Document Number
TRACOR 68-462-U

OUTLINE OF A CONCEPTUAL SEMANTICS FOR GENERATION OF COHERENT DISCOURSE

by

Roger C. Schank

March 1968

TRACOR
6500 Tracor Lane, Austin, Texas 78721, AC 512/926-2800

45

TRACOR 6500 TRACOR LANE, AUSTIN, TEXAS 78721

Document Number
TRACOR 68-462-U

OUTLINE OF A CONCEPTUAL SEMANTICS FOR GENERATION OF COHERENT DISCOURSE

by

Roger C. Schank

March 1968

ACKNOWLEDGMENT

The author has benefitted through conversations with Jacob Mey and Walter Stolz of The University of Texas, whose comments have been most helpful. E. D. Pendergraft, in his work on semiotic systems, is providing the context and encouragement for this study.

TRACOR 6500 TRACOR LANE, AUSTIN, TEXAS 78721

ABSTRACT

This paper develops a method for generating coherent sentences. A conceptual semantics is presented, that, when coupled with a conceptual dependency abstraction of meaning, allows concepts to be linked in a manner consonant with the system's knowledge of the world. The paper is part of a series of papers concerned with the problem of language synthesis for artificially intelligent systems.

PREFACE

The title of this paper is somewhat redundant since semantics as referred to here is extra-linguistic. The standard notions of a semantic component assign an interpretation to a piece of discourse. That is not our concern here, however. The problem attacked herein is one of language synthesis. Specifically, given some abstract conceptual framework, how do we go about relating it to actual linguistic behavior?

This paper deals with the generation of sentences on both a random and non-random basis. (Although the system presented here is one for generating coherent sentences, the processes involved are easily extended to larger pieces of discourse. The Conceptual Dependency framework utilized may represent discourse structures as well as sentences.) The largest concern is with just what processes are involved in generating discourse in such a way that it is coherent. That is, we expect our linguistic expressions to conform to reality. The system for generating syntactically correct utterances has been discussed elsewhere [7]. This paper is concerned with making those utterances consonant with the real world, i.e., meaningful. An outline of a semantic system is developed for doing this.

TRACOR 6500 TRACOR LANE, AUSTIN, TEXAS 78721

TABLE OF CONTENTS

1. INTRODUCTION

1.1 LANGUAGE SYNTHESIS

A synthetic system starts with the results of analytics.

> "The problem (of synthesis) is to give explicit definitions and statements of the components and rules of combinations, such that the whole can be systematically regenerated." (Hirst and Hirst [2])

With respect to language, a synthetic system must begin with some content analytic formulation of language in terms of concepts and relations between concepts, and systematically map these into a linguistic whole. The content analytic form must be a conceptualization of some linguistic input. That is, the result of analytics in this case must be some sort of conceptual parse that expresses the meaning of the input. The apparatus used in expressing this meaning deals with concepts and relations between concepts, and therefore may be said to be an abstraction of the human method for doing this.

In the process of synthesizing language then, the conceptual apparatus utilized as a receptacle for the results of analytics must be used as the starting point. The process of putting concepts together in a manner analogous to human thought processes may be thought of as conceptual generation. It is this conceptual generation and the subsequent mapping of the generated concepts and their relations into language that may be thought of as constituting the synthesis of language.

1.2 ELEMENTS OF THE SYSTEM

The conceptual apparatus utilized by the synthetics of the artificial intelligence system generates concepts by checking with a store of its knowledge of the world. The conceptual networks generated are then mapped into language on the basis of the system's previous linguistic experience.

The semantics of the system consists of lists of possible combinatory powers of the concepts. The items on these lists are functions of the concepts themselves and fall into general classifications. That is, concepts of physical objects share similarities in their possible combinations with other concepts. For example, all physical object concepts may be modified by concepts of shape and motion. Classifications may be subdivided in order to further delimit the range of combinations allowable; e.g., the category 'animal' might be divided at some point into 'human' and 'non-human' in order to restrict the combinatory powers of the category. The purpose of all this is to limit the generated conceptualizations to plausible thoughts about the real world.

After a portion of the conceptual network is partially complete, words are chosen to represent the particular concepts. Governing concepts (see below) are realized first, their dependents being chosen in accordance with a linguistic experience file. (The process of linguistic realization is dealt with by Schank [7].) The linguistic experience file contains records of previous word associations and previously encountered phrases and contexts. The point of this is to choose an acceptable construction to realize a given conceptualization. Without this file the system would appear to be a foreign speaker of a language, semantically and grammatically correct, but with a queer choice of words.

1.3 ON RANDOM GENERATION

The schema presented in Section 2 is a system for generation of random coherent discourse. Generation, if not random, must respond to some input. The input (in the form of a question) would necessarily be semantically well formed. If the input merely questioned a particular link in the system's memory, then the answer would have the required semantics already built in (see Schank [8]). However, we are concerned here with the entire process of language synthesis and therefore with the semantics required for this synthesis. It is clear that in a question-answering system, generation of large amounts of discourse not based totally on the input question must be possible. Since we do not know what our motivations in a particular choice of concept might be at any given point, we must be prepared to fully understand a system that is free of motivation. That is, we must know what a semantic theory for random generation looks like.

If generation is random, each generated concept must still conform to the system's experiential knowledge about the world. Each time a concept is generated and attached somewhere it is important to find out if such connections actually occur in the real world; e.g., Do balls have color? Are oranges edible? etc. Thus, in some sense, the semantics are nothing more than a check with reality. 'Colorless green ideas' could not be generated since neither connection (colorless → green or green → ideas) is possible. (The notion of reality expressed here is not to be taken too literally. Although "unicorns" are not members of the real world, they are part of the conceptual real world. That is, one can understand the notion of a "unicorn" and can ascertain certain properties about "unicorns." The system's real world is not necessarily one of grim reality.)

Certainly, generation of discourse by humans is not random. Along every step of the way certain aspects of the input (a question) and the internal experience files of the mind choose the appropriate concepts to be used in formulating the output (an answer). However, the lines followed in generation of free discourse (not just affirmation of a chain in a link, such as "$x \overset{?}{\Longleftrightarrow} y$; Yes $x \Longleftrightarrow y$," (for notation see Schank [7])) are the same as those used in random generation. This is true precisely because random generation is not random at all, the only random elements being particular concept choices. These concept choices, even when well motivated and entirely purposeful, still must be made from a list of possible conceptualizations, such as is found in one of the semantic lists (see below). Thus, the generation tactics employed are the same in either case. The difference between random generation and human generation of discourse is solely one of appropriate motivation.

2. THE SEMANTICS OF RANDOM CONCEPTUAL GENERATION

2.1 CONCEPTUAL DEPENDENCY

The Conceptual Dependency apparatus provides for a method of abstraction of the meaning of a piece of discourse. This abstraction may be realized in a linguistic form by means of language-bound rules that map conceptual items into linguistic items in a linear form. In the last formulation of the Conceptual Dependency model (Schank [7]), the schema presented provided the framework by which all the grammatical strings in a language could be generated, using the abstract representation of their meaning provided by Conceptual Dependency.

The Conceptual level in the stratified Conceptual Dependency model is responsible for providing a representation of the meaning of a sentence. (It is not restricted to a sentence, however.) It does this by utilizing certain conceptual categories and combining them in various ways with other categories in order to form a complete network representative of the meaning. The rules for generating and combining conceptual categories are consonant with the real world and are necessarily language-free. Consequently, a Conceptual Dependency representation of meaning may be used as the starting point for mapping into any language. In this sense, it may be thought of as an interlingua.

