# AUTOMATIC ELECTROCARDIOGRAM INTERPRETATION

A Thesis
Submitted to the
Faculty of Graduate Studies and Research
in Partial Fulfilment of the Requirements
for the
Degree of Master of Science
in the Division of Biomedical Engineering
University of Saskatchewan

by
Robert James Firby
Saskatoon, Saskatchewan
June, 1983

## ACKNOWLEDGEMENTS

I would like to thank my supervisors, Dr. J.F. Lopez and Prof. K. Takaya, for their helpful comments while writing this thesis and for their assistance in getting it printed. I would also like to acknowledge the work of Mr. Spencer Hanson in gathering the sample ECG traces used to test the programs in this thesis, and the work of Mr. Doug Hall which helped his computer talk to mine. Joanne Firby also deserves recognition as it was she who typed most of the thesis and helped to find the many quotes which shed extra light on the subject matter within it. It should also be noted that I was supported by a scholarship from the Canadian Natural Sciences and Engineering Research Council while doing this work.

Finally, I would like to thank Alfred, Lord Tennyson who helped me keep things in perspective by writing in Memorium I: "Our little systems have their day/ They have their day and cease to be."

University of Saskatchewan

Biomedical Engineering Abstract No. 83A230

# AUTOMATIC ELECTROCARDIOGRAM INTERPRETATION

Student: Robert James Firby
Supervisors: J.F. Lopez and K. Takaya

## ABSTRACT

The goal of the work behind this thesis was to design and partially implement an automatic system for the analysis of electrocardiograms.

The thesis deacribes an original model of the human ECG analysis process which demonstrates that it breaks naturally into three stages: reduction, labelling and diagnosis. It is shown that to make a correct diagnosis, each of these stages must be both expressive and complete. A review of the literature reveals that what had previously been perceived as a fundamental limitation of the machine diagnosis problem is, in reality, a lack of expressiveness at the reduction stage.

Original algorithms to overcome this problem are described for the reduction and labelling stages, with examples of their implementation being given. Future work on the diagnosis stage is also suggested and discussed.

## Table of Contents

List of Figures vii

List of Tables ix

1. INTRODUCTION TO ELECTROCARDIOGRAM INTERPRETATION 1

1.1 The Heart 2
1.1.1 The Heart Muscle System 2
1.1.2 The Electrical Control System of the Heart 4
1.2 The Electrocardiogram 6
1.2.1 Electrical Potentials from Muscles 6
1.2.2 ECG Lead Systems 8
1.2.3 The Normal ECG 10
1.3 The Abnormal Electrocardiogram 12
1.3.1 Contour 13
1.3.2 Rhythm and Arrhythmias 14
1.3.3 Summary 15
1.4 The Automation of ECG Diagnosis 16
1.4.1 Outline of Thesis 17

2. A MODEL OF HUMAN ELECTROCARDIOGRAM DIAGNOSIS 18

2.1 Introduction to the Model 19
2.2 The Details of the Human ECG Model 22
2.2.1 Reduction 25
2.2.2 Labelling 29
2.2.3 Diagnosis 30
2.2.4 Information Flow 32
2.3 Summary of the Human ECG Model 37

3. A REVIEW OF EXISTING DIAGNOSTIC ECG PROGRAMS 40

3.1 Reduction-Labelling Algorithms 41
3.1.1 Slope and Spatial Velocity Techniques 42
3.1.2 Token Techniques 47
3.2 Diagnosis Algorithms 52
3.2.1 Decision Trees 52
3.2.2 Statistical Methods 55
3.3 Summary 57

4. IMPROVED REDUCTION AND LABELLING ALGORITHMS 61

4.1 The Reduction Algorithm 63
4.1.1 ECG Trace and Derivative 65
4.1.2 Partial Wave Shape Tokens 68
4.1.3 Whole Wave Group Tokens 72
4.1.4 Summary 74
4.2 The Labelling Algorithm 75
4.2.1 QRS Position 77
4.2.2 QRS Extent 79
4.2.3 QRS Classification 83
4.2.4 P-T Wave Detection 87
4.2.5 Summary 89

