# A SYSTEM OF SEVEN COHERENCE RELATIONS FOR HIERARCHICALLY ORGANIZING EVENT CONCEPTS IN TEXT

RICHARD ETHAN ALTERMAN

TR-209

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December, 1982

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Professor Robert F. Simmons, for his invaluable guidance. I learned more about natural language processing in my weekly chats with Dr. Simmons than I did from any other single endeavor. His enthusiasm for research has been an inspiration.

I would also like to express my gratitude to the other members of my committee for their thoughtful comments and encouragement: Professors Elaine Rich, Norman Martin, Woodrow Bledsoe, and Carlotta Smith.

My sincerest thanks to the computer science department and the CMP project for the much needed financial support.

I would also like to thank past and present members of the Texas natural language crew for the good discussions and good cheer: Alfred Correira, Mark B. Fishman, Michael Hess, Michael K. Smith, Howard Smith, and Jonathan Slocum.

For their moral support, I am indebted to all of my friends:

Vincent, Mike, John, Charles, Neil, Bill, Larry, Michael, James, Peggy, Becky, Jay, Tom, Ben, Linda, and George.

My parents deserve thanks for their encouragement and the survival skills they taught me.

Finally, a special word of appreciation to Clarie for her loving support, good friendship, and the promise of more good times together.

The task of formatting this dissertation was greatly eased by use of automated document processing. The Scribe document formatter was conceived of and created by Brian Reid. The current version has been maintained and enhanced by Unilogic, Ltd. The Scribe format definitions for proper dissertation format for The University of Texas at Austin were developed by Richard Cohen.

Richard Ethan Alterman

The University of Texas at Austin
December, 1982

## ABSTRACT

A theory of event concept coherence is developed. It is shown that the pieces of event description which appear in a body of text can be gathered together and hierarchically organized using a dictionary of event/state concepts. This theory has been implemented in a computer system, NEXUS. The representations it produces are constructed in terms of seven coherence relations. The dictionary it uses was compiled from an analysis of ten folktales.

The seven coherence relations used are class/subclass, sequence/subsequence, coordinate, antecedent, precedent, consequent and sequel. Class/subclass is a taxonomic relation. Sequence/subsequence and coordinate are two kinds of whole/part relations. The other four relations are temporal; antecedent and precedent concepts come before an event, consequent and sequel concepts come after an event.

Associations between concepts in the dictionary are also organized in terms of the seven coherence relations. Associated with the concepts in the dictionary are default values for its case related arguments. Relationships between concepts in the dictionary

are refined by attaching to each relationship in the dictionary a set of constraints on matching case arguments. NEXUS uses the default values and the constraints to control the representation building process.

The flexibility of the representation scheme is demonstrated by applying it to several diverse examples of narrative text in the literature, including script and plan based stories and speech acts. The feasibility of NEXUS is shown by applying it to eight samples of text and discussing in detail the results. The utility of the hierarchical representation produced by NEXUS is validated by experiments in question answering and summarizing.

TABLE OF CONTENTS

Acknowledgements . . . . . . . . . . . . . . . . . . . . . . . . v

Abstract . . . . . . . . . . . . . . . . . . . . . . . . . . . . vii

Table of Contents . . . . . . . . . . . . . . . . . . . . . . . ix

Chapter 1. Introduction . . . . . . . . . . . . . . . . . . . . 1

1.1. Overview . . . . . . . . . . . . . . . . . . . . . . . . 1
1.2. An Empirical Study of Coherence: A Chronology . . . . . 6
1.2.1. PHASE1 - A Prototype . . . . . . . . . . . . . . . . . 6
1.2.2. Analysis of Text . . . . . . . . . . . . . . . . . . . 7
1.2.3. Compiling the Dictionary of Event/State Concepts . 11
1.2.4. TRACE - A System to Build Representations . . . . . 14
1.2.5. Question Answering and Summarizing . . . . . . . . . 17
1.3. Outline of Chapters . . . . . . . . . . . . . . . . . . . 18

Chapter 2. Representing Text with Relations . . . . . . . . . . 21

2.1. Method . . . . . . . . . . . . . . . . . . . . . . . . . . 21
2.2. Relations . . . . . . . . . . . . . . . . . . . . . . . . 25
2.2.1. Class/Subclass . . . . . . . . . . . . . . . . . . . . 27
2.2.2. Sequence/Subsequence . . . . . . . . . . . . . . . . . 28
2.2.3. Coordination . . . . . . . . . . . . . . . . . . . . . 29
2.2.4. Antecedent . . . . . . . . . . . . . . . . . . . . . . 31
2.2.5. Consequent . . . . . . . . . . . . . . . . . . . . . . 32
2.2.6. Precedent . . . . . . . . . . . . . . . . . . . . . . . 33
2.2.7. Sequel . . . . . . . . . . . . . . . . . . . . . . . . 34
2.3. Example . . . . . . . . . . . . . . . . . . . . . . . . . 35
2.3.1. Analysis . . . . . . . . . . . . . . . . . . . . . . . 35
2.3.1.1. "The pig trotted towards the stream, carrying a bundle of clothes." . . . . . . . . . 35
2.3.1.2. "The animal expertly soaked and scoured the laundry." . . . . . . . . . . . . . . . . . . 36
2.3.1.3. "The pig hung the clothes in the sun to

dry." . . . . . . . . . . . . . . . . . . . . . . . 36
2.3.1.4. "The pig gathered her laundry and trotted home." . . . . . . . . . . . . . . . . . . . . . . . 37
2.4. Comparison with Other Approaches . . . . . . . . . . . 40
2.4.1. The Restaurant Script . . . . . . . . . . . . . . . . . 40
2.4.2. Robbing a Liquor Store . . . . . . . . . . . . . . . . 43
2.4.3. The Margie Story . . . . . . . . . . . . . . . . . . . 45
2.4.4. A Black and Yellow V-2 Rocket . . . . . . . . . . . . 47
2.4.5. Speech Acts . . . . . . . . . . . . . . . . . . . . . . 49
2.4.6. Rhetorical Coherence . . . . . . . . . . . . . . . . . 50
2.4.7. Perspective Summarized . . . . . . . . . . . . . . . . 53

Chapter 3. The Construction and Use of the Dictionary . . . . . 56

3.1. The Dictionary . . . . . . . . . . . . . . . . . . . . . 56
3.1.1. The Dictionary as a Semantic Network . . . . . . . . 56
3.1.2. Notation & Terminology . . . . . . . . . . . . . . . . 63
3.1.3. Converting to 4-tuples & Inheritance . . . . . . . . 65
3.2. Overview of TRACE . . . . . . . . . . . . . . . . . . . 68
3.2.1. Find-Path . . . . . . . . . . . . . . . . . . . . . . . 71
3.2.2. TRY-PATH . . . . . . . . . . . . . . . . . . . . . . . 75
3.2.3. Some Rejected Heuristics . . . . . . . . . . . . . . . 79
3.2.3.1. Match-Foci . . . . . . . . . . . . . . . . . . . . . 79
3.2.3.2. Temporal Orderings . . . . . . . . . . . . . . . . . 81
3.3. Limitations . . . . . . . . . . . . . . . . . . . . . . . 83

Chapter 4. Results . . . . . . . . . . . . . . . . . . . . . . 87

4.1. The Tale of the PIG . . . . . . . . . . . . . . . . . . . 93
4.2. The Peasant and the Waterman . . . . . . . . . . . . . . 100
4.3. The Wishing Ring . . . . . . . . . . . . . . . . . . . . 105
4.4. The Restaurant Story . . . . . . . . . . . . . . . . . . 108
4.5. Robbing a Liquor Store . . . . . . . . . . . . . . . . . 111
4.6. The Margie Story . . . . . . . . . . . . . . . . . . . . 115
4.7. The Archer, William Tell . . . . . . . . . . . . . . . . 117
4.8. The Czar's General and the Clever Peasant . . . . . . . 117

