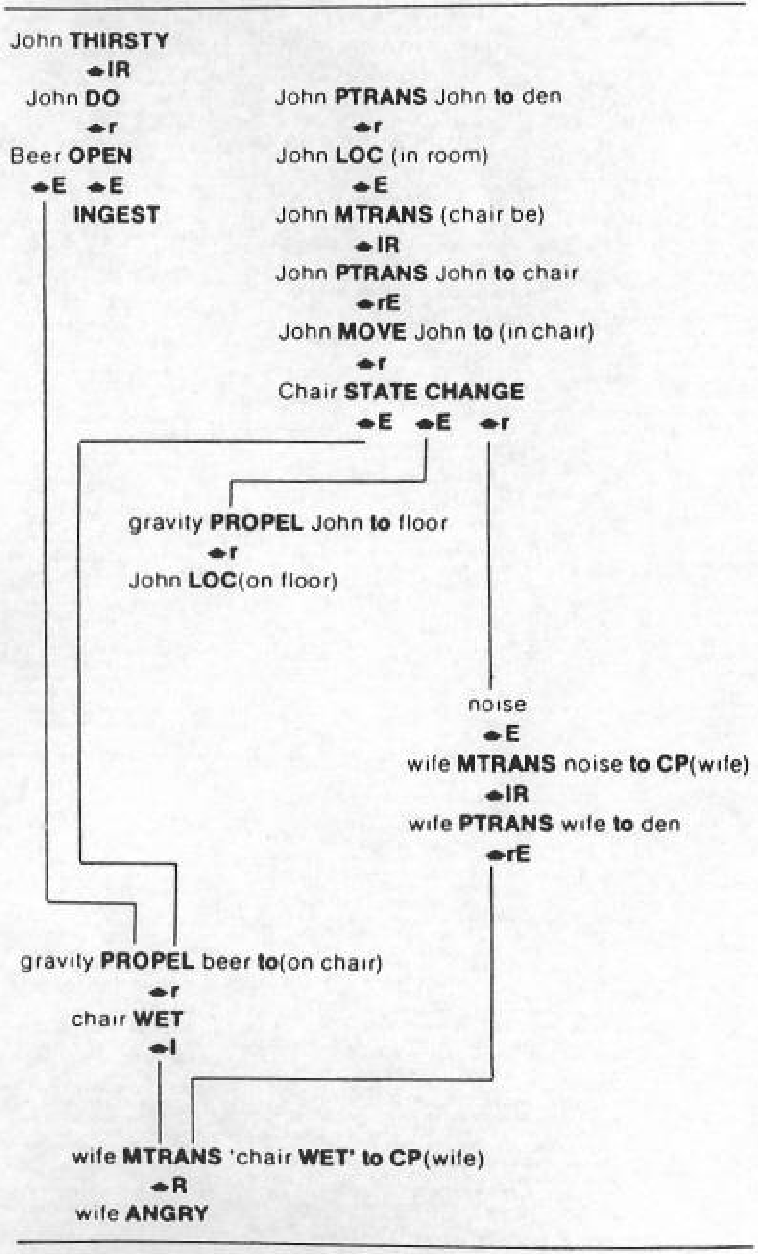
Third, and perhaps most important, we now have a way of representing connected text. We shall now explain.

When we think about what makes connected text connected, the answer does not seem obvious. However, if we reverse the question and ask if there is evidence of causal connections in text, the answer seems to be that such evidence is everywhere.

Consider the following story:

9 John was thirsty. He opened a can of beer and went into the den. There he saw a new chair. He sat down in it. Suddenly the chair tilted over and John fell on the floor. His beer spilled all over the chair. When his wife heard the noise she ran into the den. She was very angry that her new chair had been ruined.

Below is a representation of all the events in the above story connected together with the causalities that are implicit in the story. The result is a giant causal chain, that serves to relate together the events and state changes. It is this connectability that makes the story coherent. If we could not construct a causal chain we could probably not understand the story. (We use the following notational devices below: **IR** denotes an initiate followed by a reason where the intermediate mental action has been left out. Similarly **rE** denotes a result followed by an enable with the intermediate state left out. **DO** indicates an unknown action.)



The above representation says, essentially this: John's thirst caused him to decide to **DO** an action that would result in a beer being opened which would enable him to drink it. What we have is: the **THIRST** initiated an **MBUILD** (left out) to open the beer which was the **REASON** that he did the action (whatever it was) that resulted in the beer being open. The beer being open enabled another action (spilling) that happened much later in the sequence of events. (States enable events but the events that are enabled can occur any time after the state is initially present just as long as the state continues to be present.)

Thus, connecting up the actual 'state enables action' and 'action results in state' causalities is what makes sense of a text. A text is disconnected if a causal chain cannot be constructed to represent it.

Furthermore a theory of importance in text can be derived by causal chaining. Events or states that have multiple connections are likely to be highly significant with respect to the text. States or events that lead nowhere are likely to be forgotten.

The basic philosophy here then is this: Once the actual events that took place are determined, understanding is possible. The problem in understanding is how to make explicit that which has been left implicit. Conceptual Dependency was designed to handle that problem at the single thought (or sentence) level. Causal chains handle that problem at the level of interconnected thought (or texts).

2.3 Representation of Causation

We use the following causal links in our representations:

→r means an **ACT** results in a **STATE**.

→E means a **STATE** enables an **ACT**.

→I means a **STATE** or **ACT** initiates a mental **STATE**.

→R means a mental **ACT** is the reason for a physical **ACT**.

→dE means a **STATE** disables an **ACT**.

We use two standard abbreviations. These are:

⇐**rE** an **ACT** results in a **STATE** which enables an **ACT**.

⇐**IR** an **ACT** or **STATE** initiates a thought which is the reason for an **ACT**.

These abbreviations serve to leave items implicit when they are of no particular interest at a given time.

While these abbreviations help us focus on the items that interest us without dallying on irrelevant details they can also lead to problems. When we hear

3 John cried because Mary said she loved Bill.

we have an **IR**. It is important first to be able to recognize that John did an **MBUILD** of something sad and this is why he cried. However, if we make no attempt at all to figure out what John was sad about, i.e., what his actual train of thought was, we will miss the very important (though possibly wrong) inference that John loved Mary. This problem becomes more acute the less transparent the reasoning of an individual is.

There is another problem with causal chains in that people often speak of negative events causing things and in a strictly physical world this is rather odd.

That is, the causal chains that we have been presenting here are useful for figuring out chains of causality that are based on physical reality. Thus sentence (10) is easily handled with causal chains:

---

10 John gave Bill an orange for his cold.

John **ATRANS** orange **to** Bill

⇐**rE**

Bill **INGEST** orange **to** **INSIDE**(Bill)

⇐**r**

Bill **HEALTH**(POS change)

Here, as a result of the semantics of causation, we are forced to postulate an **INGEST** action that is rather important to understanding the sentence.

But, when the world is full of intentions that get only partially realized, or plans that go astray, causal chains do not work so simply. A higher level of representation is needed to account for sentences such as (11) and (12):

11 Mary liked John but not enough to agree to go to the motel.

12 John can't go because he hurt his arm.

Our simple causal world needs more information and apparatus to handle these. After Chapters 3, 4, and 5 present such apparatus, we will discuss some modifications to how we deal with causal chains.

What we have developed so far is useful nonetheless. It allows us to deal with physical events. Since we can now infer events that are missing explicitly from a causal chain, the next interesting question is when that is not possible to do. That is, even in the physical world of events and states, people still manage to skip enough intermediate steps in a chain so as to exasperate a naive chain builder. For example, there is nothing wrong with the sequence:

13 John ordered a new suit. He paid the bill with his credit card.

However, a causal chain builder would not be able to discern any obvious connection between the **MTRANS** of order and the **ATRANS** of the suit and the **ATRANS** of the money.

While there is, in fact, a causal chain there, it cannot be derived simply by using the principles of causal chain construction. To build the chain, what we need is knowledge of the social world rather than the physical world we have been discussing. This will be discussed in the next chapter.

2.4 Causal Propensity

We have concentrated thus far on situations with incompletely specified causality, that could be filled in more or less straightforwardly with a single most plausible sequence of conceptualizations. Psychologists have long been interested in the subjectively very compelling nature of certain causal sequences, for example, those arising from **PROPEL** actions (Heider, 1944; Michotte, 1963). A more complicated case of especial interest to psychologists, arises when two distinct causal chains compete as explanations for the outcome of an ambiguous situation.

One type of illustration of such causal ambiguity arises from controversial events in political conflict situations. What causes British troops to arrest Irish Republican Army leaders? The I.R.A. says it is a consequence of a British policy of oppression, and the British say it is a consequence of the I.R.A. practicing violence. What causes South African blacks to riot? The blacks say it is the humiliations of apartheid, and the South African government says it is Communist agitators. In politics, it is very frequent to see a hated group as the causal agent initiating actions which result in bad consequences. Causation is invested in actors perceived to have malevolent intentions. Explanations involving multiple causation are avoided.

