CHAPTER 2:

THEORETICAL EXPLICATION OF NONMETRIC TEMPORAL PATH ANALYSIS (NTPA)

Introduction to NTPA

NTPA is presented as a measurement theory first by example, before formal definitions are given. NTPA has been defined such that a set of computer programs termed, 'CARTLO', were written to facilitate such analysis. The programs take as primitive for input a set of classifications, each consisting of mutually exclusive and exhaustive categories, relevant to the theory of interest. The programs also require input of observational data collected by using those classifications in a manner prescribed by NTPA so that relations among categories may be investigated empirically. An investigator then queries the observational data for occurrences of patterns of interest--i.e., the relations hypothesized to exist in the data which are consistent with the theory of interest. In this manner, probabilistic measures of relations are derived in NTPA. Each measure is an estimate of the probability or propensity of a specified system pattern or process (i.e., temporal path). Statistical inferences or generalizations can be made from these query results if appropriate sampling strategies are followed during observational data collection.

Fundamental Assumptions about Observation and Data Structure

What an observer tends to notice are changes in his field of perception. These changes take on meaning given his conceptual or theoretical framework and purpose of observing. Suppose, for example, that a meteorologist is studying relations among the season, precipitation, cloud structure, temperature and atmospheric pressure. Each of these concepts can be considered as classifications which can be characterized by changing states or category changes. For example, the season might change from winter to spring, the precipitation from null to rain or to snow, the cloud structure from cumulous to nimbus-stratus or to cirrus, the temperature from 32 to 33 degrees Fahrenheit, and the atmospheric pressure from below 30 to above 30 pounds per square inch. Hence, occurrences of the weather can be characterized by recording changes of states in these classifications of interest.

What is important to note is that these classifications are conceptually independent. Of course, what one usually wants to discover or verify is how those classifications are related to each other. States in different classifications are assumed to coexist in time. That is, different classifications can be used to simultaneously characterize the weather. For example, one might observe that the season is winter and cloud structure is nimbus-stratus and precipitation is snow and temperature is below freezing and barometric pressure is less than 30 units. Each of the classifications takes on a value or category; the classification is in state or category $\underline{X}$ according to the current observation at some point in time. This is measurement in the broadest sense: characterizing an observation of some phenomenon by one or more categories from one or more classifications.

Often some numeric value is typically assigned to an observation of an event relevant to a classification--e.g., the temperature is 30 degrees. But a category or quality could also be assigned, such as just below freezing. The present author takes measurement in its broadest sense as providing information by a characterization of some occurrence with a category in a classification. Such measurement is termed herein, 'systematic observation', in order to minimize the often automatic implication that measurement involves mapping into categories which constitute a number system, because it need not be so restricted. Numbers are only one set of categories that may be used to characterize states of affairs. While numbers have useful mathematical properties, investigators often think about the world with non-numerical categories. Ultimately the measurement must make sense in terms of what is perceived and known. Even though the measurement may be far more precise than what could be perceived by sense alone (by virtue of some instrument which maps values according to some principle onto an indicator that can be directly observed--e.g., a column of mercury, a pointer on a dial), an investigator is still faced with mapping the indicator value into meaningful theoretical concepts.

In the end an investigator wants to find or verify relationships among theoretical concepts. If certain relationships are known to generally hold true, then such knowledge can be used for prediction or explanation. For example, in the summer if the cloud structure changes to nimbus-cumulous and the barometric pressure drops to below 30 and the temperature is above freezing, then it is likely that the precipitation will be rain. Thus, one might decide not to go on an outdoor picnic if the antecedent conditions obtain. Note that this relationship is stated

in words. It is not a mathematical equation (functional relationship), which is one way to symbolize a relationship, such as $E = mc^2$. While there is an elegant parsimony to such functional relationships, they still ultimately make sense in words. For example, the energy resulting from the conversion of mass is equal to the amount of mass in grams multiplied by the square of the velocity of light in centimeters per second. In turn, those words have meaning verifiable in experience (e.g., explosion of atomic bombs).

Basic Assumptions of Systematic Observation

A 'classification' is a set of mutually exclusive and exhaustive categories (states) which can be used to characterize events relevant to the classification. Following are some classifications relevant to metereology:

Classification	Categories
Season of year	Winter, spring, summer, fall
Air temperature	<-50°F, -49°F, ... 120°F, >120°F
Atmospheric pressure	Above 30 p.s.i., below 30 p.s.i.
Cloud structure	Cumulous, nimbus-stratus, cirrus, nimbus-cumulous
Precipitation	Rain, sleet, snow

A singular event is defined as beginning with a change of state (category) of a classification when observing. An event ends with a change of state in that same classification. A joint event is a simultaneous change in two or more classifications during an observation. If

nothing relevant to a particular classification is observable, then it is characterized by a 'null' state.

Specimen 1 is a sample observational record using systematic observation and the above classifications. For example, the event WINTER was recorded as beginning at 12:00 a.m. and never ended during the period of observation (through 5:40 a.m.). Hence, the duration of WINTER for this observation was equal to the length of the observation--i.e., true throughout the record. Once WINTER was recorded as the current state of the SEASON classification, it was never recorded again because there was no seasonal change during the period of observation. On the other hand, the PRECIPITATION classification changed four times during the period of observation (from NULL to RAIN, RAIN to SLEET, SLEET to SNOW, and SNOW to NULL). NULL is an operational way of indicating that there is nothing relevant to characterize in a classification at some point in time--e.g., there are no clouds present or no precipitation evident. The event SNOW occurred once (frequency of 1) in this observational record, beginning at 2:25 a.m. and ending at 4 a.m., for a duration of 1 hour, 35 minutes.

