

about the story concerning what happened, when something did happen, and why it happened.

In addition to scripts, which capture the essence of a stereotyped sequence of actions or events for a well-known situation, there are other comprehension mechanisms such as plans. Plans handle novel situations, plans describe the set of choices associated with accomplishing a goal. When a plan is frequently implemented it becomes a script. Most people do not have a "how to become president" script or a "what to do when the house burns down" script, but they could implement a plan. An example of a script common to most Americans is a restaurant script. Consider (1) below:

John went into a restaurant. He ordered a hamburger and a coke. He asked the waitress for the check and left. (1)

Many aspects of this standard script have been omitted, but are automatically inferred. If the reader of (1) were asked at a delay "Did John eat a hamburger?," the response would probably be "Yes," although that was never explicitly stated. One way to tell if an episode is a situational script is by use of the "reference test" (Schank, 1975a). When a definite article precedes a noun not previously mentioned, it is probably acceptable because the referents have been implicitly introduced by the script. Compare (2) and (3):

John went to a restaurant. He asked *the* waitress to tell *the* chef to cook him a hot dog. (2)

John went to a restaurant. He struck up a conversation with *the* bus driver. (3)

In (3) we would feel uncomfortable using "the" with bus driver and might try to augment the script to explain the anomaly.

Scripts are a set of scenes and within each scene there is a causal chain such that each action *enables* the next. For example, in the restaurant script, asking for the check *enables* the patron to receive the check. Many parts of a script can be modified and, in fact, frequently are. There are only a few aspects of a script (e.g., ingesting food) which are considered critical. Often there are deviations. Consider scenario (4):

John went to a restaurant. He ordered a hamburger. It was cold when the waitress brought it. He left her a very small tip. (4)

Since there are two deviations from the typical script, viz., "cold" and "very small," the processor would hypothesize that the deviations are related. In general, however, deviations from scripts are handled by "what if" or "whifs" associated with every situational script.

Even though we do not have scripts for every situation encountered in a story, we do need relevant conceptual structure to understand a given input and infer the appropriate connections. Consider the following passage (5):

John knew his wife's operation would be very expensive. There was always Uncle Harry ... He reached for the suburban phone book. (5)

It is unlikely that we have a "paying for an operation" script, yet we may have a "raising money for important expenditures" script. That is, the comprehension of (5) in terms of inferences made would not be very different if "son's education" were substituted for "wife's operation." In each case there is a general goal state. Trying to achieve a goal involves a plan.

A plan is a series of actions that will hopefully realize a goal. Much like the *General Problem Solver* (Newell & Simon, 1972), plans to achieve the goal state go through intermediate states, trying to reduce differences between the goal state and current state. Plans consist of a set of "deltacts," actions or subplans. When a set of actions is used often enough together, the set becomes a script. Examples of general purpose subgoals used to achieve a desired state are *D-CONT*—a change in the control of an object; *D-KNOW*—a change in what an actor knows; and *D-PROX*—a change in the proximity relations of objects and actors. Actions that are part of a chain involved in a particular action used for a general purpose belong to a "planbox." Certain plans may invoke a planbox to satisfy a goal. For example, in order to gain control of an object (*D-CONT*) one may first ask a person for it, bargain for it, threaten a person for it, steal it, or overpower a person in order to take it. Each of these actions involves a number of prerequisites and results in changes of state if the actions succeed. The preconditions that the actor can affect (e.g., one must be "near" the person one is going to ask, or at least physically capable of asking) as well as the resulting state changes are listed in the planbox.

Sometimes in the middle of a story, the actions that a character is pursuing will be frustrated and there will be an abrupt shift in plans. In order to better understand why the actor would shift from the one activity to another, Schank and Abelson (1977) described a theory of goal substitution, goal forms, goal initiations, etc. When the system cannot find an appropriate script, it tries to induce what the character's plans would be using the goal monitor. In addition to keeping track of the motivating influence of the character's goals, the monitor must recognize when a goal has been triggered, make predictions about what events will be caused by it, and keep track of a goal's fate.

In Schank and Abelson's system, the representation of a story has three levels of description: (1) the lowest level is conceptual dependency (*CD*), described earlier; (2) main conceptualizations, (*MAINCONS*), describe the important aspects of a scene and of a script; (3) the highest level is the knowledge structure (*KS*) which has the script, plan, or theme information. The *CD* level is in some ways similar to Kintsch and van Dijk's microstructure, while the *MAINCONS* are like the macrostructure. *MAINCONS* generate expectations. The three levels are connected by pointers. Like Kintsch and van Dijk and others, Schank and Abelson do not think the lowest level is involved in summarizing. Schank has also predicted that items most likely to be forgotten are those that are at "dead ends" in causal chains. However, unlike Kintsch and van Dijk, Schank has never empirically tested his ideas.

A major weakness of script theory is devising a reasonable scheme for evoking a script: How does the program know if a script is the correct one? Sometimes several scripts seem appropriate concurrently (e.g., a restaurant script and a romance script). There is no mechanism that allows both scripts to be operating and to make predictions at the same time, let alone interact. These difficulties are not unique to script theory. All extant theories that rely on "bundles" of knowledge to help direct