U.S. Department
of Transportation
National Highway
Traffic Safety
Administration

Computer & Manua Accident Typing for Bicyclist Accidents

Administrator's Guide

ADMINISTRATOR'S GUIDE

FOR

BICYLIST ACCIDENT ANALYSIS

Department of Transportation
National Highway Traffic Safety Administration
400 Seventh St., S.W.
Washington, DC 20590

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

THE PURPOSE OF THIS GUIDE

This guide provides guidelines and procedures for classifying and analyzing bicyclist-motor vehicle accidents. The approach described herein is part of a systematic effort by the National Highway Traffic Safety Administration (NHTSA) to assist states and local communities in reducing the frequency and severity of these accidents.[1] The accident classification approach is intended for use by state and local government agencies, as well as safety-concerned organizations in the private sector, to aid in problem analysis--the first step in an effective bicyclist safety program.

This guide is intended for use by the person who is responsible for organizing and supervising the conduct of the accident classification and analysis task.

Background

NHTSA's approach to bicyclist accident reduction is a three step process:

1. Analyzing bicyclist accidents in terms of the causal chain or sequence of events.

2. Grouping accidents that have similar causal patterns into "accident types."

3. Developing "countermeasures" (specific ways of intervening) that can eliminate one or more of the events or factors leading to a specific accident type.

The first of these three steps has, thus far, resulted in the definition of 44 accident types. Accident type definitions are based on causal factors and the sequence of events that led to the collision between the bicyclist and the motor vehicle. For example, a "Drive Out - Stop Sign"

---

[1] Bicycle/pedestrian, bicycle/bicycle, and single-bicycle accidents are not covered in this program. Although these are important types of accidents to address, little or no reliable accident data currently exist to support the development of a comprehensive accident-reduction program. The relatively more serious injuries sustained by victims of bicycle/motor vehicle accidents, together with the existence of well documented police records of these accidents, gives a higher priority to bicycle/motor vehicle accidents.

type is defined by the occurrence of several factors and events: The accident occurred at an intersection, the intersection was controlled by a stop sign, the motorist failed to yield to the bicyclist, and the pre-collision courses of the two vehicles were non-parallel.

A list of definitions for the 44 NHTSA accident types is provided in Appendix A.

NHTSA's second step, conducted concurrently with Step 1, used these definitions to classify a great many bicyclist/motor vehicle accidents from several areas of the country to determine the most frequently occurring types, or those with the most serious consequences.

For example, these studies indicated that about 30 percent of the bicyclist/motor vehicle accidents investigated were the result of bicyclist failing to yield to motor vehicle traffic when entering or crossing a roadway. Most of these cyclist victims were 15 years of age or younger, and about 15 percent of these victims died. Older cyclists were more likely to be involved in accidents in which the motorist overtook and struck the cyclist as they were both traveling in the same direction. Although overtaking accidents accounted for only about 15 percent of the accidents, they had serious consequences: Nearly half of them were fatal to the cyclist.

The third step involved the development of specific procedures, called "countermeasures," to reduce the incidence of specific accident types. As part of the development process, countermeasures were tested for their effectiveness in changing bicyclist behaviors or reducing accidents. Some of the countermeasures that have been developed involve engineering changes, such as creating separate bicycle facilities. Others involve passage of new ordinances, such as restricting bicycle traffic on specific routes, or enforcing existing ordinances, such as issuing citations to cyclists who fail to obey stop signs and traffic lights. Still others involve educating various target groups, such as teaching operating skills, enhancing hazard recognition, informing motorists about cyclists' rights, providing guides to road sharing, and so on.

As a result of these efforts, many effective countermeasures are now available to be applied to the reduction of bicycle accidents in jurisdictions across the United States.

Accident Typing--Why and How

As an administrator concerned with implementing a bicyclist safety program, your financial resources are almost certainly limited. You will want to choose those bicyclist accident countermeasures which will have the greatest benefits to your community. Therefore, obtaining knowledge of the frequency of accident types in your own jurisdiction is an essential first step to the effective implementation of specific countermeasures.

In order to facilitate the accident-typing process, NHTSA sponsored a research program which tested the effectiveness and feasibility of several approaches to accident typing. The program, of which this guide is a part, is a result of that research.

Two different accident typing systems emerged. In one system, a coder reads the accident report and extracts information from the report to answer multiple-choice questions about important aspects of the accident, such as where the accident occurred, what the bicyclist was doing, what the motorist was doing, and so on. These answers, in the form of a series of numbers, are fed into a computer which decodes the answers into specific accident types. This system is called "Computer Accident Typing" or CAT.

In the other system, the coder reads the accident report and extracts information about the accident, as in CAT, but instead of writing down answers to specific questions, follows a branching "decision tree" to assign a type number directly. This system is called "Manual Accident Typing" or MAT. In MAT, the coder must not only extract information, but he/she must also follow a logical pattern in assigning the type number.

Overview of Materials

This guide has been prepared for administrators who have chosen either the Computer Accident Typing (CAT) or the Manual Accident Typing (MAT) system. It will be useful for those responsible for implementing and over-seeing the program, as well as for those having direct supervisory responsibilities over accident coders.

This Administrative Guide tells you what you need to know in order to carry out the accident typing process:

- How many accidents you need.
- Which accidents to code.
- How to select and train coders.
- How to carry out the coding.
- How to evaluate the results.

It is not intended to teach you how to code accidents. For that, you must turn to other CAT system materials:

- The Training Manual. This is an illustrated volume which provides detailed self-instruction and guided practice for coding. Different versions of this manual are required for CAT and MAT.
- The Practice Cases Booklet. This volume contains the police accident reports which provide practice in the

coding process. It is used together with the Training Manual. The same booklet is used with both CAT and MAT.

- The Coder's Handbook. This manual provides procedures for typing the accident, as well as supporting explanations and definitions. Different versions are required for CAT and MAT.

In addition to these materials, a supply of data forms will be required. These may be photocopied from the originals provided in Appendix D.

Once the bicyclist accidents have been typed and an accident analysis and summary profile created (see directions in Appendix C), bicyclist accident countermeasures can be selected. Information concerning the various bicyclist accident countermeasures presently available may be obtained by contacting: National Highway Traffic Safety Administration, Traffic Safety Programs, NTS-30, 400 Seventh Street, S.W., Washington, DC 20590.

Each of the materials described in this subsection should be obtained and reviewed by the administrator prior to attempting to implement CAT or MAT. These materials may be obtained by contacting NHTSA at the above address.

OVERVIEW OF ADMINISTRATIVE AND SUPERVISORY TASKS

Successful completion of the NHTSA accident classification process requires attention to several administrative and supervisory activities prior to, during, and after the actual coding of accidents. These activities include:

1. Selecting an appropriate sample of accidents for coding.

2. Evaluating the adequacy, for coding purposes, of the information contained in the reports.

3. Selecting personnel for the coding task.

4. Training the coders.

5. Providing supervision and assistance to the coders.

6. Implementing the computer programs (CAT only).

7. Assessing the needs of your community for specific countermeasures based upon the results of the coding process.

Depending on the administrative structure of your department or agency, you may choose whether one person will be responsible for all of these activities, or whether to divide them between two individuals, an administrator (Activities 1, 2, 3, 6, and 7) and a supervisor (Activities 4 and 5).

The following paragraphs provide a brief overview of each of these activities. Detailed descriptions are given in subsequent sections.

Selecting the Sample

Previous research with accident classification has shown that an accurate indication of significant areas for improvement can be obtained by coding three to five years' accidents or 800-1000 accidents (whichever is smaller), provided that those accidents provide a true representation of that jurisdiction's overall accident pattern. For example, since bicyclist accidents may vary in number and type from season to season, you should look at accidents over several full 12-month periods. Otherwise, the sample you are looking at may reflect unnaturally high (or low) proportions of certain accident types.

Specific procedures for ensuring that the sample you choose will accurately reflect the true proportions and types of accidents your jurisdiction experiences are provided in the section titled "Selecting the Sample."

Evaluating Report Adequacy

In order for accident classifications to be assigned accurately, accident reports must contain certain elements of information. Items found to be necessary include:

1. Pre-collision direction of travel of each operator.

2. Relative pre-crash motion of the two vehicles.

3. Operator errors.

4. Characteristics of accident location.

Not all reports will contain all of the necessary information. Nonetheless, major accident types can still frequently be identified.

In a few jurisdictions, however, report forms do not ask for enough information on which to base accurate classification. In these cases, officers may be given supplemental accident report forms to complete along with the regular forms.

Procedural details for determining the adequacy of reports and for developing supplemental reports forms are given in the section titled "Evaluating Report Accuracy."

Selecting Coders

Probably the single most important factor in determining the accuracy of classification is the person who is assigned to the classification task. The coder is ultimately responsible for picking out the details of each accident which determine the type into which it is classified. If a coder regularly misses some seemingly small details, the accuracy of the classification is seriously jeopardized.

The primary characteristics of an accurate coder are:

- Good reading comprehension.

- Motivation to do a good job.

- Perseverance in the face of a repetitious task.

Procedures for the selection of coders will be given in the section titled "Selecting Coders."

Training Coders

Proper training can make the difference between an ordinary coder and an outstanding coder. The most important elements of proper training are practice and feedback. Without extensive practice, a coder may overlook important but subtle differences in accidents. Without extensive and frequent feedback, a coder may persist in making significant errors in perception or judgment.

Accordingly, a detailed training package has been developed to accompany this program. The training package consists of a Training Manual and Practice Cases Booklet which give:

- An overview of the classification process.
- Definitions and examples of important terms.
- Step-by-step instruction, using several accident reports to introduce the classification process.
- Examples of the variations in accidents which fall into the same type.
- Guided practice with feedback to establish good coding habits.
- Unguided practice with feedback to provide the range of experience necessary to code a wide variety of accident types accurately.

Details on administering the training program are given in the section titled "Training the Coder."

Supervising Coding

A review of several accident reports will convince you that coders will routinely encounter the need to make judgments which will affect the classification of the accident in question. In some cases, the report may contain conflicting information; in others, insufficient information. In some, it may be possible to make reasonable inferences about what happened, even though the report does not specifically provide that piece of information. Such problem cases call for supervisory assistance.

Other important functions of the supervisors are:

- To set work schedules.

- To verify accuracy of coders.
- To resolve disagreements between coders.
- To maintain records and data files.

Details of supervisory tasks are provided in the section titled "Supervising the Coding Process."

Implementing the Computer Programs
(CAT Only)

At or prior to the start of coding, work should begin to implement the CAT computer programs, using the information provided in Appendix E, on your computer system. A few program statements unique to your computer may have to be written to permit the programs to run on your system. The operation of the programs should be checked, using the test data provided in the appendix.

The code numbers for each case are entered in the computer and one program assigns the accident type number. All data may be entered at one time or in batches, so that early results can be generated. The second program compares the coder's judgments on the overlapping cases and indicates the case numbers on which disagreement occurs. The third program tabulates the frequency of each accident type.

Assessing Needs for Specific Countermeasures

Once the accidents have been classified according to the NHTSA system, specific countermeasure needs can be identified. The primary device for assessment of your jurisdiction's needs is the "Bicyclist Accident Analysis Summary and Profile" (provided in Appendix C). The profile provides both a numerical and a visual comparison of the major accident types for which specific countermeasures have been developed. You will prepare a profile of accidents for your community from the tabulations of accident types. The finished chart will show which types occurred most frequently in your jurisdiction during the sample period. With this information, you will be in a position to determine which countermeasures will be most beneficial to implement in your jurisdiction.

In addition to the profile, other types of analyses may be useful. Tabulation of other factors by accident type (such as injury severity, fatalities, age of bicyclist, accident location, and time of day) may assist in the local development of countermeasures tailored to the needs of your jurisdiction.

Suggestions for developing these analyses are provided in the section titled "Assessing Needs for Specific Countermeasures."

SELECTING THE SAMPLE

Since the choice of countermeasures rests on the results of the accident typing process, it is essential that the accidents chosen for classification accurately reflect the distribution of accident types that occur in your jurisdiction. Two factors are significant in this regard:

1. Quantity--you must code a sufficient number of reports to ensure that even the low frequency types (many of which typically result in serious injury) will be detected and their prevalence determined.

2. Normality--the reports must reflect "normal conditions," i.e., the samples must span enough time to "average out" seasonal variations and should not include times during which other factors may have unduly influenced accident rates, especially those involving bicyclists.

The following paragraphs provide the information you need to ensure that the sample you select for coding will give you an accurate picture of bicycle accidents in your jurisdiction.

How Many Accidents Should You Code?

The easiest answer is, of course, "as many as possible." If the number of accidents in your jurisdiction is fewer than 250 per year, then "as many as possible" may literally mean "all accidents" for a given time period. In jurisdictions sustaining more than 250 accidents per year, however, availability of time and funds place severe constraints on the number of accidents that can be coded.

Based on research on coder accuracy and frequency of particular accident types, a jurisdiction wishing to identify low-frequency types with high accuracy should code between 800 and 1,000 reports.

The minimum number of reports that you should code is related to the frequency of accident types that you are concerned about. You can reliably detect major types with a relatively small sample, say 100 reports. Using a sample of 100 reports, an accident type that accounts for 15 percent of the total would be expected to show up in your sample 15 times, plus or minus a few cases, depending on random variations in sampling and coder errors. But, if you were trying to identify low frequency accident types (because of high severity associated with those types, for example), they would be expected in a sample of 100 only once or twice. These accidents could easily be missed, either because they were left out of the sample (due to random selection) or they were misidentified by the coder. Therefore, if you wish to achieve a high level of accuracy with low frequency types, you must code more accidents.

A good rule of thumb that you might use is to select a large enough sample so that the lowest frequency accident type is expected to appear at least ten times. Thus, if you are concerned with accurately assessing types that appear in as few as one percent of the cases, you should select a sample of 1000 accidents. If your only concern is with types occurring greater than 10 percent of the time, then 100 accidents will suffice.

If you cannot muster the desired number of accidents by combining several years' reports, you can still derive the benefit of defining your jurisdiction's major pedestrian accident types. Even with only fifty reports, you can get a good idea about which accident types occur more frequently than others. In this way, you can obtain direct evidence on the general nature of bicycle accidents as they occur in your own community.

While "more is better" is generally true, the benefits that accrue to a community from coding more accidents start to taper off somewhere around 1000 reports. There are situations, however, where a jurisdiction would want to code more accidents. One such situation is the analysis of trends in accident types. To assess accident trends accurately, it is necessary to gather data over at least five consecutive years. In this case, you could obtain a reasonably clear picture of major trends by coding about 500 accidents per year, or a total of 2500 accidents. Furthermore, if you also wanted to test the effectiveness of a given countermeasure, you should code accidents for another four or five years following its implementation and maintenance.

How Many Accidents Can You Code?

The major constraints on the number of accidents that you can reasonably expect to code is coder's time. Previous research with the CAT and MAT approaches has shown that non-professional coders with a moderate amount of training can code between 25 and 35 reports an hour. Therefore, after an initial training investment of eight hours, you can expect the coding of 800 accidents to require between 22 and 32 hours.

Character of the Sample

The sample you ultimately select should be as representative of the accidents in your jurisdiction as possible. If you could code all the accidents in your jurisdiction, there obviously wouldn't be any problems. You would know exactly (within the normal range of coder accuracy) what kinds of accidents happened and how many of them there were. But, if your jurisdiction has either too many accidents to code, or if it has too few in a given year to give an accurate representation of the nature of your bicyclist accident distribution, you need to ensure that the sample you choose will be truly representative.

A good sample must meet the following criteria:

- It must be either all inclusive or the sample must be selected on a random basis.
- It must "average out" seasonal variations and random fluctations.
- It must not cover periods during which abnormal conditions existed.

A sample can easily meet these conditions if you take the proper precautions.

Random Sampling

Many techniques for obtaining a random sample have been developed but some are relatively complicated procedures. However, a random sample can also be obtained by the simple procedure of taking every "nth" report. Say, for example, that your jurisdiction had 1345 accidents during the last five years, and you want to code a sample of about 500 of them. Simply divide the number of accidents by the desired sample size to determine how many reports you should skip over when drawing your sample. In this example, 1345 ÷ 500 = 2.69. Rounding off to the nearest whole number, we find that we should code every third accident. Using this sampling rule, our sample size would be 448, which is reasonably close to our desired sample size.

If, in the above example, we wanted to select about 800 accidents to code, we would run into some difficulty. The outcome of the division procedure would indicate that we should code every second accident (1345 ÷ 800 = 2.46). But, this would result in a sample size of only 673 accidents, considerably fewer than our desired 800. In this case, we use the "nth" item approach to determine which reports should be dropped from the original group of accidents to form the sample. Thus, if we were to exclude every third accident from the sample, we would end up with a sample of 897 accidents. (The number of reports that would be dropped is found by 1345 ÷ 3 = 448. The number of accidents remaining is found by 1345 - 448 = 897.)

A general approach to determining the "n" for the "nth" item procedure is to construct a table of values following the form illustrated below.

Total Number of Accidents (N) = 1345

n	N/n	N - N/n
1	1345	0
2	672	673
3	448	897
4	336	1009
5	269	1076
	If the number is close to desired sample size, select every "nth" item.	If the number is close to desired sample size, exclude every "nth" item.

From this table, we can choose the appropriate value of "n" and whether to select or exclude every "nth" item. For example, if we desired a sample size of 1000, we would take the first three accidents and not the fourth; and so on until we had gone through all of the accidents.

If the total number of accidents is larger than the number you wish to code, reduce it by sampling. If the total number of accidents is smaller than you wish to code, increase it by including accidents from earlier time periods. If the combined total is then too large, use the "nth" item sampling technique to reduce it to the desired size.

Seasonal Variations

Bicycle accidents, like other kinds of accidents, vary with "exposure," that is, the number and kind of interactions between bicyclists and motorists. Child bicyclists are more numerous during the day in the summer months than when school is in session. During school months, child bicyclists are more likely to interact with cars before and after school hours. It is possible that these seasonal variations are reflected in variations in child-bicyclist accident types. Some traffic safety specialists believe that winter bicyclist accidents may have a different character than summer accidents. With more commuters turning to bicycle transportation, one might expect more motorist-overtaking type accidents in those months during which peak commuting hours occur at dusk or in darkness and, although wet weather will deter some bicyclists, those remaining on the road are more vulnerable to loss-of-control type accidents or failure-to-detect type accidents.

If you sample accidents from less than a full twelve-month period, you may unintentionally introduce a bias in your final results towards particular accident types--those that are prevalent during those months which are overrepresented in your sample. Similarly, accident types which are of low prevalence during certain months may be missed in a sample biased by not using twelve-month periods. By using full twelve-month periods, you will "average out" any seasonal variation that might exist in your jurisdiction.