Chapter 5. Question Answering and Summarizing . . . . . . . . . 121

5.1. Quest . . . . . . . . . . . . . . . . . . . . . . . . . . 121
5.1.1. Enhancement Questions . . . . . . . . . . . . . . . . 124
5.1.2. Goal Questions . . . . . . . . . . . . . . . . . . . . 127
5.1.3. What-Caused Questions . . . . . . . . . . . . . . . . 128

5.1.4. Motive Questions . . . . . . . . . . . . . . . . . . 131
5.1.5. What-happened Questions . . . . . . . . . . . . . . . 134
5.1.6. Enablement Questions . . . . . . . . . . . . . . . . 136
5.1.7. Perspective . . . . . . . . . . . . . . . . . . . . 139
5.2. SUM . . . . . . . . . . . . . . . . . . . . . . . . . 142
5.2.1. Single Concept Summarization . . . . . . . . . . . . 143
5.2.2. Multiple Concept Summarization . . . . . . . . . . . 147
5.2.3. Summarization . . . . . . . . . . . . . . . . . . . 149
5.2.4. Perspective . . . . . . . . . . . . . . . . . . . . 150

Chapter 6. Conclusions . . . . . . . . . . . . . . . . . . . 158

6.1. Summary . . . . . . . . . . . . . . . . . . . . . . . 158
6.2. Future Directions . . . . . . . . . . . . . . . . . . 163
6.2.1. The Dictionary . . . . . . . . . . . . . . . . . . 163
6.2.2. TRACE . . . . . . . . . . . . . . . . . . . . . . . 164
6.2.3. QUEST and SUM . . . . . . . . . . . . . . . . . . . 164

Bibliography . . . . . . . . . . . . . . . . . . . . . . . . 166

LIST OF TABLES


| | | |
|---|---|---|
| Table 2-1: | Case Relations | 23 |
| Table 4-1: | Distribution of Relations | 89 |
| Table 4-2: | Search Space Size | 89 |
| Table 4-3: | Path Lengths | 90 |
| Table 4-4: | Number of Times Concepts Appeared in TRACE's Output | 91 |
| Table 5-1: | Lehnert's Question Typology | 140 |
| Table 5-2: | Theories of Summarization | 154 |

## LIST OF FIGURES


| | | |
|---|---|---|
| Figure 2-1: | Characterizing the Relations | 26 |
| Figure 2-2: | Complete Representation of Text | 39 |
| Figure 2-3: | Restaurant Script | 41 |
| Figure 2-4: | A Narrative Schema | 48 |
| Figure 2-5: | Believe and Encourage | 52 |
| Figure 2-6: | Opening a safe | 53 |
| Figure 3-1: | A section of a semantic network. | 57 |
| Figure 3-2: | Flow of Control | 70 |
| Figure 3-3: | A subgraph of the knowledge space. | 72 |
| Figure 3-4: | A Path That FIND-PATH Suggests | 78 |
| Figure 3-5: | The Constraints Attached to Each Arc in the Path | 78 |
| Figure 4-1: | The Tale of the Pig - 24 event/state concepts | 94 |
| Figure 4-2: | The Peasant and the Waterman 14 event/state concepts | 101 |
| Figure 4-3: | The Wishing Ring - 12 event/state concepts | 106 |
| Figure 4-4: | The Restaurant Story - 16 concepts | 109 |
| Figure 4-5: | Robbing a Liquor Store 15 event/state concepts | 113 |
| Figure 4-6: | The Margie Story - 12 event/state concepts | 116 |
| Figure 4-7: | The Archer, William Tell - 10 event/state concepts | 118 |
| Figure 4-8: | The Czar's General and the Clever Peasant 7 event/state concepts | 120 |
| Figure 5-1: | Two parts of answer retrieval | 123 |
| Figure 5-2: | How did the pig clean the clothes? | 126 |
| Figure 5-3: | How did the pig wash the laundry? | 126 |
| Figure 5-4: | How did the pig dry the laundry? | 127 |
| Figure 5-5: | Why did John labor in the field? | 129 |
| Figure 5-6: | Why did John sell food? | 129 |
| Figure 5-7: | Why did the pig trot to the stream? | 130 |
| Figure 5-8: | Why did John order lobster? | 130 |
| Figure 5-9: | What caused the spade to hit the hard object? | 132 |
| Figure 5-10: | What caused the balloon to burst? | 133 |
| Figure 5-11: | What caused Margie to be unhappy? | 133 |
| Figure 5-12: | Why did John rob the liquor store? | 135 |
| Figure 5-13: | Why did Margie cry? | 135 |
| Figure 5-14: | What happened when the peasant was chopping a tree? | 137 |

Figure 5-15: What happened when the peasant dropped the axe? 137
Figure 5-16: What happened when the peasant sought the axe? 138
Figure 5-17: What did John need to rob the liquor store? 138
Figure 5-18: What did John need to eat at the inn? 139
Figure 5-19: The Restaurant Story 144
Figure 5-20: The wind carried the balloon into the tree. 146
Figure 5-21: An identity tree. 148
Figure 5-22: A preparation of ′robbing′ is ′getting a gun′. 149
Figure 5-23: "Robbing a Liquor Store" - Summary 151
Figure 5-24: "The Tale of the Pig" - Summary 152
Figure 6-1: The Temporal Relations 159

# Chapter 1

# Introduction

## 1.1 Overview

This research addresses the problem of representing the conceptual coherence of events described in text. Its goal is to gather together the pieces of event description and identify the matrix of concepts from which it was derived. Sometimes this gives rise to a single encompassing event concept, other times to a network of them. For example, the descriptions

> The pig washed the laundry.
> The pig dried the laundry.

can be gathered into a single instance of the 'cleaning' concept.

> The pig cleaned the laundry.

On the other hand, the descriptions

> The peasant was digging in his garden.
> His spade unexpectedly hit a hard object.

invoke a mesh of concepts including 'digging', 'pushing', 'moving', and 'hitting'.

To dig one must push the spade into the ground.
By pushing the spade into the ground it moves into the ground.
Therefore, if there's a hard object in the ground the spade could hit it.

The representation scheme is by no means limited to simple action sequences. Its breadth includes mental events,

After pondering his problem,
he decided to take the chest of gold to the Czar.

conversations,

The fox said, "Give me your little ones or I will knock down the tree and eat all of you."
The wren replied, "But they are so small, they could hardly fill your appetite."

continuous perceptions,

The prince spied the pig carrying her laundry to the stream.
He watched her soak and scour it.

motivated behavior,

John was angry at Bill.
He hit him.

and speech acts.

John asked Mary, "Can you reach the salt?"
She passed it to him.

Our claim is not so much that the representation has captured the meaning of the passage, but more the conceptual coherence of the event descriptions, by gathering together conceptually related descriptions into a structured packet of information. What I mean by event/state concepts are the words (terms) in a language which designate events. The lexical interactions of event/state concepts act as a model for how their corresponding extensions are perceived to interact in the world. A 'conceptual coherence relation' is a relation that identifies concepts which cohere; they connect terms that are closely allied because the types of events they designate frequently (sometimes always) occur together. Conceptual coherence relations are **minimal** with regard to the interpretation of the 'meaning' of the terms. The **coverage** of the relations include all types of event/state concept interactions. The **reasonableness** of the relations for connecting coherent event terms is demonstrated by experiments in summarizing and question answering.

The importance of the representation is that it supports the process of interpreting the text; it provides a computational handle on the text by analyzing it into structured coherent chunks of event descriptions. The situation is analogous to using case notation to represent the content of English sentences. From a computational viewpoint the case notation is relatively easy to parse into, and the resulting representation is vastly more convenient for computation than the original english form. Similarly, the coherence schemas are easier to produce than, for example, causal or speech act representations, and by producing these representations later, more difficult, interpretations of the text will be easier to establish.

So the reader should think of the event concept coherence representations as a way station along a process which begin with a lexicogrammatic parser converting sentences into a case representation, continues by assembling organized packets (schemas) of events, and ends with the packets themselves being organized into larger structures (e.g. story trees, plans, meta-plans, etc.) that are semantically enriched with world and semantic knowledge.