4.3 Summary of Reduction and Labelling Algorithms 91

5. DISCUSSION AND RECOMMENDATIONS FOR FUTURE WORK 95

5.1 The Diagnosis Algorithm 99
5.2 Improvements to the Reduction and Labelling Algorithms 101
5.3 System Testing 102
5.4 Conclusion 103

References 104

Appendix A. SIGNAL PROCESSING ASPECTS OF THE SYSTEM 111

A.1 Sampling the ECG Trace 111
A.2 Calculating the Derivative 112
A.3 Measuring the ECG Trace Noise 113
A.4 Smoothing to Eliminate Noise 115
A.5 Summary 118

Appendix B. EXAMPLES OF THE PROCESSING OF FIVE ECG TRACES 119

Appendix C. ALGORITHMS FOR THE ECG INTERPRETATION SYSTEM 125

C.1 Algorithm of the Main ECG Interpretation Program 126
C.2 Reduction Algorithms 134
C.3 Labelling Algorithms 146

## List of Figures

Figure 1-1: The Muscular Anatomy of the Heart. 3
Figure 1-2: The Electrical Control System of the Heart. 4
Figure 1-3: The Measured Electrical Potential of Muscle (from [24]). 7
Figure 1-4: The 12 Lead Diagnostic Electrode System. 9
Figure 1-5: The Effective Lead II Electrocardiogram. 11
Figure 1-6: The ECG of the 12 Diagnostic Leads. 13
Figure 1-7: Right Bundle Branch Block. 14
Figure 1-8: 2° AV Block. 15

Figure 2-1: An example of the stages of the human analysis model. 20
Figure 2-2: The problems of an insufficiently expressive rule set. 24
Figure 2-3: An example of the problems with an incomplete rule set. 25
Figure 2-4: Labelling an ambiguous wave. 32
Figure 2-5: An example of the use of feedback queries. 34
Figure 2-6: An example of feedback from a heart representation. 35
Figure 2-7: The human ECG analysis model information flow. 37

Figure 3-1: Using spatial velocity and threshold techniques to identify the waves in an ECG trace. 44
Figure 3-2: Examples of shape coding schemes. Diagram A is from [38] and B is from [21]. For details see text. 45
Figure 3-3: A missed R-wave superimposed on a T-wave. 46
Figure 3-4: Flagging the end points of slope tokens, from [2]. Diagram A shows the clipping levels for the ECG derivative and B shows the points flagged by those clipping levels. 48
Figure 3-5: An ECG trace divided into slope tokens, from [5]. 48
Figure 3-6: Grouping slope tokens into segments or bumps, from [5]. 49
Figure 3-7: Some of the logic for adding a segment to the P side of a QRS-complex, from [5]. 50
Figure 3-8: The final set of clinical labels for the slope token technique, from [5]. 50
Figure 3-9: The decision tree used in the diagnosis section of the program in [70]. 53
Figure 3-10: The decision tree used in the diagnosis program of [63]. 54
Figure 3-11: A feature vector for Bayesian classification of the QRS contour. 56

Figure 4-1: Examples of digitized ECG traces and their derivatives. 67
Figure 4-2: The seven partial wave shape tokens. 69
Figure 4-3: Overlapping partial waves and the resulting wave shape tokens. 69