Going beyond politics, actors are seen in general as having a certain degree of causal propensity in their appropriate sphere of action. In cases of causal ambiguity, actors who are credited with high causal propensity may tend to take the blame (as does the malevolent enemy), or get the credit (as, say, the professor does for a paper written jointly with a graduate student). Concepts of causal propensity come from whatever knowledge is available about the attitudes and capabilities of actors, and this knowledge can be manipulated by shadings in the verbal descriptions of actors – as every propagandist well knows, or in how closely available knowledge is scanned for clues as to what might have caused a later event. The first point is well illustrated in a subtle study by Bar-Hillel (1975), described in a paper by Tversky and Kahneman (1976) on causal thinking. In the study, the responsibility for a taxicab accident was seen differently depending on the wording of a statistical generalization about the cab company.

Tversky and Kahneman were interested in concepts of probability which do not concern us here, but we can paraphrase the Bar-Hillel study appropriately. Subjects were shown a paragraph describing an accident as below. (The key sentence is underlined here for emphasis).

14 Two cab companies, the Green and the Blue, operate in a city. The vast majority of cabs in the city are Green cabs. One night, a cab was involved in a hit-and-run accident. A witness identified the cab as a Blue cab. The court tested his ability to identify cabs under the appropriate visibility conditions. The witness was correct in most of the cases.

Subjects given this story were asked whether a Green cab or Blue cab was more likely to have caused the accident. Overwhelmingly, subjects choose the Blue cab as more likely. Now consider a variant of the story, which has exactly the same objective likelihood structure as (14), but a different wording of the second sentence:

15 Two cab companies, the Green and the Blue, operate in a city. The vast majority of cab accidents in the city involve Green cabs. One night, a cab was involved in a hit-and-run accident. A witness identified the cab as a Blue cab. The court tested his ability to identify cabs under the appropriate visibility conditions. The witness was correct in most of the cases.

Subjects shown story (15) and asked whether a Blue cab or a Green cab was more likely to have caused the accident, in this case were not so sure, and many chose Green as more likely.

Tversky and Kahneman (1976) explain the objectively unjustified difference in judgements in the two cases as follows: In (14), 'the difference in mere frequencies of Green and Blue cabs cannot be psychologically related to the propensity of any particular cab to be involved in an accident, and it is therefore ignored'. In (15), 'the difference in frequency of accident is interpreted as a difference in accident-proneness and it is therefore utilized'. In other words, the greater frequency of Green cab accidents suggests that Green cabs are dangerous and/or their drivers reckless, and these causal propensities are available to attach themselves to any individual ambiguous case.

Illustrations of a related point about causal propensity is a study by Ross et al. (1976). Subjects were induced to first 'explain' particular events (such as suicide) in the later lives of clinical patients whose case histories they had read; then they were told that the later events were not really known to have happened, but were arbitrarily attached to the cases to create the experimental task. Subjects were then asked, nevertheless, to judge the likelihood that the particular events might really happen (e.g., that the person would really commit suicide). The task of identifying known antecedents to explain a particular event considerably increased average estimates of the event's likelihood, compared to estimates made by people who read the same case history but did not engage in any 'explaining' task. Presumably the explaining task leads subjects to construct reason causations (Section 2.2, rule CS5) flowing from very powerful early mental states, i.e., strong causal propensities.

These examples are rather special, but the concept of causal propensity is important, and we shall have more to say about it in Chapter 6. The psychological misuse of causal thinking raises an important general issue for artificial intelligence. In the first study above (like others of similar character) people were misled into treating two statistically equivalent situations differently because of the active wording of (15) which suggests that drivers of Green cabs are generally more reckless or less competent, versus the passive wording of (14) which simply suggests that Green cabs are generally more numerous. In the second study, clinical material scanned under a false assumption about a future event leads to unwarranted exaggeration of particular causal linkages. A computer understanding system might conceivably be designed not to be misled by such nuances, but this would be at the probable expense of considerable computing time and effort.

However, prior experience with similar problems and instructions to think very carefully could also steer people away from errors. In other words, the tendency to make inferential errors is not something true of people and false of machines. Rather, errors occur when systems process rapidly and without much depth, and they can be minimized in systems operating in 'careful mode' (cf. Sussman, 1974). But, carefulness is a luxury which understanding systems may often not be able to afford.

Inferential concepts like causal propensity are 'quick-and-dirty' and thus useful heuristics. They help the understander make rapid sense of causal ambiguities. It is worthwhile, perhaps even mandatory, to program such short-cuts in any real-time computer understanding system, even at the cost of occasional errors. The short-cut principle of causal propensity says that in cases of causal ambiguity, pick the causal chain originating from the generally most causally active actor. Principles such as this enable people to read and understand quickly.

3 Scripts

3.1 Introduction

How do people organize all the knowledge they must have in order to understand? How do people know what behavior is appropriate for a particular situation? To put it more concretely, how do you know that, in a restaurant, the waitress will get you the food you ask for whereas if you ask her for a pair of shoes, or you ask her for food on a bus she will react as if you had done something odd?

People know how to act appropriately because they have knowledge about the world they live in. What is the nature and form of that knowledge? How is it organized? When is it brought to bear? How is it accessed? What portions of that knowledge are thought about and used, and under what circumstances?

We recognize two classes of knowledge that people bring to bear during the understanding process: general knowledge and specific knowledge. General knowledge enables a person to understand and interpret another person's actions simply because the other person is a human being with certain standard needs who lives in a world which has certain standard methods of getting those needs fulfilled. Thus, if someone asks you for a glass of water, you need not ask why he wants it. Even if he later uses it for a nonstandard but clear purpose - say he throws it in somebody's face and steals that person's watch - you have no trouble interpreting his actions. It is easy to understand what his plan was, and why he needed the water. We may never have observed such a sequence before, but our general knowledge about people and the world they live in allows us to interpret the events we see.

We use specific knowledge to interpret and participate in events we have been through many times. Specific detailed knowledge about a situation allows us to do less processing and wondering about frequently experienced events. We need not ask why somebody wants to see our ticket when we enter a theater, or why one should be quiet, or how long it is appropriate to sit in one's seat. Knowledge of specific situations such as theaters allows us to interpret the remarks that people make about theaters. Consider how difficult it would be to interpret 'Second aisle on your right' without the detailed knowledge about theaters that the patron and the usher both have. It would be rather odd to respond 'What about the second aisle on my right?' or 'Where is my seat?' or 'Is this how I get into the theater?' The usher simply takes the ticket and, assuming you understand and have specific knowledge about theatres, utters his otherwise cryptic remark without any verbal input from you.

The remainder of this chapter deals with the nature and form of such specific knowledge. We shall discuss issues related to general knowledge in Chapter 4.

With stories as well as with isolated utterances, human readers seem to have no trouble in rapidly extracting the features of the situation intended for emphasis by the writer. Consider, for example, the sentence:

1 While giving his order to the waiter at Mamma Leone's one evening, Spillane was approached by the owner, a notorious Mafia figure.

The 'while'-clause functions to set the Leone's restaurant context with its huge store of generalized and specific world knowledge. Nevertheless, the reader does not slow down to work out who 'the waiter' is or how Spillane came to be talking to him, but quickly notes that Spillane is probably at a table, in the act of ordering, in a well-known Italian restaurant.

Such specific knowledge exists in detail for every mentally competent person in the world with respect to every standard situation that he has been in many times. What form does such knowledge take?

We established in Chapter 2 that the directed inference process results in a connected causal chain of events. This causal chain is useful for representing any sequential flow of events. Since certain sequences of events frequently occur in a specific order we must postulate that people have developed special mechanisms to deal with them. That is, there are certain groupings of causal chains that exist in the form of large conceptual units.

We would anticipate that two special mechanisms are needed. First, we must be able to refer to a frequent event sequence in a sketchy manner. An event sequence with ten steps in it may be identifiable from just the first and last of those events. That means we need a special inference capability that can do more than the one described in Chapter 2. It would be a tedious and, most likely, unending process, to try to recover every missing event in a causal chain. So, the first special mechanism must be able to recognize that a script – a standard event sequence – has been mentioned.

Second, we need a mechanism for recovering steps that have been left out of a causal chain. Some of these steps may be needed to understand a given event sequence. We call this mechanism a script applier. It fills in the causal chain between two seemingly unrelated events by referring to the script recognized by the first special mechanism.

These two understanding mechanisms have their counterparts in the generation of language. When someone decides to tell a story that references a script, he recognizes that he need not (and because he would otherwise be considered rather boring, should not) mention every detail of his story. He can safely assume that his listener is familiar with the referenced script and will understand the story as long as certain crucial items are mentioned.