There are seven event/state concept coherence relations: three are taxonomic: four are temporal. The three taxonomic relations are: class/subclass, sequence/subsequence, and coordinate. The four temporal relations are antecedent, consequent, precedent, and sequel. A class/subclass relation holds between two concepts when one is a more exact description of an event than the other; for example, a subclass of the concept 'move' is 'ascend'. Coordinate and sequence/subsequence are both whole/part relations, but the coordinate of an event is concurrent with it, and a subsequence occurs only for a subinterval. There exists a coordinate relationship between the concepts 'hammering' and 'holding'; while 'hammering' one 'holds' the hammer in his/her hand. There exists a sequence/subsequence relationship between the concepts 'farming' and 'plowing'; a subsequence of 'farming' is 'plowing', another, non-concurrent, part of 'farming' is 'harvesting'. Antecedents are events that necessarily come before an event; to 'carry' a book one must first 'have' the book. Consequents are events that must immediately follow an event; after 'gathering' flowers one 'has' flowers. Precedents and sequels are weaker forms of antecedent and consequent. A precedent of an event is an event that, with some regularity, occurs before it; a precedent of 'carving' meat is 'sharpening' the knife. Similarly, a sequel of an event is an event that, with some regularity, occurs after it; if an object is 'sought' sometimes it is 'found'.

The representation system is realized as a program called NEXUS. The subsystem which produces the representations is called TRACE, the question answerer QUEST, and the summarizer SUM. The system is programmed in procedural logic using Dan Chester's HCPRVR [Chester 80a, Chester 80b].

A dictionary of event/state concepts has been compiled. Concepts in the dictionary are arranged according to the seven coherence relations. Attached to each relationship in the dictionary is a set of constraints. TRACE produces a representation by matching case encoded text against the dictionary. It uses a path finding algorithm, traversing an arc only if it can satisfy the constraints attached to the arc. The representation it produces is a copy of the relevant portion of the dictionary. QUEST answers questions about the text by retrieving the coherence patterns associated with question types. SUM summarizes text by exploiting the hierarchical organization of the TRACE-produced representations.

## 1.2 An Empirical Study of Coherence: A Chronology

The hypothesis of this research has been 1) that text is composed of structured chunks of coherent event/state concepts, 2) that it is possible to collect together events without explicitly working out all the details of their semantic (e.g. causal) inter-relationships, and 3) the initial grouping and structuring of text could be accomplished by augmenting case relations with a handful of inter-event coherence relations.

Furthermore, it was supposed that the task could be accomplished by 1) developing a dictionary of event/state concepts from an analysis of text, and 2) using the dictionary as a basis for representing text. Representations would be generated by matching text against the dictionary and then copying from the dictionary the relevant inter-concept connections.

### 1.2.1 PHASE1 - A Prototype

Even before the analysis of text began a simple system, PHASE1, was written, which used one coherence relation, schema-subschema, in addition to some case relations, to construct a representation for a paragraph of text. The system was an interpreter. Relations between concepts were represented as a 4-tuple. For example, the relationship between the concepts 'carry' and 'hold' would be represented:

[schema-subschema carry hold (list-of-constraints-on-matching-cases)] The constraints were intended to control the system's inferencing capability.

PHASE1 served as a prototype for TRACE. The two systems differed in four regards. TRACE supports seven coherence relations, and PHASE1 only one. TRACE works with a dictionary of approximately 150 concepts, and PHASE1 perhaps a dozen. TRACE uses several different match functions to test constraints, and PHASE1 used one. Finally the two differ in their search heuristics - TRACE's are both more complete and more efficient.

### 1.2.2 Analysis of Text

The next step was to analyze parts of ten folktales. The folktales come from A Children's Treasury of Folk and Fairy Tales [Protter 61] and were adapted by Eric Protter. Folktales were chosen for two reasons: they include a diverse range of event/state concepts, and the prose style is relatively simple. Two important tools aided the analysis process: Webster's New World Dictionary [Dictionary 58] and Miller and Miller & Johnson-Laird's [Miller 72, Miller & Johnson-Laird 76] partial taxonomy of verbal concepts.

The basic analysis procedure consisted of finding groups of sentences that were centered around a small set of intrinsically related event/state concepts. Scanning a dictionary to find relationships between concepts was useful, but not conclusive since a dictionary is incomplete and likely to leave out important conceptual relationships. For example, the concept chop is defined in Webster's New World Dictionary:

> To cut by blows with an ax or other sharp tool; hew.

But if a peasant is chopping wood with an axe, then he **holds** the axe - a relationship not explicitly stated in the dictionary. Miller and Miller & Johnson-Laird's taxonomy represents a more formal arrangement of concepts than the dictionary, and thereby provided a useful tool for refining relationships between concepts that are accounted for in their partial taxonomy. For example, where the dictionary defines snatch,

> To grasp or seize suddenly, eagerly, or without right, warning, etc; grab.

Miller & Johnson-Laird [Miller & Johnson-Laird 76] more precisely say -

(rapidly(take))(x,y) & hold(x,y)

The second definition is better, since it uses the core concept 'take', while also identifying a necessary consequent of 'snatching' which is 'holding'. But even here the definition fails to explicate the exact nature of the relationship that exists between the pairs of concepts snatch/take and snatch/hold; 'snatch' is a subclass of 'take', and 'hold' is a result of 'snatch'.

For that matter, neither the dictionary nor Miller & Johnson-Laird account for the relationship between 'ordering' and 'eating' which occurs in the context of a restaurant.

John ate at the restaurant.
He ordered veal.

The only way to derive connections of this sort is by an analysis of text. So, the text was the most important source for discovering relationships between concepts.

The first step of analysis consisted of converting each event/state concept into case notation. There were two motivations for choosing case notation to represent individual concepts. First, and most importantly, the notation brings the words designating events and states to the front of each representation unit. The primary role of events and the dependent status of participants was nicely reflected by the case notation. This feature seemed to be in agreement with the aim of the study, which was to find event/state coherence patterns. The other advantage of case notation was that the relationships between events and participants were limited to a small set. It was believed that a small set of role markers would, at a later stage, have computational advantages [Charniak 81].

After an event was converted to case notation the next task was to relate it to a previously described event. Originally only two relations were used, whole/part and class/subclass. The relations whole/part and class/subclass represented a splitting into two categories of the coherence relation used in PHASE1, schema-subschema. Rapidly it became apparent that whole/part and class/subclass were not sufficient. These two relations could be used to identify individual concepts adequately. For example, a pair of sentences like

The pig washed the clothes.
She dried them.

can be connected by inferring that the 'washing' and 'drying' were parts of a 'cleaning' event. But the problem was that the two relations could not be used to represent chains of events. For example,

He had no extra penny.
He could never eat at the village inn.

'having money' is a condition for 'eating' at an inn. To claim that 'having money' is part of 'eating at an inn' is misleading because 'having money' begins before the 'eating' occurs.

As the study evolved the relations were refined and slowly expanded into a set of seven. Three were taxonomic; these relations were used to organize individual concepts into a hierarchical structure. Four relations were temporal; these were used to chain together sequences of events. Causal relations were avoided. Remember, one of the goals of this research was to find a set of relations that would be sufficient for gathering together semantically related instances of event concepts without getting bogged down explicating all the details of the connections. The disadvantage of adding causal relations would have been that they are much more difficult to establish than temporal ones. Given a pair of sentences like

The pig cleaned the laundry.
She carried it home.

establishing the causal relationship between 'cleaning' and 'carrying' is at best difficult. Temporally, we can simply state that, in the world, 'carrying' is a highly predictable sequel to 'cleaning'. So the coherence relations developed during the analysis organized the text without committing to a particular understanding. Thus later, more expert, systems of knowledge could, with the advantage of working with smaller organized chunks of information, enrich (e.g. causally) the representation.

### 1.2.3 Compiling the Dictionary of Event/State Concepts

After completing the analysis phase, the next stage of research consisted of compiling a dictionary of event/state concepts. A number of difficulties had to be overcome in order to construct a dictionary. The first problem was that the representations generated by the analysis of text were at times inconsistent - this was to be expected considering the exploratory nature of the analysis phase. Each new appearance of a concept, or set of concepts, could, in principle, modify the previous representation scheme. This problem was compounded by the fact that the relations had changed in number and meaning, as the work progressed.