Another form of variation that is important to "average out" is the random fluctation that occurs in numbers and types of accidents from year to year. To counteract these variations, it is advisable to look at the records for more than one year. But, since many communities undergo long term changes that could affect the distribution of bicyclist accident types, it is also advisable not to combine records for too many years. In general, combining records for three to five years will provide enough data to average out the normal variations without introducing confounding variations due to changes over time.

Abnormal Conditions

When choosing your sample, be on the lookout for time periods during which some event occurred which might have had a significant effect on bicyclist accidents. For example, periods of acute fuel shortages should not be included in your sample because bicycle commuting sharply increases and auto usage dramatically decreases as a consequence. The ratio of bicycles to motor vehicles, and therefore exposure, is quite different than under "normal" circumstances. Other events to be aware of are local safety campaigns, long-lasting transit strikes, major changes in traffic laws (such as speed limits), traffic engineering changes, increased school busing, and unusually severe weather conditions.

Assembling the Sample

To expedite the coding process and ensure the randomness of the sample, all accident reports to be coded should be assembled in one place at the same time. Otherwise, a coder might spend as much time trying to locate the appropriate reports as he/she spends coding. Another reason for assembling the total sample is to reduce the chance of sampling errors. Interfering with the sampling procedure, even through oversight or error, may destroy the representativeness of that sample.

In collecting a set of accident reports, take care that they are readable. The more difficulty a coder has in reading a report, the lower the coding accuracy. Avoid using copies of reports if at all possible. If you must use copies, make certain that they are crisp, clean, "first generation" copies. Recopy reports that are dark, dim, run off the page, or are otherwise difficult to read. Your efforts will be rewarded in greatly increased accuracy of coding.

Summary

In general, you should follow these guidelines:

1. Code 800 to 1000 accidents, if at all possible.

2. If the number of accidents permits, code all accidents over a three- to five-year period--otherwise use random sampling.

3. If you use random sampling, choose a minimum sample size so that the lowest-frequency type will be expected about 10 times in the sample.

4. Sample over full twelve-month periods, preferably three to five years. Avoid periods of abnormal conditions.

5. Expect a coding rate of about 30 accidents per hour.

EVALUATING REPORT ADEQUACY

Accident report forms vary widely from jurisdiction to jurisdiction. In most jurisdictions, information about the causal elements in the accident may be recorded in one or more of three distinct areas of the report:

1. Structured report--boxes for checking presence of specific conditions are provided on the report form.

2. Narrative--information may be provided by the reporting officer regarding causal elements; motorist's, bicyclist's and witness' accounts of the accident; road conditions; presence of contributing factors; location of point of impact; etc.

3. Diagram--information may be provided regarding obstacles to vision, direction of travel of bicyclist and motorist, location of accident, etc.

The definitions of the NHTSA accident types often require more information than is typically provided for in the structured portion of accident report forms. The adequacy of reports for coding, therefore, lies in the relative completeness of information recorded by the reporting officer in the narrative and diagram portions of the report form.

In many cases, officers routinely provide outstanding accounts of the circumstances surrounding the accidents. These pose no problems for coding. However, there are situations in which officers fill in the boxes but are able to provide little or no additional information regarding the accident. These may be very difficult to code.

Unfortunately, the more common accident types require more information in order to be accurately classified than do the less frequent types. For example, the "Motorist Backing" type accident accounted for only about one percent of the accidents in one large sample taken from across the country, and it requires only the information that the motorist was backing. On the other hand, the "Bicyclist Ride-Out: Stop Sign" type by itself accounted for about 10 percent of the accidents. This type requires the knowledge that the pre-crash paths of the bicyclist and the motorist were crossing, that the accident occurred at an intersection, that the intersection was controlled by a stop sign, and that the bicyclist failed to stop before entering the intersection. If the report indicated only that the accident location was at an intersection, then the only accident type that could be assigned would be an "Intersection--Other" type, which doesn't provide enough detail about the accident to be useful in selecting countermeasures.

Attempts to code inadequate report forms will cause coders to be greatly frustrated and will provide little, if any, usable data about accidents in your jurisdiction. To avoid wasting resources on inadequate reports, we suggest that you evaluate the adequacy of reports in your jurisdiction prior to major investments in the coding process. If reports are currently not suitable for classification, consider changing report forms or developing supplemental report forms for collecting the required data on subsequent accidents.

Suggestions for evaluating report adequacy and improving accident reports are given in the subsections that follow.

Evaluating Current Reports

There is really only one good method for evaluating report adequacy: Have a trained coder (for example, the supervisor) code a group of accidents and observe the number of "other" or "untypable" accidents. Tabulation of bicycle/motor vehicle accidents from several different samples indicate that the "Insufficient Information" accident categories (Type numbers 55, 98, and 99) account for fewer than five (5) percent of the accidents in the sample.

By coding a representative group of accidents, following the procedures outlined in the coding manual, you will obtain an empirical measure of the adequacy of your jurisdiction's reports. If the combined frequency of 55, 98, and 99 type accidents is less than 10 percent, you need have little concern for report adequacy. However, if the number of accidents coded as these types exceeds 10 percent, you might consider either coding a larger sample[2] or taking steps to improve the completeness of reports.

In order for this assessment method to work, the accidents you code must be "representative" reports, in the same sense as the regular coding sample is representative, and they must be coded accurately. It would do no harm to select a random sample of about 200 accidents from the sample originally selected for coding.

The effort expended in this task has a payoff in addition to evaluating report adequacy: It provides a set of accident codes assigned by the supervisor that can then be used to verify coder accuracy.

---

[2] Research has indicated that if the sample is large enough (1000 reports or more), the profile of accidents (see Appendix C) is not greatly affected by either a fairly high proportion of Untypable/Other classifications (up to 40 percent) or moderately diminished coder accuracy (down to 50 percent interjudge agreement).

Improving Accident Reports

There are two methods to improve the completeness of accident reports:

1. Inform accident investigators and traffic officers of the missing information and encourage them to provide more details in the narrative section of the reports.

2. Develop new accident report forms or "Supplementary" report forms which specifically ask for information that you have found to be missing.

Both of these methods depend on your finding out what information is absent or insufficient for classifying bicycle accidents.

In general terms, the information necessary to code accidents properly falls into four major categories:

- Pre-collision direction of travel of both vehicles.
- Relative pre-crash motion of the vehicles.
- Operator errors.
- Characteristics of accident location.

Within these four categories, of course, there is a wide range of the amount of necessary detail. The minimum level of detail required in order to code the moderate- and high-frequency types is described below:

Direction of Travel. The most essential information about each vehicle's individual motion is whether the vehicle was moving straight ahead or turning. Additionally, it is important to know if either vehicle was standing or if the motorist was backing.

Relative Pre-Crash Motion. It is essential to determine if the primary paths of the two vehicles prior to the collision were parallel or crossing. This can sometimes be deduced from direction-of-travel information, sometimes from the diagram.

Operator Errors. The most important operator error to identify for bicycle accident typing is "Failure to Yield." In this regard, it is extremely important to be fully aware of the laws pertaining to bicyclists' right-of-way when operating in the roadway. Other errors include failure to detect the other vehicle, and loss of control.

Accident Location. There are two levels of important information about location, First, whether or not the accident occurred at an intersection. Then, if so, what kind of traffic-control device was present. If not, did one of the operators enter the roadway from a driveway or alley; and, if so, was it a residential or a commercial driveway or alley.

In general, the more detailed the description of the accident, the more agreement there will be between coders.

After coding the evaluation sample, compare the information contained in reports coded "Other" and "Untypable" with the above list to discover what kind of information is missing. As you review more accidents, patterns will begin to emerge showing areas for improvement.

If the combined frequency of type 55, 98, and 99 accidents was between 10 and 20 percent, briefing officers about the kind of data needed and the reason behind the reports may be sufficient to bring future reports into proper condition. However, if these categories account for more than 20 percent in your sample, then new accident report forms or supplemental report forms should be given serious consideration. These forms should be designed to provide the information which was found to be absent in the review process.

One approach to creating a supplemental form is to adapt items from the "CAT Statement List" provided as Appendix B (regardless of whether you are using the CAT or MAT approach to code the accidents). This list provides a convenient arrangement of all the information need to assign the 44 NHTSA accident types. Your supplemental report forms could consist of the entire list, or just those items which are not currently included on your regular report forms.

Regardless of the method you choose to gather more complete accident data, you may find it useful to review the "CAT Statement List" with field officers. With this process, they may gain more appreciation for the kind of additional information you are seeking.

SELECTING CODERS

The nature of the accident classification task requires that particular attention be given to the selection of the coders. Specific knowledge, motivational, and personality factors have been found to be of great importance in obtaining accurate coding.

Characteristics of a Good Coder

The major knowledge or skill factor involved in accurate coding is reading comprehension. A poor reader is not an accurate coder. There is research evidence to support the notion that the better a person can read, the more accurately he/she codes accidents. Poor readers seem to skip essential information items on reports or skip items in the coding procedure, thus assigning a high percentage of erroneous codes. Good readers, on the other hand, seem to be able to pick up small details, make adjustments for poorly copied reports, and overall, get a better grasp of the accident factors which the report is attempting to describe.

A primary motivational factor is the desire to be accurate, to "do a good job." Accuracy levels from coders who frequently say "Oh, well, that's good enough," are simply <u>not</u> good enough. Some reports require the coder to read them over many times to get all the necessary information. A person who is satisfied with a cursory reading of a report will not be as accurate as one who goes over the report to make sure he/she didn't miss some small but significant detail of the accident that might change its classification.

The most significant personality factor is perseverence; the ability to stick with a task which, even though repetitious, must be done and done well. Persons who are easily frustrated by missing data, hard-to-read reports, ambiguous cases requiring firm decision, and so on, will not make good coders.

Choosing a Good Coder

Deciding who to assign to the coding task is not easy. The trade-off between getting the best person for the job and getting someone you can afford can be a real problem.

Without administering specific psychological tests (which in themselves have many drawbacks), how can you tell who would make a good coder?

An easy answer is to choose professional staff rather than non-professional staff. Generally speaking, professionals have already demonstrated their ability to comprehend what they read, their motivation to do a good job, and their general tendency to stick to a job until it is successfully completed.

If professional staff are not available for this type of activity, then more specific selection criteria must be employed. The following guidelines may be used in the selection process:

1. Seek a person with as high an educational attainment as possible. There is a positive correlation between level of education and reading comprehension. Furthermore, a person who has pursued an education in addition to maintaining full-time employment frequently has a good measure of motivation.

2. A prior work history showing a "perfectionistic" attitude towards the final products is desirable, especially if those tasks involved recording and tabulating numerical data.

3. Select a person who has demonstrated the ability to follow instructions carefully and work methodically. People who routinely take "short cuts" will be prone to error in the accident-typing task.

In addition to these three guidelines, there is one other consideration to take into account. All other things being equal, a coder who is familiar with the report forms and the accident locations will probably be more accurate than one who is not. Coders without prior familiarity with the accident report form will soon learn where on the form they will find the information they require. However, it is helpful if the coders have some prior understanding of the rules, definitions and conventions followed by the reporting officer in completing the form. No coder is likely to be familiar from personal experience with all or even a large proportion of the accident locations in your sample. However, a coder who is familiar with the streets and highways in the community can sometimes add details from personal experience which are helpful in coding a report.

How Many Coders?

There are two major reasons for assigning more than one coder to the accident typing task:

1. The number of reports to be classified may be too great for one coder to handle.

2. Using two or more coders allows cross-validation of each coder's work.

The necessity for more than one coder increases as the number of reports in the sample increases. Classifying accidents is a repetitious job, and coder fatigue can be an important source of inaccuracy. In several informal studies, both short-term and long-term fatigue effects have appeared to be significant. Short-term effects show up after coding about fifty reports at one sitting. Coding two "blocks" of fifty reports each seems to be all the classification most coders can comfortably handle in one day. There is also a suggestion of a long-term fatigue effect. After about 200 reports, coders tend to get "bored" with the task and give it less than their full attention. Accordingly, we recommend spreading the task over a longer time period, or splitting it between several coders.

A more important reason for employing two or more coders is to provide a system for maintaining high levels of accuracy. If, for example, a jurisdiction wishes to code 800 accidents, two coders could be given 450 reports each. One hundred of the accidents would thus be coded by both coders, allowing for cross-validation and verification of each coder's work.

Summary

Generally speaking, the best coders will have the following characteristics:

1. High reading comprehension.
2. Desire to do a good job.
3. Ability to stick to repetitious tasks.
4. Familiarity with report forms and accident locations.

Whenever possible, the accidents to be classified should be divided up between two or more coders, with each coder classifying some accidents also classified by the other coders.

TRAINING THE CODERS

Thorough training is essential for obtaining high levels of coder accuracy. Coders must learn and practice the coding task and obtain feedback concerning how well they are doing. Because accident reports sometimes contain unclear, contradictory, or missing information, and because a given accident may appear to fit into more than one type, coders must learn specific definitions and conventions to help them standardize their decision-making. Also, coders must learn to avoid a few common errors identified during the development of CAT and MAT. These represent the goals of coder training.

Administering the Training

There are several aspects of administering the training program that require some consideration:

- Preparation.
- Conducting the training.
- Facilities.
- Length and spacing of training session(s).

Preparation

The following activities should be performed in preparation for training:

1. If not already decided, determine who will supervise the training and coding. The same individual should perform both functions.

2. Have the supervisor take the training course him/herself, including the completion of all practice cases. The supervisor will need to complete the training in order to evaluate report adequacy, as discussed previously.

3. Schedule the training, using the guidelines provided later in this subsection.

4. Assemble the training materials. The following materials will be required for each coder:

    a. A copy of the Training Manual.

    b. A copy of the Coder's Handbook.

    c. One Data Form (photocopied from the original in Appendix D).

    d. A copy of the Pracice Cases Booklet.

    Note that there are different versions of the first three items, depending upon whether you are training the CAT or MAT approach. The same Practice Cases Booklet is used for both approaches.

Conducting the Training

The Training Manual is designed to be self-instructional and self-paced. That is, the coder can "work through" the manual, making use of the cases provided in the Practice Cases Booklet at the appropriate points, at his/her own rate and without input from the supervisor. However, this is not the best training approach.

The recommended approach is to have alternate periods of self-instruction and group discussion. Where only one coder is being trained, the "group" should consist of the coder and the supervisor. Where more than one coder is being trained, the group should consist of all coders together with the supervisor.

The Training Manual is divided into five sections. Each section contains content presentation followed (except in Section Four) by the opportunity to code one or more practice cases. These sections are completed by the coders working independently (either in separate locations or in the same room).

However, once all coders have completed the practice case(s), they should be brought together for group discussion. The discussion process is one of the best methods through which coders may refine their interpretative skills and increase their sensitivities to small but significant details on the reports.

The group discussion should proceed as follows:

1. Each coder is asked to indicate the code he/she assigned to each practice case.

2. Where disagreements occur, coders should justify their decisions, based on re-examination of the accident

report and the manual. The group should agree on the best code for each case.[3]

3. The correct answer and accompanying discussions, as contained in the manual, should then be reviewed and further discussion encouraged if the group decision and the correct answer are different.

The self-instructional segment, and the group discussion which follows, together make up a training session. The CAT and MAT training should be divideed into four (4) such sessions as follows:

Session 1. The coders read Section One in the Training Manual and code one accident report in the Practice Cases Booklet. In group discussion, this practice case is reviewed as discussed above, and the coders read the correct answer and discussion contained in Section Two of the Training Manual.

Session 2. Working independently, the coders code two accident reports from the Practice Cases Booklet as instructed at the end of Section Two. These cases are reviewed in the group discussion, and the coders read the correct answers and discussion contained in Section Three. Allow time for questions and further discussion following the reading of Section Three.

Session 3. The coders complete the reading of Section Three and code the five accident reports as instructed. The answers are discussed in the group, and the coders read Section Four, which discusses the correct answers for each case. Again, a question and discussion period should follow the reading.

Session 4. The coders read Section Five, then code the remaining 25 reports in the Practice Cases Booklet. The cases are discussed as a group, and then the correct answers are provided, as shown on the last page of Section Five. Coders who code fewer than 16 of the 25 reports correctly should review the Training Manual, working on their own, and recode the cases missed.

Facilities

The specific requirements for training facilities will vary with the size of the group and whether or not the self-instructional segments are

---

[3] Actually, using the CAT system, a coder produces for each accident a "case code" containing (usually) eight digits. The computer uses this case code to generate the two-digit accident type code. For CAT training, the case codes are compared rather than the accident type codes.

conducted with all the coders in one location. In general, however, training facilities should:

- Be well-lighted, comfortable.
- Be relatively isolated from distraction.
- Provide sufficient table space for each coder to spread out reports, manuals, coding sheets, etc.
- Provide a chalkboard or flipchart for group discussion.

Attention to each of these aspects will enhance training by allowing the trainees to focus on the training task instead of diverting their attention to environmental factors.

Length and Spacing of Training Sessions

The total time required to complete the training will depend upon the capabilities of the coders, the number of coders, and the skill of the supervisor in leading the group discussions. However, a minimum of eight (8) hours will likely be required for training, divided as follows:

	Self-Instruction	Group Discussion	Total
Session 1	1 hour	1/2 hour	1-1/2 hours
Session 2	1/4 hour	3/4 hour	1 hour
Session 3	3/4 hour	1 hour	1-3/4 hours
Session 4	1-3/4 hours	2 hours	3-3/4 hours
		TOTAL	8 hours

While all training could be completed in a single day, some advantage is to be gained in dividing the training into two periods, holding each period on a different day. Since there are many details to be learned, the trainee coders will learn more if the training periods are spaced a day apart, thus allowing them to assimilate small chunks of information. More assimiliation ultimately means fewer inaccuracies in coding. If the two-period approach is used, the first three sessions should be performed on day one and Session 4 on the second day.

Deciding When Coders are Ready to Code

By the completion of the training program, coders will have encountered a wide range of accident types. However, the training sample was drawn from a variety of jurisdictions, each of which had a different report form. A good "final exam" for your coders is to code about 100 accidents from your jurisdiction (for example, accidents used in the evaluation of report adequacy). This exercise serves several important functions:

1. It allows coders to gain familiarity with the coding system, using report forms from your jurisdiction.

2. It gives coders the opportunity to code "on their own" without the guidance of the training program.

3. It provides the supervisor with an initial measure of each coder's accuracy (by comparing coders' work with the codes assigned by the supervisor and/or other coders.

The best available measure of coder accuracy is "intercoder agreement," that is, the percentage of accidents to which any two coders have assigned the same accident type number. Coders who agree on 60 percent or more of the cases are coding at an acceptable level of accuracy. (An intercode agreement of 60 percent roughly corresponds to an accuracy level of 70 percent.) Less accurate coders should review selected parts of the training program. Coders whose agreement rate is less than 40 percent at this point in training are probably not good candidates for more training. Consider "scrubbing" them and selecting replacements.