Let us look at some simple stories to see how this can be done:

2 John went to a restaurant. He asked the waitress for coq au vin. He paid the check and left.

3 John got on a bus. He fell asleep. He woke up in New York.

4 John was the quarterback. As time ran down, he threw a 60-yard pass into the end zone. His team won the game.

5 John went to Bill's birthday party. Bill opened his presents. John ate the cake and left.

These stories are understandable because they make reference to frequently occurring scripts. Much more than the three lines given in each story is understood by someone listening to it. A story understander must fill in the parts of each story that were left out. A story understander can do this by implicitly or explicitly referring to the referenced script.

Consider stories (6) and (7):

6 John went in to a restaurant. He saw a waitress. He went home.

7 John was walking on the street. He thought of cabbages. He picked up a shoe horn.

The connectivity of stories (6) and (7) is altogether different from that of stories (2)-(5). Stories (2)-(5) make reference to a script. Story (6) seems to reference a script but never quite gets there. By this we mean that the point or main goal of the script cannot safely be inferred. Did John eat or didn't he? You can't tell from this story. That is, in story (6) the events seem disconnected because of uncertainty that the referenced script should actually be instantiated. Story (7) does not reference a script and in any case it makes little sense. Stories need not reference scripts to make sense. Consider story (8):

8a John wanted a newspaper.
 b He found one on the street.
 c He read it.

Although (8) does not reference a script it is understandable. The events in it can be easily connected to each other with information readily obtainable from the story. In order to connect (a) to (b) it is necessary only to hypothesize 'John saw and picked up the newspaper.' One of these conditions is given by the story explicitly and the other is readily inferable. They do not violate necessary conditions or other conditions of the story itself (as discussed in Chapter 2). It is possible to connect (b) to (c) by the inferences that 'find' results in 'have', and 'have' enables 'read'.

Such simple connectability is present in (2)-(5) only by virture of the existence of appropriate scripts. Contrast Story (2) with Story (9):

2 John went to a restaurant. He asked the waitress for coq au vin.

9 John went to a park. He asked the midget for a mouse. He picked up the box and left.

In Story (9) we are unprepared for the reference to 'the' midget rather than 'a' midget and 'the' box rather than 'a' box. Further, we are incapable of connecting the last two lines of the story, without a great deal of effort. Story 9 allows us no reference to a standard situation in which midgets, mouses, boxes and parks relate. The story is not understandable, simply by virtue of the fact that we have no world knowledge that serves to connect its pieces. If there were a standard 'mouse buying script' that averred that only midgets in parks sold mice which were always packed in boxes, then we would be able to apply that script to Story 9 and connect the pieces of the story. What scripts do, then, is to provide connectivity.

In Story 2, which is superficially quite similar to Story 9, we get a great deal of connectivity. We are not suprised when 'the' waitress or 'the' check are mentioned. We understand exactly the relationship between asking for coq au vin and paying the check. Further, we assume that John ate coq au vin, that he waited a while before being served, that he looked at a menu, and so on. All this information is brought up by the restaurant script. Further, it is brought up by a particular part or track of the restaurant script, namely the kind of restaurant in which one orders coq au vin. This 'fancy restaurant track' of the restaurant script includes within it the possibility of a maitre d', a wine steward, tablecloths, paying with credit cards, fancy desserts and so on.

Thus the restaurant script must contain a tremendous amount of information that encompasses the enormous variability of what can occur in a restaurant. There must also be a 'fast food restaurant' track, a cafeteria track, etc. in the restaurant script, that includes the entering, ordering and paying scenes, but has a different set of possibilities than the fancy restaurant. In the 'fast food track', paying can occur immediately after ordering and before eating; eating may occur inside or outside the restaurant; the person who takes the order must be approached by the patron rather than going to where the patron is seated.

The presence of such tracks in the restaurant script is indicated by the understandability of stories that make use of those tracks. For example, consider Story (10):

10 John went into the restaurant. John ordered a Big Mac. He paid for it and found a nice park to eat in.

This story is understandable precisely because it calls up the track of the restaurant script that states that you don't have to be inside a fast food restaurant to eat there. However, if a reader does not understand that 'Big Mac' calls up the fast food track, he will have difficulty understanding the story. That is, the same story, with 'coq au vin' substituted for 'Big Mac', would seem rather odd. A story with this substitution would in principle be understandable, but the lack of applicability of available scripts would make it harder (and take more time) for a hearer to understand.

Thus while it is possible to understand a story without using a script, scripts are an important part of story understanding. What they do is let you leave out the boring details when you are talking or writing, and fill them in when you are listening or reading.

We shall now describe a script in more definite terms. A script is a structure that describes appropriate sequences of events in a particular context. A script is made up of slots and requirements about what can fill those slots. The structure is an interconnected whole, and what is in one slot affects what can be in another. Scripts handle stylized everyday situations. They are not subject to much change, nor do they provide the apparatus for handling totally novel situations. Thus, a script is a predetermined, stereotyped sequence of actions that defines a well-known situation. Scripts allow for new references to objects within them just as if these objects had been previously mentioned; objects within a script may take 'the' without explicit introduction because the script itself has already implicitly introduced them.

Stories (2)-(5) all make use of scripts. There are scripts for eating in a restaurant, riding a bus, watching and playing a football game, participating in a birthday party, and so on. These scripts are responsible for filling in the obvious information that has been left out of a story. Of course, it is obvious only to those understanders who actually know and can use the script. For example, these questions might be asked of hearers immediately after respective stories (2)-(5) with the full expectation of an accurate and fast reply.

Q1 What did John eat?

Q2 Where did the bus go?

Q3 What happened to the pass John threw?

Q4 Where did the presents come from?

Every script has associated with it a number of roles. When a script is called for use, i.e., 'instantiated' by a story, the actors in the story assume the roles within the instantiated script. If no actor has been specifically mentioned when a particular script is instantiated, his presence is nonetheless assumed and a default unnamed actor is used in his place. All this happens whenever a script is called up. This explains the use of the definite article in reference to 'the waitress'. She has been implicitly mentioned before by the initial instantiation of the script. (Roles are discussed at greater length in Chapter 6.)

A script must be written from one particular role's point of view. A customer sees a restaurant one way, a cook sees it another way. Scripts from many perspectives are combined to form what might be considered the 'whole view' of the restaurant. Such a 'whole view' is rarely, if ever, needed or called up in actual understanding, although it might well constitute what we may consider to be one's 'concept' of a restaurant.

We have built, at Yale, a computer program called SAM ('Script Applier Mechanism') that understands simple stories about script-based situations. It is described in detail in Chapter 8. Much of what we have to say here about script application has been influenced by our experience with that program. It has been tested most extensively with stories about restaurants. Let us consider the restaurant script in detail.

3.2 The Restaurant Script

The following is a sketch of one track of the restaurant script (the coffee shop track) from the point of view of the customer. Since the particular verbs that might best describe each action may not always fit in a given story that calls up a script, the actions of a script are described in terms of the underlying events that take place. The primitive ACT is the core of each event in the chain of events being effected. One of the scenes (ordering) is given below with a good deal of optional detail. The options to the right provide a single coherent path through the scene; shortcuts and loops are indicated on the left.

Track: Coffee Shop
Props: Tables
Menu
F-Food
Check
Money

Roles: S-Customer
W-Waiter
C-Cook
M-Cashier
O-Owner

Entry conditions: S is hungry.
S has money.

Results: S has less money
O has more money
S is not hungry
S is pleased (optional)

Scene 1: Entering

S **PTRANS** S into restaurant
S **ATTEND** eyes **to** tables
S **MBUILD** where to sit
S **PTRANS** S **to** table
S **MOVE** S **to** sitting position

Scene 2: Ordering

(menu on table) | (W brings menu) | (S asks for menu)

S **PTRANS** menu **to** S

S **MTRANS** signal **to** W
W **PTRANS** W **to** table
S **MTRANS** 'need menu' **to** W
W **PTRANS** W **to** menu

W **PTRANS** W **to** table
W **ATRANS** menu **to** S

S **MTRANS** food list **to** **CP**(S)
* S **MBUILD** choice of F
S **MTRANS** signal **to** W
W **PTRANS** W **to** table
S **MTRANS** 'I want F' **to** W

W **PTRANS** W **to** C
W **MTRANS** (**ATRANS** F) **to** C

C **MTRANS** 'no F' **to** W
W **PTRANS** W **to** S
W **MTRANS** 'no F' **to** S
(go back to *) or
(go to Scene 4 at no pay path)

C **DO** (prepare F script)
to Scene 3

Scene 3: Eating

C **ATRANS** F **to** W
W **ATRANS** F **to** S
S **INGEST** F
(Optionally return to Scene 2 to order more;
otherwise go to Scene 4)

Scene 4: Exiting

S **MTRANS** **to** W
(W **ATRANS** check **to** S)

W **MOVE** (write check)
W **PTRANS** W **to** S
W **ATRANS** check **to** S
S **ATRANS** tip **to** W
S **PTRANS** S **to** M
S **ATRANS** money **to** M
(no pay path): S **PTRANS** S **to** out of restaurant

Scene 2 may seem very detailed. In fact we have left out considerable detail and possible options in each of the scenes. We have left out whole scenes (the 'wait to be seated by the hostess' scene, for example). Everybody who has been to a restaurant often enough is aware of many more details and can use them if the occasion arises.