Another difficulty was that the taxonomic relations had not been fully exploited. One of the goals of compiling the dictionary was to avoid, as much as possible, including redundant information. Here the concept of inheritance played a key role. Event/state concepts inherit relationships from their ancestor concepts; if 'keeping' is a descendant of 'having' and 'having' and 'giving' have a relationship, then 'keeping' should inherit that relationship without having to state it explicitly.

The final problem was that it was not possible to gather together all the concepts on a single piece of paper. There were too many concepts, too many relationships, too much editing to do, and not enough space - so in short order pencil and paper became cumbersome to work with.

To aid the compilation process a LISP program, TANG (i.e. tangled hierarchy), was written which could display, from a given concept, concepts spanning outward at an arbitrary depth. TANG worked interactively. In addition to displaying relationships between concepts, TANG allowed the user to insert, delete, and modify the dictionary. At the user's request, TANG would print out a copy of the dictionary. At the end of the compilation process a LISP function was added to TANG that converted TANG's internal form of the dictionary into the relational form described in chapter three.

The actual entry of data worked in a straightforward manner. Piece by piece the representations generated during the analysis phase were considered. Before adding a relationship to the dictionary TANG was used to determine how the new relationship would interact with the relationships previously established between concepts. In particular, several questions needed to be answered: Was the new relationship consistent with previous ones?, Was it an improvement? was it redundant? (i.e. already implicitly in the net because of inheritance), and What constraints and default values should be added to prevent erroneous inferences? The compilation of the dictionary was a painstaking process that resulted in a dictionary of approximately 150 event/state concepts.

The final form of the dictionary was a set of 4-tuples which represent the relationships between event/state concepts, and a set of 3-tuples for defining the default values of a concept. Each 4-tuple is of the form:

[relation event/state event/state list-of-constraints]

The constraints describe how the case arguments of the two event/state concepts should match. So, for example, the relationship between 'snatch' and 'take' would be represented in the dictionary:

```
[subclass take snatch ((match the agents)
                       (match the objects) . . .) ]
```

Each 3-tuple is of the form:

[TEMPLATE event/state list-of-default-values]

The default values for a 'drinking' event would be stored in the dictionary:

[TEMPLATE drink ((obj liquid) . . .)]

### 1.2.4 TRACE - A System to Build Representations

It then became necessary to show that the event/state dictionary could, in fact, be used as a basis for chunking text into organized subnets of concepts. There were several technical questions that needed to be answered, and the project weighed in the balance. At issue was the interplay of concepts. Effectively the dictionary had been constructed by patching together pieces of text into an elaborate network of concepts. But how would relationships, developed independently, affect one another when they were knotted together into a single net? Could an interpreter use the net to build representations of text or would a relationship developed from one analysis somehow interfere, or block, a relationship discovered in a different piece of text?

To answer these questions TRACE was written, a system that uses the dictionary as a basis for constructing representations of text. The inputs to TRACE are hand-constructed case representations for the sentences of a sample of text. Its output is a set of concept trees, a map, copied from the dictionary, which describe the interconnecting paths between the event/state concepts used in the text.

TRACE was written in procedural logic using Dan Chester's HCPRVR [Chester 80a, Chester 80b]. The advantage of writing in procedural logic was that it promoted clarity of thought. At a high conceptual level the program could be specified without serious loss of efficiency. The effectiveness of HCPRVR, and procedural logic in general, as a developmental tool can be attributed partially to the advantages of unification as a parameter passing scheme, partially to its automatic provision of backtracking, and partially because it predisposes the user towards the practice of good program decomposition.

TRACE was written as an interpreter of the 3-tuple and 4-tuple rules contained in the dictionary. The separation of knowledge, the dictionary, from the mechanism that used it greatly aided development. Once TRACE was set up it was possible to improve its output, without modifying its program, by changing relationships between concepts or adding new constraints.

TRACE has several tools available for controlling the inferencing process. First there are the constraints. A constraint between two event/state concepts describes how two case arguments, one from each concept, should match. If the arguments do not match the relationship is rejected. TRACE supports several different match functions. Constraints on the relationship between the agents of 'angry' and 'shout' would prevent TRACE from interpreting

Mary was angry at John.
She shouted at him.

to mean that "John shouted at Mary".

Another tool that TRACE uses to control inferencing is the set of default values associated with the case arguments of each event state concept in the dictionary. A combination of default values and constraints prevents TRACE from falsely connecting the two event descriptions

The peasant opened the sack
and then entered the hut.

while allowing TRACE to connect the two descriptions

The peasant opened the door
and then entered the hut.

Testing the dictionary settled into a routine. Each new example created mild perturbations in the net. A host of responses were possible. Sometimes new constraints, default values, or relationships needed to be added to the net. Other times a set of relationships in the dictionary was reorganized. Occasionally bugs were found in TRACE. Always, before moving onto the next example, TRACE was retested on previous examples to insure that the subtle ramifications of each modification were not left undetected. Gradually the integrity of the dictionary increased.

### 1.2.5 Question Answering and Summarizing

The relative success of TRACE represented an answer to the questions that had motivated this research. A dictionary of event/state concepts can be derived from an analysis of text. It is possible to use the dictionary to collect and organize event descriptions in text. The task can be accomplished without a complete (e.g. causal) description of the relationships.

But before the experimental work could be declared complete it was decided that further demonstrations of the utility of the representation produced by TRACE, beyond its intended purpose of capturing the coherency of event descriptions in text, would be useful. First a question answerer, QUEST, was written. QUEST's input is a question in case notation form, and a list of concept trees produced by TRACE. Its output is a case notated version of an answer. QUEST is able to exploit the structure of the schemas for answering questions. For example, 'how' questions can be answered by finding a class/subclass relationship between the instances of two concepts. For the event descriptions

The peasant moved the wood,
he carried it on his back.

and the question

How did the peasant move the wood?

QUEST would produce the answer

By carrying it on his back.

Lastly a summarizer, SUM, was written. For SUM the input is a list of schemas produced by TRACE, and the output a summary. Here TRACE's structuring of the text could be thought of as a hierarchical arrangement of concepts. Concepts at the top of the trees represent a summary of the concept the tree represents.

The additions of QUEST and SUM to NEXUS represent only the initial exploration of the possible uses of the TRACE produced schemas (concept trees). The conclusion of this thesis will suggest other areas of exploration.

### 1.3 Outline of Chapters

The seven relations that are described in detail in chapter two account for the conceptual coherence of events. After defining the relations and citing numerous examples, the relation scheme is applied to a paragraph of text from <u>The Tale of the Pig</u>. There, in a sentence by sentence analysis, we will see a representation evolve.

The last chore of the chapter, and perhaps the most important, will to be to look at the breadth of the representation scheme by comparing it to other schemes cited in the literature. In particular, we will discuss concept coherence representations for script-based stories (Schank, Abelson, Cullingford), plan-based stories (Schank, Abelson, Wilensky), text covered by story trees (Simmons, Rumelhart), text covered by rhetorical coherence relations (Hobbs), and text involving speech acts (Allen, Cohen, and Perrault).

In the third chapter we will show how the dictionary was constructed. The chapter begins by motivating the choice of semantic networks as a knowledge representation paradigm. Next, the notation is described and some issues concerning the inheritance of properties and relationships between concepts is discussed. Finally, the two stage algorithm that TRACE uses for finding the coherence of events is described.

Chapter four discusses the results of the experiments with TRACE. Eight examples are described. Some references that are resolved are pointed out. Rejected paths between concepts are cited.

Chapter five will show how the schemas produced by TRACE can be used by QUEST to answer questions, and by SUM to produce summaries. QUEST's basic procedure is to seek the pattern of relationships associated with the answer of a particular question-type. It will be shown that QUEST can be used to retrieve answers for all of the event-relationship questions in Lehnert's taxonomy of questions [Lehnert 77].