Figure 4-4: Various partial wave shape tokens for very similar ECG traces. 71
Figure 4-5: The partial wave shape tokens generated by actual ECG traces. 72
Figure 4-6: Cases of partial wave shape tokens which define whole wave group tokens. 73
Figure 4-7: Removing ambiguous information with whole wave group tokens. 73
Figure 4-8: Examples of whole wave group tokens. 74
Figure 4-9: An example to show that some whole wave tokens represent waves while others represent the space between waves. 77
Figure 4-10: Detecting QRS-complexes. 79
Figure 4-11: Features sought by the group-property feedback query. 80
Figure 4-12: Results of the QRS-complex extension algorithm. 82
Figure 4-13: Classifying QRS-complexes. 85
Figure 4-14: Comparing QRS classes. 85
Figure 4-15: A possible QRS extension error. 87
Figure 4-16: Examples of actual QRS-complex identification. 87
Figure 4-17: Significant whole wave group tokens. 88
Figure 4-18: A missed R-wave. 89
Figure 4-19: Examples of actual initial P and T-wave detection. 89
Figure 4-20: Actual ECG traces labelled after a single diagnosis feedback query. 91

Figure A-1: The elimination of some small waves by the smoothing algorithm 117

Figure B-1: The first sample trace. Sampling rate is 100Hz, and the noise value is 6 digitizer units. 120
Figure B-2: The second sample trace. Sampling rate is 100Hz, and the noise value is 15 digitizer units. 121
Figure B-3: The third sample trace. Sampling rate is 250Hz, and the noise value is 10 digitizer units. 122
Figure B-4: The fourth sample trace. Sampling rate is 250Hz, and the noise value is 2 digitizer units. 123
Figure B-5: The fifth sample trace. Sampling rate is 250Hz, and the noise value is 1 digitizer unit. 124

## List of Tables

Table 2-1: A comparison of human ECG analysis and language learning. 22

Table 2-2: Summary of the Human ECG Analysis Model 38

Table 3-1: A summary of the major ECG analysis algorithms from the literature, and their important characteristics. 60

Table 4-1: Sample algorithm execution times. 94

# Chapter 1

# INTRODUCTION TO ELECTROCARDIOGRAM INTERPRETATION

*It is a step toward health to know the disease.*
[Erasmus, Adagia, No. 9]

*The mechanism of the human heart, when you thoroughly understand it, is, like all the other works of nature, very wonderful, but very simple. When it does not work well, the fault is not in the machinery, but in its management.*
[T.C. Haliburton, Nature and Human Nature, 1855]

The heart has always been one of the most enigmatic parts of the body because of its inaccessability. However, in the late 1800's Einthoven began exploring the activity of the heart by measuring the electric field it generates while contracting. Since then, the electrocardiogram, or ECG, has become one of the most important clinical tools for the diagnosis of heart disease. This importance, along with the now-extensive knowledge of heart function, has led naturally to many attempts at automating the ECG diagnostic process. While these programs have been very successful in some areas, there remain many ECG problems that appear quite difficult to computerize. This thesis is concerned with understanding why some ECG features are more difficult to diagnose than others and how solutions to these problems might be approached. First, however, to provide

a common background, a brief introduction to the heart and the origin of the electrocardiogram is necessary.

## 1.1 The Heart

The human heart consists of two pumps, one to send freshly oxygenated blood from the lungs to the rest of the body, and one to send the blood, on its return, back through the lungs. Each pump consists of two chambers, and the action of the four chambers is coordinated by a system of specialized electrically conducting tissue. A basic description of the heart chambers and conduction system is given in this section.

### 1.1.1 The Heart Muscle System

The muscular anatomy of the heart is simple in concept, as shown in figure 1-1.

The two input chambers of the heart are the atria. Blood flows from the body through the superior and inferior venae cavae into the right atrium and from the lungs through the pulmonary vein into the left atrium. The ventricles are the output chambers with the right ventricle pumping blood into the pulmonary artery and through the lungs, and the left ventricle pumping into the aorta and the rest of the body.