In summary, in the frame of reference of an observer the current state of affairs is characterized by the simultaneously occurring categories in different classifications. A singular occurrence of an event begins with a change of state (category) in a classification and ends with another change of state in that same classification. The duration of that occurrence is the elapsed time from beginning to end. Therefore, when querying such observational data, two fundamental kinds of counts (or measures) are considered: 1) the counting of changes within a particular classification or joint classification of occurrences, which is

Specimen 1. An Observational Record of the Weather Using Systematic Observation

Time	Cloud Structure	Precipitation	Atmospheric Pressure	Temperature of air	Season of year
12:00	(NULL)	(NULL)	ABOVE 30	35°F	WINTER
1:30			BELOW 30		
1:35	NIMBUS-STRATUS				
1:50		RAIN		34°F	
2:00				33°F	
2:20				32°F	
2:21		SLEET		31°F	
2:22				30°F	
2:25		SNOW			
4:00	CIRRUS	(NULL)			
5:00	(NULL)				
5:40			ABOVE 30		

termed, 'event frequency', and 2) the counting of seconds of duration of a singular or joint classification of event occurences which is termed, 'time'.

Making Queries

A query is a question about frequency and time of occurrences or patterns of occurrences in a systematic observation record. As with any descriptive measure of a relationship, the results of a query are assumed to make sense to the investigator on the basis of: 1) other knowledge about how and why the data were collected, 2) the assumptions stated above about the characteristics of a systematic observational record, and 3) how the results of the measurement are derived, to be explained below. The query program cannot detect meaningless or inappropriate questions, although it will detect and report improper syntax of questions. In addition, the results are descriptive of the data. Whatever inferences or generalizations to be made from the data are, of course, up to the investigator and depend upon the design of the study and sampling methods. Following are some sample queries, using the weather observation system described above:

a) IF PRECIPITATION IS RAIN?

b) IF SEASON IS WINTER, THEN PRECIPITATION IS RAIN OR SLEET?

c) IF SEASON IS WINTER AND CLOUD STRUCTURE IS NIMBUS-STRATUS OR NIMBUS-CUMULOUS AND ATMOSPHERIC PRESSURE IS BELOW 30, THEN TEMPERATURE IS 33°F OR 32°F OR 31°F OR 30°F, THEN PRECIPITATION IS SLEET, THEN PRECIPITATION IS SNOW?

d) IF ATMOSPHERIC PRESSURE IS NOT ABOVE 30, THEN CLOUD STRUCTURE IS NOT CUMULOUS OR CIRRUS?

Each query consists of one or more 'phrases'. A phrase always begins with the word 'IF' or 'THEN' and ends with a comma or question mark. A

query must always begin with 'IF' and end with a question mark. The phrase is the fundamental unit of evaluation when a query is performed on observational data. When scanning the observation data file, the truth or falsity of a phrase is evaluated at each relevant datum. A true instance in the data is counted as a 'hit', false instance as a 'miss', and an irrelevant instance is not counted at all.

For example, the first query above (a) would produce the following results, give the observational data in Specimen 1:

FREQUENCY	LIKELIHOOD	TIME(IN SEC'S)	PERCENT TIME
IF PRECIPITATION IS RAIN?			
1 OUT OF 3	.33	1860 OUT OF 20400	9.12

The only classification relevant to this one-phrase query is PRECIPITATION. Therefore, data pertaining to other classifications are irrelevant to this query and ignored. In addition, the NULL code is always ignored when determining frequency, but is not ignored if relevant to a phrase when counting time (because NULL is used to terminate a prior occurrence when there is currently nothing to code in a classification). The query results are interpreted: Given the data in Specimen 1, RAIN occurred once out of a total of three occurrences of some kind of precipitation. Given these data, the likelihood (probability, propensity) that precipitation was RAIN was 1/3. The duration of RAIN was 1860 seconds out of the total time observed, 20400 seconds. The percent time that RAIN occurred was 9.12 (1860*100/20400).

By casually surveying Specimen 1, it should be readily apparent how these results were obtained for the first query. However, it is important to understand the decision making algorithm that the computer program uses. From the point of view of the computer, the observational data are a series of datum/time pairs. Each datum has associated with it the clock time at which it was recorded. For example, the datum WINTER in Specimen 1 began at 12:00 a.m. Likewise, the temperature was recorded as 35°F at 12:00 a.m., the atmospheric pressure as ABOVE 30 at 12:00 a.m., the precipitation as NULL at 12:00 a.m., and the cloud structure as NULL at 12:00 a.m. The next change recorded was when the atmospheric pressure dropped to BELOW 30 p.s.i. at 1:30 a.m. At 1:35 the cloud structure was recorded as changing to nimbus-stratus, and so on.

The query program evaluates each data point in the sequential order entered even though some data may be recorded at the same clock time. The program first determines to which classification the datum belongs and it then updates the current 'context' register with that datum. The context registers contain the current state (category) of each classification--the context information is important when evaluating multi-phrase queries and phrases containing the connecting word, 'AND'.

Next, the program determines whether that datum is relevant to the current phrase of the query being evaluated. 1) If the datum is irrelevant to the current phrase, the program checks first to see whether the datum is relevant to any prior phrases, if any, of the query. If not, it does a time accumulation update, then it looks at the next datum, updates the 'context', etc. (If the datum is relevant to a <u>prior</u> phrase, decisions are made depending on the nature of the query, to be discussed below.) 2) If the datum is relevant to the current phrase, then the

program proceeds to evaluate that phrase as being true or false by checking the state of each classification included in the phrase. The phrase is considered true if and only if a complete match is found between the category or categories of each classification specified in that phrase and their respective states in the data context registers. A partial match or complete non-match is considered as false.

Next the hit/miss accumulators are updated for the current phrase. If the phrase is found to be true in the data at this point, the 'hit count' and the 'total count' are each incremented by one; whereas, if false (a miss), only the 'total count' is incremented by one. If the current phrase of the query is the last phrase and also true, then the time accumulator is updated by incrementing it by the interval found by subtracting the current clock time from the clock time of the next datum. Note also that the time accumulator is automatically updated subsequently as long as the current phrase is true and is the last phrase of the query.