Summary

Proper training requires a trained supervisor, a complete set of training materials, and a distraction-free environment. Training should involve both self-instruction and group discussion and should be distributed over two periods.

At the end of training, coders should be given a "final exam" consisting of a sample of 100 reports drawn from the same pool of reports from which the coding sample was drawn.

Coders are ready to code when they reach at least 60 percent agreement with other coders, including the supervisor.

SUPERVISING THE CODING PROCESS

In addition to the factors listed in earlier sections, coding accuracy also depends on the supervisor's familiarity with the overall goals of the program, the accident type definitions, and the mechanics of the coding process. The tasks of the supervisor include:

- Setting work schedules.
- Answering questions.
- Verifying coders' accuracy.
- Resolving disagreements between coders.
- Maintaining records and files.

These duties are described in detail in the following subsections.

Setting Work Schedules

The optimum work schedule is a balance between the number of reports to be classified in each coding session and the amount of time between sessions. If the coder is required to classify too many reports per session, coder accuracy will decrease due to fatigue. If too much time elapses between coding sessions, accuracy may decrease due to forgetting important definitions and conventions.

Although the exact limits for these factors will vary, depending on report legibility, conditions of the environment, and personal differences between coders, previous experience with the coding task indicates that 50 to 75 reports are all that a coder can be expected to code in a single session. Furthermore, a coder can reasonably be expected to complete only two such sessions per day. The limits on elapsed time are even less well defined. It appears that, if more than two to three weeks elapse between sessions, a "brushing up" on definitions and conventions is warranted before continuing coding sessions. Long intervals, for example six months, would seem to justify a brief retraining session prior to new coding.

The number factor becomes most important when coding a large number of accidents in a relatively short period of time, as would be encountered in classifying a sample of accidents from the last several years.

If, for example, a jurisdiction had 1000 reports to code, and had two coders each coding 600 reports (with a sizeable overlap to establish intercoder agreement), the shortest amount of coding time that should be allowed would be about eight person-days. Each coder, working at maximum capacity, could code 150 reports in two sessions per day. A more realistic schedule, based on one session per day of 50 reports per session, would require 12 sessions per person. In the former case, the coders would not have time to do anything except code. In the latter, however, each session would require only about one and one-half to two hours, leaving time for performing other important duties.

The elapsed-time factor becomes most important when coding a few accidents at a time over long periods of time, as would be encountered in keeping current accidents classified as they occurred. If a jurisdiction had 300 bicyclist accidents a year and wanted to maintain a current record of accident types, it might make sense to code on a monthly or a bi-monthly basis. Although the number of accidents will vary each session, setting up a regular coding time will reduce the problems associated with long intervals between coding sessions. Coders will need only a "brush up" if coding is performed regularly; if too much time goes by, "retraining" may be necessary to maintain accuracy.

Answering Questions

An accident report, by its very nature, does not provide complete details of the events surrounding the accident. It is, after all, a short summary of a complex happening recorded by a person who wasn't present, based on missing or conflicting information. It should not, therefore, be surprising that a coder will have some questions as to how to handle certain situations. It is the supervisor's responsibility to clarify the obscure and explain the confusing. Accordingly, in addition to knowing the definitions of each accident type and the conventions used for resolving conflicts, the supervisor must have a thorough knowledge of how accidents happen. With this knowledge, the supervisor can make reasonable inferences from the information on the report to help the coders assign problem reports to reasonably likely categories.

The best way for supervisors to prepare to answer questions is to code a wide range of accidents themselves. A minimum of 200 accidents is recommended; 400 would be better. In general, the supervisor should code the same accidents as are included in the "overlap" between coders. These may be the same reports that were used to evaluate report adequacy.

In response to some questions, you may want to establish rules or conventions which can be applied in general by all coders. In specifying these conventions, be sure to follow these guidelines:

- Conformance. Your conventions should conform with the coding rules and definitions as presented in this guide, the Coder's Handbook, and the training program.

- Documentation. The conventions you define should be written down and given to each coder. The conventions should be assembled in one document and in a logical sequence as soon as they are defined.

- Consistency. Coders should use the same conventions in evaluating all cases. This implies that if a convention is established part way through the coding process, some cases coded earlier may have to be reviewed and retyped, if necessary.

Verifying Coder Accuracy

In order to evaluate a coder's accuracy intelligently, it is essential that both supervisor and coder understand what "accuracy" means in the context of accident classification. An "accurate" accident classification is one that agrees with the classification assigned by a consensus of experienced coders.

All coders, even "experts," sometimes overlook important information or jump to conclusions about an accident type instead of following strict procedures. Therefore, the only way to determine if an accident type is correctly assigned is for two or more coders to assign codes independently and to examine in detail reports on which they disagree. If coders agree, it may be because they all overlooked the same information. Thus, the more coders that code the same accidents, the more likely you are to generate disagreement, and the more accurate your accident classification will become.

Certainly, there are a few accidents for which conventions have not yet been developed that might fit into more than one classification category. Depending on which elements the coder deems most important, coders may rightly disagree among themselves. However, even though there is some room for legitimate disagreement among coders, most of the cases you will encounter will have a "right answer," that is, a type on which coders can generally reach agreement. With careful attention to detail, strict adherence to procedures, frequent reference to definitions of terms and conventions, and a fair amount of discussion between coders, types assigned by different coders can agree 75 percent of the time or more.

Because discussion between coders and their supervisor is a significant part of the coders' training experience, it is important to check accuracy more frequently early in the coding process, with less frequent accuracy checks later. Thus, most of the "overlapping" accident reports should be given to the coders fairly soon after they begin coding, with the remaining ones spread out over the duration of the task.

At first consideration, accuracy levels in the range of 70-75 percent do not appear to be very good. However, research indicates that the errors that most coders in that range make are distributed randomly across accident categories. Thus, an accident category may "lose" one case due to random errors, but it will probably pick up another due to another random error. Consequently, the overall percentages of the various categories do not differ much from coder to coder. Since countermeasure approaches are selected on the basis of the overall percentages, 70-75 percent accuracy is, for all practical purposes, as good as higher "accuracy" levels.

Resolving Disagreements

When two different coders classify the same accidents, chances are that their assigned type numbers will disagree in about 30-50 percent of the cases they have coded.[4] Some of the disagreements can be attributed to perceptual or reading errors: One of the coders missed a significant piece of information that was contained in the report. Other disagreements can be attributed to procedural errors: One of the coders took a shortcut that caused him or her to overlook a relevant step in the procedure. Some of the disagreements may be due to a coder not understanding the definitions or conventions. Disagreements based on these problems can be resolved rather easily.

Part of the supervisor's quality-assurance effort is to compare the accident types independently assigned by different coders to the same accidents (the "overlapping" part of the sample). If the coders do not assign the same code number to a given accident, the supervisor should review the accident report in question and discuss with the coders what information on the report was overlooked, what term was interpreted differently than as defined, or what convention was not employed. In general, the supervisor should seek a consensus of all coders, and comparisons of the overlapping reports should be conducted on an ongoing basis so that disagreements between coders can be detected as soon as possible. Resolving disagreements provides outstanding opportunities for refining coders' judgments and sharpening their eye for detail. Careful attention to resolution of disagreements can make the difference between "good" coders and truly "expert" coders.

Maintaining Records and Files

The supervisor is responsible for the maintenance of the various records and files that are necessary to complete the accident

[4] Comparing the CAT eight-digit case codes frequntly shows a greater amount of disagreement between coders than comparing accident types. Since no one accident type classification requires all eight digits of the code, coders may disagree on the non-critical (for that type) statements without affecting the accident type number assigned by the computer.

classification task and to determine which of the specific countermeasures should be selected. These records fall into two categories:

1. Accident reports.

2. Accident codes.

In order to maintain the purity of the random sample, it is essential that coders have access only to reports in the sample. Thus, the supervisor should juxtapose him or herself between the coders and the general file of accident reports. If coders can obtain accident reports to code only from the supervisor and the supervisor has only reports included in the sample, then errors due to biased sampling are eliminated.

Furthermore, the supervisor should maintain records of which reports were coded by which coder. This step ensures that a coder will not suddenly discover that he or she is coding a group of accidents that he/she has already coded. Also, the supervisor can keep close tabs on the "overlap" of reports in order to stay current on the resolution of disagreements. A good way to make this kind of recordkeeping easier is for the supervisor to prepare coding sheets in advance by typing in the name of the coder and the identifying number of accidents to be coded. Using this procedure will eliminate coder errors, such as accidently recoding a set of reports or entering incorrect identifying numbers.

Evaluation of coder accuracy demands a listing of assigned accident types by accident-identifying number for each coder. By maintaining this list, a coder's accuracy can be checked at any time by having a second coder classify a sample of the accidents under question.

Finally, in order for the selection of specific countermeasures to proceed, the accident codes must be tallied to provide a count of the number of accidents that were assigned to each accident type. As discussed in the next section, the CAT computer program provides you with this information. MAT users must perform this tally manually.

A number of auxilliary records may also be desirable, depending on what supplemental data analyses are to be employed. Such items as date, time of day, age of victim, locations, etc., may be important elements in developing countermeasures locally.

IMPLEMENTING THE COMPUTER PROGRAMS
(CAT Only)

Inputting the CAT computer programs to your computer should begin enough in advance of coding that your reports can be computer typed without delay once the coders have completed their work. The computer programs, as contained in Appendix E, are relatively simple and may be run on any computer system capable of compiling ANSI Standard Fortran programs.

However, since every computer system has its own job control language and data input/output specifications, your computer personnel will want to review the program listings, as well as the program flow chart (also provided in Appendix E) to help them install the program in your computer. A set of test data is also included in the appendix so that the correct operation of the program can be verified prior to inputting your data.

There are three computer programs provided in Appendix E:

1. The typing program checks to assure that the data format for each case is correct and indicates which cases have improper format. The typing subroutine, which assigns the bicyclist accident code number to each case, is run on each properly formatted case. The case number and bicyclist accident code are output for each case.

2. The comparison program compares the codes assigned to the same accident by two different coders (i.e., the overlap cases) calculates the number of agreements and indicates those case numbers on which the coders disagreed. The typing of up to 10 coders can be compared. The program outputs the percent agreement for all possible pairs of coders, as well as the average agreement across all pairings.

3. The tabulate program calculates and prints out the total frequency and percent of each accident type.

For those cases which are only coded by one coder, you can send the code sheets to your computer facility for input at one time or in several small batches as they are completed. The advantage of the small-batch submission is that the program can be run on each batch as submitted and/or all cases entered prior to and including the batch. The accident tabulation output can provide you with early results. Once all cases have been entered, the program can be run to provide the final output.

The overlap cases require special treatment. The cases you select to be coded by more than one coder should be entered on a separate computer file. The data sheets containing the same case numbers should be entered for each coder, the typing program run, and the comparison output produced. The printout will show those case numbers upon which the coders disagreed, and you can resolve these disagreements with your coders as discussed on page 30. Resolving disagreements will involve changing one or more code values for each case on one of the coder's data sheets. The results of this process will be one corrected set of code sheets for the overlapping cases. This set is then submitted for computer entry on the regular data file (the comparison file can be erased).

As noted previously, you will want to run the majority of your coder accuracy checks early in coding.

ASSESSING NEEDS FOR SPECIFIC COUNTERMEASURES

Selection of specific countermeasures may proceed as soon as the accident sample has been coded, coders' accuracy verified, and the number of accidents are tabulated by accident type. One minor adjustment to the data will simplify the evaluation task:

1. For MAT users, express numbers as percentages of all accidents in the sample. For example, if there were 71 "motorist overtakes undetected cyclist" accidents in a sample of 920 accidents, the percentage would be:

   $$\frac{71}{920} \times 100 = 7.7\%$$

   The CAT computer program provides the percentages.

2. Combine accidents having the same countermeasures into a single group (for example, accident types 1, 2, 3, and 4 can be grouped under one countermeasure: "anti-ride-out training").

Once these adjustments have been made, displaying the results as a bar graph will be helpful in visualizing the extent to which each accident category poses a problem in your jurisdiction. One way to represent a "profile" of your jurisdiction's pattern of bicyclist accidents is to use the "Bicycle Accident Analysis Summary and Profile" form. An example of the use of this form, along with a blank copy for your use, is provided in Appendix C.

As noted on page 4 of this guide, assistance in selecting countermeasures can be obtained from NHTSA. In selecting countermeasures, attention should be given not only to the frequency of accidents but also to accident severity, and the potential for gaining wide community support, funds and personnel available, among other factors.

In addition to preparing a profile of accident frequency, other analyses may be desirable in order to develop countermeasures that are unique to your jurisdiction. A cross-tabulation of accident-types by location may pinpoint high accident locations. This analysis could provide focus for engineering or enforcement needs that otherwise would have gone unnoticed. Examining accident types for age-involvement might provide insight into the bicycle safety needs of specific age-groups (for example, novice adults, children aged 8-11).

Multiple cross-tabulations may also be useful. For example, tabulating high accident locations by age and accident type may reveal specific problems. For example, a given location may show an abundance of junior high children involved in "Ride-out - Midblock" type accidents. Such a situation might be the result of bicyclists crossing a busy street on the way to or from school at a midblock pedestrian crosswalk without activating the walk signal. Examination of the site may reveal that the pushbutton is not accessible to a mounted bicyclist. Relocation of the pushbutton, strengthening enforcement at that location, or instituting bicycle-safety instruction in the school are all suitable countermeasures that could be employed in this specific instance. Without having analyzed the accident data, however, only diffuse, generalized countermeasures could have been applied in this situation.

APPENDIX A

DEFINITIONS OF NHTSA ACCIDENT TYPES

SPECIFIC CIRCUMSTANCES

CODE	TYPE	DESCRIPTION
36	Weird	The accident involved one of the following aspects: . The motorist or cyclist intentionally caused the accident. . The officer indicated no accident actually occurred. . The accident did not involve a cyclist. . The cyclist was struck by falling cargo.
40	Play Vehicle	The cyclist was riding a child's vehicle, such as a "Big Wheel" type tricycle, other tricycle, or a bicycle with training wheels. (But not an adult tricycle.)
11	Backing	The accident involved a motor vehicle which was backing.
29	Non-Roadway	The accident occurred: . In a parking lot or open area. . Other non-roadway location, such as a gas station, alley, lot, etc.

PARALLEL PATH

These types all involve situations in which the initial paths of the cyclist and motorist (prior to any turns which caused or avoided the accident) were parallel. Parallel paths may be either same direction or opposite direction of travel.

CODE	TYPE	DESCRIPTION
THE MOTORIST TURNED OR MERGED INTO THE PATH OF THE CYCLIST		
35	Drive-Out - On-Street Parking	The motorist was exiting or entering on-street parking.
22	Motorist Left Turn in Front of Cyclist	Left, going in the same direction as cyclist.
23	Motorist Left Turn Facing Cyclist	Left, facing each other as approached.
24	Motorist Right Turn	Right, either going in the same or opposing directions.
THE CYCLIST TURNED OR MERGED INTO THE PATH OF THE MOTORIST		
3	Ride-Out From Sidewalk	Onto the street from a residential driveway or alley. Cyclist coming from sidewalk.
18	Cyclist Left Turn, in Front of Traffic	Left, going the same direction as the motorist.
19	Cyclist Left Turn, Facing Traffic	Left, facing each other as they approached.
21	Cyclist Right Turn, From Wrong Side of Street	Right, and the cyclist was riding on on the wrong side of the street.
THE OPERATOR WAS ON THE WRONG SIDE OF THE STREET		
30	Head On, Counteractive Evasive Actions	Either operator was going the wrong way, the approach was head on, and the evasive actions countered each other.
28	Wrong Way Motorist	The motorist was going the wrong way.
26	Wrong Way Cyclist	The cyclist was going the wrong way.

CODE	TYPE	DESCRIPTION

* *

THE MOTORIST WAS OVERTAKING THE CYCLIST

13	Motorist Overtakes Undetected Cyclist	The motorist failed to detect the cyclist.
15	Motorist Overtaking, Counteractive Evasive Actions	The evasive actions were counter-active.
16	Motorist Overtaking, Misjudges Passing Space	The motorist misjudged the space, length, or width required to pass cyclist.
17	Motorist Overtaking Cyclist, Path Obstructed	The cyclist's path was obstructed. Cyclist struck obstruction or over-taking motorist.
39	Motorist Overtaking	Other situations involving a motorist overtaking a cyclist.

THE CYCLIST WAS OVERTAKING A MOTOR VEHICLE

27	Cyclist Overtaking	Cyclist struck a slow or stopped vehicle in a traffic lane.
41	Cyclist Strikes Parked Vehicle	Cyclist struck a vehicle in park-ing lane.

THE OPERATOR LOST CONTROL AND INADVERTENTLY SWERVED INTO THE PATH OF THE OTHER VEHICLE BECAUSE OF ANY OF THE FOLLOWING REASONS:

Mechanical failure, such as brakes, steering, tires, or other vehicle problems.
Road conditions, such as ice, potholes, mud, sand, or other surface conditions.
Prior collision with moving or stationary objects.
Operator impairment due to drugs or alcohol.
Operator error due to oversteering or improper braking.

14	Motorist Lost Control	Motorist loss of control.
20	Cyclist Lost Control	Cyclist loss of control.

CROSSING PATH

These types all involve situations in which the initial paths of the cyclist and motorist (prior to any turns which caused or avoided the accident) were crossing. Crossing paths may be either same direction or opposite direction of travel.

CODE	TYPE	DESCRIPTION
THE CYCLIST DID NOT CLEAR INTERSECTION BEFORE LIGHT TURNED GREEN FOR CROSS TRAFFIC		
6	Trapped	The motorist's view of the cyclist was not obstructed.
7	Multiple Threat	The motorist's view of the cyclist was obstructed by standing traffic.
THE MOTORIST FAILED TO YIELD TO THE CYCLIST		
8	Drive Out, Driveway/ Alley	At a driveway or alley or other midblock location.
12	Drive Through	At a controlled intersection. Motorist ran a sign or signal.
9	Drive Out - Stop Sign	At an intersection controlled by a stop sign or flashing red light, motorist obeyed the sign but failed to yield to cyclist.
10	Right on Red	At an intersection controlled by a a signal, motorist obeyed signal but failed to yield to cyclist.
48	Drive Out - Intersection	At an intersection, situation not covered above.
THE CYCLIST FAILED TO YIELD TO THE MOTORIST, MIDBLOCK		
1	Ride-Out - Residential Driveway	At a residential driveway or alley.
2	Ride-Out - Commercial Driveway	At a commercial driveway.
4	Ride-Out - Midblock	At a shoulder or curb - midblock location. (Cyclist not using driveway.)