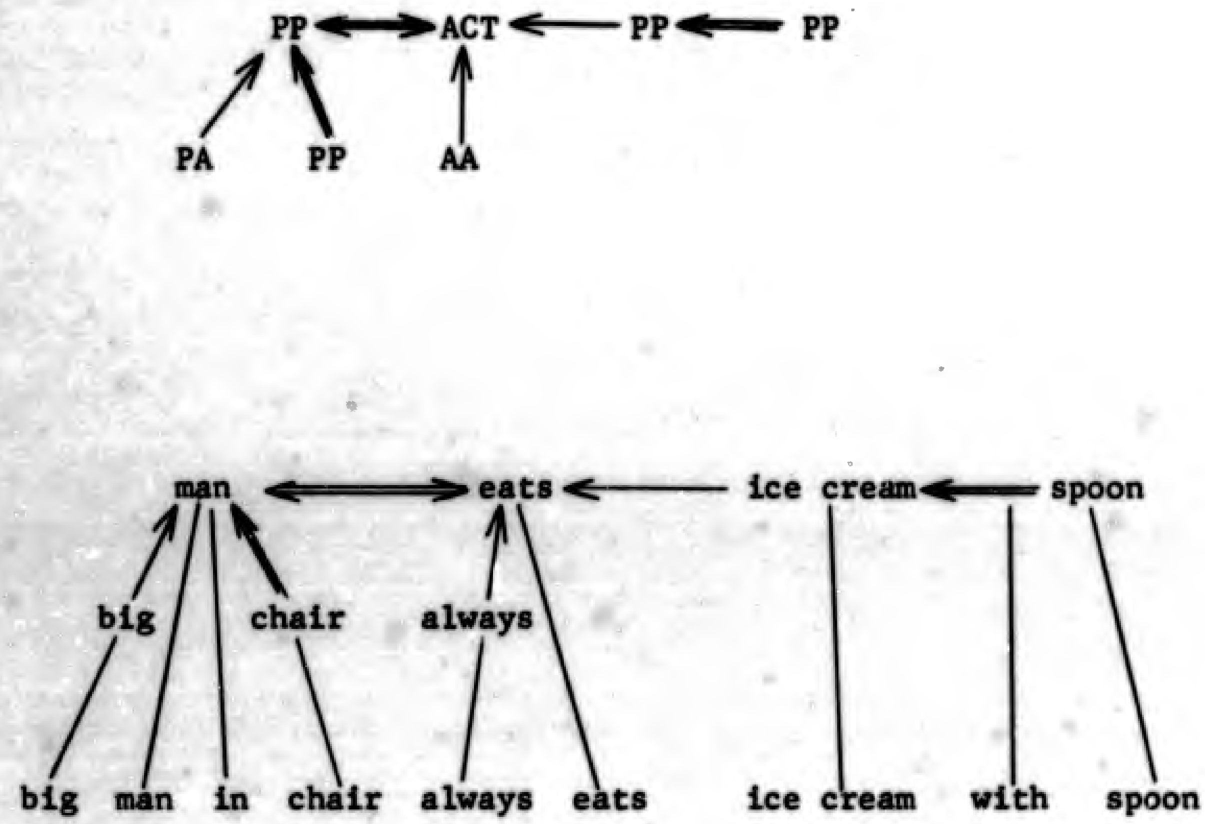
The categories used in the Conceptual level are PP, PA, ACT, AA, M, T, and LOC. These correspond in some sense to the syntactic categories Noun, Adj, Verb, Adv, Modal, Time, and Locative respectively, but they are not necessarily instances of those categories. Upon realization into language certain conceptual items may not appear, while others may appear more than once. Different types of arrows are used to symbolize the different types of connections possible between conceptual categories. Every concept in the completed network is dependent on every other concept. A 'blue ball' would be represented conceptually by 'ball ← blue' where 'ball' is a PP governing the dependent PA 'blue.' Every completed network has a two-way dependency link as its head. This link connects the mutually dependent concepts that serve as the head of the construction. Figure 1 shows a completed network and its realization. Sentential rules are applied to the realized output in order to insert articles, take care of verb agreement, etc.

The conceptual structure shown in Figure 1 can be used to generate an utterance with that meaning in any language. The relationship might not always be as simple, however (as indeed many English sentences are not simple one-to-one mappings from the conceptual structure).

Figure 2 shows all the conceptual generations and combinations possible. They may be used to represent any human conceptualization.

2.2 THE PLACE OF SEMANTICS

It is clear that if the schema outlined here is to be operable it must make use of some sort of a semantics that will limit the range of possible conceptualizations to the real world. When a conceptual connection is generated, it is a statement about the real world, such as 'red ball' (balls have color) or 'man runs' (men are capable of motion). The semantic system must limit the range of conceptual connections to statements about the real world, so that all conceptualizations will be meaningful. Thus, the semantics must be extra-linguistic, i.e., language free. The semantic rules must be tied up in the system's experience of the real world and thus be part of every conceptual rule in the system. Then, if the rule PP ← PA exists,



The big man in the chair always eats ice cream with a spoon.

Figure 1

The notation used here is read as stating that the item on the left may have the item on the right dependent upon it in the manner shown. The symbol (...) means that the previous item may have many occurrences dependent in the manner shown.