Finally, the cycle being completed, the program looks at the next datum, and the whole decision-making process is repeated until the end of the observational data file is reached. At that time the results are printed, and the program is ready to accept another query.

Results of prior queries are not stored by the computer. While this may appear to be a limitation, the program was designed to answer only the questions asked and no more. Whether the results of a query are what were expected is dependent on the type of query and the data. However, the complexity of a query, other than being limited by syntax rules, is limited only by available computer memory. In most cases, memory limitation need not be of concern. Since the possible number of patterns is

practically unlimited, the program was designed to search for occurrences of any one of them at a time, rather than answer all of a limited set of questions in one pass through the data file.

Examples of Query Calculations

For the query, 'IF PRECIPITATION IS RAIN?', the program looks at the first element (cloud structure is NULL at 12:00), sets the context register element for the classification CLOUD STRUCTURE to NULL, and determines that is is irrelevant to the current and only phrase of this query. The program continues in this manner until the code for RAIN is encountered at 1:50. Note that at this point that the context is: CLOUD STRUCTURE IS NIMBUS-STRATUS AND PRECIPITATION IS RAIN AND ATMOSPHERIC PRESSURE IS BELOW 30 AND TEMPERATURE IS 35°F (not 34°F--yet) AND SEASON IS WINTER. Since the code for RAIN is relevant to the current phrase, it is evaluated and it is true. Therefore, the hit accumulator (=1), the total (=1), and the time interval are incremented. Since the current phrase is the last phrase, the query cannot advance to the next phrase, even though it is true. Now, the next code, 34°F is evaluated and the context updated. This code is irrelevant to the current phrase. However, since the current phrase is the last phrase and it is still true, the time accumulator is incremented by 600 seconds because it is 10 minutes to the next code (33°F at 2:00). The program proceeds similarly until it encounters SLEET at 2:21 (at this point the time accumulator is 1860). Since SLEET is relevant to the current phrase, the phrase is evaluated as false in the data, and only the 'total' is incremented (now = 2). The current and only phrase is not advanced to the next phrase because it is not true that the current phrase is both a hit and not the last

phrase. The program continues past 31°F and 30°F (which are irrelevant) until it gets to SNOW at 2:25. SNOW is relevant, but is evaluated as a miss, and only the hit total is incremented (now = 3). The remaining codes are found to be irrelevant.

Now the program will scan the next observation data file, if present. Note that for each data file the context register is reset to an intitial zero state, but all accumulators are left as they were, when proceeding to continue evaluating the query against a subsequent data file. Since there are no more data files in this example, the program proceeds to print results. Before it does so, likelihoods are calculated for each phrase and the percent time that the last phrase was true is also calculated. Then each phrase is printed with the results of that phrase on the line immediately below. Note that multi-phrase queries will have counts and likelihoods for each phrase, but only time results for the last phrase. Note also that counts and times are printed only once for multiple data files. No subtotals are printed in the current version.

Rules for Query Syntax

Before illustrating the computational process for more complex queries, some basic rules for acceptable query construction are presented. Rules are necessary, of course, to prevent ambiguity when searching data files during query evaluation. Depending on the version or options available on the computer, the investigator may use as elements of a query either classification and category names or only category code numbers which were defined previously to be associated uniquely with categories. In addition, only the following keywords are permitted: IF,

THEN, IS, ARE, NOT, OR and AND. Only two punctuation marks are allowed: commas and a question mark. Commas are used to separate phrases, and the question mark informs the computer of the end of a given query.

The phrase is the fundamental unit of evaluation when scanning the observational data. A query is comprised of one or more phrases. A phrase is comprised of one or more phrase segments. A phrase segment consists of one classification and its category or categories of interest. In Schema 1 the CARTLO query syntax rules are illustrated.

In interpeting the syntax diagrams, words and punctuations in rectangles are required keywords if they lie in a path taken. Words in ellipses indicate that a substitution is to take place at that point in the syntax. What is substituted is either another syntax diagram or defined classification names and respective category names, or defined category code numbers. The syntax rules can be summarized verbally as follows:

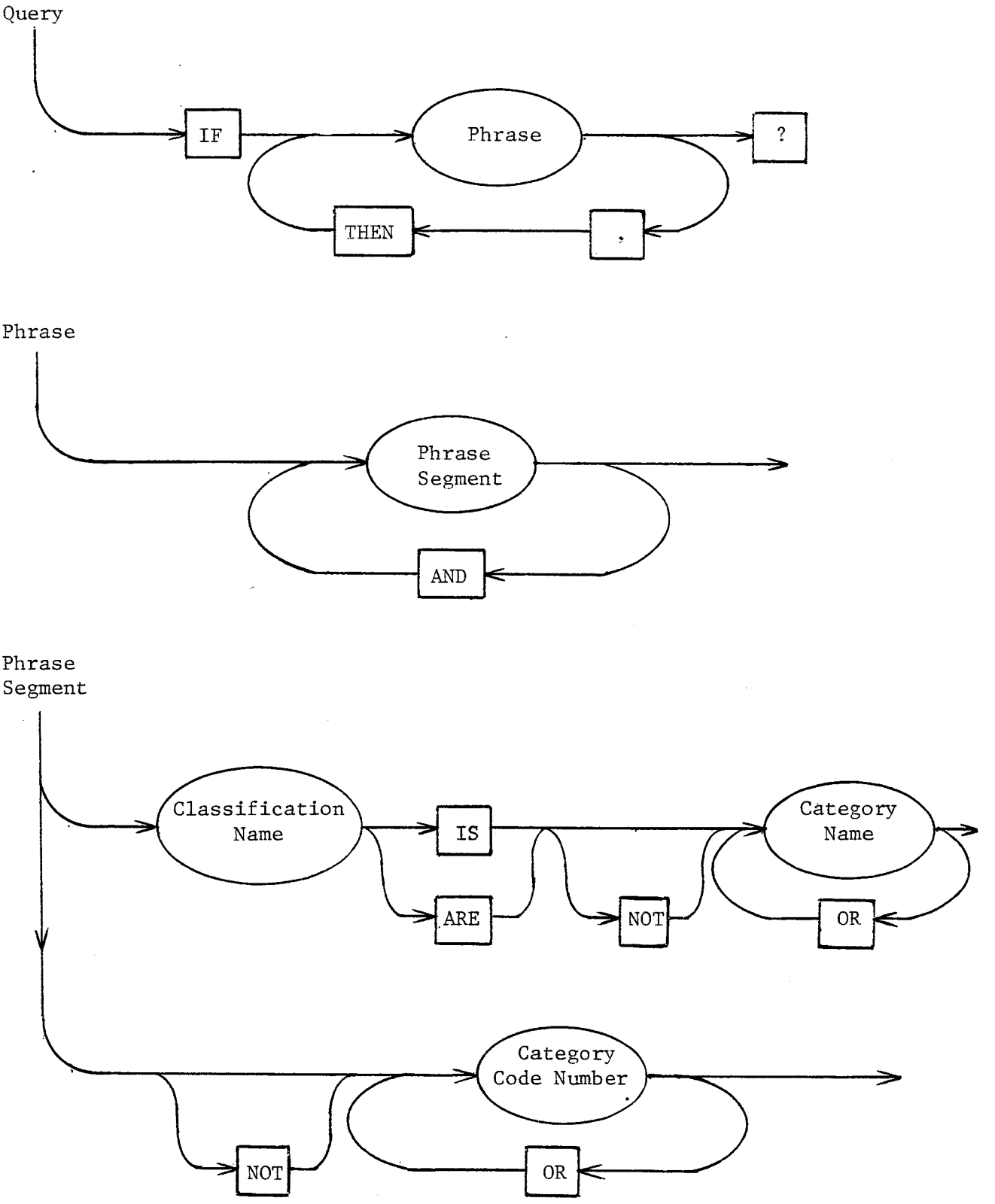
1. A query begins with the word, IF, contains one or more phrases, and ends with a question mark.