CODE	TYPE	DESCRIPTION
* *		
THE CYCLIST FAILED TO YIELD TO THE MOTORIST AT AN INTERSECTION		
5	Ride-Out - Stop Sign	At an intersection controlled by a stop sign or flashing red signal.
49	Ride-Out - Intersection	At an intersection, situation not covered above.
THE MOTORIST WAS TURNING		
33	Motorist Cuts Corner	Left, cut the corner.
34	Motorist Swings Wide	Right, swung out too wide.
THE CYCLIST WAS TURNING		
31	Cyclist Cuts Corner	Left, cut the corner.
32	Cyclist Swings Wide	Right, swung out too wide.
THE ACCIDENT OCCURRED AT AN INTERSECTION		
55	Controlled Intersection, Other	That was controlled by stop signs or signals.
25	Uncontrolled Intersection, Other	That had neither sign nor signal.

APPENDIX B

CAT STATEMENT LIST

CAT STATEMENT LIST

1. Exceptions:

    1. Motorist or cyclist intentionally caused the accident.

    2. Officer indicated no accident occurred.

    3. Accident did <u>not</u> involve a cyclist.

    4. Cyclist was riding a child's vehicle such as a "Big Wheel" type vehicle, tricycle or bicycle with training wheels (not adult tricycle).

    5. Cyclist was struck by falling cargo.

    0. Unknown or none of the above.

2. Motorist primary motion:

    1. Backing.

    2. Stopped, or standing.

    3. Going forward.

    4. Turning left.

    5. Turning right.

    0. Unknown or none of the above.

3. Cyclist primary motion:

    1. Stopped or standing, no direction.

    2. Going forward.

    3. Turning left.

    4. Turning right.

    0. Unknown or none of the above.

4. Operator actions:

    1. Motorist and cyclist attempted evasive actions which were counter-active.

    2. Cyclist attempted to avoid obstruction.

3. Motorist drove out of driveway or alley.

4. Motorist exited or entered on-steet parking.

5. Cyclist on sidewalk entered road from driveway.

6. Cyclist entered road from residential driveway or alley.

7. Cyclist entered road from a commercial driveway.

8. Cyclist entered road over curb or shoulder (midblock).

9. Cyclist did not clear intersection before light turned green for other traffic.

0. Unknown or none of the above.

5. Accident location:

1. Parking lot or other non-roadway.

2. Intersection controlled by a signal.

3. Intersection controlled by a stop sign.

4. Intersection, no control.

5. Intersection, other.

6. Midblock.

0. Unknown or none of the above.

6. Initial approach paths:

Parallel paths:

1. Same direction, cyclist overtaking.

2. Same direction, motorist overtaking.

3. Facing approach.

Crossing paths:

4. Cyclist right-of-way.

5. Motorist right-of-way.

0. Unknown or none of the above.

7. Motorist error:

   1. Ran sign or signal.
   2. Misjudged passing space.
   3. Failed to detect cyclist--stopped or standing traffic.
   4. Swing wide on right turn.
   5. Cut corner on left turn.
   6. Driving on wrong side of the street.
   7. Lost control.
   8. Failed to detect cyclist, other reason.
   0. Unknown or none of the above.

8. Cyclist error:

   1. Ran sign or signal.
   2. Swing wide on right turn.
   3. Cut corner on left turn.
   4. Struck a parked vehicle.
   5. Riding on wrong side of street.
   6. Lost control of cycle.
   0. Unknown or none of the above.

APPENDIX C

BICYCLIST ACCIDENT ANALYSIS SUMMARY AND PROFILE

EXAMPLE OF THE USE OF THE
BICYCLIST ACCIDENT ANALYSIS
SUMMARY AND PROFILE

In order to develop a profile of bicycle accidents, it is necessary to tally the number of accidents in each classification and express those numbers as a percentage of the total number of accidents. Additionally, it is useful to group certain accident types together. These three steps are illustrated in the following hypothetical example.

The Planning Department of Representative County coded all 958 accidents that occurred in the 36-month period starting 1 January 1976 and ending 31 December 1979. The number of accidents that were classified as belonging to each type is shown in Table C-1, along with the percentage of the total represented by each category.

The profile form shown in Table C-2 groups accidents according to seven clases of bicyclist accidents as originally defined by NHTSA[1] with certain additional types included. To complete the profile form, as shown in Table C-2, we perform the following steps:

1. Calculate and enter the counts and percent figures for the seven accident types included in the "Other Low Frequency Types" category. We sum the counts, 2 + 1 + 4 + 2 + 2 + 3 + 2 = 15. The percent figure is obtained by dividing 15 by the total sample 958, which equals 0.0156 or 1.6 percent, rounded off. Alternately, we could have added the percent figures for the seven types.

2. Calculate and enter the count and percent figures for the five "Weird/Insufficient Information" types as in step 1, above.

3. Enter the count and percent figures for the individual accident types in the proper spaces in the profile.

[1]For a discussion of these classes and the accident types as originally defined, see Cross, K. A study of bicycle/motor-vehicle accidents: Identification of problem types and countermeasure approaches. Washington, DC: U.S. Department of Transportation, September 1977. Contract No. DOT-HS-00982. (Available from NTIS)

Table C-1

Representative County:
Number of Accidents by Accident Type

Code	Type	Count	Percent
1	Ride-Out - Residential Driveway	50	5.2
2	Ride-Out - Commercial Driveway	28	2.9
3	Ride-Out From Sidewalk	20	2.1
4	Ride-Out, Midblock	26	2.7
5	Ride-Out - Stop Sign	109	11.4
6	Trapped	28	2.9
7	Multiple Threat	21	2.2
8	Drive Out - Driveway/Alley	50	5.2
9	Drive-Out - Stop Sign	111	11.6
10	Right on Red	16	1.7
11	Backing	9	0.9
12	Drive Through	5	0.5
13	Motorist Overtakes Undetected Cyclist	40	4.2
14	Motorist Lost Control	5	0.5
15	Motorist Overtaking, Counteractive Evasive Actions	19	2.0
16	Motorist Overtaking, Misjudges Passing Space	21	2.2
17	Motorist Overtaking Cyclist, Path Obstructed	23	2.4
18	Cyclist Left Turn, in Front of Traffic	81	8.5
19	Cyclist Left Turn, Facing Traffic	27	2.8
20	Cyclist Lost Control	16	2.7
21	Cyclist Right Turn, From Wrong Side Of Street	9	0.9
22	Motorist Left Turn in Front of Cyclist	12	1.3
23	Motorist Left Turn Facing Cyclist	65	6.8
24	Motorist Right Turn	50	5.2
25	Uncontrolled Intersection, Other	20	2.1
26	Wrong Way Cyclist	31	3.2
27	Cyclist Overtaking	9	0.9
28	Wrong Way Motorist	6	0.6
29	Non-Roadway	8	0.8
30	Head On, Counteractive Evasive Actions	2	0.2
31	Cyclist Cuts Corner	1	0.1
32	Cyclist Swings Wide	4	0.4
33	Motorist Cuts Corner	2	0.2
34	Motorist Swings Wide	1	0.1
35	Drive-Out - On-Street Parking	3	0.3
36	Weird	9	0.9
39	Motorist Overtaking	2	0.2
40	Play Vehicle	2	0.2
41	Cyclist Strikes Parked Vehicle	1	0.1
48	Drive-Out - Intersection	2	0.2
49	Ride-Out - Intersection	9	0.9
55	Controlled Intersection, Other	1	0.1
98	Parallel Path Unknown	2	0.2
99	Intersecting Paths Unknown	2	0.2
	TOTAL	958	100

Table C-2

BICYCLIST ACCIDENT ANALYSIS
SUMMARY AND PROFILE

Jurisdiction *Representative County* Date Prepared *1/9/80*
Prepared by *John Jones*
Covering Period *1 Jan* 19*76* to *31 Dec* 19*79*
Total Accident Cases Analyzed *958*

Class	Code	Type	Count	Percent	Profile 2%	4%	6%	8%	10%	12%
A	1	Ride-Out – Residential Driveway	50	5.2						
	2	Ride-Out – Commercial Driveway	28	2.9						
	3	Ride-Out From Sidewalk	20	2.1						
	4	Ride-Out, Midblock	26	2.7						
B	5	Ride-Out -- Stop Sign	109	11.4						
	6	Trapped	28	2.9						
	7	Multiple Threat	21	2.2						
	49	Ride-Out – Intersection	9	0.9						
C	8	Drive Out – Driveway/Alley	50	5.2						
	9	Drive-Out – Stop Sign	111	11.6						
	10	Right on Red	16	1.7						
	11	Backing	9	0.9						
	12	Drive Through	5	0.5						
	48	Drive-Out – Intersection	2	0.2						
D	13	Motorist Overtakes Undetected Cyclist	40	4.2						
	14	Motorist Lost Control	5	0.5						
	15	Motorist Overtaking, Counteractive Evasive Actions	19	2.0						
	16	Motorist Overtaking, Misjudges Passing Space	21	2.2						
	17	Motorist Overtaking Cyclist, Path Obstructed	23	2.4						
	39	Motorist Overtaking	2	0.2						
E	18	Cyclist Left Turn, in Front of Traffic	81	8.5						
	19	Cyclist Left Turn, Facing Traffic	27	2.8						
	20	Cyclist Lost Control	16	2.7						
	21	Cyclist Right Turn, From Wrong Side of Street	9	0.9						
F	22	Motorist Left Turn in Front of Cyclist	12	1.3						
	23	Motorist Left Turn Facing Cyclist	65	6.8						
	24	Motorist Right Turn	50	5.2						
G	25	Uncontrolled Intersection, Other	20	2.1						
	26	Wrong Way Cyclist	31	3.2						
	27	Cyclist Overtaking	9	0.9						
	28	Wrong Way Motorist	6	0.6						
	29	Non-Roadway	8	0.8						
		Other Low Frequency Types (Sum Types 30, 31, 32, 33, 34, 35, and 39)	15	1.6						
		Weird/Insufficient Information (Sum Types 36, 40, 55, 98, and 99)	16	1.7						

In order to turn the summary into a profile, we find the point in the Profile Section which corresponds to the number in the percent column and fill in the space from the left side of the Profile Section to that point. For example, the percent of "Ride-Out - Residential Driveway" type is 5.2. The point corresponding to 5.2 is slightly more than halfway between the line representing four percent and the line representing six percent. So, we place a vertical line there, and fill in the space between the zero line and 5.2 line which we just drew, forming a horizontal "bar." The other "bars" on the profile are created using the same procedure.

A quick visual scan of the completed profile will reveal which types are most prevalent in the jurisdiction. In this example, "Ride-Out - Stop Sign" (type 5) and "Drive-Out - Stop Sign" (type 9) accidents are most prevalent. "Cyclist Left Turn in Front of Traffic" (type 18) accidents are the third most frequent type. Thus, the profile provides a convenient, and often striking, representation of the results of the bicyclist accident analysis process.

Research has shown that the bicyclist accident typing may vary widely in terms of their severity. That is, some accident types, such as "Motorist Overtakes Undetected Cyclist" (type 13), "Cyclist Left Turn in Front of Traffic" (type 18), and "Ride-Out - Stop Sign" (type 5) may result in more fatalities than the other types. The most frequently occurring accidents do not necessarily have the highest fatality rates. Therefore, in planning a bicyclist safety program, you may also want to emphasize prevention of those types in your community which have the highest fatality rates. To help in this determination, you can follow the process just described to develop a profile of only the fatal accidents in your sample.

A blank copy of the Bicyclist Accident Analysis Summary and Profile is included on the following page for use with your accident sample.

______________ County:
Number of Accidents by Accident Type

Code	Type	Count	Percent
1	Ride-Out - Residential Driveway		
2	Ride-Out - Commercial Driveway		
3	Ride-Out From Sidewalk		
4	Ride-Out, Midblock		
5	Ride-Out - Stop Sign		
6	Trapped		
7	Multiple Threat		
8	Drive Out - Driveway/Alley		
9	Drive-Out - Stop Sign		
10	Right on Red		
11	Backing		
12	Drive Through		
13	Motorist Overtakes Undetected Cyclist		
14	Motorist Lost Control		
15	Motorist Overtaking, Counteractive Evasive Actions		
16	Motorist Overtaking, Misjudges Passing Space		
17	Motorist Overtaking Cyclist, Path Obstructed		
18	Cyclist Left Turn, in Front of Traffic		
19	Cyclist Left Turn, Facing Traffic		
20	Cyclist Lost Control		
21	Cyclist Right Turn, From Wrong Side Of Street		
22	Motorist Left Turn in Front of Cyclist		
23	Motorist Left Turn Facing Cyclist		
24	Motorist Right Turn		
25	Uncontrolled Intersection, Other		
26	Wrong Way Cyclist		
27	Cyclist Overtaking		
28	Wrong Way Motorist		
29	Non-Roadway		
30	Head On, Counteractive Evasive Actions		
31	Cyclist Cuts Corner		
32	Cyclist Swings Wide		
33	Motorist Cuts Corner		
34	Motorist Swings Wide		
35	Drive-Out - On-Street Parking		
36	Weird		
39	Motorist Overtaking		
40	Play Vehicle		
41	Cyclist Strikes Parked Vehicle		
48	Drive-Out - Intersection		
49	Ride-Out - Intersection		
55	Controlled Intersection, Other		
98	Parallel Path Unknown		
99	Intersecting Paths Unknown		
	TOTAL		

BICYCLIST ACCIDENT ANALYSIS
SUMMARY AND PROFILE

Jurisdiction ______________________ Date Prepared ______________

Prepared by ______________________

Covering Period ____________ 19 ____ to ____________ 19 ____________

Total Accident Cases Analyzed ______________________

Class	Code	Type	Count	Percent	Profile						
					2%	4%	6%	8%	10%	12%	
A	1	Ride-Out – Residential Driveway									
	2	Ride-Out – Commercial Driveway									
	3	Ride-Out From Sidewalk									
	4	Ride-Out, Midblock									
B	5	Ride-Out – Stop Sign									
	6	Trapped									
	7	Multiple Threat									
	49	Ride-Out – Intersection									
C	8	Drive Out – Driveway/Alley									
	9	Drive-Out – Stop Sign									
	10	Right on Red									
	11	Backing									
	12	Drive Through									
	48	Drive-Out – Intersection									
D	13	Motorist Overtakes Undetected Cyclist									
	14	Motorist Lost Control									
	15	Motorist Overtaking, Counteractive Evasive Actions									
	16	Motorist Overtaking, Misjudges Passing Space									
	17	Motorist Overtaking Cyclist, Path Obstructed									
	39	Motorist Overtaking									
E	18	Cyclist Left Turn, in Front of Traffic									
	19	Cyclist Left Turn, Facing Traffic									
	20	Cyclist Lost Control									
	21	Cyclist Right Turn, From Wrong Side of Street									
F	22	Motorist Left Turn in Front of Cyclist									
	23	Motorist Left Turn Facing Cyclist									
	24	Motorist Right Turn									
G	25	Uncontrolled Intersection, Other									
	26	Wrong Way Cyclist									
	27	Cyclist Overtaking									
	28	Wrong Way Motorist									
	29	Non-Roadway									
	Other Low Frequency Types (Sum Types 30, 31, 32, 33, 34, 35, and 39)										
	Weird/Insufficient Information (Sum Types 36, 40, 55, 98, and 99)										

APPENDIX D

BLANK CAT AND MAT CODING FORMS

COMPUTER ACCIDENT TYPING
ACCIDENT DATA RECORDING FORM

CODER ______________________ PAGE ______________________ DATE ____________________

REPORT NUMBER	1	2	3	4	5	6	7	8

REPORT NUMBER	1	2	3	4	5	6	7	8

MANUAL ACCIDENT TYPING
ACCIDENT TYPE RECORDING FORM

CODER ____________________ PAGE ____________________ DATE ____________________

REPORT NUMBER	ACCIDENT TYPE	REPORT NUMBER	ACCIDENT TYPE

APPENDIX E

COMPUTER PROGRAMS AND SUPPORT INFORMATION

COMPUTER PROGRAMS AND SUPPORT INFORMATION

Typing (Data Input and Decoding) Program

A LIST for this program is provided in Table E-1. This program requires data lines to be entered in the following syntax:

1. Accident number--may be alphabetic or numeric characters, up to nine (9) characters in length.
2. Comma--delimiter required to separate accident number from accident data.
3. Accident data codes--must be numeric characters, up to eight (8) characters in length.

For example, accident number 84031 was coded 95954829. This would be entered as:

84031,95954829

If there is no comma in the data line, the program prints the accident number and the message "no comma." If there are alphabetic characters in the Accident Data codes, the program prints the message "Character in Data" followed by the data line. In either case, no decoding is performed.

If the data line is entered in the proper syntax, then the program calls subroutine TYPE for decoding the data. The decoded accident type is printed following the accident number. A LIST of subroutine TYPE is provided in Table E-2, and a flow chart is provided as Figure E-1.

The program continues to accept entries and decode data until an "end of file" (control 7) is entered. After the end-of-file has been entered, the program prints the number of cases typed.

The input file is defined in Line 4 of the program, and the printer is defined in Line 5. These two statements are machine specific and will have to be modified for use with your specific computer.

Statement 41 may not be supported in Batch Mode processing, but has become standard for interactive applications.

Sample data inputs and program outputs are provided in Table E-3.

The sample data tests that all of the accident codes have been programmed correctly. The data has the accident code included in the accident number for easy comparison. If all the statements are right, the accident code and the accident number should agree, as shown in the table.

Tabulation Program

A LIST of this program is provided in Table E-4. This program tabulates and provides a sum and percent-of-total for each accident type.

A sample run of this program, using the sample data from Table E-3, is provided as Table E-5.

Coder Comparison Program

A LIST of this program is provided in Table E-6. The program compares two to ten files of accident codes generated by the Data Input and Decoding program. The program prints the accident number in the first column and up to ten columns of accident codes corresponding to the number of files entered. If any disagreement occurs between any two coders for a given accident, an asterisk (*) is printed between the accident number and the first column of accident codes.

At the conclusion of the comparison, the program prints interjudge agreement for all possible pairs of coders and overall interjudge agreement.

This program was written for interactive application and would have to be substantially altered for batch application.

Program Lines 5 and 16 are machine specific. Line 5 assigns the printer to unit number one. Line 16 assigns 1-10 to unit numbers 1-10.

A sample printout of this program is provided as Table E-7.