Human listeners have available another kind of information which we will not systematically treat, namely imagery (mainly visual) associated with each action in the sequence. Often, descriptive visual information is given in a story, but even if it is not, the listener hearing about a restaurant will typically call to mind impressions of the shapes, colors, relative positions and other properties of objects implicitly or explicitly present in the scene: tables, tablecloths, how the waiter or waitress is dressed, how the food looks (and smells), the check, the cash register, etc. With each action a 'vignette' containing auxiliary information is stored. The nature of the information in images has been the subject of much controversy (Pylyshyn, 1973; Kosslyn and Pomerantz, 1977), and we do not wish to stir up this hornet's nest here.

The restaurant script is a giant causal chain. Although the details have been left out, each action in the above script results in conditions that enable the next to occur. To perform the next act in the sequence, the previous acts must be completed satisfactorily. If they cannot be completed the hitches must be dealt with. Perhaps a new action not prescribed in the straightforward version of the script will be generated in order to get things moving again. This 'prescriptive' behavior, to be discussed later, is an important additional component of scripts. Script preconditions are another important part of the causal sequence in scripts. In the restaurant script, for example, we must inquire whether the main actor has money. If we have no evidence to the contrary, we proceed normally. Otherwise, we must find out if the main actor knows he has no money. If the answer is negative, we must predict that an interference will arise when the main actor tries to pay his bill; otherwise we must predict that the main actor may try to leave without paying. Such predictive powers are often used in understanding. Events with strong future implications are 'kept in mind' – like Charniak's (1972) 'demons' – so that they can resolve later inferential ambiguities.

In a text, new script information is interpreted in terms of its place in one of the paths within the script. Thus in story (2):

2 John went to a restaurant. He asked the waitress for coq au vin. He paid the check and left.

The first sentence describes the first action in scene 1 of the restaurant script. Sentence 2 refers to the crucial action of scene 2, and sentence 3 to the last two actions of scene 4. The final interpretation of story (2) would contain a chain through the restaurant script that included all the principal actions (or MAINCONS, for main conceptualizations) needed to connect the events.

MAINCONS are determined by their importance in a scene. For every scene there is at least one MAINCON. In scene 2 above, the MAINCON is the customer stating his order (**MTRANS** 'I want F' **to** W). If a scene is 'instantiated' its MAINCON must have happened.

Most real stories that deal with scripts relate events that are unusual with respect to a standard script. The problem in script application then, besides deciding how much of a script to infer, is to know how to tie together events that are not directly in the script.

Consider story (11):

11 John went to a restaurant. He ordered a hamburger. It was cold when the waitress brought it. He left her a very small tip.

In story (11) the first two sentences describe actions in scenes 1 and 2. Part of the third sentence is in the script as an action of scene 3, but there is also the information that the hamburger is cold. The fourth sentence ('He left her a very small tip') is a modification of the 'S **ATRANS** tip **to** W' action of scene 4. The modifier, 'very small' is presumably related to the unexpected information about the 'cold hamburger'. Even an unknowledgable script applier, checking story (11) against the standard restaurant script, could come up with the low-level hypothesis that the small size of the tip must have something to do with the temperature of the hamburger, since these two items of information are the only deviations from the script.

But we do not want our processor to lack knowledge. In slightly more complex examples, adequate understanding requires attention to the nature of deviations from the script. A smart processor can infer from a cold hamburger that the **INGEST** in scene 3 will not lead to the result of S having pleasure. The concept of a very small tip can be stored with the restaurant script as a reaction to the violation of pleasure. Thus the processor might even infer that a cold hamburger was unsatisfactory by working backwards from its understanding of a small tip. This might be necessary if the food description were ambiguous in desirability, say, a 'very rare steak', rather than a 'cold hamburger'.

3.3 Script Application

To define when a script should be called into play, script headers are necessary. The headers for the restaurant script are concepts having to do with hunger, restaurants, and so on in the context of a plan of action for getting fed. Obviously contexts must be restricted to avoid calling up the restaurant script for sentences that use the word 'restaurant' as a place ('Fuel oil was delivered to the restaurant').

Even if a proper header is encountered, however, it may not be appropriate to call up all the details of a script or even its MAINCONs. This is because script references in stories are often to 'fleeting scripts'.

12 John took a bus to New York.
In New York he went to a museum.
Then he took a train home.

In this example, the names of scripts are mentioned and it is presumed that each script proceeded normally. (Alternatively, it is possible that some abnormal things happened which were considered unworthy of narration by the author of the story. This alternative makes no practical difference except in very special cases.) There is a serious question about what 'proceeded normally' means in terms of what really is stored in the long-term memory of an understanding system.

Here we have the three explicitly stated scripts, **BUS**, **MUSEUM-GOING**, and **TRAIN**. (From this point, we shall indicate a script name by a **$** in front of the name; thus **$BUS** indicates the bus script). It is unlikely that people would fill in the default paths of each of these scripts if exposed to story (12). What is more likely is that they simply remember that the script occurred by establishing a pointer to the entire script. In this manner, the information about the script is available if needed, but memory is not cluttered with gratuitous detail. The story can be stored as a sequence of three pointers.

For a script to be non-fleeting, two of its lines must occur, a header and one other line. When a header is found, requests (as in Riesbeck's (1975) parsing system) are called up that connect possible inputs with events within the script. If such an input is found, then the script is 'instantiated'; that is, a copy of some of its general details is made, with slots filled in by the known properties of the story at hand. The role references are concretized. For example, a reference to 'the bus driver' results in the creation of a token with a pointer to the script role Driver. General information about specific roles such as Driver is stored under role themes, (see Chapter 6) and can be accessed if a role person undertakes some action not already in the script.

The nature of instantiated detail depends upon the story event(s) found after the header is found. Consider this story:

13 John went to a restaurant.
He ordered chicken.
He left a large tip.

The action of ordering calls in the ordering scene of the restaurant script. Since the entering scene lies on the path to ordering, we assume that its main conceptualization has taken place. Then the MAINCONs between ordering and tipping are assumed, as well as the final exit. Consequently our understanding system will treat example (13) as if it had actually been:

14 John went to a restuarant.
He sat down.
He read a menu.
He ordered chicken.
He ate the chicken.
He left a large tip.
He paid the check.
He left the restaurant.

That is, we fill in, as if we had actually heard them, the events on the default path of the applied script, as long as we are simply filling in the steps between explicitly stated points. Thus, in order to get from 'entering' to 'ordering', it is safe to assume 'sitting' and 'reading'. In order to get from 'ordering' to 'tipping' it is safe to assume 'eating'. Since 'tipping' is a prelude to 'paying' and 'leaving', we also assume 'leaving'. We do not want to assume too many steps when we are told of events that are far apart in the script. Thus, the story, 'John went to a restaurant. He left a large tip.', is considered odd. Do we want to assume that he ate? It is highly likely that John did eat in this story. Nonetheless, we might not want to simply assume it.

So, the rules for dealing with instantiated scripts are directly related to how many steps are left out. Essentially, instantiated scripts are those that make explicit one or more specific steps in the script itself. It is then our job to fill in the surrounding steps that ought to be explicitly inferred and treat them as if they were said.

The rules for activating a script are dependent on certain key concepts or conceptualizations when found in certain contexts. The restaurant context should not be called up simply because an input sentence refers to 'restaurant', but this is not to say that access to the script should be completely suppressed, because script-related information may be useful in later stages of understanding. For example, in 'I met a bus driver in the restaurant', remembering that one of the persons in the story has a role in the bus script may be crucial for interpreting what he might say or do afterwards. (Such role information is discussed further in Chapter 6.)

The conceptualizations which invoke a script are its headers. These headers come in four varieties, which are classified on the basis of how strongly they predict that the associated context will in fact be instantiated.