SUM's input is the representation produced by TRACE. It uses two types of strategies for summarizing text. Single concepts are summarized by selecting the nodes at the top of concept tree that represents them. Multiple concepts are summarized by using heuristics to delete concepts that are relatively less important. To demonstrate SUM's techniques, a number of working examples are provided. The chapter concludes by comparing SUM to other summarizers discussed in the literature.

Chapter six summarizes the findings of this research and proposes new directions.

# Chapter 2

# Representing Text with Relations

## 2.1 Method

In this chapter we will discuss the relations supported by Nexus. We will briefly outline the method used for deriving them. We will formally examine them, citing several examples of each relation type. In detail we'll see how they can be used to represent a paragraph of text from The Tail of the Pig. We'll conclude by showing the breadth of the relation scheme by comparing it to several representation schemes discussed in the literature - including scripts, plans, speech acts, schema/narrative trees, and story trees.

The study began with an analysis of parts of ten folktales. The first step of analysis consisted in converting by hand each event/state concept in the text into case notation. The head of each representational unit designates an event or state. Associated with the head is a set of roles, marked by names like: agent, instrument, object, location, source, destination, and affected entity. For,

John rode a bicycle to the market.

the case representation is:

(ride agt John instr bicycle dest market)

Roughly this means that John was the agent of the riding event, the instrument was a bicycle, and the destination the market. The case relations we'll use primarily derive from Simmons [Simmons 83], but also include pieces of the case systems of Fillmore [Fillmore 68], Grimes [Grimes 75], and de Beaugrande [Beaugrande 80]. Table 2-1 shows a catalogue of the case relations that **Nexus** uses.

After an event was converted to case notation the next task was to relate it to a previously described event. Originally only two relations were used, whole/part and class/subclass, but as the study evolved the relations were refined and expanded to a set of seven: class/subclass, sequence/subsequence, coordination, antecedent, consequent, precedent and sequel. For a pair of sentences like:

John opened the door. He entered the shack.

we would construct the case representations,

(open agt John obj door)
(enter agt John loc shack)

The connection between 'opening a door' and 'entering a shack' is that opening a door enables one to enter a shack; it is an event that occurs with some regularity before an 'entering'. So opening a door is a precedent of entering. The resulting representation would be:

| Case | Abbreviation | Definition |
|---|---|---|
| Object | Obj | The thing moved or transferred. |
| Source | Source | The location of a thing at the beginning of a motion. |
| Destination | Dest | The location of a thing at the end of a motion. |
| Agent | Agt | The entity which instigates the action. |
| Instrument | Instr | The tool used in performing the action. |
| AffectedEntity | AE | The entity affected by an event. |
| Location | Loc | The place where an event occurs. |
| Theme | Thm | An event or state embedded in a perception or communication. |
| Time | Time | The time of an event. |
| Stateof | Stateof | The entity which the state describes. |
| beneficiary | ben | The entity on which the event has a secondary effect. |
| recipient | rec | The receiver in a transfer of possession. |

Table 2-1: Case Relations

```
(enter agt John loc shack
       precedent (open agt John obj door))
```

Most of the time it is necessary to infer intermediate events to connect a pair of events. Consider the sentences:

John bought food. He ate it.

A consequent of 'buying food' is 'having food'. An antecedent of 'eating food' is 'having food'. To represent the connections between 'buying food' and 'eating food' we must infer the intermediate state that John 'had food'. We can represent this connection as follows[1]:

```
(eat agt John obj food
     antecedent (have agt John obj food
                      consequent* (buy agt John obj food)))
```

Another example is:

John chased the ball.
It hit the wall.

A co-occurrence (coordinate) of 'chasing' is that the object being chased must be 'moving'. An antecedent of 'hitting' is 'moving'. To connect 'chasing' and 'hitting' we must infer a 'moving'. We can represent the connection as follows:

```
(chase agt John obj ball
       coordinate  (move obj ball
                         antecedent* (hit obj ball ae wall)))
```

---

[1] Consequent* is the inverse of consequent.

## 2.2 Relations

Before discussing each relation in detail we will give an overview, characterizing their differences and similarities (see figure 2-1). One way is by classifying some of the relations as being taxonomic, others as being temporal. Class/subclass is obviously taxonomic. Coordination and sequence/subsequence are two types of whole/part relations. Antecedent, precedent, consequent, and sequel are temporal. Antecedent and precedent concepts come before an event, and consequent and sequel concepts come after an event.

The sequence/subsequence and coordinate relations can be differentiated by their intervals. Subsequence event concepts occur for a subinterval of their parent concept, and coordinated concepts are concurrent with it.

A third dimension of characterization has to do with a logical/plausible split. Antecedent and consequent related concepts are logically related. Precedent and sequel concepts are plausibly connected.

We can also characterize the relations by describing how they can sometimes overlap one another. In some instances the class/subclass relation clearly overlaps the whole/part relation. For example, one could classify the relationship between 'carrying'

```
               Event/State Coherence Relations
                 /                             \
                /                               \
               /                                 \
           Taxonomic                             Temporal
            /     \                               /      \
           /       \                             /        \
          /         \                           /          \
          |          \                         /            \
          |           |                       /             |
      Whole/part     sc                   Before         After
      /     \                             /    \         /    \
     /       \                         n /     p|     n|     p\
    /        |                          /       |      |       |
subseq     coord                      ante    prec   consq    seq
```

Key:

| Abbreviation | Fullname |
|---|---|
| subseq | sequence/subsequence |
| coord | coordination |
| sc | subclass |
| ante | antecedent |
| prec | precedent |
| consq | consequent |
| seq | sequel |
| n | necessary |
| p | plausible |

Figure 2-1: Characterizing the Relations

and 'holding' as either class/subclass (e.g. a type of 'holding' is 'carrying') or coordinate (e.g. 'carrying' is a coordination of 'holding' and 'travelling'). In the case of 'trading' and 'bargaining' the sequence/subsequence and sequel relations overlap; either 'bargaining' is a subsequence of 'trading', or 'trading' is a sequel of 'bargaining'.

### 2.2.1 Class/Subclass

If two concepts can describe the same event, but one is a more exact description than the other, their relationship should be classified as class/subclass (<u>sc</u>). For example,

The elevator finally moved, descending rapidly.

```
(move obj elevator
      sc (descend obj elevator mod rapidly))
```

Every event which could be described as a 'descending', could also be described as a 'moving' and obviously not every 'moving' is a 'descending' - thus 'descending' is a subclass of 'moving'. Formally we can state the conditions for establishing a class/subclass relationship as follows:

> For all event/state concepts x & y, a subclass of x is y if, i) any event which can be described as y can be described as x, and ii) not all events which can be described as as x can be described as y.

Additional examples of the class/subclass relation are:

1. John walked slowly, hobbling on his sore foot.

2. John put the china on the table, setting each piece carefully in place.

3. John hit the ball, striking at it viciously.

4. John took the cash, snatching it from the dealer's hands.

5. The eagle flew away, soaring into the sky.

### 2.2.2 Sequence/Subsequence

If one event is a part of another event, and it occurs for a sub interval of time, then the corresponding concepts are in a sequence/subsequence (<u>subseq</u>) relationship. For example,

He dug deeper, breaking the earth with his spade.

```
(dig agt he dir deeper
     subseq (break ae earth instru (spade possby him)))
```

Whenever the concept 'digging' applies the concept 'breaking' (e.g. breaking earth) also applies, and there are no instances of 'digging' without a 'breaking'; consequently, 'digging' and 'breaking' have a sequence/subsequence relationship.

Our conditions for establishing a sequence/subsequence relationship are:

> For all event/state concepts x & y, a subsequence of x is y if, i) events which can be described as x always contain, as a proper subset of x's interval, events which can be described as y, and ii) there are no instances where x is a subsequence of y.

The second condition distinguishes sequence/subsequence from the class/subclass relation. In effect, the second condition guarantees that y is only a part of the sequence x, and cannot refer to all of x.