Table E-1

Typing (Data Input and Decoding) Program

```
FORTRAN IV      V02.1-1                                          PAGE 001
      C  PROGRAM CAT ---C1453 TYPING--C. J. COX, MAY 1980.
      C
      C
             INTEGER CODE,ITEM(8)
             BYTE COMMA, TEMP(8),LINE(20),NUMBER(10),BLANK
             DATA COMMA/',', ICNTER/0/,BLANK/'    '/
      C
      C
      C
      C      ASSIGN FORTRAN LOGICAL UNIT NUMBER 3 TO INPUT FILE
             CALL ASSIGN(3,'INPUT.DAT',9,'RDO')
      C
      C      ASSIGN FORTRAN LOGICAL UNIT NUMBER 6 TO OUTPUT DEVICE
             CALL ASSIGN(6,'OUT.DAT')
      C
      C
      1      DO 25 LOOP=1,10
      25     NUMBER(LOOP)=32
      C
      C
      C          READ IN A RECORD
      C
             READ(3,100,END=99,ERR=333)LINE
      100    FORMAT(20A1)
             DO 150 IMP=1,20
             IF (LINE(IMP).EQ.BLANK)GOTO 175
      150    CONTINUE
             IMP=21
      175    IMP=IMP-1
             ICNTER=ICNTER+1
             DO 200 NUMCHR=1,10
      200    IF(LINE(NUMCHR).EQ.COMMA)GOTO 300
             TYPE 250,LINE
      250    FORMAT(' ',20A1,' NO COMMA')
             GOTO 1
      300    DO 350 LOOP=1,NUMCHR
      350    NUMBER(10-NUMCHR+LOOP)=LINE(LOOP)
             INITEM=IMP-NUMCHR
             ICNT=0
             DO 375 LOOP=(NUMCHR+1),IMP
             ICNT=ICNT+1
             TEMP(ICNT)=LINE(LOOP)
      375    CONTINUE
      C
      C  CHANGE CHARACTER DATA TO INTEGER DATA
      C
             DECODE(INITEM,400,TEMP,ERR=600)ITEM
      400    FORMAT(8I1)
      410    IF(INITEM.EQ.8)GOTO 475
             DO 450 LOOP=(INITEM+1),8
      450    ITEM(LOOP)=-9
      475    CONTINUE
      C
      C
```

Table E-1 (Cont'd)

```
FORTRAN IV      V02.1-1                                          PAGE 002

      C      DATA IN PROPER FORMAT CALL TYPING SUBROUTINE
      C
0038         CALL TYPE(CODE,ITEM)
      C
      C
      C      WRITE OUT CODE
      C
0039         WRITE(6,500)NUMBER,CODE
0040  500    FORMAT(' ',10A1,X,I3)
      C
      C      LOOP BACK TO READ NEXT CASE
      C
0041         GOTO 1
0042  333    TYPE 334,LINE
0043  334    FORMAT(' ERROR ON INPUT',20A1)
0044         GOTO 1
      C
      C   DISPLAY ERROR MESSAGE
      C
0045  600    TYPE 650,LINE
0046  650    FORMAT('  CHARACTOR IN DATA ',20A1)
0047         GOTO 1
0048  99     TYPE 98,ICNTER
0049  98     FORMAT(T5,I5,' CASES CODED')
0050         STOP
0051         END
```

Table E-2

Decoding Subroutine Type
(Used in Data Input and Decoding Program)

```
FORTRAN IV      V02.1-1     SAT 01-AUG-81 00:05:11                PAGE 001

            SUBROUTINE TYPE(CODE,ITEM)
            INTEGER ITEM(9), CODE
      C
      C  WEIRD
      C
            CODE=36
            IF(ITEM(1).EQ.1.OR.ITEM(1).EQ.2.OR.ITEM(1).EQ.3.OR.
           1 ITEM(1).EQ.5)RETURN
      C
      C  PLAY VEHICLE
      C
            CODE=40
            IF(ITEM(1).EQ.4)RETURN
      C
      C  BACKING
      C
            CODE=11
            IF(ITEM(2).EQ.1)RETURN
      C
      C  NON-ROADWAY
      C
            CODE=29
            IF(ITEM(5).EQ.1)RETURN
      C
      C  GO TO CROSSING PATHS, IF CROSSING
      C
            IF(ITEM(6).EQ.4.OR.ITEM(6).EQ.5)GOTO 500
      C
      C  ASSUME PARALLEL PATH
      C
      C  DRIVE-OUT, ON-STREET PARKING
      C
      C
            CODE=35
            IF(ITEM(4).EQ.4)RETURN
      C
      C  MOTORIST LEFT TURN IN FRONT OF CYCLIST
      C
            CODE=22
            IF(ITEM(2).EQ.4.AND.(ITEM(6).EQ.1.OR.ITEM(6).EQ.2))
           1 RETURN
      C
      C     MOTORIST LEFT TURN FACING CYCLIST
      C
            CODE =23
            IF(ITEM(2).EQ.4.AND.ITEM(6).EQ.3)RETURN
      C
      C     MOTORIST RIGHT TURN
      C
            CODE=24
            IF(ITEM(2).EQ.5.AND.(ITEM(6).EQ.1.OR.ITEM(6)
           1 .EQ.2.OR.ITEM(6).EQ.3))RETURN
      C
      C     RIDEOUT SIDEWALK
```

```
      C
0029          CODE =3
0030          IF(ITEM(4).EQ.5)RETURN
      C
      C       CYCLIST LEFT TURN IN FRONT OF TRAFFIC
      C
0032          CODE=18
0033          IF(ITEM(3).EQ.3.AND.(ITEM(6).EQ.1.OR.ITEM(6).EQ.2))
             1 RETURN
      C
      C       CYCLIST LEFT FACING TRAFFIC
      C
0035          CODE =19
0036          IF(ITEM(3).EQ.3.AND.ITEM(6).EQ.3)RETURN
      C
      C       CYCLIST RIGHT FROM WRONG SIDE
      C
0038          CODE=21
0039          IF(ITEM(3).EQ.4 .AND. (ITEM(6).EQ.1 .OR. ITEM(6)
             1 .EQ.2 .OR. ITEM(6).EQ.3) .AND. ITEM(8).EQ.5) RETURN
      C
      C       HEAD-ON COUNTERACTIVE EVASIVE ACTIONS.
      C
0041          CODE=30
0042          IF(ITEM(6).EQ.3 .AND. ITEM(4).EQ.1)RETURN
      C
      C       WRONG WAY MOTORIST
      C
0044          CODE=28
0045          IF(ITEM(7).EQ.6)RETURN
      C
      C       WRONG WAY CYCLIST
      C
0047          CODE=26
0048          IF(ITEM(8).EQ.5) RETURN
      C
      C       OVERTAKING MOTORIST SKIP TO 100 IF NOT OVERTAKING
      C
0050          IF(ITEM(6).NE.2)GOTO 100
      C
      C   MOTORIST OVERTAKES UNDETECTED CYCLIST
      C
0052          CODE=13
0053          IF(ITEM(7).EQ.3.OR.ITEM(7).EQ.8)RETURN
      C
      C   COUNTER ACTIVE EVASIVE ACTIONS
      C
0055          CODE=15
0056          IF(ITEM(4).EQ.1)RETURN
      C
      C   MISJUDGE PASSING SPACE
      C
0058          CODE=16
0059          IF(ITEM(7).EQ.2)RETURN
```

Table E-2 (Cont'd)


```
      C
      C   PATH OBSTRUCTION
      C
0061          CODE=17
0062          IF(ITEM(4).EQ.2)RETURN
      C
      C   OVERTAKING OTHER
      C
0064          CODE=39
0065          IF(ITEM(7).NE.7 .AND.  ITEM(8).NE.6)RETURN
      C
      C   CYCLIST STRUCK A PARKED VEHICLE
      C
0067  100     CODE=41
0068          IF(ITEM(8).EQ.4)RETURN
      C
      C   CYCLIST OVERTAKING
      C
0070          CODE=27
0071          IF(ITEM(6).EQ.1)RETURN
      C
      C   MOTORIST LOSS OF CONTROL
      C
0073          CODE=14
0074          IF(ITEM(7).EQ.7)RETURN
      C
      C   CYCLIST LOSS OF CONTROL
      C
0076          CODE=20
0077          IF(ITEM(8).EQ.6)RETURN
      C
      C   PARALLEL PATHS--UNKNOWN
      C
0079          CODE=98
0080          IF(ITEM(6).EQ.3)RETURN
      C
      C   MULTIPLE THREAT
      C
0082  500     CODE=7
0083          IF(ITEM(4).EQ.9.AND.ITEM(7).EQ.3)RETURN
      C
      C   TRAPPED
      C
0085          CODE=6
0086          IF(ITEM(4).EQ.9)RETURN
      C
      C   DRIVE OUT DRIVEWAY ALLEY
      C
0088          CODE=8
0089          IF(ITEM(4).EQ.3)RETURN
      C
      C   DRIVE
      C
0091          CODE=12
```

Table E-2 (Cont'd)

```
FORTRAN IV      V02.1-1     SAT 01-AUG-81 00:05:11                PAGE 004

0092          IF(ITEM(7).EQ.1)RETURN
      C
      C   DRIVE OUT STOP SIGN
      C
0094          CODE=9
0095          IF(ITEM(5).EQ.3.AND.ITEM(6).EQ.4)RETURN
      C
      C   RIGHT ON RED
      C
0097          CODE=10
0098          IF(ITEM(5).EQ.2.AND.ITEM(6).EQ.4)RETURN
      C
      C   INT.  OTHER
      C
0100          CODE=48
0101          IF(ITEM(5).EQ.5.AND.ITEM(6).EQ.4
         1 .AND.  (ITEM(7).NE.  4 .AND.  ITEM(7).NE.  5 .AND.
         2 ITEM(8).NE.  2 .AND.  ITEM(8).NE.3))RETURN
      C
      C   RIDE-OUT RESIDENTIAL DRIVEWAY
      C
0103          CODE=1
0104          IF(ITEM(4).EQ.6)RETURN
      C
      C   RIDE-OUT COMMERCIAL DRIVEWAY
      C
0106          CODE=2
0107          IF(ITEM(4).EQ.7)RETURN
      C
      C   RIDE-OUT MIDBLOCK
      C
0109          CODE=4
0110          IF(ITEM(4).EQ.8)RETURN
      C
      C   RIDE-OUT STOP SIGN
      C
0112          CODE=5
0113          IF(ITEM(5).EQ.3.AND.ITEM(6).EQ.5)RETURN
      C
      C   INT.  RIDE-OUT
      C
0115          CODE=49
0116          IF(ITEM(5).LT.4.AND.ITEM(5).GT.0.AND.ITEM(6).EQ.5
         1 .AND.  (ITEM(7).NE.4 .AND.  ITEM(7).NE.  5 .AND.  ITEM(8).NE.  2
         2 .AND.  ITEM(8).NE.  3))RETURN
      C
      C   MOTORIST CUTS CORNER
      C
0118          CODE=33
0119          IF(ITEM(7).EQ.5)RETURN
      C
      C   MOTORIST SWING WIDE
      C
0121          CODE=34
```

Table E-2 (Cont'd)

```
FORTRAN IV      V02.1-1     SAT 01-AUG-81 00:05:11                    PAGE 005

0122            IF(ITEM(7).EQ.4)RETURN
        C
        C  CYCLIST CUTS CORNER
        C
0124            CODE=31
0125            IF(ITEM(8).EQ.3)RETURN
        C
        C  CYCLIST SWINGS WIDE
        C
0127            CODE=32
0128            IF(ITEM(8).EQ.2)RETURN
        C
        C  UNCONTROLLED INT.
        C
0130            CODE=25
0131            IF(ITEM(5).EQ.4)RETURN
        C
        C  INT. OTHER
        C
0133            CODE=55
0134            IF(ITEM(5).GT.0 .AND. ITEM(5).LT.6)RETURN
        C
        C  INSUFFICIENT--UNKNOWN
        C
0136            CODE=99
0137            RETURN
0138            END
```

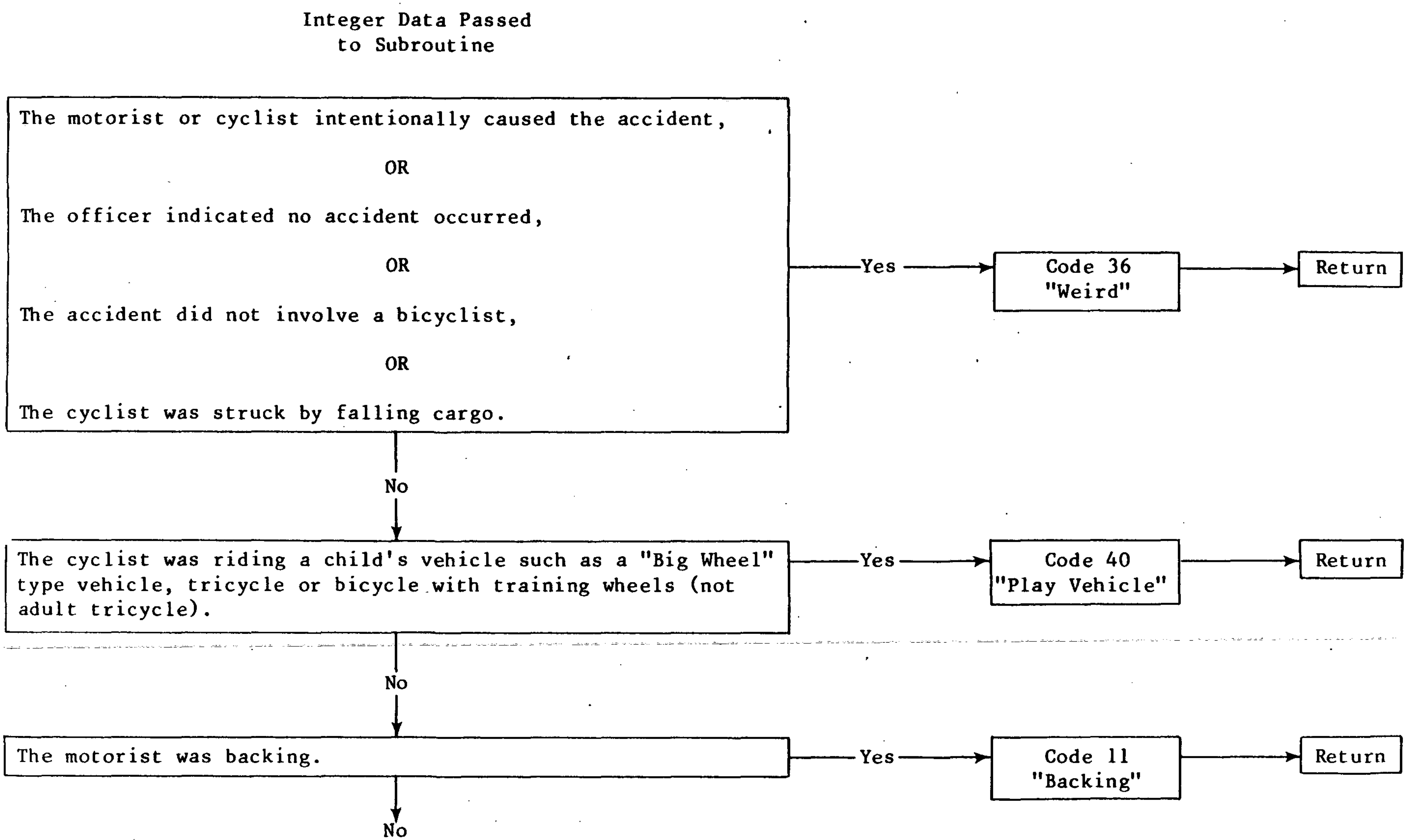

Figure E-1. Accident Typing Subroutine Flow Chart

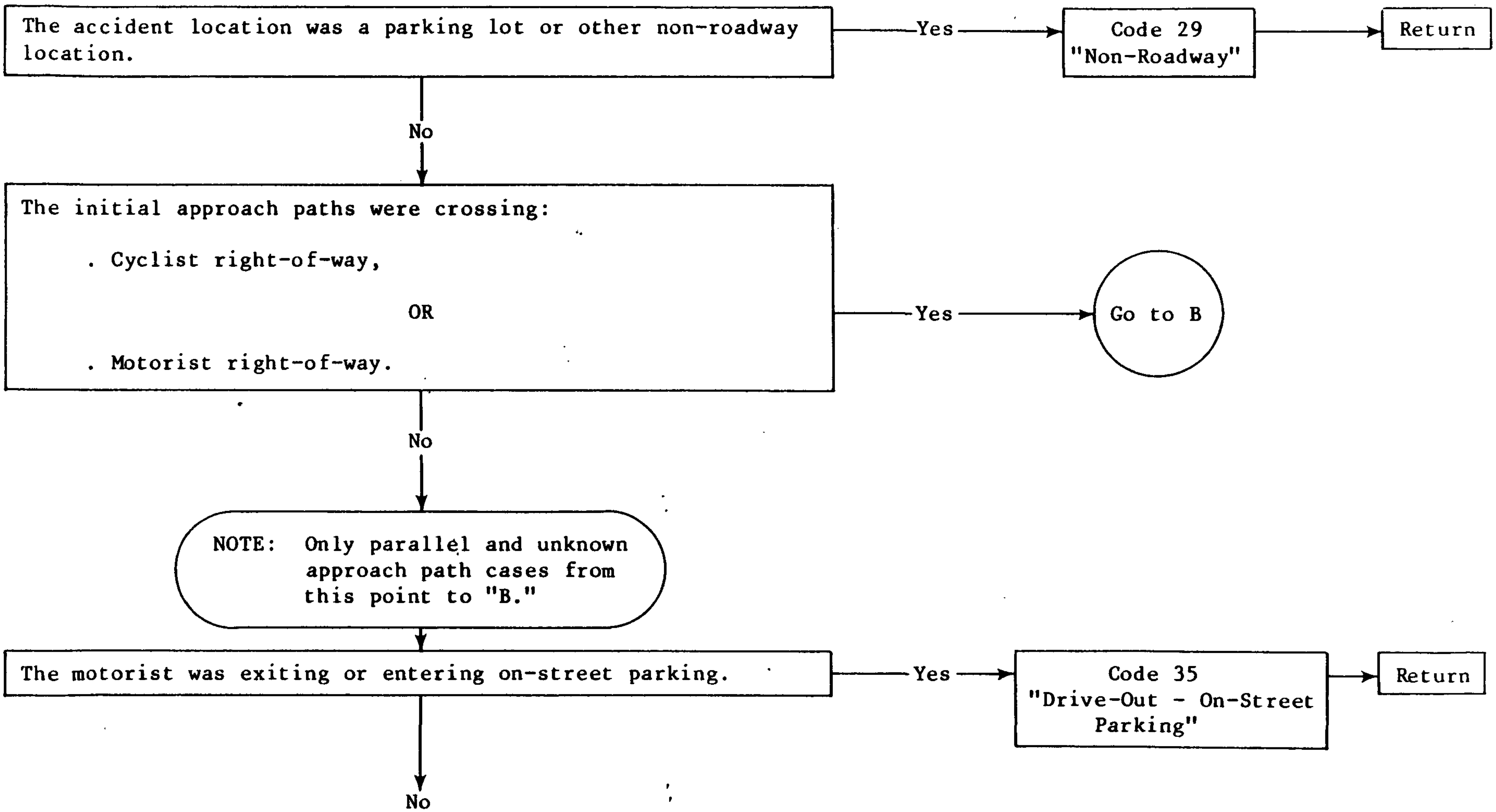

Figure E-1 (Cont'd)