The first type is called a Precondition Header (PH) because it triggers the script reference on the basis of a main script precondition being mentioned in the text. For example, the sentence 'John was hungry' is a PH for the restaurant script because it is the goal condition for the MAINCON (**INGEST** food), which is normally assumed to be true when the script is instantiated. A story understander having access to both scripts and plans would make the prediction (a relatively weak one, to be sure) that the restaurant context would come up because this script is known to be a common means of implementing a plan of action for getting fed. A related PH would be an actual statement of the goal the script is normally assumed to achieve, or one from which the goal could easily be inferred. In 'John wanted a Big Mac', or 'John wanted some Italian food', the inference chain to the script precondition is straightforward. Knowledge about the existence of an Italian food subtrack of the restaurant script would make the PH prediction about the probable invocation of that script even more forceful.

A second type of Header making stronger predictions than a PH about the associated context is called an Instrumental Header (IH). An IH commonly comes up in inputs which refer to two or more contexts, of which at least one can be interpreted as a 'instrumental' for the others. For example, in 'John took the subway to the restaurant', both the subway and restaurant contexts would be predicted, since subsequent inputs about either make perfectly good sense. Here, the reference to the restaurant is anticipatory, and the subway is a recognized instrumental means of reaching locales in which more important script goals can be expected to be achieved. In turn, we understand that the restaurant script is in some sense instrumental to the business context in a sentence like 'John went to a business lunch'. An important function of scripts is to provide the background in which more planful activities are carried out.

The notion of a time-place locale for situations leads to the third and most strongly predictive type of header, the Locale Header (LH). Many situations are known to have a 'residence', a place or building where they characteristically go on. Indeed, many organizations have distinctively designed buildings (for example, McDonald's Golden Arches) which signal their script to the public. When an understander reads that an actor is in the proximity of such a residence, or better yet, inside the residence, expectations about the occurrence of the script are correspondingly reinforced. Examples of LH are 'John went to the soccer field' or 'John went into the Museum of Modern Art'. It is important to note that LH's need not be complete sentences: certain kinds of prepositional phrases ('At Leone's, John ordered a hot dog', 'On the bus, John's pocket was picked') are often used as a shorthand to define locale. Sentences like these can usually be paraphrased as a temporal clause of the form 'When X was at locale Y' attached to the main conceptualization.

The conceptual pattern that is being looked for here is X be **LOC**(script header). This pattern also occurs in places where we do not want to invoke a script. For example in (15):

15 The delivery man brought fifteen boxes of doughnuts to the restaurant. He went inside and spoke to the manager.

Clearly, the delivery man is in the restaurant here, but we do not want to predict that he will now eat (although he might). In (15), we call up the **$DELIVERY** script first. While the restaurant script can be a subpart of a larger script (such as **$TRIP**) it must be marked as not being capable of being subsumed by **$DELIVERY**. This marking calls off the restaurant script initially, but keeps open expectations for subsequent calls to the restaurant script. Thus if we see another scene of the script (e.g., if the delivery man sits down and orders) we must be prepared to initiate the full restaurant script. Thus, script headers can be suppressed by certain contexts.

The fourth kind of header is the Internal Conceptualization Header (ICH). Any conceptualization or role from a script may occur in a text. It will sometimes call the script up and sometimes it will not. The most obvious cases of these alternatives are when a role name (such as waitress) is used in the locale of the role or away from the role (as in 'I went out with a waitress').

A problem occurs when we have a story such as (16):

16 John went to visit his friend Mary who was a waitress. While he was waiting for her, he ordered a hamburger.

The reference to restaurant here is only by inference. However, that inference is enough to set up a possible expectation for the activities of the restaurant script. As we said earlier, two items are really necessary to be certain a script has been invoked. Here, the second item is a conceptualization internal to the script. Its recognition comes from the mention of waitress, plus the recognition of the conceptual sense of 'order' (i.e., the restaurant sense). Only with these two key concepts in context can the script be called.

3.4 Interferences and Distractions

Suppose that a script has been instantiated, and then a sentence comes along which does not relate to anything in the script:

17 John went to a restaurant.
He ordered veal scallopini.
The weather was rather poor.

There is no way in which the instantiated script helps the understanding of the third sentence. That unexpected sentence refers to a new topic which might be another script or might not. In any case, the computer or human understander must simply wait to see what comes next.

Often a sentence which does not seem to fit anything directly in an instantiated script can be related to the script indirectly.

18 John went to a restaurant.
He sat down and signaled the waitress.
He got mad.
He left.

In the story above, we must be careful not to assume all of the events on the default path of the restaurant script. The sentence 'He sat down and signaled the waitress' leads us to assume the default entering scene and the beginning of the ordering scene. But on seeing 'He got mad' we must stop processing the script in the normal fashion. At this point we must find out what could have made John mad – was it something within the scriptal context, or in some new context? The answer is inferred via a simple rule about anger, namely that it is ordinarily caused by something some other person either did or did not do. We immediately look at the script to see if some action is called for on the part of another person at this point in the script. The answer is that a waitress should come to John at this point. So we can assume that this did not happen, and that this is why John got mad and left. It is important to remember the point in the script where the exit took place. We do not want to infer the rest of the default path of the script (i.e., that he paid the check before leaving).

The above inference is a weak one. John may have gotten mad about something else. But text is usually presented so as to be understood correctly. That is, people don't intentionally mislead in stories of this kind. If something non-standard had occurred it probably would have been mentioned explicitly. In filling out scripts, we are relatively safe with weak inferences precisely because it is usual for non-standard occurrences to be explicitly mentioned.

In order to relate an unexpected sentence to an instantiated script we need to know what kinds of events can cause detours or abrupt endings in scripts. We recognize two broad classes of such events: interferences and distractions. Interferences are states or actions which prevent the normal continuation of a script. There are two types of interferences: obstacles, where some enabling condition for an impending action is missing, and errors, where an action is completed with an unexpected and inappropriate result.

The actor encountering an obstacle may respond by taking corrective action to try to produce the missing enabling condition. Such corrective actions we call prescriptions. Alternatively, the actor may give up, either immediately or after one or more prescriptions fail, and exit from the scene. The actor encountering an error is in a different situation. The usual correctives are loops – repetitions of the action to try to get it to come out right. Often a prescription must accompany the loop. For example, when the waitress brings you a hot dog after you order a hamburger, it is unreasonable merely to order a hamburger again as if nothing had happened. If the error is to be fixed, the standard prescription would be to explain to the waitress (or perhaps argue with her) that you did not order the hot dog. Alternatively, the actor may tolerate an error and proceed through the script anyway. Thus if the waitress brings the wrong order, the customer might either send it back and reorder, or accept the substitute or trade with a friend. Bad errors may of course present obstacles, so that the next action is not even enabled, e.g., if the waitress brings an empty casserole, then the option of eating anyway is removed.

Beyond responding instrumentally to an obstacle or error an actor may often also respond emotionally. He may express frustration, sadness, or anger at obstacles. After certain errors, he may be indignant, after others, amused. These emotional states are all reactions to interferences. They may be intense enough on occasion to abort the initial goal(s) of the script, as when the disgusted customer loses his appetite, and/or they may initiate derivative goals (e.g., punishing the guilty role person, say, by leaving the waitress a very small tip as in story (11)).

Distractions are unexpected states or actions which initiate new goals for the actor, carrying him temporarily or permanently out of the script. By their nature, distractions are not tied to a particular script – any number of things can distract a customer in a restaurant, for example. It is possible for some event to be both an interference and a distraction, such as the waitress dropping the soup, which fails to complete the **ATRANS** of soup to the customer, and which may initiate a new goal of getting the customer's clothes dry.

The above concepts provide a set of questions which a processor can ask when it encounters an unexpected input within a script:

a Does it specify or imply the absence of an enablement for an impending script action? (Obstacle)

b Does it specify or imply that a completed action was done in an unusual manner, or to an object other than the one(s) instantiated in the script? (Error)

c Does it specify an action which can be understood as the corrective resolution of an interference? (Prescription) This question would be activated when an obstacle is inferred from or described directly in the text.

d Does it specify or imply the repetition of a previous action? (Loop) This is activated when an error is inferred from or described directly in the text.

e Does it specify or imply emotional expression by the actor, likely to have been caused by an interference? (Reaction)

f Does it specify or imply that the actor will have a new goal that has nothing to do with the original script? (Distraction)

g Does it specify or imply the motivated abandonment of the script by the main actor? (Abandonment)

If any of the questions a – f are answered in the affirmative, then a detour is established within the script. New expectations will now guide the processing of subsequent inputs. A detour path will be followed until the original script either is reentered or abandoned. Scriptal deviations can thus be handled in a well-structured way.

The identification of inputs as obstacles, errors, etc., often depends upon having scripts available as points of reference. If we were not in a script, we might not recognize certain states as interferences, or if we did, we still might not know with what they were interfering. Compare, for example, the two stories:

19 John went to a restaurant.
He sat down.
He discovered he didn't have his magnifying glass.

20 John went for a walk.
He turned into Main St.
He discovered he didn't have his magnifying glass.

In story (19), it is easy to understand that the magnifying glass might be important because the menu is expected to arrive next. In story (20) we have no clear idea about the significance of the magnifying glass. 'A walk' does not specify enough of a sequence of events to be a script (unless we know John's personal habits). With no anticipated next event, we have no information on why the magnifier might be used.