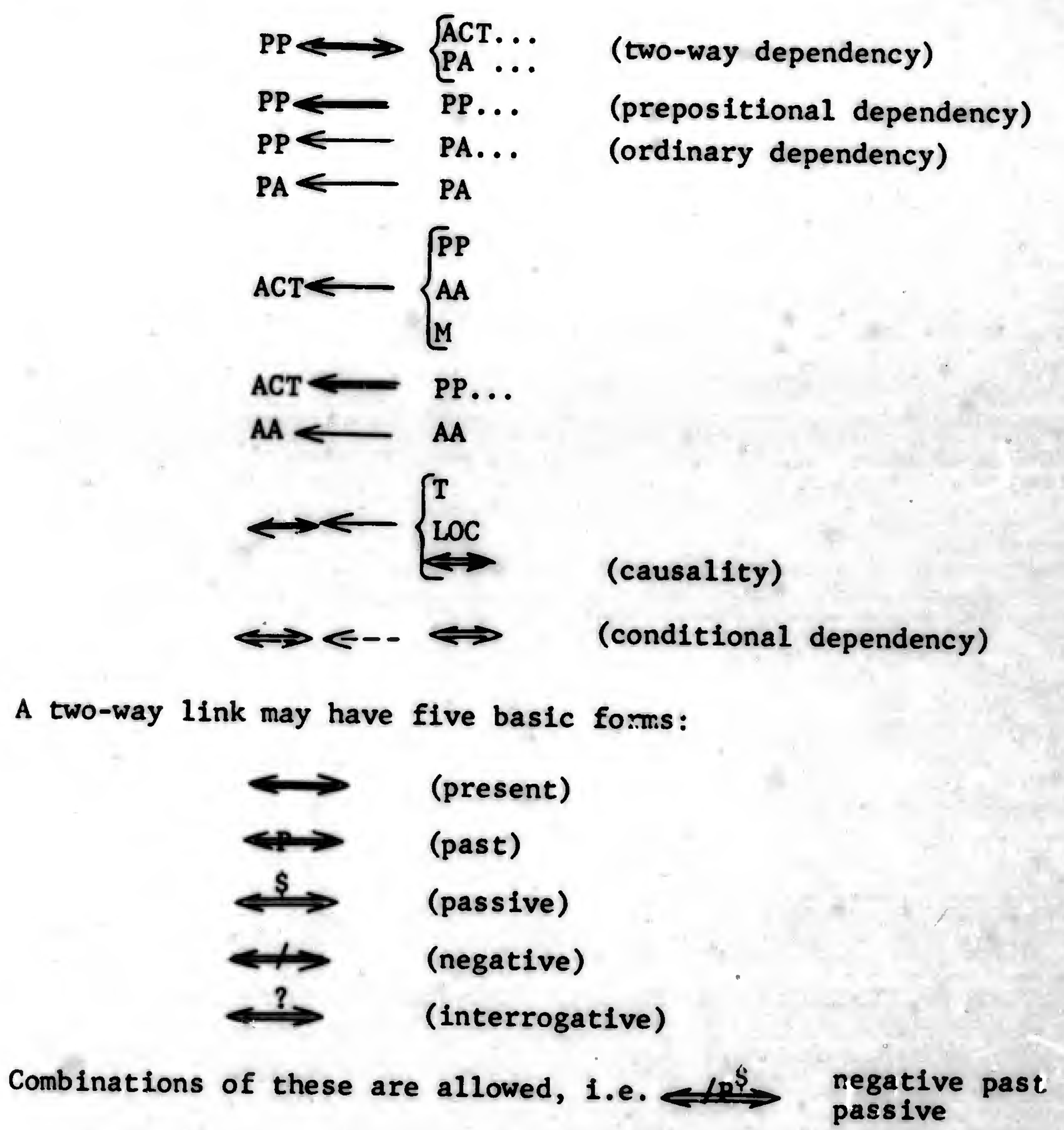


Figure 2

some limitation on which type of PA (see below) can connect to which type of PP must be present. This is necessary to prevent 'green ideas' from occurring.

There must be some starting point to this system so that there is no unnecessary duplication of information. That is, we would not want our 'green' file to have in it the information that what cannot be green (or have color) are 'ideas.' Rather, that which is modified should have the limitations for what can be modified built within it. The real world would then be divided up in terms of what PP's may be and do, what ACT's can act upon, etc. In addition, some memory system must be employed that keeps track of where it has been, so that when two complete (no more dependents are to be added) concepts affect each other (such as causality or other sequential activity) the plausibility of their affecting each other is maintained.

2.3 OUTLINE OF A SEMANTIC SYSTEM

A semantic system for use in the Conceptual Dependency framework must begin after the first conceptual rule is selected. (There is a choice of PP ⟺ ACT, and PP ⟺ PA.) At this point the particular PP is selected from a list of concepts that are PP's. Let us choose '$ball_1$.' (Since this is a concept and not an English word, it is necessarily unambiguous with reference to the individual system. The fact that this particular concept's English realization is the same as some other concepts' English realization is irrelevant for our purposes.) The system then checks to see which concepts may be dependent on 'ball' by delimiting the conceptual structures available for dependency on a 'PP.' These are (from Figure 2) ← PA, ⇐ PP and ⟺ ACT. (⟺ PA functions in exactly the same way as ← PA, the difference lying in the notion of predication involved with ⟺.) One of these is chosen. The particular conceptual item corresponding to the chosen conceptual category is selected utilizing the '$ball_1$' file:

PP - [$ball_1$]

PA	PP	ACT
has texture	located anywhere	rolls
has color	for anyone	bounces
has beauty	belongs to people	hits
has size	(generally children	push
has shape-round	or athletes)	knock down
.	.	(words of
.	.	inanimate
.	.	nonmechanical
		motion)
		.
		.
		.

The file shown above is not intended to be representative of the contents of a file that would actually be used by the system but rather of the structure of that file. It is clear that a more precise formulation of this type of information is needed to limit conceptual selections.

Let us say that a PA is to be chosen. It must come from the associated PA list of general modifiers. Whatever system is used to structure this list must refer to particulars such as 'red' and 'large.' If shape is referred to no choice would be available since the shape is part of the definition of 'ball.' Thus, the specified features of all the lists would constitute a definition of the concept. A PA such as 'intelligent' could not be chosen as a modifier of 'ball' since it is not on the PA list.

Since PA's have PA's as possible modifiers the associated list for the selected PA is checked. If 'has color' were the selection, a list would be checked for 'color' PA's.

PA - [color]

PA
has shades
has beauty
has intensity
.
.
.

One of these categories and an instance of that category such as 'shade:light' is selected.

Another modifier for the original PP may or may not be chosen. If one is chosen, a category and an instance of that category is selected as before. For example, 'belongs to people:of David' would be a possible choice.

When no more dependents are to be selected for the original PP, the form of the two-way dependency link is decided upon; e.g., past. Then an ACT is selected in the same manner as before. ACT's may have dependent AA's, and if a concept of motion were chosen as the ACT, the AA file might look as follows:

ACT - [motion]

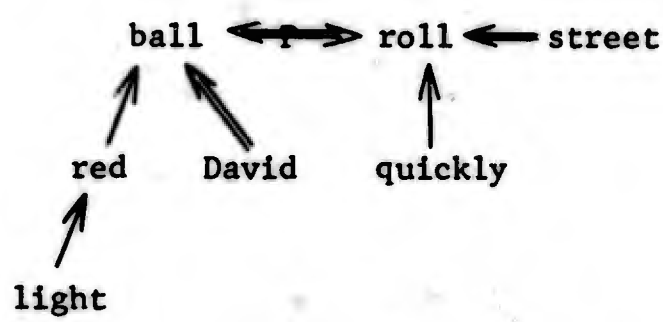
AA
speed
sound
direction
pattern

Another dependent for ACT may be chosen from a list of possible ⟸ PP's and ← PP's. Assuming that a concept of motion such as 'rolls' were chosen for ACT, ← PP would be eliminated as a possible dependent as part of the marking of 'rolls,' corresponding to a verb that may not take a direct object. The associated file for the 'rolls' aspect of the category 'motion' is as follows:

ACT - [rolls] ← PP

on a surface
around an item
off an item
down an incline
.
.
.

A possible conceptual structure utilizing the above rules might be:



The appropriate realization and sentential rules would yield:

David's light red ball rolled quickly down the street.

2.4 THE STRUCTURE OF THE SEMANTIC FILES

2.4.1 General Discussion

It is clear that if we are to use semantic files of the sort mentioned above, they will have to serve as well-structured definitions of the concepts available to the system. In addition, the external structure of individual conceptual files must be able to relate to a larger whole. We also have a right to expect that a good deal of the information in our non-linguistic experience files (see Deese [1] and Schank [8]) will be found in the external structure and the internal structure and contents of these files.

In our 'ball ← PA' file we find the entry 'has texture.' Some place in this file it will be necessary to insert the stored memory of what textures balls have been known to have. That is, we may not select randomly from a list of textures as we did when we needed a color. Clearly a ball may have any color, but probably not just any texture. This makes clear the problem of which textures to eliminate from the range of possibilities. Also it will be necessary to have the implied features marked. That is, a ball may be assumed to have a 'smooth' texture unless some other texture is specifically mentioned. That is not necessarily a defining characteristic however, as 'round' was. If a ball is not round in some sense, it ceases to be a ball, but this is not the case if it does not have a smooth texture. It would seem that a list of 'conceptual associations' would be needed as an adjunct to the list of defining attributes.

We have seen that each listing under a possible connecting conceptual category for a given concept references certain features. We expect that the number of these features is finite and that the features exhibit some grouping structure over a number of concepts. The marked features (such as shape: round) serve to define the concept in the strict sense of the word. But the entire listing actually serves as a complete definition. The inclusion of the possibility of a given concept having 'beauty' is an essential part of a definition in terms of the system. As Deese [1] notes

> "Categorical structures place the ultimate limits upon the interpretations human beings find it possible to make."

Clearly the system's interpretation of a piece of discourse will be based upon the categorical limitations present in the overall definition of the concepts used. As Thompson [9] has noted, the imposition of organization determines to as great an extent as experience itself the information that is yielded by the system. What we are doing then, by constructing semantic lists, is structuring our system's view of the outside world.