2. Subsequent phrases are separated by a comma and each begins with the word, THEN.

3. A phrase consists of one or more phrase segments, each separated by the word, AND.

4. A phrase segment must take one of the following forms consistently: 4.1. A previously defined classification name followed by the keyword, IS (or ARE), followed optionally by the keyword, NOT, which is followed in turn by one or more category names previously defined as members of that classification, each separated by the keyword, OR. 4.2. The preceding keyword, NOT, is optional, which is followed by one or

Schema 1. CARTLO Syntax Rules



more previously defined category code numbers, each separated by the keyword, OR.

A query will only be evaluated against the observation data file if its syntax is valid, all keywords are spelled correctly, and classification and category names or code numbers match identically with those previously defined. Otherwise an error message is generated and the term of the query where the error occurred is displayed. It is up to the investigator to determine how to correct the error. If the same classification is used more than once within a phrase, either an error message will be generated or the results should be suspect because by definition categories within a classification are mutually exclusive and exhaustive (e.g., the season cannot be characterized as both SUMMER and WINTER at the same time).

Evaluation of Query Phrases

1. The truth or falsity of the current phrase is only considered when the current datum is relevant to one of the phrase segments in that phrase.

1.1. A datum is relevant to a phrase segment if it represents one of the categories in the classification specified or implied by the phrase segment.

2. A phrase segment is true if and only if the current state of its respective classification at that point in the data file matches a category code number specified in that phrase segment of the query.

3. A phrase is true if and only if:

3.1. The current datum is relevant to the phrase; and

3.2. All phrase segments of that phrase are true at the point in the data file; and

3.3. All antecedent phrases, if any, were found in the data to be true and have remained true in the order specified by those antecedent phrases.

4. A phrase is false if:

4.1. The current datum is relevant to the phrase; and

4.2. One or more phrase segments are not true at that point in the data file.

These evaluation rules may appear somewhat unwieldy, but they do mirror what most people mean when they use IF, THEN, AND, OR, and NOT in ordinary language.

For example, the empirical likelihood of 'IF SEASON IS WINTER AND ATMOSPHERIC PRESSURE IS BELOW 30, THEN PRECIPITATION IS SLEET OR SNOW?' will be evaluated. Suppose that it is observed that the TEMPERATURE is now 35°F. Can this datum be used to evaluate the truth of falsity of the query? It cannot, because TEMPERATURE is not relevant to this query as stated. Suppose that the SEASON is now observed to be SPRING. Can this datum be used to evaluate the first phrase of the query? It can, since SPRING is relevant to a phrase segment in the first phrase (SEASON IS WINTER). The first phrase is evaluated as false, given the datum, SPRING, because of rules 4.1. and 4.2. Suppose that is now observed to be WINTER, but the barometer is ABOVE 30. Can these data be used to evaluate the first phrase? They can, since each is relevant to the first phrase. However, the phrase is evaluated as false because all segments are not true--i.e., that the ATMOSPHERIC PRESSURE IS BELOW 30 is not true at this time (Rule 4.2.).

Now suppose that during the WINTER that the barometer is observed to be BELOW 30. Can these data be evaluated according to the first phrase of the query? They can, and the phrase is evaluated as true at this point in time because of Rules 3.1. and 3.2. (Rule 3.3. does not apply at this point because there are no phrases antecedent to the first phrase). Now, since the first phrase was found to be true, data at subsequent times relevant to the second phrase (THEN PRECIPITATION IS SLEET OR SNOW) become relevant. Suppose subsequently that it is observed to be RAINing. Is this datum relevant to the second phrase? It is. Is the second phrase true? It is not, since RAIN was not one of the categories of the PRECIPITATION classification specified in this phrase segment.