The detour categories a-f tend to occur in certain standard patterns. One common type of sequence involves successful resolution of an interference:

(Obstacle) – (Prescription) – (Success),
or (Error) – (Loop) – (Success)

These success sequences return processing to the script at the point of the previously blocked action, in the case of an obstacle, or following the previously flawed action, in the case of an error. The category 'Success' may often be implicit, but sometimes it is explicitly marked, and we need to recognize it if it occurs. Consider an elaboration of story (19):

21 John went to a restaurant.
He sat down.
He discovered he didn't have his magnifying glass.
He asked the waitress to read him the menu.
She agreed.

The obstacle in the third sentence by inference relates to the action 'S **MTRANS** food list **to CP**(s)' in the Ordering Scene. The normal instrumental action for this is **ATTEND**ing eyes to the menu, and sometimes this **ATTEND** in turn has an instrumental action, namely **GRASP**ing the magnifying glass in proper position. The enablement of having the glass is here missing. The customer chooses a prescription of a type which is of general utility, namely asking someone else to produce a result difficult to produce oneself. (Prescriptions, like medicines, are sometimes general in their applicability, sometimes specific.) The waitress agrees to his request, returning processing to the script with the action 'W **MTRANS** food list **to CP**(S)' anticipated as a substitute for 'S **MTRANS** food list **to CP**(S)'.

It was of course possible for the customer to choose to restore the missing enablement rather than to modify the act to be enabled. (These two distinct categories are both generally pertinent in overcoming obstacles.) The fourth sentence might have been, 'He borrowed a magnifying glass from his old friend Moody at the next table', or even, 'He went home to get it.' These prescriptions return processing to the original action, 'S **MTRANS** food list **to** S', although the going home alternative has the interesting property that the customer leaves the restaurant. We of course should expect him to return, and it is only by understanding the nature of detour paths that it is possible to realize that leaving the restaurant does not here terminate the script.

Some Obstacle-Prescription pairs are so common that they may come to be recognized as a path of the script itself. In the Ordering scene if S needs a menu but it is not on the table and the waitress doesn't spontaneously bring it, then we have an Obstacle – the enabling menu for knowing the food list is missing. An obvious prescription is to signal the waitress to bring a menu. Anyone who has eaten with any frequency in restaurants knows that this ordinarily works. Therefore it is unnecessary and somewhat odd to use alternative prescriptions – say, searching by yourself for where the menus are kept – unless the primary prescription fails.

Later on in the ordering scene there is a common Error-Loop pair. If the customer orders something which is not available, then from the point of view of the restaurant, he has made an error. The loop which is initiated, namely ordering something else, is virtually unavoidable. Thus we treat it as part of the main script.

Every act in the restaurant (or any other) script is potentially subject to obstacles and errors, each of which suggests its own appropriate prescriptions or loops. A few of these will occur with sufficient frequency that a person repeatedly exposed to the script situation will learn them along with the rest of the script. This is the major way in which scripts grow. In time, he may learn a sizeable number of alternative script paths which were once detours, even to the point of having prescriptive sub-branches to follow if there are anticipated interferences to prescriptions themselves. Indeed, occupational role members must have very elaborate scripts from their situational point of view, e.g., a trial lawyer's conception of the courtroom script. Occasional or new participants in the same situation, or those knowing it only from hearsay, naturally have much simpler scripts. In a child's early experiences in restaurants, for example, there is no appreciation of many of the details we have listed, such as the waitress bringing the check. (Some aspects of a child's learning of scripts are discussed in Chapter 9.) To the extent that experiences in certain situations are different, then, different scripts would be appropriate. When we refer to 'the' restaurant script, therefore, we are relying on those stereotyped details which are culturally consensual.

Returning to our discussion of detours, another common sequence is a chain of attempts to remove a stubborn obstacle:

(Obstacle) – (Prescription) – (Failure) – (Prescription) – (Failure)...

This chain terminates either in a final success, or in a last straw (Failure) – (Abandonment) sequence. In the latter case, control does not return to the script, of course, because the script is terminated.

Failures are prone to elicit emotional reactions, albeit emotional reactions also occur in direct response to obstacles and errors. Reactions in turn may or may not interrupt the instrumental sequence. Thus we might have either:

(Obstacle) – (Prescription) – (Failure) – { (Reaction) / (Prescription)... }

or

(Obstacle) – (Prescription) – (Failure) – (Reaction) – (Distraction)

The (Reaction) – (Distraction) pair is meant to express those cases where the actor is carried away by his anger, annoyance, etc., initiating some nonscriptal action as a consequence.

Distractions need not occur only as a result of emotional reactions. Indeed, distractions can come from many sources. If the distracting events play out their course within the location of the situational script, then it is likely that control will return to the script at the point it was interrupted. The restaurant script may be said to be 'in abeyance' in the middle of the following story, for example:

22 John was eating in a restaurant.
Suddenly a thief tried to run off with several coats.
The manager tackled the thief.
The police came and arrested the man.
John paid the check and left.

With a script held in abeyance, the problem is to postpone the requests that were looking for completion of the script that was started (here, restaurant). That is, once the distraction scene starts we really do not expect the restaurant script to continue until the substory has ended. Nonetheless, it could continue at any point and requests to handle those inputs must be around.

A peculiar problem, though, is that the distraction substory may take the main actor out of the restaurant (or other script locus), and there is no telling whether or not he will return. One of the authors well remembers a personal experience at a modest restaurant with three friends some years ago. One of our group was not too hungry, and he asked the waiter just for a plate to share some of our spaghetti. The waiter said this would cost $.25 (a princely sum in those days), and we considered this an unreasonable pretension for such a humble place. Indignant, we decided to cancel our orders and leave. However, the waiter maintained that since the cook was already making the food, we were responsible for paying the bill. Now we were really mad, and we refused to pay. Thereupon he telephoned the police and we were led a block away to the station house and held on $200 bail. We were charged with failure to pay our obligation of $5.50, and were threatened with a night in jail. This sobered us, but we still did not want to surrender meekly to the restaurant manager who was standing by awaiting action. Did we return to the restaurant?... Well, yes and no. We solved our problem by asking if we could have the food as a take-out order. That was agreed, and we went happily home with no extra plate and no extra charge.

Such a 'realistic' story involves the interaction of three scripts in abeyance at once (**$RESTAURANT**, **$PETTY CLAIM**, and **$JAIL HOUSE**), with an unexpected resolution of the interferences in all three.

3.5 Script Interactions

There are several ways in which more than one script can be active at once. In the previous section, we discussed the possibility of a 'script in abeyance', with a distracting script occurring within its boundaries. Another possibility is that the second script does more than merely distract from the first, but actually interferes, preventing the occurrence of normal actions:

23 John was eating in a dining car.
The train stopped short.
John's soup spilled.

The first sentence activates two scripts simultaneously, **$RESTAURANT** and **$TRAIN**. (One could postulate a dining car script that we would expect someone who eats on dining cars to have. If we had that script here, these problems would not occur for this example).

When two scripts are active at once they compete for incoming items of information. Sometimes the events that fit in one affect the events of the other. The second sentence of (23) is clearly part of the train script. However, it causes a problem that doesn't usually occur in the restaurant script, namely that the table moved suddenly. We cannot expect the restaurant script to contain information about what to do or even what happens when a table moves suddenly. The third sentence (John's soup spilled) must be handled by means other than a script. This is done easily enough (in principle) by inferring the physical effects of a sudden train stop, and knowing that soup spills when moved abruptly. The problem here is the serious effect this sentence has on the restaurant script. It is as if the waiter has done something wrong, such as bringing the wrong order. That is, the customer can, at this point, ask for a replacement. A next sentence such as 'John called the waiter' would have to be handled as an Error-Loop detour path within the restaurant script. Remember that in this example two scripts are active at the same time. Any new item is potentially in either one. Here this means handling the spilling soup as an inference from the train moving and sending information to the instantiated restaurant script that the food is now no good (and perhaps that the customer is now wet). Such new inputs trigger detour paths in the restaurant script that are capable of handling them even though the impetus for them came from outside the script itself.

24 John was wooing his girlfriend in the restaurant.
He asked her for the salt.
Then he asked her for her hand.

In example (24), a similar problem occurs. Here again we have two scripts (**$ROMANCER** and **$RESTAURANT**) active at the same time. The next inputs don't affect each other, but which new input belongs to which script? The problem in this example is obviously not too serious and is quite a bit like semantic ambiguity in the disambiguation of isolated sentences. Resolution is possible as long as there is enough information in the script applier about the requirements of the two scripts. A serious problem occurs when a new event could occur in either of the two scripts. For example, suppose we had 'He asked her for money'. This might fit a path in the wooing script (as in 'he is after her for her money'), or it might be part of the restaurant script (as in the path that handles what to do if you discover you can't pay). This is a case of ambiguity within scripts which we call Scriptal Ambiguity. Often, other knowledge (for example, about John's personal character) will help disambiguate such sentences. If not, future inputs usually will.