Additional examples of the sequence/subsequence relation are:

1. John was travelling to N.Y. His plane departed at 4 a.m.
2. John visited N.Y. He travelled by train.
3. John cleaned his laundry. He washed his clothes in a machine.
4. John farms an acre of land. He'll be planting in the spring.
5. He dug deeper, removing dirt by the spadeful.
6. They exchanged gifts. He gave her a new hat.
7. Patco negotiated with the government. They argued that they should be compensated for stressful working conditions.

### 2.2.3 Coordination

If an event has a part which occurs for the same time interval, then the two corresponding event concepts are in a coordinate (<u>coord</u>) relationship. For example,

John carried the book.
He held it in his hands.

```
(carry agt John obj book
      coord (hold agt John obj  book instr  hands))
```

Every event which conveys a sense of 'carrying' in part also conveys a sense of 'holding', and there are 'holding' events that are not part of a 'carrying', and no 'carrying' event can completely be described by 'holding' - thus 'holding' is a coordinate (coord) of 'carrying'. A formal statement of this relation is:

> For all event/state concepts x & y, a coordinate of x is y if, i) every event which can be described as x can in part be described as y, and ii) if x occurs y must necessarily occur simultaneously, and iii) there are no instances where an event which can be described by x can be completely described by y.

If we say y is a subsequence of x then y describes a part of the time interval that can be described by x, if we say that y is a coordinate of x then y is one of the concurrent activities that compose x. (e.g.. 'travelling' is made up of the successive events 'depart', 'move' and 'arrive', but 'carrying' is a continuous coordination of 'holding' and 'travelling'.). Note that the third condition insures that y is not in a subclass relation to x.

Additional examples of coordination are:

1. The wren moved slowly down the road; the fox followed.
2. He carried the box of books. He travelled slowly to the car.
3. She rode the horse along the path. She sat high in the saddle.
4. Mary glared at John, showing him her feelings.
5. Mary glared at John, staring at him fiercely.
6. John brought the dessert. He travelled by car.
7. He chopped the wood, holding the axe in one hand.

### 2.2.4 Antecedent

If one event must necessarily occur before another event, the relationship between their corresponding concepts is classified as antecedent (<u>ante</u>). For example,

John had some food.
He ate it.

```
(eat agt John obj food
     ante (have agt John obj food))
```

To eat one must first 'have' the thing which is to be eaten; thus an antecedent of eating is having food. Our formal statement for establishing an antecedent relationship is:

> For all event/state concepts x & y, an antecedent of x is y if, i) every event which can be described as x is preceded by a sequence which can be described by y, and ii) y is logically necessary.

Other examples of the antecedent relation are:

1. John lost his axe, but then he recovered it.
2. Mary had the book, but she gave it to John.
3. John's car moved forward and hit Mary's car.
4. The trumpet sounded and he heard it.

### 2.2.5 Consequent

If one event always, necessarily, occurs immediately after the other, then the relationship between their corresponding concepts should be marked as consequent (consq). Take the following example,

John gave Mary a red kite.
Mary had a red kite.

```
(give agt John rec Mary obj kite
     consq (have possby Mary obj kite))
```

Events of 'giving' are necessarily immediately followed by states of 'having'. Formally we say that:

> For all event/state concepts x & y, a consequent of x is y if, i) every event that can be described as x is always immediately followed by an event which can be described as y, and ii) y logically follows from x.

Listed below are several examples.

1. John threw the baseball. It moved across the diamond to the third baseman.
2. They seized the fort, and held it for 3 months.
3. He grabbed the gun and held it in his right hand.
4. Mary has some wild flowers. She gathered them in the park.
5. John dropped the axe and it fell into the lake.
6. John persuaded Mary. She agreed to buy a ticket for the fund raiser.

### 2.2.6 Precedent

If one event, with some regularity occurs before another event, the relationship between their corresponding concepts can be classified as precedent (prec). For example,

John sharpened the knife.
He carved the meat.

```
(carve agt John obj meat instr knife
      prec (sharpen agt John instr knife))
```

Events which can be described as 'carving' are sometimes preceded by sequences which enable the 'carving' and can be described as 'sharpening'. Formally we can say,

> For all event/state concepts x & y, a precedent of x is y if, i) a sequence which we can describe as x is sometimes, with some regularity, preceded by a sequence which can be described as y.

Note, y is an antecedent of x if y is a necessary condition for x to occur; y is a precedent of x if y regularly occurs before x.

Additional examples of precedent are:

1. John released the bird. It travelled northwards.
2. John was admitted to the university. He'll enter in the fall.
3. John opened the door. He entered.
4. John was hungry. He ate.
5. John was thirsty. He drank.
6. John wanted money. He robbed the bank.

Notice precedent, a coherence relation, covers the semantic relations enablement, examples (1) thru (3), and motive, examples (4) thru (6).

### 2.2.7 Sequel

If one event with some regularity follows another, the relationship between their corresponding concepts is sequel (seq). An example of a sequel relationship is:

John cleaned the laundry.
He carried it home.

```
(clean agt John obj laundry
       seq (carry agt John obj laundry dest home))
```

Sequences of 'cleaning' are typically followed by sequences where the laundry is 'moved'. Formally we say,

> For all event/state concepts x & y, a sequel of x is y if y is one out of a set of things that can, and do with some regularity, follow x.

If one event always, immediately, follows the other the relationships between the concepts which described them is consequent; sequels are neither necessary nor need they immediately follow. Additional examples of sequel are:

1. He searched for the clue and found it.
2. After negotiating for a month, they agreed to terms.
3. Mary and John argued for an hour. Eventually John agreed with Mary.
4. The cup hit the floor, breaking into pieces.

5. John shouted at Mary. She became angry.

6. Mary broke her toy. She was sad.

7. John watched the movie. He went home.

The coherence relation, sequel, covers outcomes, examples (1) thru (4), reactions, examples (5) and (6), and highly predictable temporal relationships , example (7).

## 2.3 Example

In this section we will examine our relation scheme by applying it to a sample of text from the folktale, "The Tale of the Pig". It is hoped that from a step by step analysis the reader will be better able to grasp a sense of the structured content coherence that we have been advocating.

> The pig trotted towards the stream, carrying a bundle of clothes. The animal expertly soaked and scoured the laundry. The pig hung the clothes in the sun to dry. The pig gathered her laundry and trotted home.

### 2.3.1 Analysis

#### 2.3.1.1. "The pig trotted towards the stream, carrying a bundle of clothes."

There are two events described in this sentence, 'trotting' and 'carrying'. Individually we can represent them as follows:

(trot agt pig dir stream)

(carry obj (bundle of clothes))

'Carrying' has two coordinates, 'holding' and 'travelling'. 'Travelling' has three subsequences 'departing', 'moving', and 'arriving'. A subclass of 'moving' is 'trotting'.

```
(carry agt pig obj (bundle of clothes)
       coord (travel agt pig twd stream
                    subseq (move1# agt pig twd stream
                             sc (trot agt pig
                                      twd stream))))
```

**2.3.1.2. "The animal expertly soaked and scoured the laundry."**

The sentence describes two events:

```
(soak agt pig obj laundry)

(scour agt pig obj laundry)
```

'Soaking' and 'scouring' are both subsequences of a 'washing' event:

```
(wash agt pig obj laundry
      subseq1 (soak agt pig obj laundry)
      subseq2 (scour agt pig obj laundry))
```

**2.3.1.3. "The pig hung the clothes in the sun to dry."**

Here the author explicitly states that the consequent of 'hanging clothes in the sun' is to dry them:

```
(dry obj clothes
     consq* (hang obj clothes agt pig loc sun))
```

'Washing' and 'drying' are two subsequences of a 'cleaning' event,

```
(clean agt pig obj clothes
       subseq1 (wash agt pig obj laundry
                     subseq1 (soak agt pig obj laundry).
                     subseq2 (scour agt pig obj laundry))
       subseq2 (dry obj clothes
                    consq* (hang obj clothes
                                 agt pig
                                 loc sun)))
```