Assuming that the SEASON and ATMOSPHERIC PRESSURE have not changed, the first phrase is still true though the second phrase has now been evaluated as false. Since the first phrase remains true, data relevant to the second phrase (a change in PRECIPITATION) are still of concern. Suppose that it begins to SNOW. The second (and last) phrase now is evaluated as true, because Rules 3.1., 3.2., and 3.3. have been met. If there were a third phrase to the query, it would be evaluated next, since both the first and second phrases have been found to be true in the data in that order. However, there are no more phrases in this query, so data relevant to this phrase or to an antecedent phrase are of concern. Suppose that the SEASON is now observed to be SPRING. The current point of evaluation regresses to the most recent relevant phrase (the first phrase in this example), which is now evaluated to be false. Data relevant to the first phrase now become of concern, since the data have forced a query phrase regression, and since the phrase evaluation cannot be advanced when a given phrase has been evaluated as false.

A word about NOT. The keyword, NOT, is used to exclude categories in a classification from consideration in a phrase segment. For example, 'SEASON IS NOT WINTER' is equivalent to specifying, 'SEASON IS SPRING OR SUMMER OR FALL'. Note also that the NOT is distributed if multiple categories are specified. For instance, 'SEASON IS NOT WINTER OR SPRING', is equivalent to 'SEASON IS SUMMER OR FALL'. The query program always considers phrase segments positively. It automatically converts the NOT category or categories into their complement in the classification for each phrase segment of this type. Thus, the keyword, NOT, is for convenience in query input. Query results are always stated in the complementary positive.

A word about AND. The counting of hits and misses in a phrase containing at least one AND (i.e., it has two or more phrase segments) needs further explication. In this situation, the number of relevant category changes and the number of different relevant events in the data may not be equivalent. This is due to the fact that each singular or conjoint event is considered to be at a different point in time. Thus, an event is equivalent to each horizontal row in the observation record data listing--e.g., see Specimen 1. Such joint occurrences constitute only one event. In Specimen 1 joint occurrences were coded at 12:00, 1:50, 2:21, and 4:00 a.m. Each of these is considered as a single conjoint event. Thus, there are a total of 12 events (or lines) in Specimen 1, although 19 codes were recorded altogether. Of these 12 events, only some may be relevant to the particular phrase containing AND(s). For example, suppose the query is, 'IF SEASON IS WINTER AND CLOUD STRUCTURE IS NIMBUS-STRATUS?'. There are four events in Specimen 1 which are

relevant to this phrase, only one of which is true. The relevant events occurred at 12:00, 1:35, 4:00 and 5:00 a.m., with the hit at 1:35.

Further examples of query results. A printout of the results of further queries on the data file in Specimen 1 is presented in Table 1.

An Excursion into Recursion

A recursive query is one which has elements of the same classification occurring in different phrases. That is, one or more classifications recur in the query. The elements in different phrases need not be identical as long as they are from the same classification(s). A simple recursive query for the observation of weather in Specimen 1 would be, 'IF PRECIPITATION IS RAIN, THEN PRECIPITATION IS SLEET?'

Since RAIN and SLEET are from the PRECIPITATION classification and occur in different phrases, this query is recursive. The results, given Specimen 1 would be:

	FREQUENCY	LIKELIHOOD	TIME(IN SEC'S)	PERCENT TIME
IF PRECIPITATION IS RAIN,				
	1 OUT OF 3	.33		
THEN PRECIPITION IS SLEET,				
	1 OUT OF 1	1.00	240 OUT OF 20400	1.18

What is important to note is that the same general decision-making algorithm presented above applies. However, by definition of an observation system and the assumptions previously mentioned, certain restrictions obtain because of logical impossibilities. For example, in searching the data file given the above query, the first relevant instance

Table 1. Results of NTPA Queries of Data in Specimen 1

b)

	FREQUENCY	LIKELIHOOD	TIME(IN SEC'S)	PERCENT TIME
IF SEASON IS WINTER,				
	1 OUT OF 1	1.0		
THEN PRECIPITAT IS RAIN OR SLEET,				
	2 OUT OF 3	.67	2100 OUT OF 20400	10.29

c)

	FREQUENCY	LIKELIHOOD	TIME(IN SEC'S)	PERCENT TIME
IF SEASON IS WINTER AND CLOUD STRC IS NIMBUS-STR OR NIMBUS-CUM AND ATMOS PRES IS BELOW 30,				
	1 OUT OF 6	.17		
THEN TEMPRTURE IS 33°F OR 32°F OR 31°F OR 30°F,				
	3 OUT OF 4	.75		
THEN PRECIPITAT IS SLEET,				
	1 OUT OF 2	.5		
THEN PRECIPITAT IS SNOW?				
	0 OUT OF 0	NO DATA	0 OUT OF 20400	0.0

d)

	FREQUENCY	LIKELIHOOD	TIME(IN SEC'S)	PERCENT TIME
IF ATMOS PRES IS BELOW 30,				
	1 OUT OF 3	.33		
THEN CLOUD STRC IS NIMBUS-STR OR NIMBUS-CUM?				
	1 OUT OF 2	.5	11100 OUT OF 20400	54.41

encountered is RAIN at 1:50 a.m., which is evaluated as a hit. Next, an instance relevant to phrase 2 is found for SLEET.

In a non-recursive phrase which is the last phrase, the program would ordinarily continue to look for instances of it in the data as long as the prior phrase is still true. However, in a recursive phrase which is immediately adjacent to the prior phrase (which contains elements from the same classification as the current phrase), it is logically impossible by definition for the current phrase (2) to occur again with the first phrase still having been immediately true. This is due to the assumption that categories from the same classification are mutually exclusive and hence cannot coexist in time. In the data it is assumed that the beginning of a new category in a classification terminates, by definition, the occurrence of the prior coded category. For example, the occurrence of SLEET terminates the occurrence of RAIN. If it is now sleeting, then it can no longer be raining. Obviously, one must be careful to define categories such that they are truly exclusive. Otherwise a new category (e.g., sleet mixed with rain) would need to be added to the classification.