An amusing kind of scriptal ambiguity occurs when the players think they are in different scripts, or when two players in a single script each have two different roles, one real and the second a figment of the other person's imagination – as in the following anecdote:

25 A traveling salesman found himself spending the night at home with his wife when one of his trips was unexpectedly canceled. The two of them were sound asleep, when in the middle of the night there was a loud knock at the front door. The wife woke up with a start and cried out, 'Oh, my God! It's my husband!' Whereupon the husband leapt from the bed, ran across the room, and jumped out the window.

If we regard husband-surprising-wife's-lover as a script, then we can readily understand this anecdote as the husband seeming to the wife to be her secret lover, and the wife seeming to the husband to be the spouse of the jealous husband outside. For a script application mechanism to appreciate this duality, it would have to infer one version of the script from the wife's point of view, and a second from the husband's. It would have to understand, in other words, that there can be a script in someone's mind – a personal script – which is at variance with the actual or situational script because of systematic distorting factors.

Of course, in order to really understand the joke in story (25), the listener must apply a rule about personal scripts, namely that in order for a personal script to override serious discrepancies with reality, it must have been very well practiced by the individual. Thus we infer that both husband and wife are well versed in adultery. We will have more to say about personal scripts in Section 3.6.

The concurrent activation of more than one script creates rather complex problems. A slightly less troublesome type of script interaction arises at the boundary where one script leaves off and another begins. Consider this example.

26 John was robbed on the train.
At the restaurant he couldn't pay the check.

In this example, the robbery is an unpredicted event in the train script. The new event does not affect the normal completion of the train script, so we simply have a pointer to the train script (since it is a fleeting script) and a pointer to a robbery script embedded within it. However, when the next line of the story is seen, it is affected by the earlier robbery script. This is noticed by the entry conditions on a script. In order to perform a role in a script, certain conditions must be met. To take a train ride, one must be able to get to the station and into the train (i.e., the doors of the train must be open at the appropriate time). Further, one must either have a ticket or the money to purchase one (in which case someone must be selling a ticket and one must be able to find him). The entry conditions for a customer in a restaurant are similar to those for a train. Obviously an important entry condition for restaurants is that the customer have the means to pay the check. When a script is begun, it is necessary to check the entry conditions. If an entry condition has been violated it must be noticed immediately upon instantiating the script or when a pointer to that script is created. So in (26), the robbery, from which one must infer John has no money, violates one of the entry conditions of the next script to be activated. This violation sets up an expectation for the no-pay path of the resturant script with a link back to the robbery as the reason for taking this path of the script.

A final script interaction type concerns indeterminacy in script-endings.

27 Yesterday John was in New York.
He went to a restaurant.
He ate a large lobster.
Then he bought a watch.

In (27) we have the problem of recognizing when an active script has been ended and a new script has begun. Once the restaurant script has been instantiated in (27), we expect it to be ended in normal fashion. When a new input comes in that does not normally fit in that script without an ending being perceived, we have a problem. If the new input is something which is unexpected but could possibly occur in a restaurant, (i.e., is a Distraction), should the restaurant script end? If the new input were, 'The waitress did a dance', we would have no reason to end the restaurant script since this could occur in a restaurant, and would not be likely to initiate a customer reaction leading to a **PTRANS** from the restaurant. The restaurant script would simply be held in abeyance. In the example given in (27), ('Then he bought a watch') we have something that can normally be handled by a script, but that takes place in watch stores and not restaurants. We have to assume that the restaurant script has ended and infer 'He left the restaurant'. This would cause all the normal MAINCONs of the restaurant script to be inferred. The watch-buying event thus serves double duty: it activates a new script at the same time it terminates an instantiated old one. We call this a 'script-ending script'.

Of course, it is conceivable that one could buy a watch in a restaurant. Because of this possibility, with script ending scripts we still keep the requests active from the original script. Thus, if we next encounter 'Then he paid the check', if we have marked our previous inference with a lack of certainty, we can undo what we have inferred and place the 'watch' event inside the restuarant script as a Distraction.

The problem of script-ending scripts is a difficult one, partially because it occurs frequently, and partially because one can never be certain that the right decision has been made. Time span seems to play a role in the decision process too. For example, if 'Then he bought a watch' were (a) 'He bought a watch' or (b) 'He bought a watch an hour later' we have different solutions. In (a) we would probably assume that John was still in the restaurant and in (b) we would feel more certain that the restaurant script was ended because of the time gap.

3.6 Types of Scripts

So far we have examined situational scripts in which 1) the situation is specified; 2) the several players have interlocking roles to follow, and 3) the players share an understanding of what is supposed to happen. The waitress typically does what the customer expects, and the customer typically does what the waitress expects. There is great social economy when both parties know the script because neither party need invest effort deciding what the actions of the other mean and how appropriately to respond. Indeed, it is characteristic of institutionalized public situations with defined goals (the customer eating, the restaurant making money) that the social interactions be stylized. This is one reason why scripts are so common, and so helpful in understanding.

Suppose, however, that one of the parties wants to direct the interaction into channels other than those defined by the situational script. He may have some Personal Script which he is following, over and above the actions needed to conform to the situational script. The customer for example, may have the goal of making a date with the waitress. If he has pursued such a goal often, then the actions involved may (for him) be very stylized and scriptal. The main path might involve friendly conversation, casual kidding, finding out if she is unattached, displaying interest, and asking when she gets off work. The waitress might or might not respond in the anticipated way. She might be friendly, but misinterpret the motive for the customer's friendly overtures. She might remain aloof and business-like. She might play hard to get.

Personal scripts do not behave in the stylized fashion of situational scripts. All the participants in personal scripts are not necessarily aware of their participation. The seducee, say, or the victim of a swindle is often not aware until the very end of the enactment of the actor's personal script of their participation in it. The personal script exists solely in the mind of its main actor. It consists of a sequence of possible actions that will lead to a desired goal. It is different from a plan (to be discussed in Chapter 4) in that there is no planning involved for the actor in a personal script. He is participating in a sequence of events much like other sequences he has used many times before. He could teach his method to anyone who wanted to know it. There is very little planning involved because he has done this personal script repeatedly.

There is, of course, no limit to the mental projections a person can bring to a situation in the hope of attaining some goal. These are frequently not script-like, but suited *ad hoc* appropriately to the particular situation. If the customer says to the waitress, 'If you see a tall man with a walrus mustache later this evening, a Mr. Robinson, please tell him that John and Mary found his umbrella', we don't want to try to interpret this request as part of a script. It is unique to this particular situation. Indeed, even if this customer for some reason often says such things to waitresses as part of a peculiar personal script, an understanding system would have great difficulty perceiving this without intimate knowledge of the mental world of this customer. From the standpoint of an artificial intelligence system, therefore, the useful personal scripts to store are those which are common to many individuals, and can therefore be conjectured for new characters in a story. There are many such common personal scripts. They tend to have the character of roles or parts which people assume as the occasion arises, for example, **$FLATTERER**, **$JEALOUS SPOUSE**, **$GOOD SAMARITAN**, etc., or stealthy occupations like **$PICKPOCKET** or **$SPY**.

Personal scripts are usually but not always goal-oriented. A personal script also might be followed as a matter of ritual (e.g., **$PRAYER**), or as an elaborated emotional and behavioral reaction following a situational outcome. An example of the latter would be the **$JILTED LOVER** who (say) discovers he has been jilted, disbelieves it, confirms it, is furious at his rival, curses all women, feels depressed, gets drunk, and throws himself in the river. This non-goal-oriented type is not quite as interesting from an artificial intelligence point of view as the goal-oriented type, because other individuals do not 'get into the act'. The personal script can be very personal indeed.