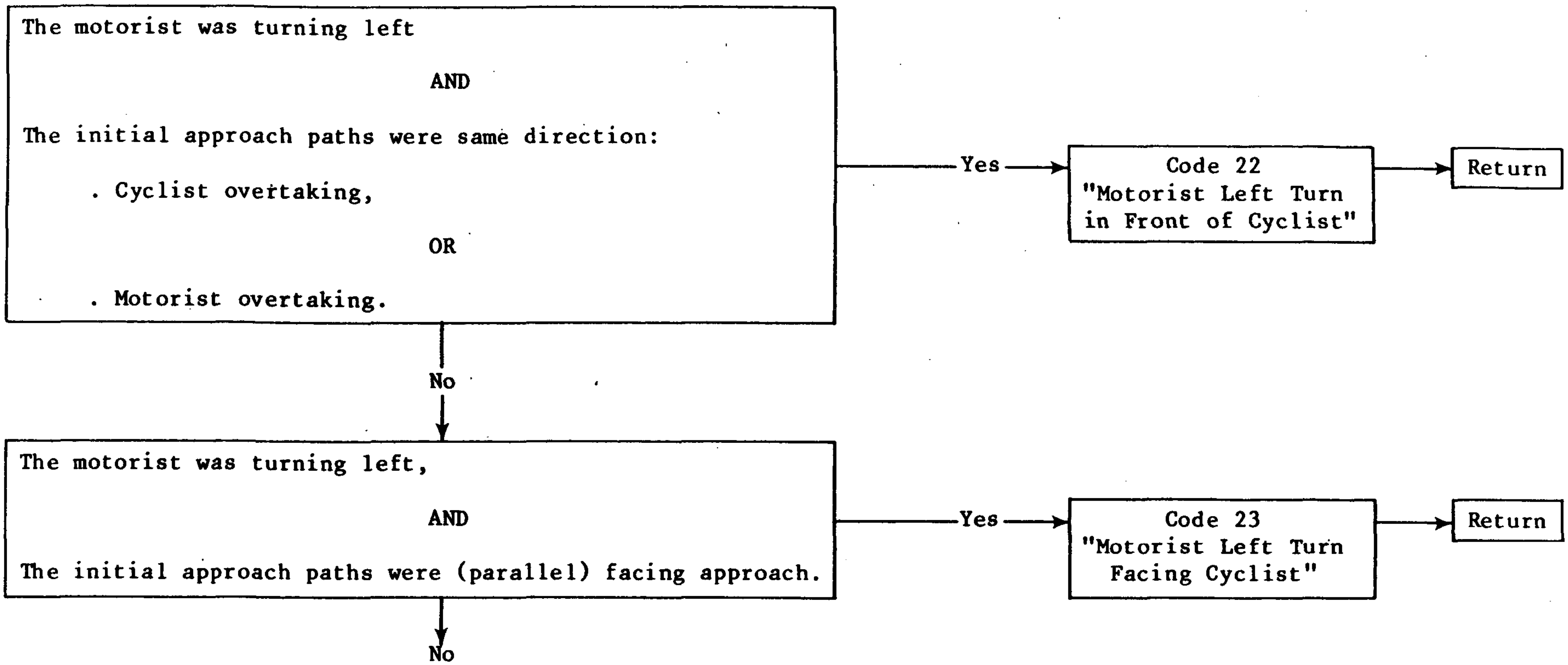

Figure E-1 (Cont'd)

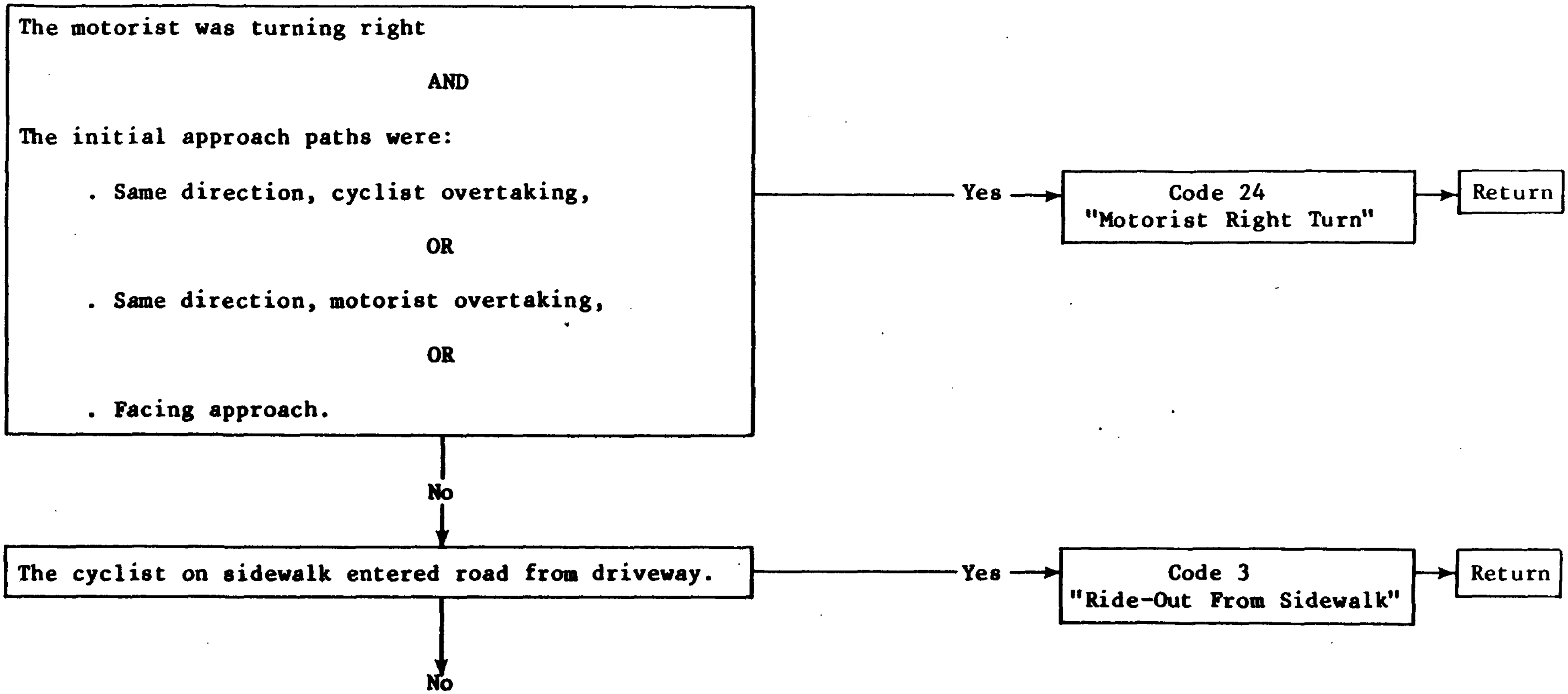

Figure E-1 (Cont'd)

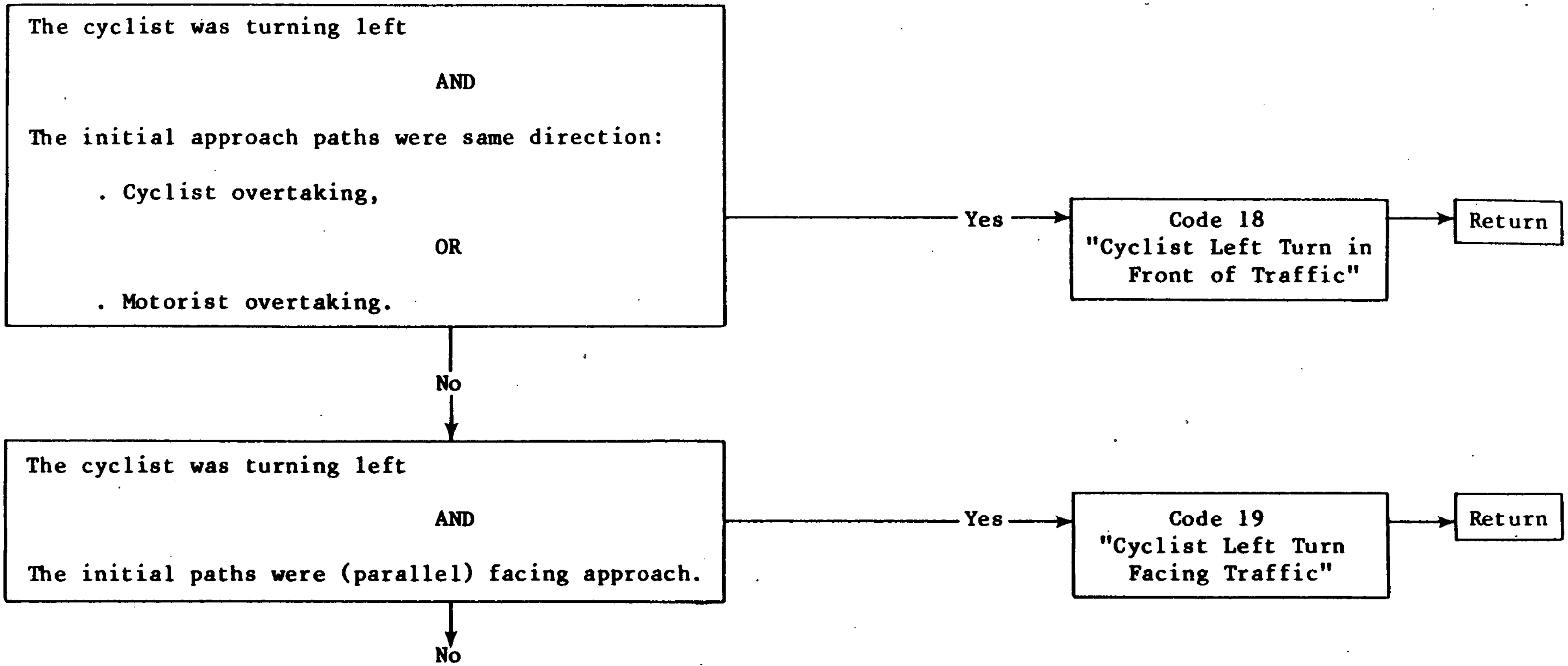

Figure E-1 (Cont'd)

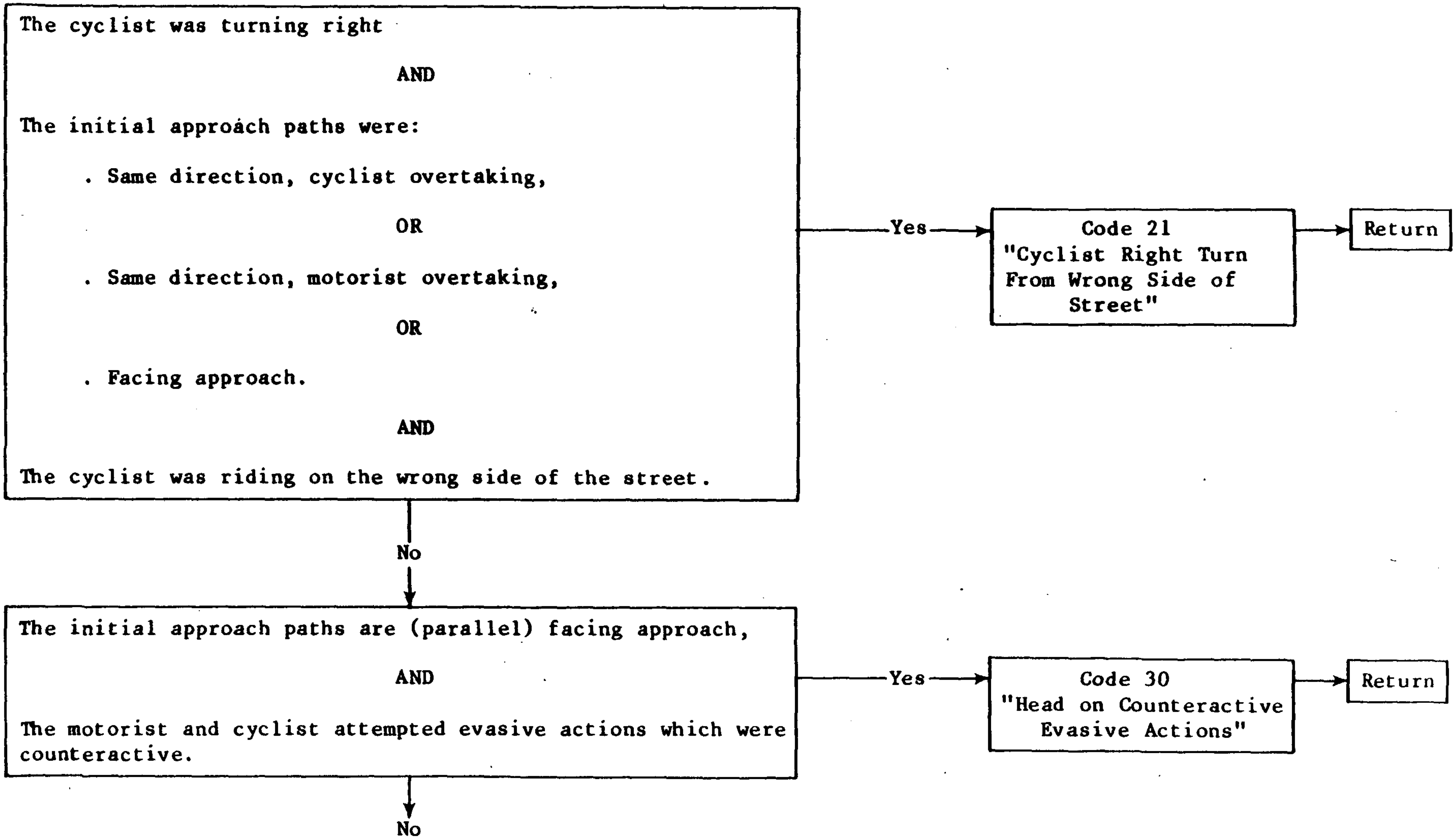

Figure E-1 (Cont'd)

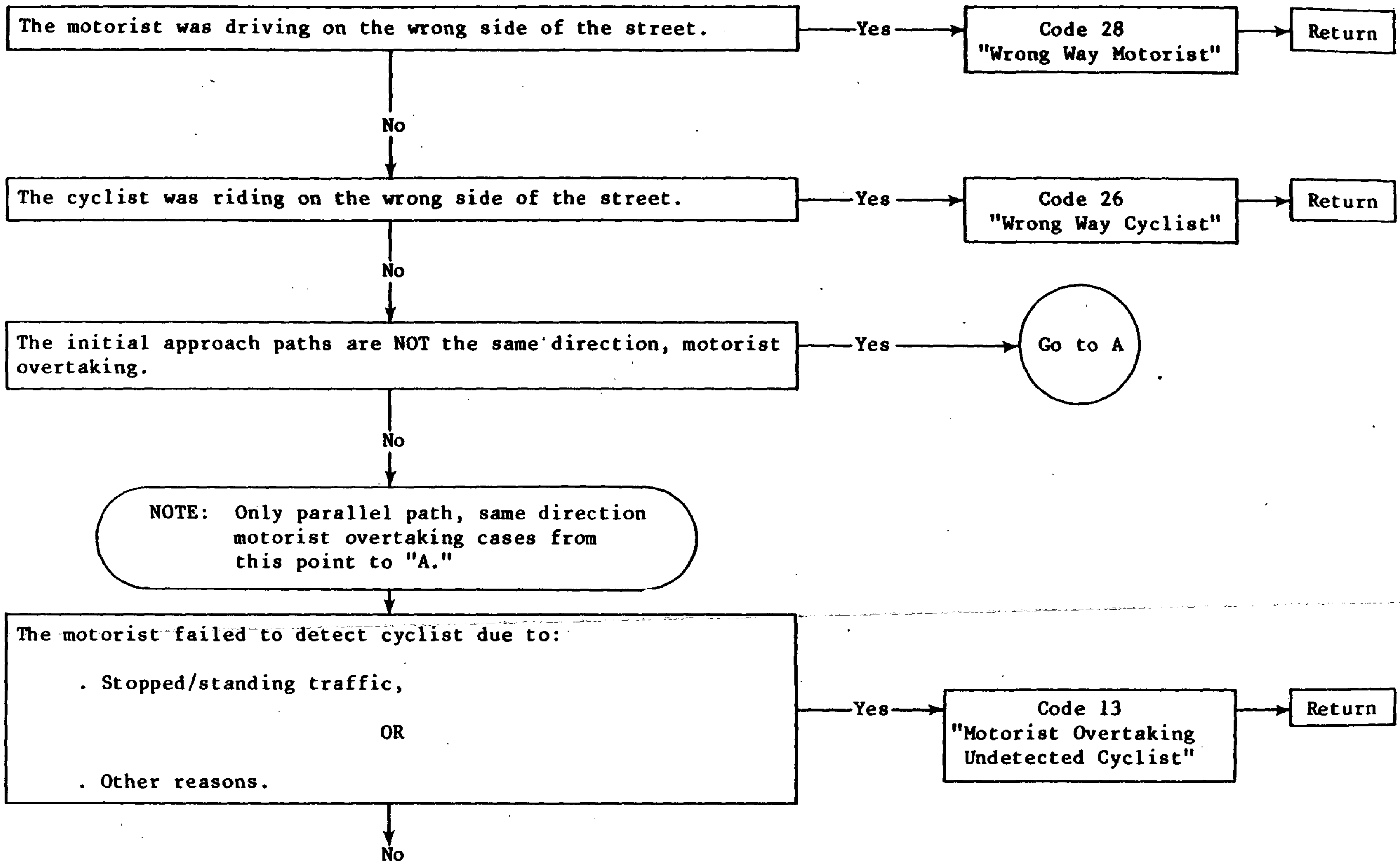

Figure E-1 (Cont'd)