An enablement of cleaning clothes is that the clothes are moved to the location where they will be cleaned. A subclass of 'moving' is 'carrying'.

```
(clean agt pig obj clothes
    prec (move2# agt pig obj clothes
            sc (carry agt pig obj (bundle of clothes)
                  coord (travel agt pig twd stream
                           subseq (move1# agt pig
                                          twd stream
                                          sc (trot agt pig
                                                   twd stream))))
       subseq1 (wash agt pig obj laundry
                     subseq1 (soak agt pig obj laundry)
                     subseq2 (scour agt pig obj laundry))
       subseq2 (dry obj clothes
                    consq* (hang obj clothes
                                 agt pig
                                 loc sun)))
```

2.3.1.4. **"The pig gathered her laundry and trotted home."**

Two events are described in this sentence:

(gather agt pig obj (laundry possby her))

(trot agt pig dest home)

A consequent of 'gathering the laundry' is 'having the laundry'. An antecedent of 'carrying laundry' is 'having laundry'. A coordinate of 'carrying' is 'travelling'. A subsequence of 'travelling' is 'moving'. A subclass of 'moving' is 'trotting'.

```
(carry agt pig obj laundry
       ante (have agt pig obj laundry
                  consq*(gather agt pig obj laundry))
       subseq (travel agt pig dest home
                      subseq (move agt pig dest home
                                   sc (trot agt pig
                                            dest home))))
```

The 'moving' event is a sequel of 'cleaning'; after cleaning the laundry at the stream it is usually moved somewhere else. A subclass of 'moving' is 'carrying'. If we pull together all the various subgraphs we have constructed, we will end up with the graph in figure 2-2.

There are a few noteworthy features of this representation. In itself the representation shows the events of the story are coherent; the events can be collected together under the concept 'cleaning'. So the representation is de facto evidence of the text's coherency.

Also notice that our analysis has produced a single tree. In general the samples we look at will produce several interconnecting trees (one per major concept). In this case we have only one tree because the text describes only one major concept 'cleaning'. Because the representation is a tree it would be easy to summarize the text; the top node of the tree represents a summary of the story.

```
[clean agt pig obj clothes
   prec (move2# agt pig obj clothes
              sc (carry agt pig obj (bundle of clothes)
                    coord (travel agt pig twd stream
                             subseq (move agt pig
                                        twd stream
                                        sc (trot agt pig
                                              twd stream))))
                            .dest home))))
   subseql (wash agt pig obj laundry
                 subseql (soak agt pig obj laundry)
                 subseq2 (scour agt pig obj laundry))
   subseq2 (dry obj clothes
                consq* (hang obj clothes
                             agt pig
                             loc sun)))
   (seq (move2# agt pig obj laundry source stream dest home
           ante (have agt pig obj laundry
                      consq*(gather agt pig obj laundry))
           sc (carry agt pig obj laundry
                 subseq (travel agt pig dest home
                           subseq (movel# agt pig dest home
                                     sc (trot agt pig
                                            dest home))))))]
```

Figure 2-2:   Complete Representation of Text

The top node of the tree is (clean agt pig obj laundry loc stream), and could be used to produce the summary 'The pig cleaned the laundry at the stream'.

Also the relations used to organize the events in the text can be used to suggest answers to question. For example the combination of consequent and precedent relationships can be used to answer 'goal' questions. For a question like 'Why did the pig carry the laundry to the stream?', QUEST will be able to use the coherence relations to suggest the answer 'To clean the laundry at the stream.' Both summarization and question/answering capabilities will be discussed in greater detail in chapter 5.

## 2.4 Comparison with Other Approaches

### 2.4.1 The Restaurant Script

Schank et. al. [Schank & Abelson 77, Cullingford 78] model situations in terms of scripts. In Scripts, Plans and Goals they are defined (p41):

> A script is a structure that describes appropriate sequences of events in a particular context.

A script couples a particular context with sequences of events which, from experience, are associated with that context. If we think of semantic knowledge as a space of verbal concepts, then scripts describe a view on that space. From the vantage point of a particular script the descriptions of events that we can see in the space of semantic memory are only those that are highly likely to occur in the script's context. Scripts provide meta-event-knowledge about events and contexts strongly coupled together. Figure 2-3 shows a sample restaurant script.

Script: Restaurant

Track: Coffee Shop

| | |
|---|---|
| Props: Tables | Roles: Customer |
| Menu | Waiter |
| Food | Cook |
| Check | Owner |
| Money | |

Entry Condition: Customer is hungry.
Customer has money.

Results: Customer has less money.

Typically occurring events in the script:

Scene 1: Entering

Scene 2: Ordering

Scene 3: Eating

Scene 4: Exiting

Figure 2-3: Restaurant Script

The major difference between scriptal and coherence versions of eating at a restaurant is a foreground-background distinction. With scripts the context (e.g. location, props, and roles) is brought to the foreground. For NEXUS events are in the foreground and contexts become filters that constrain the coherence inferencing between events.

Lehnert [Lehnert 77] provides an example of text for which a restaurant script would apply:

> John went to a restaurant. The hostess seated John. The hostess gave John a menu. The waiter came to the table. John ordered a lobster. He was served quickly. He left a large tip. He left the restaurant.

If we apply the seven coherence relations to Lehnert's example the text gathers into a single instance of the concept of eating.

```
[eat agt John obj lobster loc restaurant
      prec (travel agt John dest restaurant
                   subseq (went agt John dest restaurant))
      subseq (seated ben John agt hostess
                subseq (give agt hostess rec John obj menu))
      subseq (order agt John obj lobster
                prec (have agt John obj menu
                        consq* (give rec John agt hostess obj menu))
                prec (travel agt waiter dest table
                        subseq (came agt waiter dest table)))
      subseq (serve ben John obj lobster)
      sequel (travel agt John source restaurant
                     sc (left agt John source restaurant))
      seq (tip agt John rec waiter
               sc (give agt John object tip rec waiter
                       sc (leave agt John obj tip))) ]
```

First, notice that in NEXUS a script-based story is a single concept tree. All the events in the story can be subsumed by the concept eating.

Also notice that some of the connections must be limited to restaurant contexts; ordering food and leaving tips are events that only occur when eating at a restaurant. In NEXUS constraints are attached to the arcs connecting 'eating' to 'ordering' and 'tipping' which insure that TRACE will only make these inferences if it concludes John is eating at a restaurant. This is an important point. It helps to clarify the difference between the general concept 'eating' and a scripted 'eating'. 'Eating at a restaurant' is a special case of 'eating'. The relationships that distinguish it from the general concept 'eating' are controlled by tightly restricting the matching of case arguments, so for 'tipping' to be a sequel of 'eating' the locations of the 'tipping' and 'eating' must be a restaurant.

### 2.4.2 Robbing a Liquor Store

For Schank and Abelson [Schank & Abelson 77, Wilensky 78, Wilensky 81] plans are made up of information about how actors achieve goals. The distinction between scripts and plans is that plans are general and scripts are specific. In planned behavior there is associated with every kind of D-Goal (Delta Goal) a list of planboxes and scripts that can achieve that goal. If an actor wants to have (D-Cont{rol}) a bicycle s/he can steal it. Stealing is a planbox associated with D-Cont. In NEXUS the relationship between 'having' and 'stealing' is consequent; a consequent of 'stealing' is 'having'. The planboxes associated with a goal are, in NEXUS, coherence chains which result (have as a consequent or sequel) in the goal concept.

An example of a script which achieves a goal is the connection between the car script and the goal of going someplace (D-Prox{imate}); an actor can go to a restaurant by travelling in a car. Again, in NEXUS, the relationship between travelling and being at a location is consequent.

Not all planned behavior can be accounted for by temporal relations. When the plan is concept-oriented NEXUS represents it the same way it represents scripts. The difference is that where a script requires tight restrictions on some of the relationship between concepts, plan do not. So when Schank and Abelson say plans are general and scripts are specific, we can interpret it, in terms of NEXUS, to mean that the matching of arguments for scripts is more tightly constrained then it is for plans.

Wilensky [Wilensky 78] provides an example of text for which a plan scheme would apply.