The blood flow between the heart chambers is controlled by four valves. The mitral and tricuspid valves, between the left atrium and ventricle, and between the right atrium and ventricle respectively, prevent backflow of blood from the ventricles to the atria when the ventricles contract. The

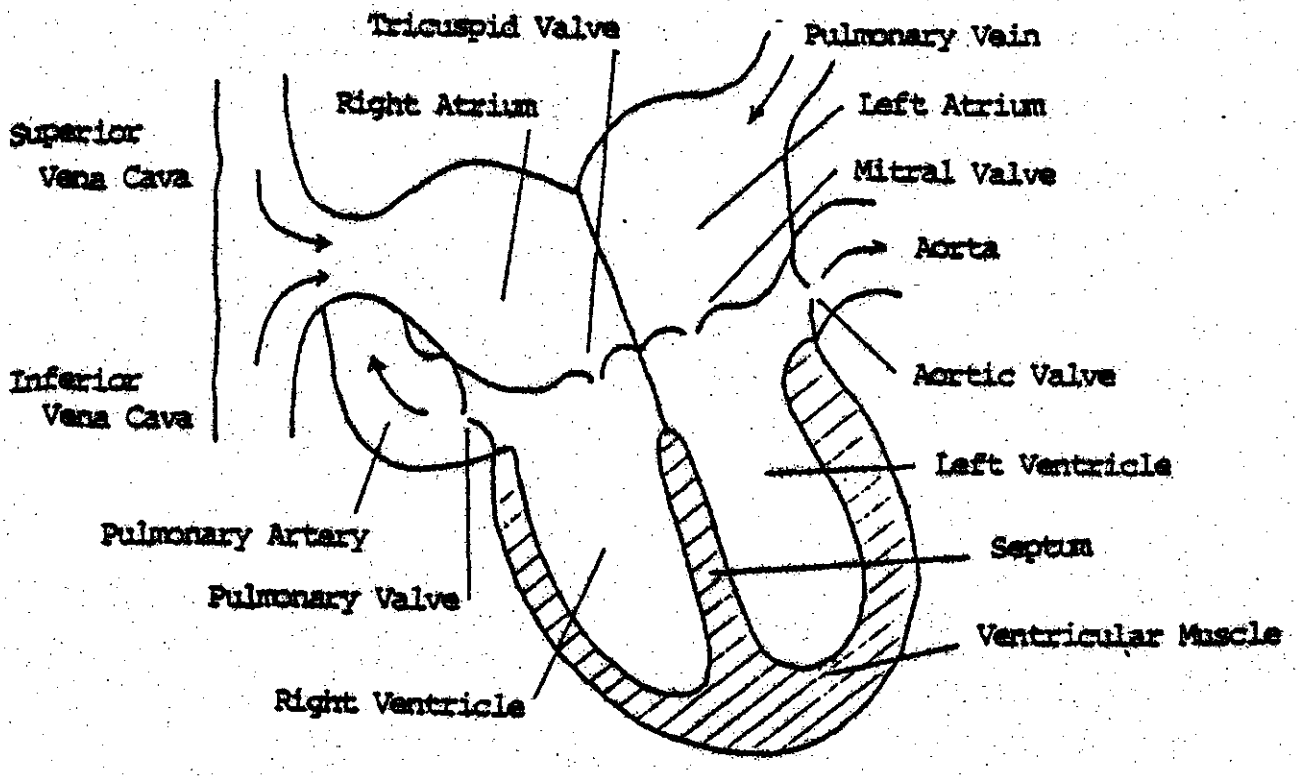


**Figure I-1: The Muscular Anatomy of the Heart.**

pulmonary valve in the right ventricle and the aortic valve in the left prevent the backflow of blood from the circulatory system into the ventricles when they expand (i.e., relax).

The pumping action of the heart occurs each beat, and has three phases. At the beginning of the beat, the atria contract together, pushing blood through the mitral and tricuspid valves into the two ventricles. When this contraction is complete, back pressure closes the valves and the two ventricles contract together. This forces blood through the pulmonary and aortic valves into the pulmonary artery and aorta, and then into circulation. In the final phase, the heart muscle relaxes and blood flows from the vena cava and pulmonary vein into the atria and ventricles.