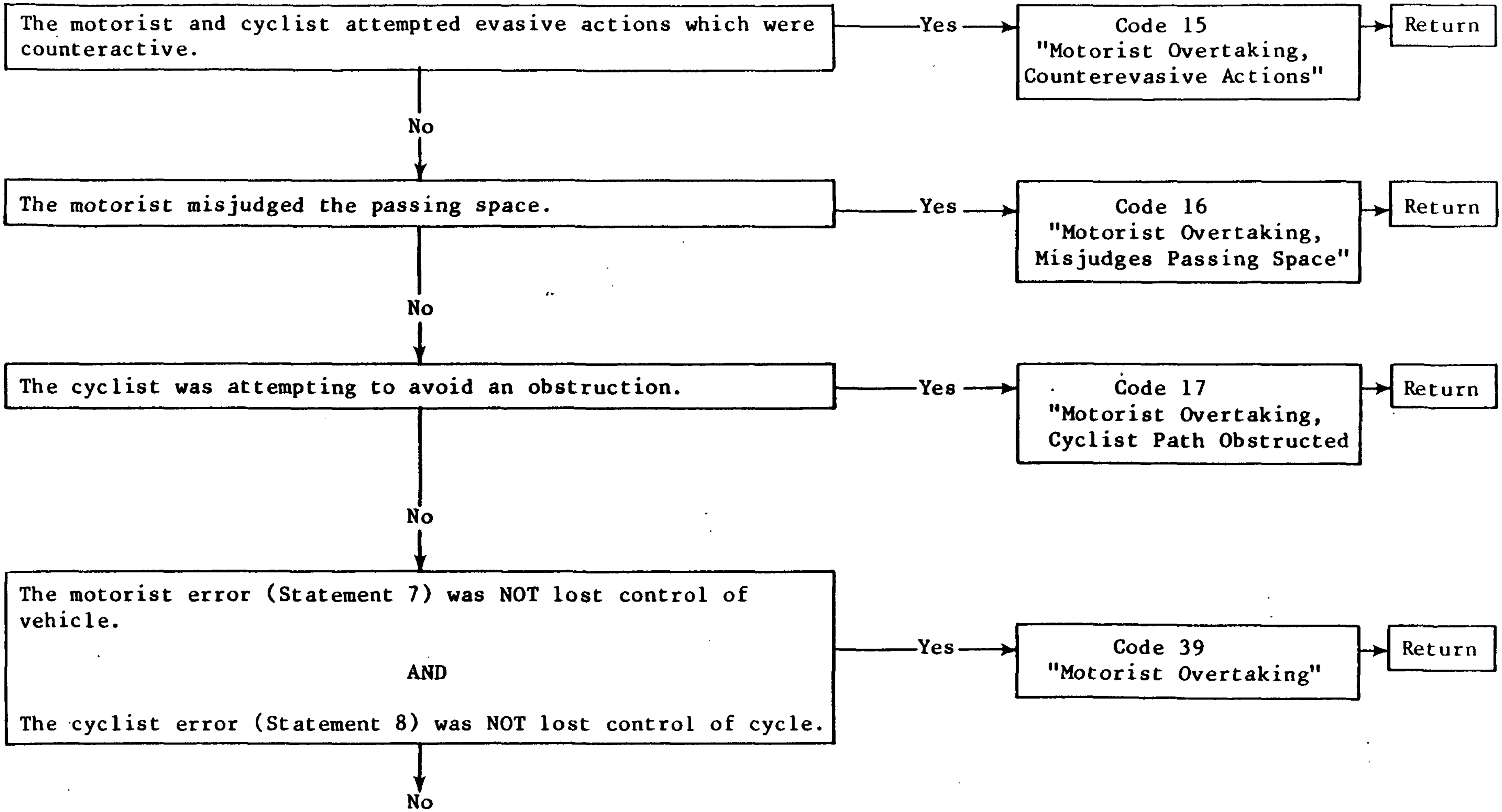

Figure E-1 (Cont'd)

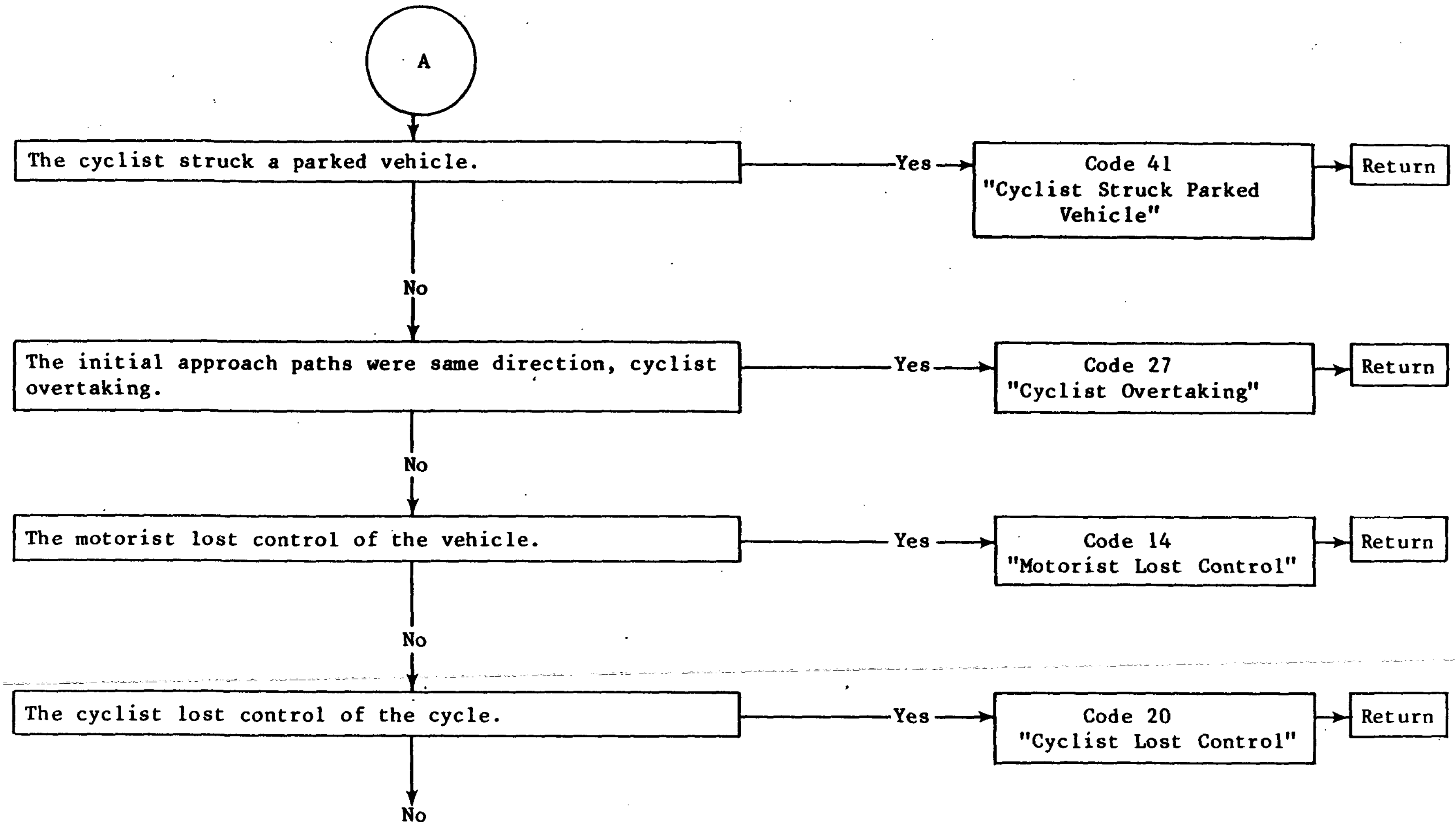

Figure E-1 (Cont'd)

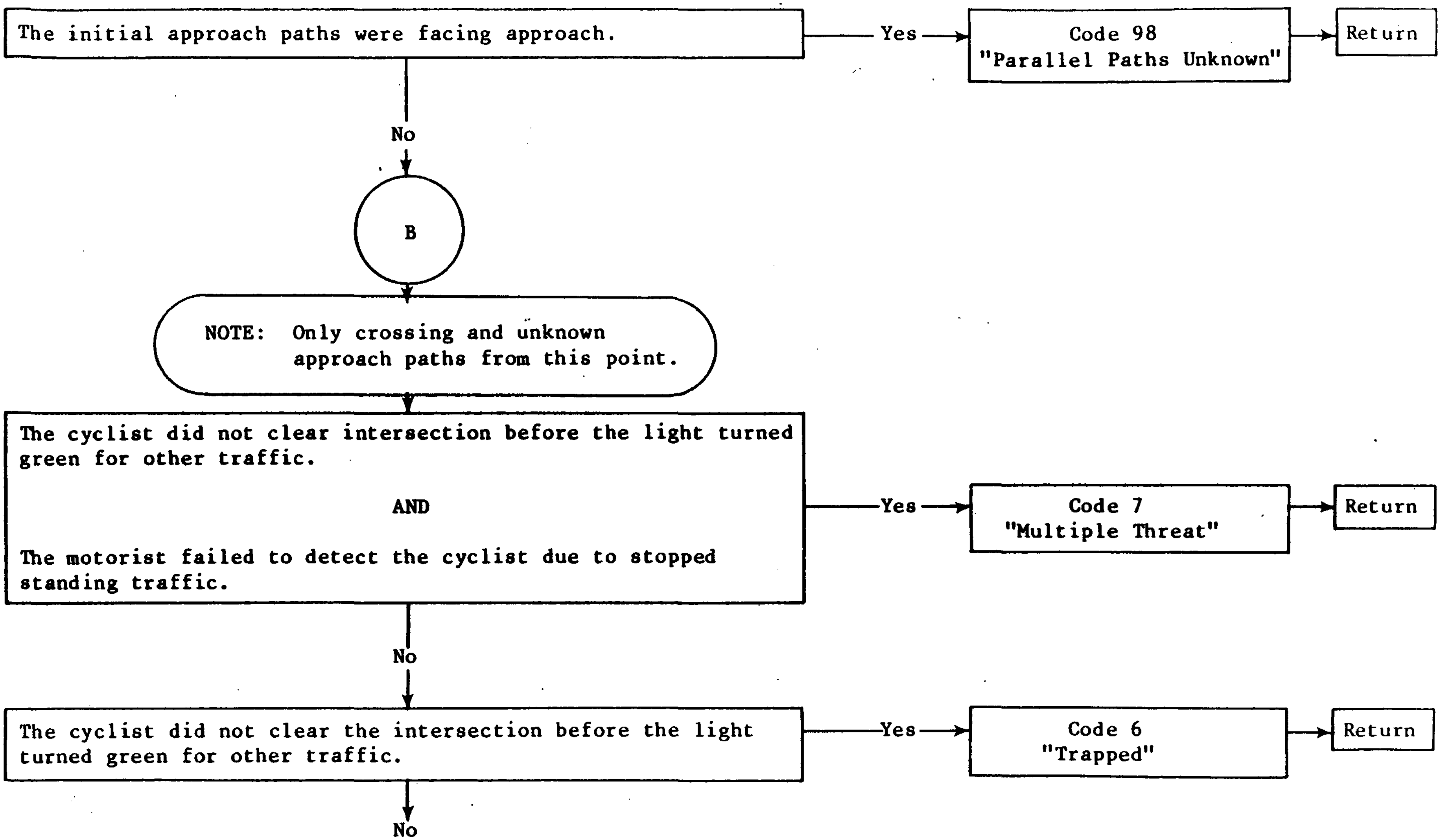

Figure E-1 (Cont'd)

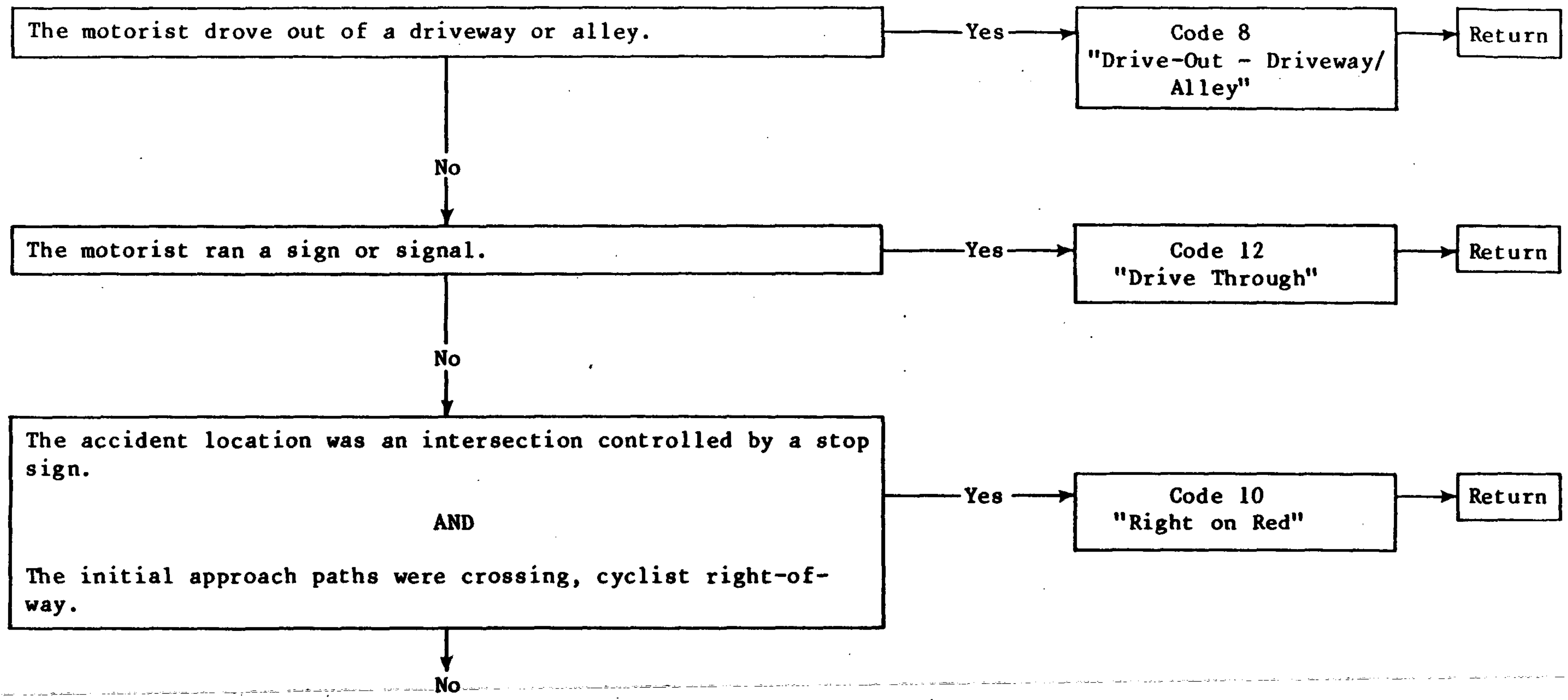

Figure E-1 (Cont'd)

The accident location was intersection, other.

AND

The initial approach paths were crossing, cyclist right-of-way.

AND

The motorist error (Statement 7) and the cyclist error (Statement 8) were NOT:

- Swing wide on right turn,

OR

- Cut corner on left turn.

→ Yes → Code 48 "Drive-Out, Intersection" → Return

↓ No

The cyclist entered the road from a residential driveway or alley.

→ Yes → Code 1 "Ride-Out, Residential Driveway" → Return

↓ No

The cyclist entered the road from a commercial driveway.

→ Yes → Code 2 "Ride-Out Commercial Driveway" → Return

↓ No

Figure E-1 (Cont'd)

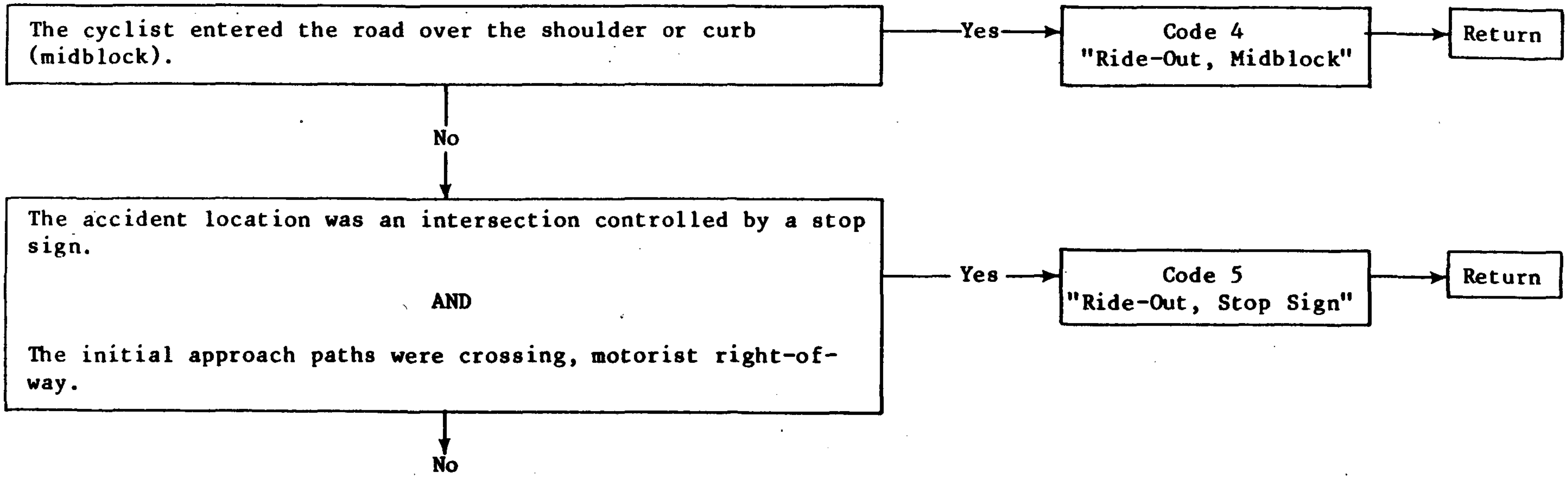

Figure E-1 (Cont'd)

The accident location was:

- An intersection controlled by a signal,

OR

- An intersection controlled by a stop sign.

AND

The initial approach paths were crossing, motorist right-of-way.

AND

The motorist error (Statement 7) and the cyclist error (Statement 8) were NOT:

- Swing wide on right turn,

OR

- Cut corner on left turn.

Yes → Code 49 "Ride-Out, Intersection" → Return

No ↓

The motorist cuts corner on a left turn.