Therefore, the point in the data at which SLEET is evaluated against phrase 2 must be marked as a 'jump back' spot in the data, because it is relevant to a prior phrase. Since it is logically impossible to continue evaluating phrase 2 and since it is the last phrase, the 'jump back' is executed immediately. The phrase to now evaluate regresses to the earliest prior with which it is recursive (i.e., phrase 1 in this example), and instead of evaluating the next relevant data point (which would be SNOW at 2:25), the data pointer is reset to the previously marked 'jump back' point. In this instance it happens to be the data point (i.e., at

2:21) that was just evaluated for phrase 2. However, it is now evaluated for phrase 1 as a miss. Similarly, SNOW at 2:25 is also a miss for phrase 1.

Thus, the same data point was evaluated more than once but for different phrases. SLEET was a hit for phrase 2 but a miss for phrase 1. This is very important to recognize. A strict, non-recursive pattern matching procedure would not perform such an overlapping counting algorithm. A singular data point will be evaluated more than once in NTPA if relevant to different recursive phrases, but never more than once for a given phrase.

As another example, suppose the data were as follows:

SEQ.	TIME	PRECIPITATION	(COMMENTS)
0	00:00:00	NULL	(NOT COUNTED, BECAUSE NULL)
1	00:00:10	SLEET	-P1
2	00:00:20	RAIN	+P1
3	00:00:30	SLEET	+P2, -P1
4	00:00:40	RAIN	+P3, +P1
5	00:00:50	SLEET	+P2, -P1
6	00:01:00	RAIN	+P3, +P1
7	00:01:10	RAIN	-P2, +P1
8	00:01:20	SLEET	+P2, -P1
9	00:01:30	SNOW	-P3, -P1
10	00:01:40	RAIN	+P1
11	00:01:50	SLEET	+P2, -P1
12	00:02:00	RAIN	+P3, +P1
13	00:02:10	NULL	NOT COUNTED

Suppose further that the query is, 'IF PRECIPITATION IS RAIN, THEN PRECIPITATION IS SLEET, THEN PRECIPITATION IS RAIN?'. This three-phrase query is multiply recursive. Phase 3 is recursive with 2 and in turn 2 is recursive with 1. The results would be as follows:

FREQUENCY	LIKELIHOOD	TIME(IN SEC'S)	PERCENT TIME
IF PRECIPITATION IS RAIN,			
6 OUT OF 12	.5		
THEN PRECIPITION IS SLEET,			
4 OUT OF 5	.8		
THEN PRECIPITATION IS RAIN?			
3 OUT OF 4	.75	30 OUT OF 120	25.0

The comments in the right-hand column indicate the phrases evaluated at each relevant datum (e.g., -P1 = miss on phrase 1, +P3 = hit on phrase 3, etc.) for the above query. These recursive counting procedures should be easy to understand in cases of simple recursion. Queries with complex recursion become more difficult and the investigator should exercise some caution because it is easy to make unconscious assumptions about how the counting should proceed and therefore be surprised at the results obtained. Research to date by the author concerning the nature of queries has resulted in the identification of two parameters describing the relationship of a phrase with a prior phrase in a query: proximity and commonality. A particular phrase's relationship to a prior can be partioned as illustrated in Figure 2. These joint parameters must be considered in decision making on whether and where in the data to 'jump back' and to which phrase to regress during a query evaluation, and to do it in such a way as to optimize the counting procedure consistently. When the IF, THEN, AND, OR, and NOT operators can be mixed in a practically infinite number of combinations (given the previously defined syntax rules), one may begin to appreciate the complexity of the decision-making algorithm necessary.

To date no algorithm has been invented which will count all possibilities of recursive queries for all investigator needs. Sometimes one

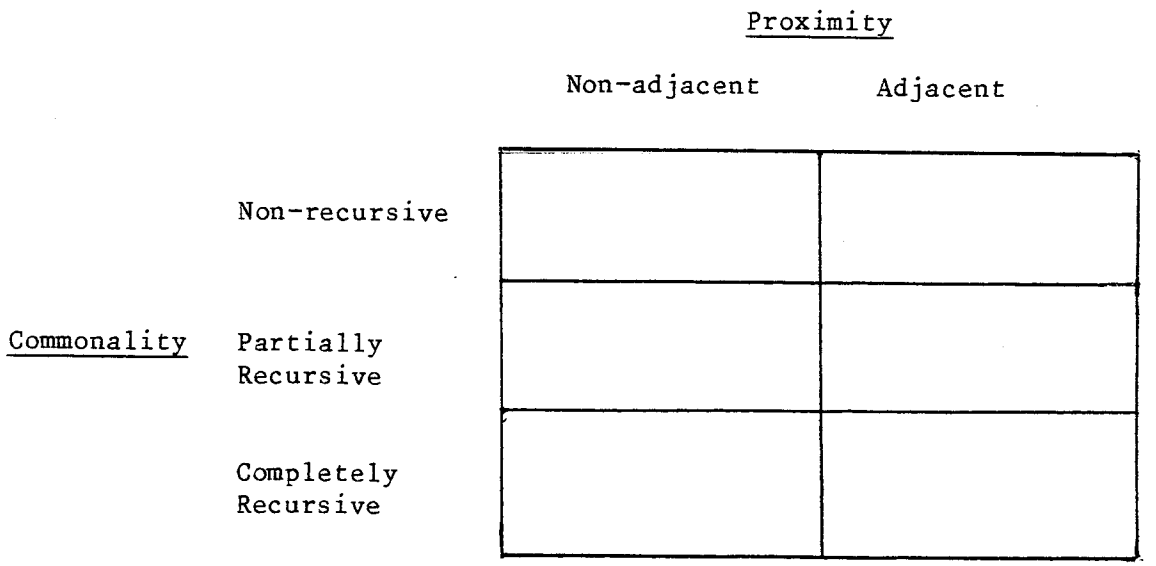


Figure 2. Partitioning of Query Phrase Relationships

decision-making algorithm is preferable for one research problem, whereas the same algorithm is viewed as counterproductive for a different problem. A future version of the query program is planned where the investigator can adapt the algorithm preferred for handling complex recursive queries by using an options specification routine. Otherwise, the current version's algorithm will be in effect, as described below.