> John wanted money. He got a gun and walked into a liquor store. He told the owner he wanted some money. The owner gave John the money and John left.

Again, with the coherence relations, most of the text gathers into a single instance of the 'robbing' concept.

```
[rob agt John loc (liguor store)
     prec (want agt John obj money)
     ante (have agt John obj gun
                consq* (get agt John obj gun))
     ante (travel agt John dest (liquor store)
              subseq (move agt John
                         sc (walk agt John dest (liquor store))))
     subseq (say agt John to owner thm (want agt John obj money)
                 sc (told agt John thm (want agt John obj money)))
     subseq (give agt owner rec John obj money)
     seq (getaway agt John source (liquor store)
             sc* (escape agt John source (liquor store)
                   sc* leave agt John source (liquor store)))]
```

Antecedents of 'robbing' are 'having a weapon' and 'travelling to the location where the robbery will occur'. A precedent of 'robbing' is 'wanting money'. Subsequences of 'robbing' are 'asking for the money' and 'giving money to the robber'. A sequel of 'robbing' is the 'escape' of the robber.

### 2.4.3 The Margie Story

Rumelhart has presented a story grammar [Rumelhart 75] that produces two different trees; one tree represents the syntactic organization of a story, the other a form of semantic organization. He uses the semantic organization to produce summaries. One of the stories he discusses is the "Margie Story".

> Margie was holding tightly to the string of her beautiful new balloon. Suddenly, a gust of wind caught it. The wind carried it into a tree. The balloon hit a branch and burst. Margie cried.

The "Margie Story" is a little trickier than the previous examples. The representation that NEXUS would produce is:

```
(carry agt wind obj balloon dest tree
   ante (have agt wind obj balloon
            consq* (take agt wind obj balloon)
                      sc (catch agt wind obj balloon)
                      ante (hold agt margie obj balloon
                                 instr string)
                      coord (move1# obj balloon dest tree
            ante* (hit obj1 balloon obj2 branch
                     consq (break obj balloon
                              sc (burst obj balloon)
                              seq (unhappy stateof margie
                                    prec* (cry
                                            agt margie))))))
```

Notice Margie's reaction to the balloon breaking is handled by a combination of sequel and precedent arcs; a sequel of a balloon breaking is that the owner of the balloon is unhappy, and a precedent of 'crying' is 'unhappiness'. Another interesting thing about the "Margie Story" is that its event content coherence representation does not result in a simple tree. For example, the instances of 'hitting' and 'crying', via 'breaking' and 'bursting', intersect at 'unhappiness'. Similarly the concepts of 'carrying' and 'hitting' intersect at 'moving'.

There are two important points to be made about this example. First, because it connects events like 'carrying' and 'bursting', it shows NEXUS' capability for structuring events beyond what can be accounted for by scripts or plans. Secondly, it shows the nature of the difference between a story tree representation and a concept oriented representation; the story tree represents the overall structure of the story by dividing it into categories like setting and episode, and NEXUS represents the events event/state concepts and their implicit connections. We will have more about this in chapter five when we discuss summarizing.

### 2.4.4 A Black and Yellow V-2 Rocket

Simmons' [Simmons 82, Simmons & Correira 80, Correira 80, Levine 80] schema/narrative trees represent a mixture of story trees[2] and schemas. Each subtree is composed of a setting and a sequence of events. Where scripts represent an explicit bridge between events and contexts, schema/narrative trees bridge the gap between schemas and story trees. Consider Simmons' example "A Black and Yellow V-2 Rocket" (abbreviated).

> A great black and yellow V-2 rocket stood in a New Mexico desert. Empty it weighed five tons. For fuel it carried eight tons of alcohol and liquid oxygen. Everything was ready. Scientists and generals withdrew to some distance and crouched behind earth mounds. Two red flares rose as a signal to fire the rocket. The rocket rose. Radar tracked it. It plunged into earth.

Simmons' organizes the text into a flight system schema/narrative tree (see figure 2-4). In this case, the setting of the flight schema includes the text, which describes the size, location, position, weight when empty, type and weight of fuel. For us the interesting thing about this example is that the setting portion of the flight schema is exactly the portion of the text that NEXUS would

---

[2] The term tree is used loosely here to mean a rooted directed graph that may in fact have other connections among its nodes.

```
(FLIGHT TPC ROCKET LOC (DESERT PREP IN DET A LOC (NEWMEXICO)
                         NBR SING)
  SETT (FLTSYSTEM EVT ((ROCKET POSIT STAND
                               LOC (DESERT PREP IN
                           DET A LOC (NEWMEXICO) NBR SING)
                         DET A SIZE (GREAT)
                         COLOR (BLACK *AND (YELLOW))
                         TYPE V-2
                         LGTH (LONG LGTH (FOOT  ...))
           (ROCKET WT (TON QU (FIVE) NBR PL)
                   ST (EMPTY) NBR SING)
           (ROCKET CO (ALCOHOL WT .....))
  SEQ (ASCENT TPC ROCKET FROM (DESERT ...)
              SETT ...
        EVT ((RISE TPC ROCKET ...
```

Figure 2-4: A Narrative Schema

fail to organize; Nexus successfully accounts for only the sequence portion of Simmons' schema/narrative tree.

```
(flight obj rocket
        subseq (take-off obj rocket
                  prec (stand obj rocket loc (New Mexico desert))
                  ante (signal
                           sc (rise obj (two red flares)))
                  coord (protect ae (and scientists generals)
                           prec (withdraw agt( S & D))
                           subseq(shield ben (S & D) obj earth-mounds
                                    sc (crouch
                                           behind earth-mounds)))
                  subseq (fire obj rocket)
                  subseq (ascend obj rocket
                                 sc (rise obj rocket))
        coord (track instr radar object rocket)
        subseq (descend obj rocket
                    sc (plunge obj rocket dest earth))
```

One of the things discussed in the introduction was that the packets of information which NEXUS produces can be organized into story trees. Simmons' work shows the nature of the bridge which combines story structure information with sequence (event coherence) information.

### 2.4.5 Speech Acts

Recent approaches [Allen 79, Cohen and Perrault 79, Cohen, Perrault, and Allen 81] to question-answering have attempted to treat the interactions between man and machine as speech acts [Austin 69, Searle 69, Searle 76]; for a computer system to engage successfully in a useful dialogue with a human it must have the capability to go beyond the explicit meaning of the user's utterance and infer implicit plans and goals. Allen [Allen 79] has written a dialogue system that provides gate information for train travellers. His research takes a speech act approach to recognizing the traveller's intentions when s/he makes a request for information. For a request like

When does the Montreal train leave?

his system would produce the answer:

3:15 at gate 7.

Notice that the system infers that the traveller not only wants to know the time that the train is scheduled to leave, but also the location of its departure. Speech acts also occur in narrative text: a modified example from Allen's dissertation is:

John asked Mary, "Can you reach the salt?"
She passed the salt.

NEXUS takes a schematic approach to building a coherence structure which would encompass both John's request and Mary's action: NEXUS has to find a connection between the theme of John's speaking and Mary's action. The connection it should find is that 'reaching the salt' is a antecedent of 'passing the salt'.

```
[pass1 agt Mary rec John obj salt
      prec (request1 agt John to Mary
              thm (reach1 agt Mary obj salt))
      ante (have agt Mary obj salt
              consq* (get agt Mary obj salt
                          ante (reach1 agt Mary obj salt))) ]
```

From a speech act point of view a number of things have been left out, for example that John believes that Mary can reach the salt, and that Mary knows John believes that Mary can reach the salt, etc. The point here is that a speech act can be understood to be coherent without an explicit analysis of the speaker's goals or intentions. NEXUS builds a representation which shows the coherence of John's request and Mary's action, but does not commit itself to an interpretation of John's intention or plan. The packet that NEXUS produces gathers together and structures conceptually related event/state descriptions.

### 2.4.6 Rhetorical Coherence

Chapter one tried to distinguish rhetorical from conceptual coherence. This section will make that distinction clearer by providing some example. We will base our discussion Hobbs' theory of inter-sentential coherence.

Hobbs has described the process which establishes coherence