Clinical psychologists tend to be interested in very personal scripts. These are the stuff of neuroses, especially when they are activated inappropriately and create interferences in the ongoing social behavior of the individual. Behavior governed by unconscious motivation stems from a script hidden from an individual's conscious self. It is not our task in this volume to pursue such matters, other than to note the potential relevance of the script concept, properly explicated, to the demystification of neurotic behavior. Indeed, a school of clinical psychologists – the 'transactional school' – has already used the term 'script' for use in the analysis of behavior (cf. Steiner, 1975) although their use of it is looser than ours. In any case, we will not try to deal with idiosyncratic or unconscious personal scripts. The knowledge needed to handle their occurrences is too specialized and unparsimonious and is of little use in predicting and understanding actions at the level at which we are interested. Suppose we encounter a story like (28):

28 Mary's friends offered heroin to her. She shot up.

Here the simplest answer to the question, 'Why did Mary shoot up heroin?' is that she wanted to (or that it was her habit), and the next simplest is that her friends persuaded her. Without very specialized additional context, we would not likely assume (say), that she did it because as a child she hated her overbearing older sisters, and now she lets her peers talk her into bad behavior in the hopes that they will all be caught and punished, thus getting even with the sisters and also expressing her own guilt for hating them. This could conceivably be correct, but it is so much explanation based on so little substance that, like syndicated newspaper columns offering psychiatric advice, it puts one off as being gratuitous. Our policy in developing a theory of knowledge structures is to get as far as we can with fairly simple and general constructs.

With personal scripts, then, we would in practice restrict our attention to the most common readily inferred type. Often, personal scripts are used in otherwise novel situations, where there are no other scripts around. Interesting interactions occur when more than one script type is around at a time. Here we simply sketch some possible interactions and their properties.

a One actor with a concealed personal script within a situational script.

This is a very common type of interaction. One actor behaves with stylized duplicity, maintaining a public front while pursuing a personal motive. Persistent toadying to superiors in hope of a promotion, the affectation of virtues by political candidates, the rapid-fire friendly conversation of reporters, salesman, con artists and spies – these are all potential examples.

This type of script interaction is different in an important respect from most of the types discussed in previous sections. Here if a story understander is aware of the personal script, he is set to expect certain interferences during the progress of the situational script. This is in contrast to stories such as (16), (21) – (23), (27), in which unexpected or accidental events arise without prior warning. With prior knowedge of a personal script, the understander can prime appropriate questions or requests of both non-scriptal and scriptal input.

b Two or more actors with competing concealed personal scripts within a situational script.

This is an extension of the previous type. It might involve spy and counterspy, or a group of dishonorable thieves in a bank robbery, or any number of other situations of competition and double-cross. Here again the interesting questions for the understander are how the protagonists manage their double roles, and whether either or both of them develops awareness of the other's hidden agenda. The complexities of monitoring the input are much greater than in the previous case, but there are no new conceptual features.

c One or more actors with personal scripts whose nature is known by the other actors.

We have discussed the possibility that a personal script be concealed, but it is not unusual for a personal script even to be known in advance by other players. There are many stylized interactions in which one or both parties know the other to be dissembling, as in polite social invitations known not to be intended. Quite often if the deceit is relatively harmless, the knowing victim will pretend not to know, in order to save the face of the other. This is what Goffman (1959) calls the 'face work' of everyday social interaction. Rather complicated patterns of social misrepresentation (both script-like and non-script-like) can occur, and it is not our purpose here to try to trace these. At this stage in the development of script concepts, perhaps we can hope only to cope with very obvious personal-situational script interactions where foreknowledge or discovery of the other's personal script leads immediately to a prescribed action. For example, if a dope peddler discovers that some people acting like customers are in reality members of the dope squad, he will avoid the incriminating rendezvous.

There is one further type of script we recognize, namely the INSTRUMENTAL SCRIPT. Instrumental scripts are quite like situational scripts in structure, that is, they describe prescribed sequences of actions. However, the kinds of actions they describe, the variability of the ordering, and the use of the script in understanding, differs. Examples of instrumental scripts are **$LIGHTING A CIGARETTE, $STARTING A CAR, $WORKING A KEYPUNCH, $FRYING AN EGG**. There is little variability with instrumental scripts. The order of events is very rigid, and each and every one of the events in the script must be done. There are obviously an extremely large number of instrumental scripts. Every cookbook contains hundreds of them.

There are, of course, situational scripts, in which the variability is about the same as some instrumental scripts. In the coffee shop track of the restaurant script there is little possibility for actions varying. In fact, there is probably more possible variability in frying an egg. The crucial differences between instrumental and situational scripts are with respect to the number of actors, and the overall intention or goal of the script.

Usually, situational scripts take mulitple actors, while instrumental scripts have only one participant. Things can and do happen in a situational script that are not expected and are often the point of any story that invokes situational script. For example, a fight in a restaurant is of interest because of its unusualness. The restaurant is just so much context for the story. Barring a bomb going off, we don't expect stories about 'what happened while I was starting my car'. The actions in instrumental scripts are fixed and uninteresting for the most part. Unusual or other interesting events do not usually take place with instrumental scripts as their context. In an instrumental script, nearly the only thing that can be related within it is the failure of the intended goal and what was done to correct it. This is usually itself a standard prescription. There are only a few things to do when a car doesn't start and they are part of the script as well.

Making the distinction between situational and instrumental scripts enables us to make some choices that facilitate processing. When we instantiate a situational script, we must set up prediction mechanisms that will: be able to handle definite references to characters that have not yet been mentioned (e.g., the waitress); be able to infer the presence of important (or goal) scenes that have not been instantiated (e.g., 'eating' in a restaurant); find the appropriate detour path for unexpected inputs. Most importantly, mechanisms of memory must be set up to remember the unexpected events of the situational script together with the explicit and inferred MAINCONS.

An instrumental script has available to it much of this apparatus, but it is unreasonable to bring it to the fore every time that an instrumental script is referenced. We simply don't expect that 'I fried an egg' is the beginning of a story about an interesting thing that happened in the process of egg frying. To bring powerful prediction mechanisms to bear at this point would be a mistake.

Perhaps more important is our treatment of these two script types after they have been processed. We have stated what we would like to remember after having made use of a situational script. What is the analogy with instrumental scripts? It is not unreasonable to expect that, except under very unusual circumstances, we would want to forget the details of an instrumental script and remember only the goal. In fact, it is not implausible to even forget the script entirely, to save memory space and processing time. The reason that this can be done is that an instrumental script can always be rediscovered. If someone is smoking a cigarette, it must have gotten lighted somehow. If, for some reason we ever need to use this fact, it can be inferred and found as easily as if we had been told it.

The three script types described in this chapter can occur together in a juxtaposition that might include them all. For example, John could take Mary to dinner at a restaurant, doing various instrumental scripts along the way (lighting her cigarette, starting the car). However, during the meal he is affecting the personal script of **ROMANCER**. This affects his behavior every now and then, in what he says, how he walks, what wine he orders, but probably not in that he orders or pays the check (situational) or how he cuts his meat (instrumental).

3.7 Script-based Understanding

By subscribing to a script-based theory of understanding, we are making some strong claims about the nature of the understanding process. In order to understand the actions that are going on in a given situation, a person must have been in that situation before. That is, understanding is knowledge-based. The actions of others make sense only insofar as they are part of a stored pattern of actions that have been previously experienced. Deviations from the standard pattern are handled with some difficulty.

To illustrate this, consider the following example that recently happened to one of us. I received a phone call from an old friend who lives about 100 miles north of me. He said 'Hi, I'm on I-91'. (The highway that connects our respective cities.) I asked him if he would like to drop by and he said he would. I then asked him where he was exactly so I'd know when to expect him. He answered 'Well let's see, oh there's a sign coming up, it says, wait a minute, Wallingford next, uh, oh there it is, exit 14'. At this point I was totally baffled. I had visions of my friend having lost connection with reality. My 'telephoning before visiting' script did not have room in it for uncertainty about one's location that could be resolved by signs 'coming up'. It took me a while to realize that my friend, being a gadget-oriented person, had a phone installed in his car. What we had was the telephone script mixed with the driving script, a mix that I was quite unfamiliar with.

Of course, people can adapt to situations with which they do not have previous experience. This adaptability comes from knowledge of plans and goals discussed in Chapters 4 and 5. However, even there the point remains the same. People need a great deal of knowledge in order to understand. That knowledge can be of two kinds: specific and general. Scripts are intended to account for the specific knowledge that people have. Most of understanding is script-based.

Understanding then, is a process by which people match what they see and hear to pre-stored groupings of actions that they have already experienced. New information is understood in terms of old information. By this view, man is seen as a processor that only understands what it has previously understood. Our script-based program, SAM, works this way. It thus can be faulted on the basis that anything that it understands was preprogramed into it in gory detail. We will meet other bases for understanding, but we view human understanding as heavily script-based. A human understander comes equipped with thousands of scripts. He uses these scripts almost without thinking.

We will discuss how scripts are acquired in Chapter 9. A simple example will suffice here. One of us (RS) recently bought a new car. My daughter Hana (age 4) was with me when we bought it and asked if I was going to get a new key chain. I asked her what she meant. She replied that when we had gotten our old car in Rhode Island (where it had arrived off the boat 2 years earlier) I had bought a new key chain. This was her only experience with getting a car and already the events in it were a script for her. When you get a new car you get a new key chain. If people are building scripts at such an early age, it seems easy to imagine that the number they possess is great.