2.4.2 Specific Attributes

In order to get an idea of some of the problems involved in the actual functioning of the non-random system it is useful to take a look at an input question and the problems that it raises:

Are oranges edible?

We assume that something analogous to Quillian's Semantic Memory [6] would disambiguate the sentence by finding the most likely path between the definitions of the two concepts involved. However, this will not necessarily affirm the link, and it certainly will not produce more information. It is probable that a sufficient answer to this question would not be a "yes." Rather we expect to find some more information concerning who would and who would not eat oranges, etc. Therefore, we must relate the unambiguous conceptual apparatus

oranges $\overset{?}{\longleftrightarrow}$ edible

to the real world.

We begin with the dictionary definition of 'orange' and 'edible.' (taken from [10]):

$orange_1$ - a globose, reddish-yellow, bitter or sweet edible citrus fruit

edible - fit to be eaten as food

These in turn, correspond to machine definitions as discussed above that fill in the associated lists.

PP - $[orange]_1$

PA	PP	ACT
shape-round	location	nonmechanical motion
color-reddish -yellow	possession	
taste-{bitter, sweet}		
edibility-edible		

Here, 'citrus fruit' serves as a class name and will be used to refer us to the proper conceptual category to be used (see below). The PA category 'edibility' has been added as a possible modifier of 'orange.' It is clear that ACT's such as 'throw, sell, buy, eat ...' do not belong on the ACT list simply because 'oranges' do not do that sort of thing. That is, 'oranges' cannot have a transitive relationship with certain ACT's. (We are still dealing with concepts here. 'Oranges sell good,' for example, is not an instances of 'oranges ⟺ sell.') In order for 'oranges' to appear in a conceptualization adjacent to 'sell,' it would be necessary for 'oranges' to be the concept that was acted upon. Thus, if 'orange' were the topic of the sentence, corresponding to the main PP of the conceptualization, the sentence and its conceptualization would have to be in passive form. That is, the direction in which the concepts act upon each other would have to be reversed. Therefore, the defining semantic lists to be used would be determined by the object of the ACT, and the ACT's associated lists would serve to determine the range of possibilities for the main PP.

In the sentence "The orange was sold,"

orange $\overset{\$}{\Longleftrightarrow}$ sold

the 'sold' file would be checked for acceptable PP's. Here 'orange' might appear under the 'fruit' category. Thus, the passive marker in the conceptual apparatus determines which file to check for the necessary dependencies.

In the case that we were discussing above, 'edible' is found to modify 'orange' in a PA → PP relationship so the $\overset{?}{\Longleftrightarrow}$ becomes $\Longleftrightarrow$ and the conceptualization is realized. Consider the question:

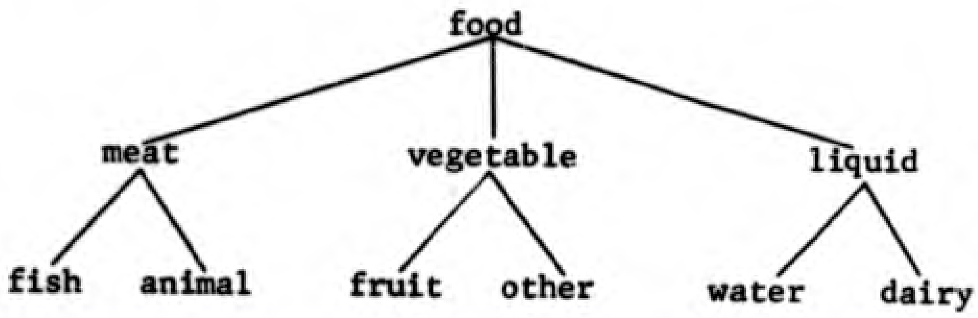
Can oranges be eaten?

The analytic system would be responsible for discovering that this was a passive construction, implying that some object is needed. The representation would be

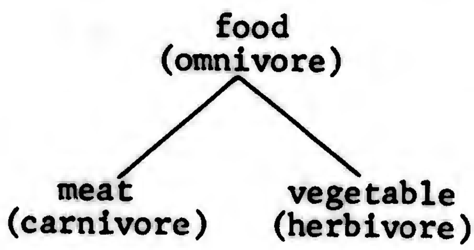
oranges $\overset{\$?}{\Longleftrightarrow}$ eat $\Leftarrow$?

Thus a complete answer should not only affirm the link but delimit the range of objective PP's as well. We cannot expect that the 'orange' list would be checked for edibility in its list of PA's. Were the ACT 'throw' instead of 'eat' we would not check for 'throwability' in a PA list. In other words we do not necessarily expect that 'eat' is a defining characteristic of 'orange' and we proceed normally, i.e., in the reverse order, since we are dealing with a passive.

The 'eat' file, as any other ACT file, specifies what may be acted upon in general and what may do the acting. We assume that 'food' is the general classification allowable and so we consult the hierarchical tree for 'food.' This tree serves to break the category 'food' into smaller categories with which we may work.



'Orange,' of course, would be listed under a subclassification of 'fruit.' It is clear that some sorts of limitations such as 'carnivore,' 'herbivore,' and 'omnivore' must be included in the 'food' tree in a node-naming capacity. 'Food' is only valid as 'food for somebody,' in terms of the eater. What is food for one animal is not necessarily food for another. Thus the tree must be labeled at each node with the class of PP that will serve as the replacement for ⟸ ? (by whom?).



The system would then be able to output the name of the node that was used to fill the first slot (in this case - 'orange') as the objective PP. All higher nodes would also be outputted; i.e., omnivore also serves as a correct answer to 'by whom?' The system could then pick from the list of omnivores and herbivores the appropriate PP, or if no contextual motivation were supplied, the applicable class names. An acceptable output could be either.

Oranges can be eaten by people.

or

Oranges can be eaten by omnivores and herbivores.

Clearly then, categorical arrangements, by hierarchies and other means, are intrinsic parts of the functioning of the semantic system. For the system to function effectively, the world will have to be divided up into categories in terms of their combinatory powers with other categories. With this in mind, we can proceed to a further discussion of random generation.

2.5 ASPECTS OF THE SEMANTIC SYSTEM

An example of the semantic operations with a specific PP was given in Section 2.3. It is clear that what were given there as possible semantic lists for 'ball' will probably serve as a defining framework for some larger category that includes 'ball,' e.g., 'physical object.'

After a PP is picked one of the possibly dependent categories is chosen. An item for the particular category of the PP is then selected from its associated semantic list. If the particular form of the selected item (e.g., color:red) is not specified, one is chosen. This conceptual item is then attached appropriately. (If it is specified, then it is redundant (a defining characteristic), and another is selected.) Then modifiers are selected for the most recent item, and a PA list is consulted for a PA. If an item is not picked, the dependencies are traced back to the original governor and another dependent item may or may not be selected (exception: the main PP must select a concept connected by a ⟷). This process continues until an entire conceptualization is completed.

What we are really in need of then are certain types of experience files for use in relating conceptual experience to linguistic experience. These files are delimited by the type of concept with which we are dealing. In addition, these files are used in conjunction with linguistically oriented hierarchical files (as shown in Section 2.4) in order to produce a concept. That is not to say that the hierarchical files are actually dealing with linguistic items, but rather, that linguistic divisions of the real world are used as conceptual headings.

As an example, after a PP is picked from a list of PP's its defining framework is used to limit the range of possibilities for its proposed dependents. The fact that the item is chosen from a list of concepts implies that the particular word sense of an ambiguous word is defined. Suppose that the concept 'ball' is chosen with the dictionary definition [10]:

> A round body of various size and materials
> hollow or solid, used in games.