Yes → Code 33 "Motorist Cuts Corner" → Return

No ↓

Figure E-1 (Cont'd)

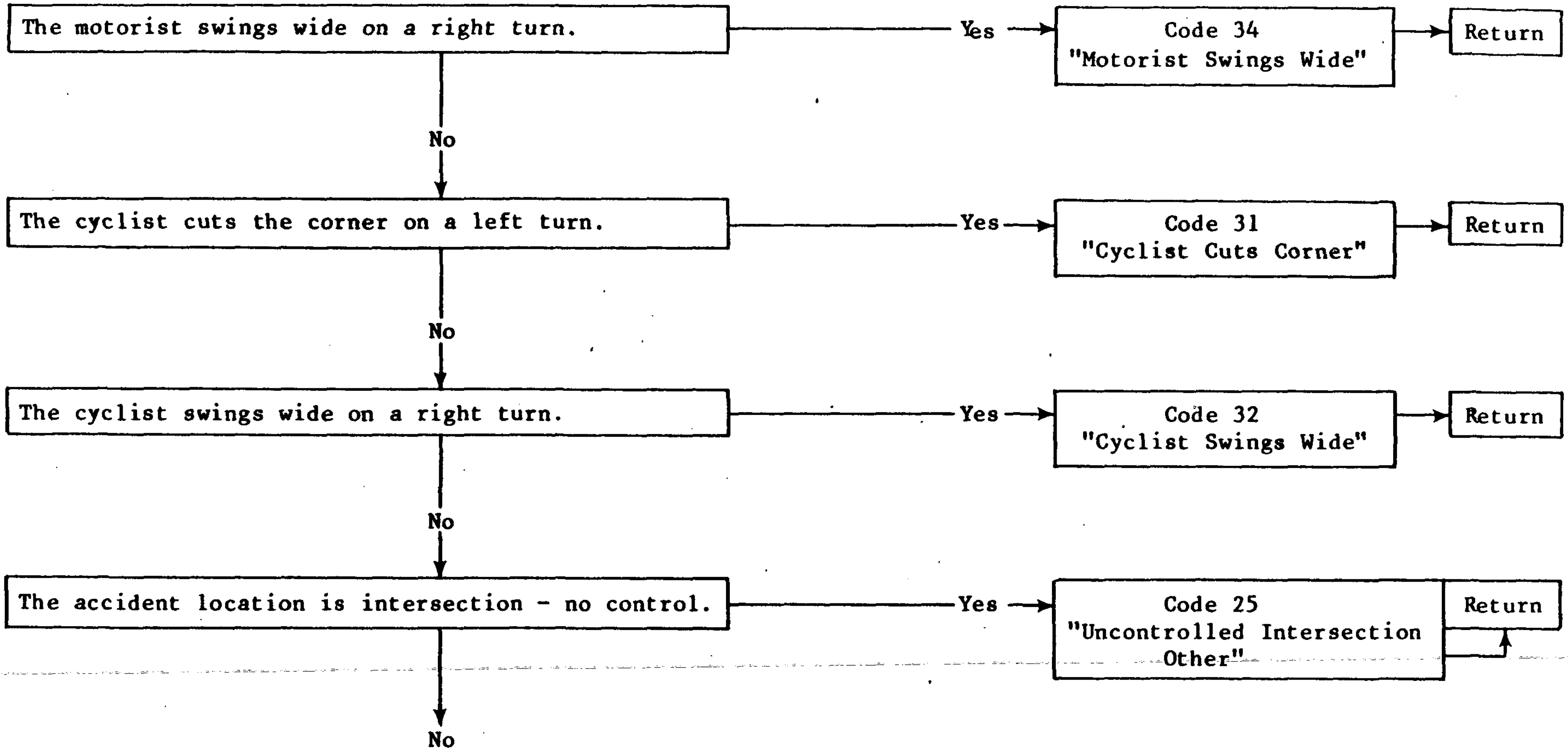

Figure E-1 (Cont'd)

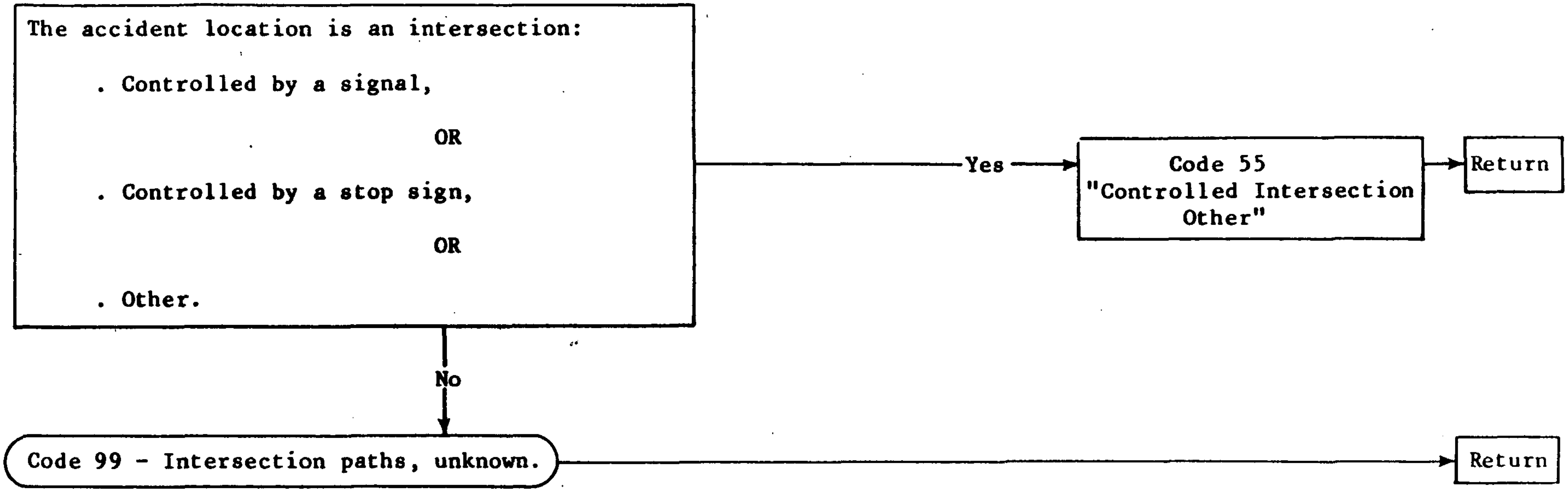

Figure E-1 (Cont'd)

Table E-3

Sample Data Input and Decoded Output

TST-36, 10000000	TST-36, 36
TST-40, 49999999	TST-40, 40
TST-11, 01000000	TST-11, 11
TST-29, 00001000	TST-29, 29
TST-35, 00040100	TST-35, 35
TST-22, 04000100	TST-22, 22
TST-23, 04000300	TST-23, 23
TST-24, 05000300	TST-24, 24
TST-3, 00050000	TST-3, 3
TST-18, 00300100	TST-18, 18
TST-19, 00300300	TST-19, 19
TST-21, 00400305	TST-21, 21
TST-30, 00010300	TST-30, 30
TST-28, 00000060	TST-28, 28
TST-26, 00000005	TST-26, 26
TST-13, 00000230	TST-13, 13
TST-15, 00010200	TST-15, 15
TST-16, 00000220	TST-16, 16
TST-17, 00020200	TST-17, 17
TST-39, 00000200	TST-39, 39
TST-41, 00000004	TST-41, 41
TST-27, 00000100	TST-27, 27
TST-14, 00000070	TST-14, 14
TST-20, 00000006	TST-20, 20
TST-7, 00090030	TST-7, 7
TST-6, 00090000	TST-6, 6
TST-8, 00030000	TST-8, 8
TST-12, 00000010	TST-12, 12
TST-9, 00003401	TST-9, 9
TST-10, 00002400	TST-10, 10
TST-48, 00005400	TST-48, 48
TST-1, 00060000	TST-1, 1
TST-2, 00070000	TST-2, 2
TST-4, 00080000	TST-4, 4
TST-5, 00003500	TST-5, 5
TST-49, 00002500	TST-49, 49
TST-33, 00000050	TST-33, 33
TST-34, 00000040	TST-34, 34
TST-31, 00000003	TST-31, 31
TST-32, 00000002	TST-32, 32
TST-25, 00004000	TST-25, 25
TST-55, 00003000	TST-55, 55
TST-99, 00000000	TST-99, 99
TST-98, 00000300	TST-98, 98

Table E-4

Tabulation Program

```
FORTRAN IV      V02.1-1     SUN 19-JUL-81 06:32:57                PAGE 001

      C  BICYCLE ACCIDENT TABULATION PROGRAM
      C
0001         PROGRAM BIKTAB
0002         LOGICAL NUM(10)
0003         LOGICAL SPACE
0004         REAL  SUM(45), EAN(45)
0005         INTEGER TO(20), FROM(20)
0006         INTEGER MAT(45), CODE,  LABEL(45)
      C
      C  ASSIGN FORTRAN UNIT #3 TO INPUT FILE
      C
0007         CALL ASSIGN (3,0,-1,'RDO')
      C
      C  ASSIGN FORTRAN UNIT #6 TO OUTPUT FILE/PRINTER
      C
0008         CALL ASSIGN (6,'KB:')
      C
      C  SET UP ACCIDENT TYPE CODES
      C
0009         DATA LABEL/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,
            1 18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,
            2 34,35,36,39,40,41,48,49,55,98,99,63/
0010         DATA  EAN/45*0./, SUM/45*.0001/
0011         DATA SPACE/'   '/,  TOTY/0./, MAT/45*0/
0012         WRITE(6,10)
      C
      C  OUTPUT HEADER
      C
0013  10     FORMAT(T25,'BICYCLE ACCIDENT TYPING'////'  ')
      C
      C  READ IN A LINE
      C
0014  1      READ(3,20,END=99,ERR=1)NUM,CODE
0015  20     FORMAT(X,10A1,I5)
      C
      C  LOCATE ACCIDENT TYPE SUBSCRIPT
      C
0016         DO 50 LOOP=1,44
0017         IF(CODE.EQ.LABEL(LOOP))GOTO 75
0019  50     CONTINUE
      C
      C  USE 45 FOR UNKNOWN TYPE
      C
0020         LOOP=45
0021  75     MAT(LOOP)=MAT(LOOP)+1
0022         GOTO 1
      C
      C  CASES ALL READ SUM AND CALCULATE PERCENT
      C
0023  99     ACCUM=0.
0024         DO 100 J=1,45
      C
      C  COMPUTE TOTAL
      C
```

Table E-4 (Cont'd)

```
FORTRAN IV       V02.1-1     SUN 19-JUL-81 06:32:57                    PAGE 002

0025          TOTY=MAT(J)+TOTY
      C
      C  MOVE INTEGER DATA TO REAL VARIABLE
      C
0026          SUM(J)=SUM(J)+MAT(J)
0027    100   CONTINUE
      C
      C  CALCULATE PERCENT
      C
0028          DO 150 J=1,45
0029          EAN(J)=SUM(J)/TOTY*100.
0030    150   CONTINUE
      C
      C  OUTPUT LABELS AND VALUES
      C
0031          DO 200 L=1,3
0032          IF(L.EQ.3)GOTO 300
0034          WRITE(6,500)(LABEL(J),J=((15*L)-14),(15*L))
0035    250   WRITE(6,550)(SUM(J),J=((15*L)-14),(15*L))
0036          WRITE(6,600)(EAN(J),J=((15*L)-14),(15*L))
0037          GOTO 200
0038    300   WRITE(6,650)(LABEL(J),J=((15*L)-14),(15*L-1)),63
0039          GOTO 250
0040    200   WRITE(6,450)SPACE
0041          WRITE(6,400)TOTY
0042    500   FORMAT(' ACC.  TYPE',T11,15I5)
0043    550   FORMAT(' # CASES',T11,15(F5.0))
0044    600   FORMAT(' % OF TOT. ',T12,15(F4.1,X))
0045    450   FORMAT(4(/),'     ',A1)
0046    650   FORMAT(' ACC.  TYPE',T11,14I5,3X,A1)
0047    400   FORMAT(////'          CASES  TYPED = ',F6.0)
0048          STOP
0049          END
```

Table E-5

Sample Tabulation
(Using Sample Data Input

ACC. TYPE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# CASES	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.
% OF TOT.	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3

ACC. TYPE	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
# CASES	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.
% OF TOT.	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3

ACC. TYPE	31	32	33	34	35	36	39	40	41	48	49	55	98	99	?
# CASES	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	0.
% OF TOT.	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	0.0

CASES TYPED = 44.

Table E-6

Comparison Program

```
FORTRAN IV      V02.1-1     SUN 19-JUL-81 06:33:47                PAGE 001

      C
      C       PROGRAM COMPAR.FOR
      C         COMPARES ACCIDENT CODES GENERATED BY TYPING PROGRAM.
      C         THIS PROGRAM HANDLES FROM 1 TO 10 FILES, LISTING THE
      C         DIFFERENCES AND AVERAGE AGREEMENT RATE BETWEEN EACH FILE.
      C
      C
      C       VARIABLES: NUMFIL--NUMBER OF FILES TO BE COMPARED
      C                  NAME  --NAME OF INPUT FILES
      C                  ACCNUM--ACCIDENT NUMBER FROM EACH FILE
      C                  ICODE --ACCIDENT TYPE CODE
      C                  IDIFF --SPACE TO NOTE DIFFERENCES
      C                  IMATCH--MATRIX OF ACC. TYPE MATCHES
      C                  ITOT  --MATRIX OF TOTAL POSSIBLE MATCHES
      C
      C
      C
      C
0001          LOGICAL ACCNUM(10,10), IDIFF, NAME(4), BLANK, ASTRIX
0002          INTEGER ITOT(10,10),MATCH(10,10),ICODE(10)
0003          DATA ITOT/100*0/, MATCH/100*0/, ALLMAT/0./,
             1 ALLTOT/0./, ALLMAT/0./, BLANK/'    '/, ASTRIX/'*   '/
      C
      C       DEFINE OUTPUT DEVICE/PRINTER
      C
0004          CALL ASSIGN (6,'KB:',0)
      C
      C
      C    REQUEST NUMBER OF FILES TO COMPARE
      C
0005          TYPE 50
0006  50      FORMAT(T5,'NUMBER OF FILES TO COMPARE--',$)
      C
      C       FOR BATCH APLICATIONS SET NUMFIL EQUAL TO THE
      C       NUMBER OF FILES TO COMPAR.
      C
      C
      C       FILL VARIABLE NAME WITH FILE NAMES
      C
      C
0007          ACCEPT 75, NUMFIL
0008  75      FORMAT(I5)
0009          DO 100 LOOP=1,NUMFIL
0010          TYPE 80,LOOP
0011  80      FORMAT(T5,'FILE ',I2,'  NAME ',$)
0012          ACCEPT 160, NAME
0013  160     FORMAT(4A4)
0014          INFILE=LOOP+9
      C
      C
      C
0015          CALL ASSIGN(INFILE,NAME,10,'RDO')
      C
      C
```

Table E-6 (Cont'd)

```
FORTRAN IV        V02.1-1     SUN 19-JUL-81 06:33:47                PAGE 002

      C      ASSIGN FORTRAN UNIT NUMBER 1-10 TO FILE NAMES
      C
      100    CONTINUE
      111    IDIFF=BLANK
             READ(10,150,END=999,ERR=115)(ACCNUM(1,M),M=1,9),ICODE(1)
      150    FORMAT(X,9A1,2X,I3)
      115    DO 400 LOOP=2, NUMFIL
             INFILE=LOOP+9
             READ(INFILE,150,END=200,ERR=215)(ACCNUM(LOOP,M),M=1,9),
            1 ICODE(LOOP)
      215    DO 225 INNER=7,9
      225    IF(ACCNUM(1,INNER).NE.ACCNUM(LOOP,INNER))GOTO 200
      250    IF(ICODE(1).NE.ICODE(LOOP))IDIFF=ASTRIX
             GOTO 400
      200    INFILE=LOOP+9
             REWIND INFILE
      210    CONTINUE
             READ(INFILE,150,END=300,ERR=260)(ACCNUM(LOOP,M),M=1,9),
            1 ICODE(LOOP)
      260    DO 275 INNER=7,9
      275    IF(ACCNUM(1,INNER).NE.ACCNUM(LOOP,INNER))GOTO 210
      300    ICODE(LOOP)=0
      400    CONTINUE
      C
      C
      C      COUNT MATCHES AND TOTAL POSSIBLE.
      C
             DO 500 I=1,(NUMFIL-1)
             DO 500 J=(I+1),NUMFIL
             IF(ICODE(I).EQ.0.OR.ICODE(J).EQ.0) GOTO 500
             ITOT(I,J)=ITOT(I,J)+1
             IF(ICODE(I).EQ.ICODE(J)) MATCH(I,J)=MATCH(I,J)+1
      500    CONTINUE
             WRITE(6,600,ERR=111)(ACCNUM(1,M),M=1,9),IDIFF,
            1 (ICODE(M),M=1,NUMFIL)
      600    FORMAT(' ',5X,9A1,2X,A1,10(4X,I3))
             GOTO 111
      999    DO 1000 I=1,(NUMFIL-1)
             DO 1000 J=(I+1),NUMFIL
             TEMP=MATCH(I,J)
             TOT=+ITOT(I,J)
             AVER=(TEMP/TOT)*100
             ALLMAT=ALLMAT+MATCH(I,J)
             ALLTOT=ALLTOT+ITOT(I,J)
      1000   WRITE(6,1100)I,J,MATCH(I,J),ITOT(I,J),AVER
      1100   FORMAT(/T5,'MATCHES BETWEEN FILE',I2,' AND FILE',I2,' =',I4,
            1 ' OUT OF',I4,' POSSIBLE = ',F5.1,'%')
             TEMP=(ALLMAT/ALLTOT)*100
             WRITE(6,1200)TEMP
      1200   FORMAT('   TOTAL AGREEMENT RATE = ',F5.1,'%')
             STOP 'END OF OUTPUT'
             END
```

Table E-7

Sample Coder Comparison

TST-36		36	36	36
TST-40		40	40	40
TST-11		11	11	11
TST-29		29	29	0
TST-35		35	0	0
TST-22		22	0	0
TST-23		23	0	0
TST-24		24	0	0
TST-3		3	3	3
TST-18	*	18	19	19
TST-19		19	19	19
TST-21		21	21	0
TST-30		30	30	30
TST-28	*	28	28	12
TST-26		26	26	26
TST-13		13	13	13
TST-15		15	15	15
TST-16		16	16	16
TST-17		17	0	0
TST-39		39	0	0
TST-41		41	41	41
TST-27		27	27	27
TST-14		14	14	14
TST-20	*	20	20	26
TST-7		7	7	7
TST-6		6	6	6
TST-8		8	8	8
TST-12	*	12	3	3
TST-9		9	9	9
TST-10		10	10	10
TST-48		48	48	48
TST-1		1	1	1
TST-2		2	2	2
TST-4		4	4	4
TST-5		5	5	5
TST-49		49	49	49
TST-33		33	33	33
TST-34		34	34	34
TST-31		31	0	0
TST-32		32	0	0
TST-25		25	25	25
TST-55		55	0	0
TST-99		99	0	0

MATCHES BETWEEN FILE 1 AND FILE 2 = 31 OUT OF 33 POSSIBLE = 93.9%

MATCHES BETWEEN FILE 1 AND FILE 3 = 27 OUT OF 31 POSSIBLE = 87.1%

MATCHES BETWEEN FILE 2 AND FILE 3 = 29 OUT OF 31 POSSIBLE = 93.5%
TOTAL AGREEMENT RATE = 91.6%

*U.S. GOVERNMENT PRINTING OFFICE : 1983 0-401-287/859