Characterizing Queries for Recursive Decisions

Logic dictates some recursive decisions and optimization governs others. The first principle followed is that in a terminating condition of phrase evaluation (i.e., after evaluation, the current phrase is the last one or it is a complete miss; therefore evaluation cannot advance to the next phrase), a decision needs to be made as to whether to evaluate the current phrase again on the next data point or to perform a query regression and/or data 'jump back'.

The problem is further compounded if the current datum is irrelevant to the current phrase but relevant to a prior phrase--i.e., before evaluation of the current phrase. There is no problem if the datum is relevant to the current phrase, and the current phrase is not the last one, and it is evaluated as a complete hit. In this case, the query is advanced to the next phrase and the next datum is considered. A further consideration is the number of 'jump back' pointers which have been previously established in queries which are multiply recursive, as well as the data 'context' at each of those points (i.e., the state of each classification at each 'jump back' point).

Moreover, recursions must uncoil properly. As in a wound clock spring, the spiraling coil cannot cross over itself; otherwise it would

not be able to unwind in the reverse order in which it was initially wound. Hence, subsequent recursions must be nested within prior recursions. That is, as the most recent recursion "unwinds", all prior recursion "windings" cannot be allowed to become entangled during the current unwinding process. Without this first-in-last-out stacking procedure, chaos might prevail. The computer program might never finish the evaluation task because it became stuck in an entanglement indefinitely. This principle of avoidance of entanglement is termed herein as the 'non-crossover criterion'. The major factors to be considered in recursive decision making follow:

1. Evaluate the current datum.

1.1. Is the datum relevant to the current phrase? If so, then update the context registers and go to 3.

1.2. If 1.1. is false, is the datum relevant to a prior phrase? If so, then update the context registers and go to 2.

1.3. If 1.1 and 1.2. are false, then ignore the datum, accumulate time in prior phrase, advance to next datum, and go to 1.

2. Decide what to do when the datum is relevant to a prior phrase.

2.1. Has the phrase regression crossed over the phrase most recently evaluated or are there any 'jump backs' stacked up at this time? If not, then go to 3.

2.2. Else consider the phrase last evaluated and the current phrase. If in this interval inclusively there is a phrase which meets the 'non-crossover criterion' (i.e., legally recursive) or which is an adjacent recursive phrase, then go to 4 (i.e., execute an immediate jump back).

2.3. Else if in this interval inclusively there is a phrase which was previously marked and stacked for a valid recursion, then go to 4.

2.4. Else it must be true that the query regression did not cross over any phrases which were legal recursion points or which were adjacently recursive or which had been previously marked in the data as 'jump back' points. Go to 3.

3. Evaluate the current phrase.

3.1. Evaluate this phrase as a hit or miss. Update accumulators for this phrase. Go to 3.2.

3.2. Is this phrase non-recursive? If not, then go to 3.3. If yes, then if a hit and not the last phrase, then advance to the next phrase and next datum; If yes but a miss, then advance to the next datum. Go to 1 in either case.

3.3. Else is this phrase an illegal recursive phrase? If not, then go to 3.4. If so, then is it a hit and not the last phrase? If yes, then advance to next phrase and datum and go to 1. If so, but a miss or it is the last phrase, then is it a non-adjacent recursive phrase? If yes, then go to 1. Otherwise, execute a 'jump back'--i.e., go to 4.

3.4. Else this phrase is a legal recursive phrase. Therefore, mark the spot in the data and this phrase number and place them on the top of their respective 'jump back' stacks only if the recursion depth has not been exceeded and it is not true that the current phrase crosses over the phrase most recently placed on the stack. In either case, if the current phrase is a hit and not the last, then advance to the next phrase and to the next datum; or if a miss then only advance to the next datum. Go to 1 in either case.

4. Execute a 'jump back'.

4.1. Pull the 'jump back' data pointer from the top of its stack. Pull the 'jump back' phrase pointer from the top of its stack. Update the context registers to what they were at the 'jump back' data pointer. Go to 3.

One consideration ommitted from the above decision-making algorithm is the outcome of an evaluation of a multi-segment (AND) phrase which is both true and false. That is, part of it is true, but not all of it. 'Jump back' decisions can be further elaborated depending on whether the evaluation outcome is completely true, completely false, or partially true. Moreover, the current algorithm considers partially recursive phrases as legal only if the latter is a completely inclusive subset (i.e., proper subset) of the former with respect to their classifications.

Perhaps at this point the reader will begin to appreciate the complexity of decision making necessary for handling recursive queries in an unambiguous manner. Several years of research and testing have been necessary for developing this evolving algorithm. Those experiences have eliminated a large number but not all counting problems which are logically or empirically invalid. One may see the dangers of allowing inexperienced users to set optional parameters in a future version of the program, because of the complex interrelationships of the parameters. Needless to say, certain combinations of decision parameters and data configurations might cause the clock's spring to "sprong" or become hopelessly entangled while unwinding.

Formal Definition of NTPA

Classification, C, is defined as a set of categories, c_1 through c_n, which are mutually exclusive and exhaustive:

$$C = \{c_1, c_2, \ldots c_n\}$$

Categories c_i and c_j are exclusive iff:

$$c_i \ \& \ c_j = \{\Phi\} \quad \text{(for all i,j)}$$

Categories c_1, c_2, ... c_n are exhaustive iff:

$$c_i \ U \ c_j \ U \ \ldots \ c_n = \{C\}$$

The probability (or propensity) of c_i is defined:

$$P(c_i) = \frac{m(c_i)}{m(C)} \qquad [2]$$

The measure function, $m(c_i)$, is defined as the frequency of observed occurrences of events coded by category c_i in classification C. Note that:

$$m(C) = m(c_1) + m(c_2) + \ldots + m(c_n)$$

Also,

$$P(\sim c_i) = 1 - P(c_i)$$

$$P(c_i \ \& \ c_j) = 0 \quad \text{(by definition)}$$

$$P(c_i \ U \ c_j \ U \ \ldots \ c_n) = 1 = P(C)$$

$$P(c_i \ U \ c_j) = P(c_i) + P(c_j)$$

The probability of c_i then c_j is defined:

$$P(c_i,c_j) = \frac{m(c_i,c_j)}{m(c_i)} \qquad [3]$$

The measure function, $m(c_i,c_j)$ is defined as the frequency of observed occurrences of event patterns of the form 'IF c_i, THEN c_j', where c_i occurs at time t_1 and c_j occurs at time t_2 and $t_1 < t_2$, and no other c_n in classification C occurs during the interval t_1 through t_2.