This definition is coded in terms of its semantic category, i.e., physical object. (Note: This is all still language-free.) That is, the machine's definition has as its first marker, 'physical object.' This steers us to the 'physical object' file, which limits the range of dependents to 'ball' by considering the system's knowledge of the real world. (This has nothing to do with linguistic experience.) Since any concept listed under 'physical object' is a 'PP' the range of its possible dependents is limited by the conceptual apparatus. Any 'PP' file has three sections denoting the three categories that may be dependent on a PP.

physical object	← PA	⟵ PP	⟷ ACT
	1. shape	1. location	1. nonmechanical motion
	2. color	2. possession	
	3. texture		
	4. material		
	5. consistency		
	6. beauty		
	7. size		

This file must correspond to a similar file pertaining to the specific object and thus defining it.

ball	← PA	⟵ PP	⟷ ACT
	1. round	1. any	1. any
	2. any	2. animate/human; domestic animal;	
	3. smooth		
	4. any		
	5. solid or hollow		
	6. any		
	7. any		

This file serves to point out which aspects of the physical object 'ball' serve to define its place in the real world and which are variable. When a dependent category is chosen, one of its items is selected. A defining characteristic may not be selected (e.g., round ball) since it is already implied. If a PA is chosen let us say that (7):size is selected. Since ball:(7) says "any" we look up the size file, where we find:

size
1. large
2. small
3. medium

These are concepts. When one is chosen it is placed in the conceptual apparatus as dependent on ball. Upon realization into language it does not necessarily keep its present form. 'Large' for example, might be realized as 'large' or 'big' or 'huge,' etc. The actual selection is done within the confines of the linguistic experience file that is consulted upon realization. Here all connections are checked to see if they conform to linguistic experience. For example,if 'medium' had been chosen for 'ball' the linguistic experience file would not permit the construction 'medium ball' to be its English realization. This would be done by a check of the file to see if this construction had ever occurred before. Upon finding that it hadn't, the construction would be rearranged to include the category from which it came (i.e., size) in the realization, thus producing the acceptable 'ball of medium size.' Admittedly this type of file would be enormous, but it is a reasonable assumption that humans utilize a file of this sort to eliminate 'queer' forms. This is what makes the difference between ordinary speech and the use of 'foreign' constructions. In addition, it seems that it is in the linguistic experience file that word associations enter the picture. It would be necessary to avoid linguistic realizates that express the wrong connotation.

So we see that we must use three separate files in addition to a linguistic experience file in order to obtain a conceptually correct dependent for a given concept. The first file delimits the defining framework for the category of the governing concept for all possible dependents. The second file fills in the defining characteristics for the particular form of the conceptual category to be used. The third file contains the actual concepts that may be chosen in accordance with the two previous files. This is the basis of our semantic system.

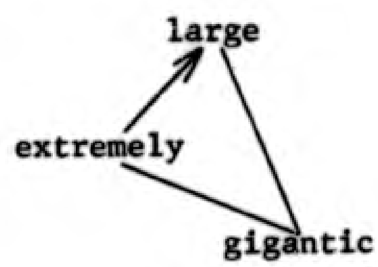
After the link between the PA and PP is made, dependents may be selected for the PA. The categorical feature to be used has already been employed, namely - 'size of physical object.' The file uncovers

PA - [size of physical object]

PA

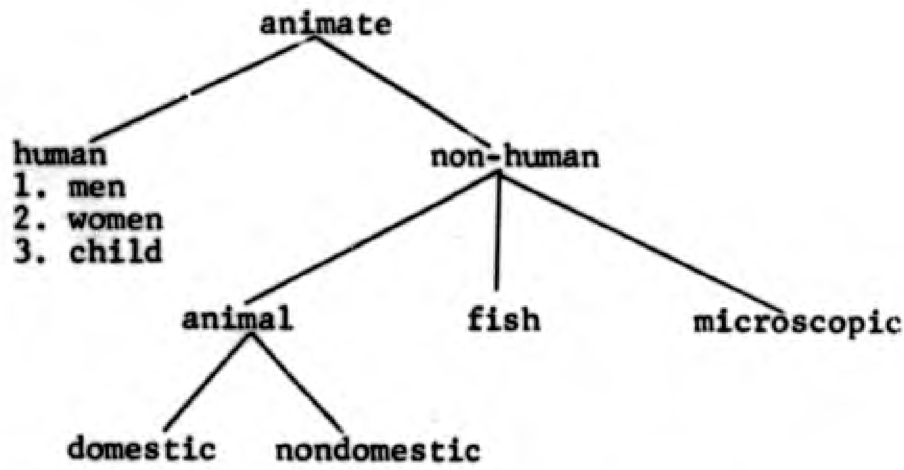
1. degrees

and a degree is selected from a list of degrees. If 'extremely' is chosen, for example, the linguistic realization could be:

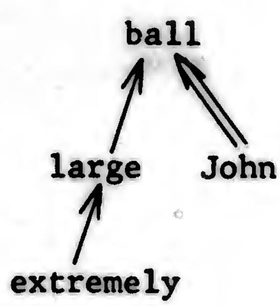


That is, lexical items may be chosen that express more than one concept. This is a basic fact in semantics as expressed in Lamb's example [4] (taken originally from Hjelmslev [3]) of the lexeme 'mare' being made up of the two sememes 'female' and 'horse.' The linguistic experience file would note that while 'extremely large' may be realized as 'gigantic', 'female horse' may be realized only as 'mare.' In effect then, what we are doing is establishing a pattern of speech for our system.

After all possible dependents are chosen for a governor, the system traces back to the last previous governor (in this case 'ball'). We may now select additional dependents. Let us say that a ⇐ PP is chosen to be dependent on ball. Again we check the associated files (shown above) and select an item (e.g., (2)). In (2) our choice of possession is neither a defining characteristic nor a free choice as before. Here the selection is limited to a certain hierarchical file that we have, called 'animate.'



In addition the selection of items along the file is delimited to two specific subtrees, that labeled human, and that labeled domestic animal. When one item is chosen, it too may have dependents, as before. However, as any of these items may have names, selection may or may not be over. If 'man' is chosen, it may be used or a particular form of it may be used, say 'John.' The structure will then be as follows:



If no more dependents are to be chosen, a form of the two-way dependency link is selected (e.g., ↔P, past tense). The choice of the item to the right of the two-way link is made, utilizing the same file for 'ball' as before. In this case if ↔ACT is selected we know that we must choose from a list of concepts of nonmechanical motion.

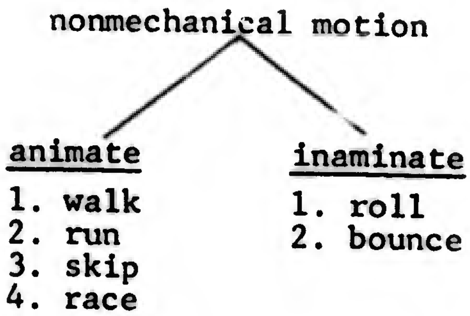
[nonmechanical motion]

1. roll
2. fall
3. hit
4. bounce

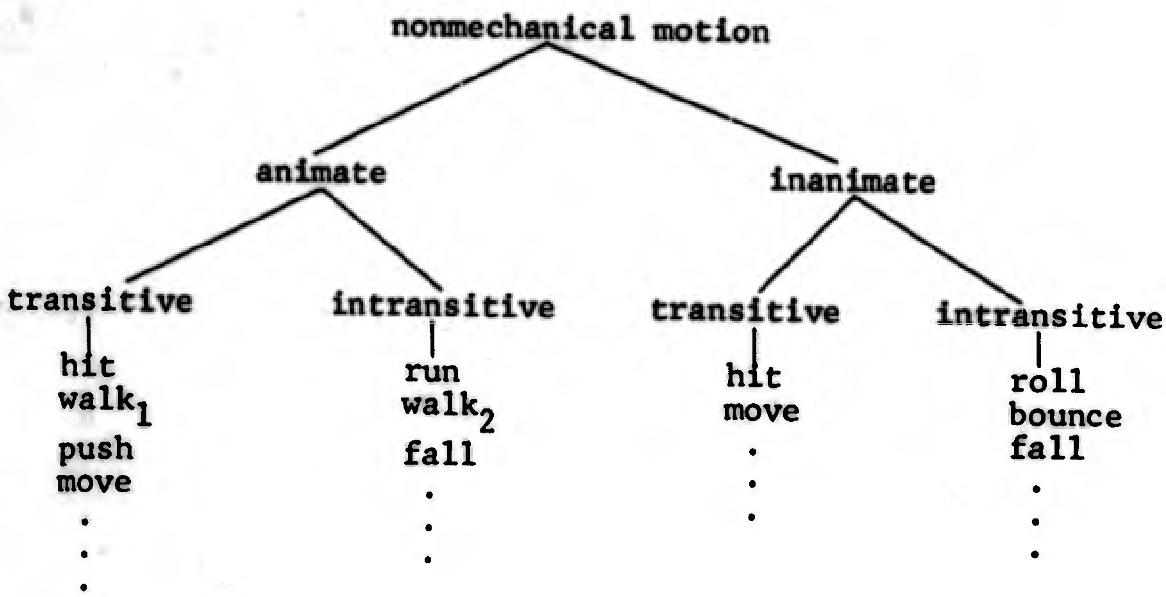
Let us choose 'fall.' The problem now is how to choose the dependents of 'fall,' using what we know about the structure of the left hand side of the two-way link, and the semantic restrictions of 'fall.'

To do this it is necessary to introduce some sort of notion of transitivity or intransitivity of ACT in order to specify which ACT's may have a ← PP and which a ⇐ PP dependent upon them. It is clear, for example, that if 'fall' is to have a dependent PP it must be connected by a prepositional link, i.e., it must fall <u>on</u> something. In the first place, the old notion, that intransitivity for verbs depends on whether they can stand without an object, must be discarded for use in the conceptual level. Any two concepts that may be connected by a two-way dependency link may stand without dependents. It is a function of the particular language to decree that certain verbs must take objects. "I love" in English may not be an acceptable sentence but it is a complete conceptualization. The fact that English sentential rules may place an "am in" between "I" and "love," or force an object after "love" is of no concern in the original conceptualization. Therefore some other method of defining the differentiation of ACT's must be found.

One thing is clear. The choice of the original PP strictly delimits the range of possible objects for the ACT and must therefore do so in our semantic system. Verbs of non-mechanical motion, with which we are working, might be divided into an animate and inanimate hierarchy, i.e., ACT



Clearly there are concepts that are neither animate nor inanimate, such as hit, fall, knock down, etc. These also divide into a transitive and intransitive class to the extent that as transitivity is defined as ability to take $\leftarrow$ PP and intransitivity as ability to take $\Leftarrow$ PP. (It might be noted that although some English verbs may be either transitive or intransitive, concepts may be only one or the other. This is because we are working with unambiguous word senses and not with words themselves.) The structure of the file would be as follows:



All ACT files then, must be divided according to their ability to take ⇐ PP or ← PP. The presence of the conceptual category in question is determined by the defining characteristic of the ACT used. It is clear that the category 'verbs of non-mechanical motion' that was part of the 'physical object' listing for ACT must be changed to 'verbs of inanimate nonmechanical motion' as a result of the previous discussion.

Then, since 'fall' came from the intransitive list, its file will be structured as that for all inanimate non-mechanical motion:

		⇐ AA	⇐ M	⇐ PP
⌜fall⌝ -	[inanimate nonmechanical motion]	sound	may	from
		speed	will	to
		force	The M file	on
		.	is always	off
		.	the same and	.
		.	may always	.
			be used.	.
			.	
			.	
			.	

The AA file is utilized in the same way as the PA file above. The M file is also used in the same way, although it is likely that there are no restrictions on it, but rather just a listing of the possibilities for it. It is also likely that these possibilities are unrestricted with respect to particular ACT's.

A problem crops up when ⇐ PP is considered as a dependent. The structure of the file must specify all possible prepositions that may be used with verbs of nonmechanical motion.

⟸ PP

1. off-ness
2. to-ness
3. on-ness } the same (on the table, not in the table)
4. in-ness } ("in the soup" not "in Brooklyn" which is covered by LOC)
5. along-ness
6. by-ness (as in "rolled by the boy," not passive and not causal)
7. from-ness (a place, not the same as 'off')

Again, this file is only of conceptual prepositions. 'With' for example in "fell with a thud" would come from the same place as "fell with a noise," that is, from the AA - "noisely." The fact that "thudily" is not a word in English would be discovered by the linguistic experience file, which would restructure the sentence (not the concept) accordingly. Also 'by' is not included in the sense "fell by being pushed" since this would be taken care of by the Conceptual Dependency abstraction for causality. It is only a quirk of English that places a preposition in that slot.

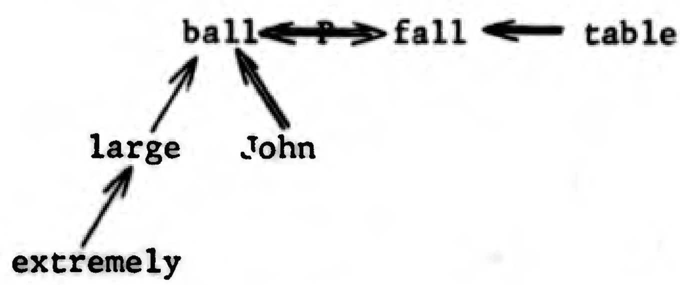
As before, the file used must coordinate with a specific file for the actual concept.

fall ⟸ PP

1. elevated physical object
2. stationary physical object
3. } any
4. } any
5. not allowed
6. not allowed
7. not allowed

Here, the hierarchical classification of pnysical object must be utilized. It is evident that two of the defining characteristics must be "normally elevated from ground" and "normally stationary." The realized preposition for (3) and (4) is determined by the concept chosen. Fall cannot take (5), (6), or (7).

Let us choose (1) and "table" for the particular concept. Clearly "table" will have to be derived from the concept of "table top" by the rules of English realization, but we may use "table" for the time being. We now have as our concept:



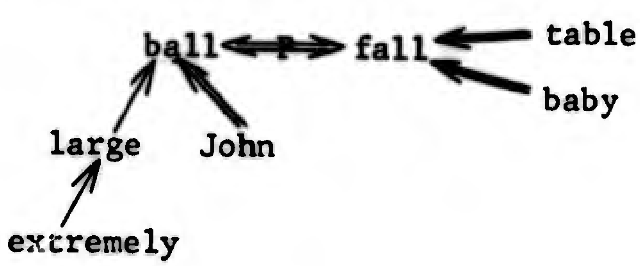
Conceptually we may choose to generate another ⇐ PP string. Let us say that we wished to generate the string "on the baby." According to the conceptual rules the ⇐ PP may be attached to an ACT or another PP. It is clear that "The ball fell from the table on the baby" is a perfectly well-formed semantically correct sentence. It is also ambiguous, even though it would be hard to take it so (tables are not usually on babies). Clearly the correct form of the conceptualization is permitted by precisely the same reasoning as was used above with "table." In this case the "any" restriction of (3,4) is used from the specifications of "fall."