In general:

$$P(c_i,c_j,\ldots,c_m,c_n) = \frac{m(c_i,c_j,\ldots,c_m,c_n)}{m(c_i,c_j,\ldots,c_m)} \quad [4]$$

Assume classifications Cl, C2, ... Cn, where cl_i is a member of Cl, $c2_i$ is a member of C2, and cn_i is a member of Cn. The probability of cl_i and $c2_j$ is defined:

$$P(cl_i \ \& \ c2_j) = \frac{m(cl_i \ \& \ c2_j)}{m(Cl_i \ \& \ C2_j)} \quad [5]$$

The measure function, $m(cl_i \ \& \ c2_j)$, for joint classification is defined as the frequency of the observed joint occurrences of events coded as cl_i and $c2_j$, where cl_i begins at time t and $c2_j$ is also occurring at time t. Note also that:

$$m(Cl_i \ \& \ C2_j) = \Sigma_i \Sigma_j \ m(cl_i \ \& \ c2_j).$$

The $P(cl_i \ \& \ c2_j \ \& \ \ldots \ cn_m)$ can be likewise defined by extending the above definition for two classifications to n classifications.

The time measure function, $tm(c_i)$, is defined as the total duration of observed events coded as c_i in classification C. The duration of c_i, where c_i begins at t_1 and ends at t_2 is defined:

$$d(c_i) = t_2 - t_1.$$

Thus, the total duration of c_i over a period of observation T is defined:

$$tm(c_i) = \Sigma_t d(c_i)_t.$$

Note that tm(C) = T = total duration of the observation = $tm(c_1)$ + $tm(c_2)$ + ... + $tm(c_n)$. It should be also noted that the above definitions of probabilities of nonmetric temporal paths, based on frequency measure functions, can be likewise used for estimating probabilities based on time measure functions, simply by substituting the time measure

functions (tm) for the frequency measure functions (m) in equations [2] through [5]. Thus, probabilities may be estimated in NTPA by either relative frequency, relative duration, or both, depending on the nature of the inquiry.

An event, $E(S_t,C)$, is defined as a change of state of system S relevant to classification C at time t. A joint event, $E(S_t$, C1 & C2 & ... Cn) is defined as a simultaneous change of one or more states of system S relevant to classifications C1, C2, ... Cn at time t.

Systematic observation of S is the mapping of system events relevant to n classifications into their respective categories as they are observed to occur in time. Hypotheses take the form of queries, specified as nonmetric temporal paths, to be verified by analysis of observational data. The results of queries are in the form of estimated probabilities (propensities, likelihoods) of a system's processes. Thus, hypotheses are assumed to be probabilistic and reflect the likelihood of the occurrence of an individual system's processes and/or system-environment transactions, estimated by measures of relative frequency, relative duration, or both. Generalizations across systems of a given type can be made if appropriate sampling strategies are employed. If the systems are independent, then the probability measures derived from observations of each system can be averaged; a population mean and confidence interval can be estimated for the specified nonmetric temporal path by using extant procedures of statistical inference. The unit of analysis on which this mean is based is the measure of the probability of the specified process in each individual system sampled.

General Remarks about NTPA

The measurement theory implicit in NTPA is standard, with the exception that event relations are enumerated. The path (process, pattern) which indicates the relation among factors of interest is itself nonmetric and temporal--hence, the name, 'nonmetric temporal path analysis'. However, the resulting frequency or duration measure derived from observations of a nonmetric temporal path is metric. The measure derived from observations of the relation is numerical, but the relation itself is not. NTPA differs from the LMA in measurement of relations, where the parts of the relation are measured separately in the LMA and the strength of the relationship is estimated by a statistical measure of association. In short, the LMA relates the measures by a linear or curvilinear function, whereas NTPA measures the relation in terms of the uncertainty of its occurrence, expressed as a probability.

The unit of measure in NTPA is based on the occurrence of events or patterns of events characterized by categories in classifications during systematic observation. To obtain a frequency measure, each event or pattern is assigned a weight of one. Thus, a frequency measure is simply an enumeration of occurrences of events or event patterns. This is no different than the measurement of length, for example, where the units of measurement (e.g., inches) are counted when making an observation of the length of some object. The resulting measurement is the frequency of the units of measurement observed. Duration is measured in NTPA using conventional units of time (seconds) elapsed from the beginning of an event or event pattern to its end.

It should also be noted that the formula for conditional probability does not in general hold for NTPA relative frequency estimates, since

there is not necessarily a one-to-one correspondence between event occurrences in different classifications. In other words, the $P(cl_i,c2_j)$ is not equivalent to the $P(c2_j|cl_i)$. However, conditional probabilities may legitmately be constructed from NTPA probabilities based on time measure functions. For example, the $P(c2_j|cl_i)$ can be determined by dividing the $P(c2_j \ \& \ cl_i)$ by the $P(cl_i)$, but only when the latter two probabilities are based on the NTPA time measure (tm) function. Finally, conditional probability should not be conflated with the probability of a sequential occurrence. Conditional probability depends on joint occurrence--it does not matter whether cl_i or $c2_j$ begins first, insofar as both can co-occur. However, the order of occurrence is patently relevant for estimating the probability of a temporal sequence (e.g., $P(cl_i,c2_j)$), regardless of whether a frequency or time measure function is used as a basis of estimation.