However, it is worthwhile to try to generate the string "← table ⟸ baby" for explanatory purposes. (This also yields some direction to a possible conceptual parser.) In order to generate something dependent on table we must utilize the physical object file as we did with PP 'ball.' This will tell us just what a table is usually on.

PP - table ⟸ PP

1. off-ness: not allowed
2. to-ness: not allowed
3.
4. in/on-ness: a physical object-inanimate
5. along-ness: not allowed
6. by-ness: not allowed
7. from-ness: any place

The defining characteristic for (3) eliminates an animate object from consideration, thus making it impossible to generate ⟸ table ⟸ baby. The complete conceptualization generated by the random semantic generator is:



This may be realized as:

John's extremely large ball fell off the table on the baby.

3. THOUGHTS ON NON-RANDOM GENERATION

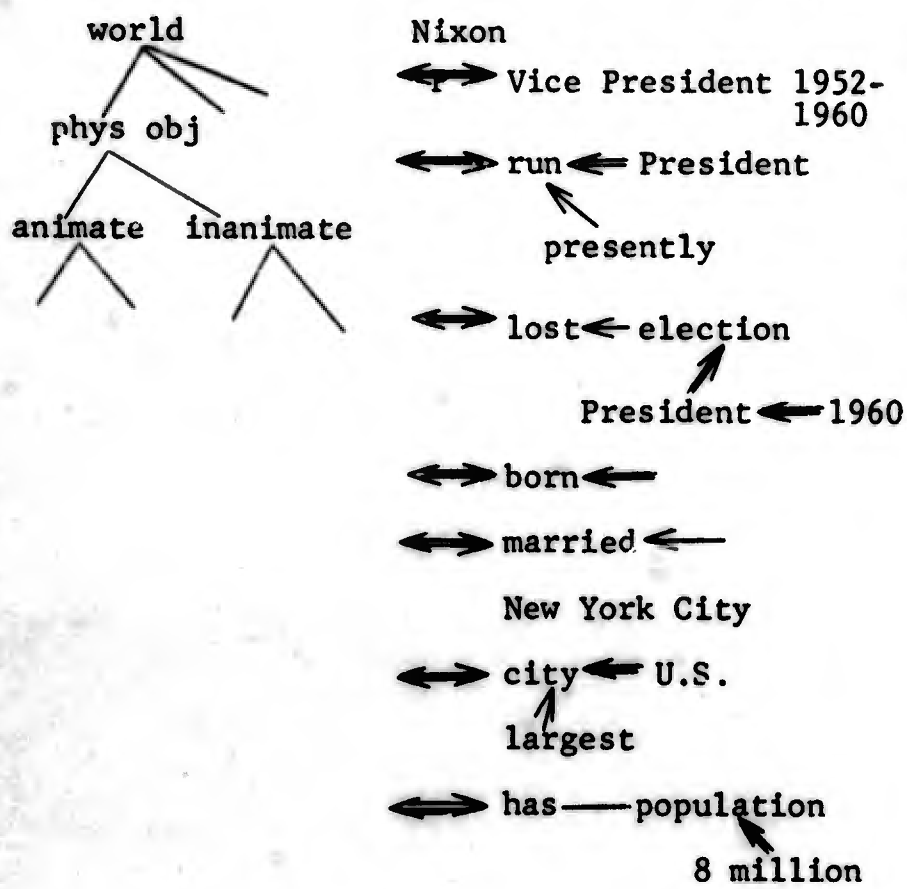
Generation, if not random, must respond to some input. If we assume a completed Conceptual Dependency analysis of the input, with assigned interpretations to each concept, the problem is that of looking up data, storing the new information from the input, and checking to see if the dependencies proposed as answers may exist in the real world. The completed conceptualization is then mapped into language.

Random generation techniques are utilized in non-random generation, however. You would not expect a complete file for 'ball' and 'orange' as one would find for a proper name. Rather, information about concepts such as these would be stored in marked versions of the 'physical object' file, the choice of possible dependents being delimited by that file. Only specific entities such as 'Richard Nixon or ' New York City' have information files. Thus, the dictionary component is composed of semantic techniques and lists, as discussed in the random system. The 'proper name' file must be a differently structured aspect of the non-linguistic experience file. It in fact comprises a significant part of our non-random generation scheme since we can expect that a question-answer system would be largely concerned with information from this file.

We can expect that the non-linguistic experience file will have three principal components as shown:

NON-LINGUISTIC EXPERIENCE FILE

dictionary | nominal information | static information



characteristics of mice, flowers, etc. that are not intrinsic to their defining characteristics and combinatory powers but are known by the system.

Upon obtaining required concepts and the relations between them from the input, the non-random generation begins by checking PP $\Longleftrightarrow$ ACT cluster. The form of the two-way link is then checked.

For example, in 'Did Nixon run for President in 1964?' the experience file (analytic) makes 'President' into 'President of U.S.' (because of common contextual association as in Semantic Memory [6]); then we have

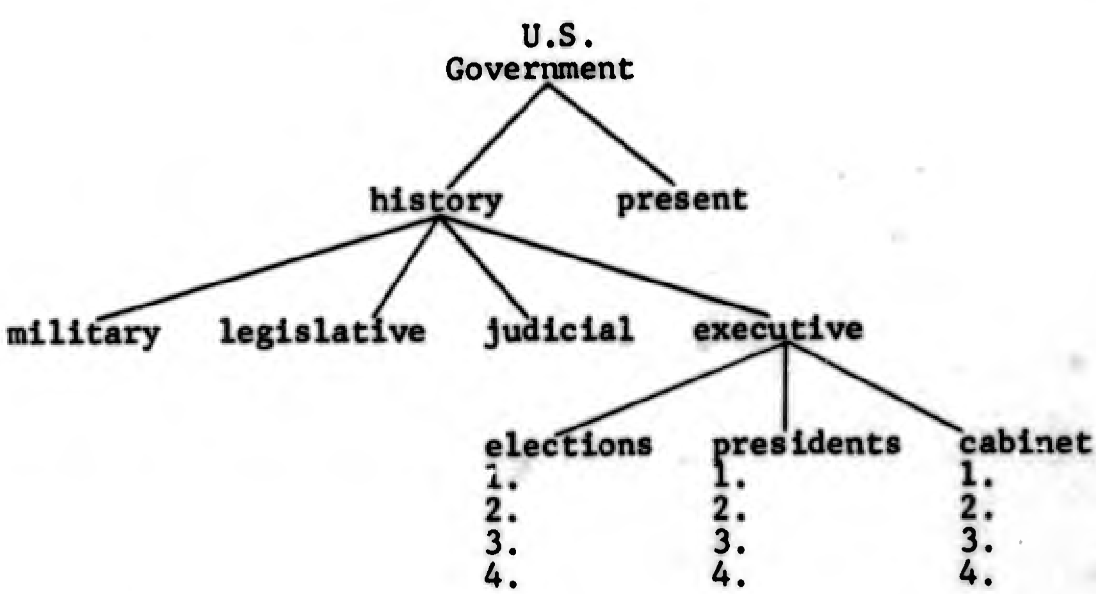
1964
?
Nixon ⟷ ran ⟸ President ⟵ U.S.

A check of Nixon of the file uncovers

1960
Nixon ⟷ ran ⟵ President ⟵ U.S.

1968
and Nixon ⟷ run ⟵ President ⟵ U.S.

There is always the possibility that the Nixon file is incomplete and a 'no' cannot be outputted at this point. It is therefore necessary to check another file. The 'U.S.' file might be a start. It is clear that ran ⟸ President must somehow imply 'election,' and that the 'U.S.' file must be subdivided to include 'Presidents' and 'elections for.' This file then, must be hierarchically ordered in the following way:



The search down the tree for the 'election' node would be aided by the definition of 'election' in eliminating fruitless branches.

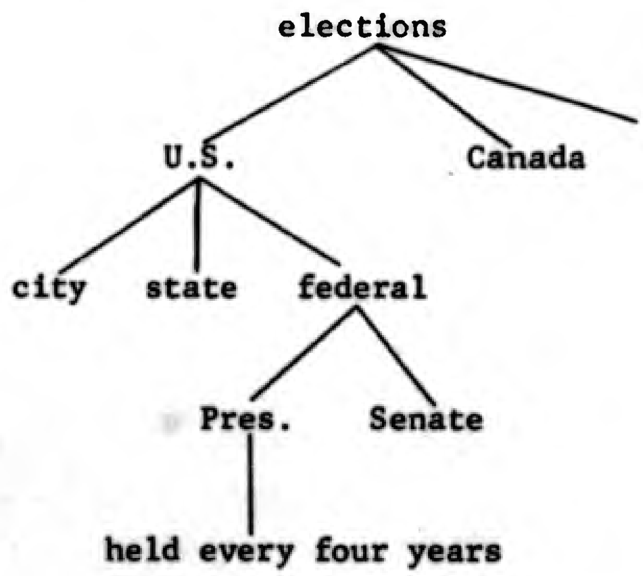
Though our content analysis is conceptual and therefore would not be different with synonymous words, it must still be able to get into the appropriate file by specifying 'election' or 'opponent of President' or something like that. It is assumed that a Quillian type analysis [6] will have found the appropriate sense of 'run' by finding the applicable connection between 'run' and 'President.' More importantly, the dictionary entry would need to include 'as in an election' at some place in the intersection. It is then necessary to look at the election entry to note that an election is between candidates, that one is the winner and is elected, and that the others are his defeated opponents. The actual information needed would depend on the exact structure of the 'U.S.' hierarchical file above. In other words, if there is substructure called 'elections' we may use it, or if the 'President' list contained therein has as part of its information the candidates that were defeated by the particular President then we may use it. It is possible that the particular entries in the President list might better serve as references to other information stores headed by that President. This would not be useful here, however, as 'Nixon' would not be contained in that file in any other way than as an object of the file. It would be most inefficient to search each file looking for the appropriate object, and it is doubtful that humans actually do this. If the question were 'Who ran against Nixon?' then it is possible that the object would serve as the starting point, since the question might just as well have been stated as 'Nixon ran against who?' However, in a question such as 'Who do you know that lives in New York?' it is clear that the 'N.Y.' file could not be traced for its inhabitants, but rather the list of acquaintances would be checked for objects. This would appear to be the way humans perform, and in any event would appear to be more efficient. The procedure might be different if the place named were Taconic, Conn.; however, the accessible information for the machine is not likely to contain the populace lists of either city.

It is possible that the choice of which search procedure to follow is aided by the notion of transitivity versus intransitivity discussed in Section 2.5. It appears that in the cases in which dependencies may be reversed, i.e., ← rather than $\longleftarrow$, objects can become subjects and the less efficient search methods may be eliminated.

Eventually, we may assume, the information that the only two candidates in 1964 were Johnson and Goldwater is found. Since Nixon was not found, the $\overset{?}{\Longleftrightarrow}$ becomes $\not\Longleftrightarrow$ and the sentence is outputted (with or without the correct information).

Another problem presented by this question is the matter of the date allowing us the possibility of shortening our search. If it is 1864 or 1963, the task is different. Again the main link 'Nixon $\overset{1864}{\xleftrightarrow{P}}$ ran' is checked to see if it is compatible with the system. It is clear that all dates must be checked in such a manner that the information 'Nixon was not alive in 1864' would be outputted.

The more interesting problem is 1963. It is necessary for the system to know that Presidential elections were not held in 1963, or any other year not divisible by four. This implies the sentence must be used in a form that includes elections, as before, and that information about each term be fully investigated before use. It seems that $\Longleftarrow$ election should be substituted in between ran and President. That is, in the Semantic Memory [6] search that uncovers the sense of 'run' to be used, those significant concepts crossed must necessarily become a part of the implied information that is readily available to the system. It is the election file that would contain under U.S., Presidential, the information about when they are held:



So we see that much of our information in answering questions must come from the static-information file. This file, coupled with the dictionary component, should comprise a workable non-random generation system that commands a workable knowledge of the real world. The nominal-file is extrinsic to the system operation since it depends on the purposes of the system.

4. CONCLUSION

Eric Lenneberg has stated that:

> "Words are not the labels of concept completed earlier and stored away; they are the labels of a categorization process or family of such processes. The task of cognitive organization never comes to an end. Referents of words can change easily, meanings can be extended, and categories are always open. Words tag the processes by which the species deals cognitively with its environment. If we hear a given word used in connection with a given object or phenomenon, we are able to intuit the general usage of that word. It does not have to be paired with 200 similar objects before we can make predictions. Natural languages always have universally understandable types of semantics, but may easily have different extensions of meaning." [5:333]

This paper has been an attempt to build the universally understandable semantics referred to by Lenneberg by attacking the problem of cognitive organization. The different extensions of meaning across languages referred to above also holds between individual speakers. Since we are attempting to have our machine become an individual speaker we can expect that it will develop its own slightly different conceptual structure based on its own experience. The attempt here has been to start that structure on its way. With the conceptual semantics outlined here, the effective classification schema referred to more fully developed, and a linguistic experience file connected to the previous realization schema [7], we can expect that our machine will begin to speak coherently.

REFERENCES

1. Deese, J., "Some Contributions of Psycholinguistic Studies to Content Analysis," presented to the National Conference on Content Analysis, Philadelphia, Pa., November 16-18, 1967.

2. Hirst, N., and Hirst C., Towards the Logical and Metaphysical Foundations of Organismic Theory: A Prerequisite to Artificial Intelligence, TRACOR 68-252-U, Tracor, Inc., Austin, Texas, February 1968.

3. Hjelmslev, L., Prolegomena to a Theory of Language, University of Wisconsin Press, Madison, Wisconsin, 1961.

4. Lamb, S., Lexicology and Semantics, Voice of America Forum Series (mimeographed), 1966.

5. Lenneberg, E., Biological Foundations of Language, John Wiley, New York, 1966.

6. Quillian, R., Semantic Memory, Bolt, Beranek and Newman, Inc., Cambridge, Massachusetts, October 1966.

7. Schank, R., "Conceptual Dependency as a Framework for Linguistic Analysis," Linguistics (in press).

8. Schank, R., The Use of Conceptual Relations in Content Analysis and Data Base Storage, TRACOR 68-347-U, Tracor, Inc., Austin, Texas, January 1968.

9. Thompson, F., "The Organization is the Information" presented to 30th Annual Convention, American Documentation Institute, October 22-27, 1967.

10. The Random House Dictionary of the English Language, Random House, New York, 1966.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) TRACOR, Inc. 6500 Tracor Lane Austin, Texas	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

OUTLINE OF A CONCEPTUAL SEMANTICS FOR GENERATION OF COHERENT DISCOURSE

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Roger C. Schank

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1968	37	10

8a. CONTRACT OR GRANT NO. Independent Research b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) TRACOR 68-462-U 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

This paper develops a method for generating coherent sentences. A conceptual semantics is presented, that when coupled with a conceptual dependency abstraction of meaning, allows concepts to be linked in a manner consonant with the system's knowledge of the world. The paper is part of a series of papers concerned with the problem of language synthesis for artificially intelligent systems.

DD FORM 1 NOV 65 1473

Unclassified
Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computational linguistics						
Semantics						
Conceptual Dependency						
Conceptual Semantics						
Sentence Generation						
Question-Answering						
Language Synthesis						
Artificial Intelligence						

Security Classification