### 1.1.2 The Electrical Control System of the Heart

The synchronization of the heart chambers required for the sequencing described above is achieved through an electrical control system. The essentials of this system are shown in figure 1-2.

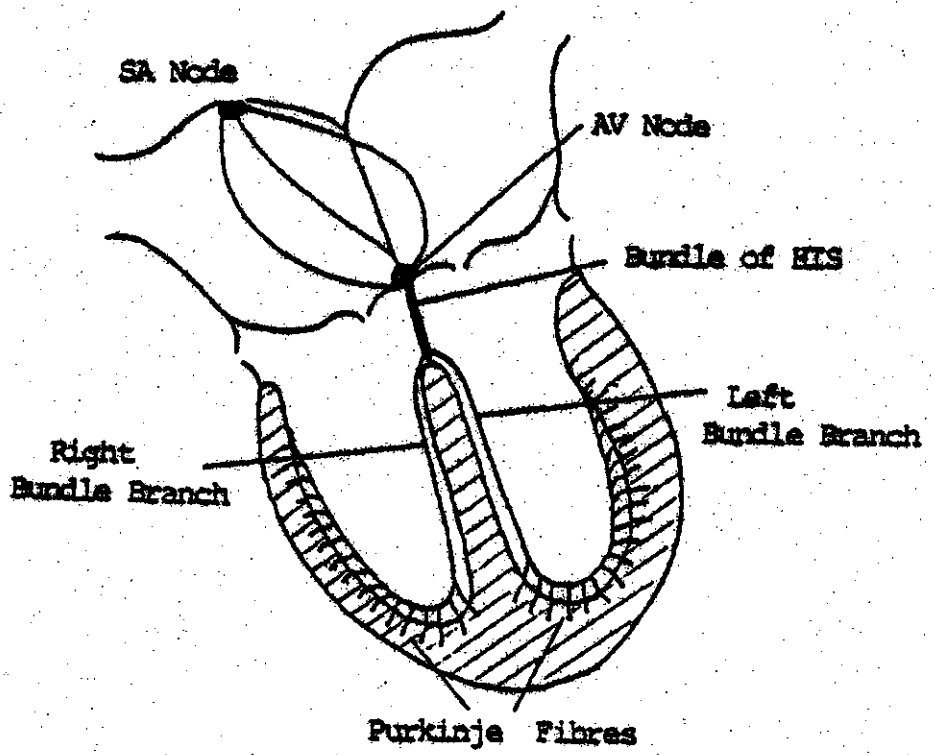


**Figure 1-2:** The Electrical Control System of the Heart.

The heart muscle, like all muscle in the body, begins to contract when it receives an external negative electrical impulse or action potential. In a normal heart, the sino-atrial (SA) node positioned on the surface of the right atrium spontaneously produces a regular series of action potentials. All specialized heart tissue has this pacemaker property, with the natural period between potentials being longest for the Purkinje fibres, then the bundle of HIS, and

shortest for the SA node. Because the SA node has the fastest rhythm, and pacemaker cells are reset when they receive an action potential, the rate of SA node action potentials is the rate of the heartbeat. Although the rate is thus controlled within the heart, it can be increased or decreased by the action of the autonomic nervous system on the SA node. Therefore, in a normal heart, each beat begins with an action potential from the SA node.

The action potential is conducted away from the SA node by the muscle tissue of the atria. The potential causes the muscle to contract, and a wave of action potential and muscle contraction spreads over the atria, pumping blood into the ventricles. On reaching the boundary between the atria and ventricles, the action potential stops everywhere except at a bundle of tissue called the atrio-ventricular (AV) node. Here it is delayed to allow complete emptying of the atria.

From the AV node, the action potential travels along the Bundle of HIS, splits along the left and right bundle branches, and spreads over the inside of the ventricles along the Purkinje fibres. This conduction is at high speed, about six times faster than along ordinary muscle, and reaches the entire ventricular muscle almost simultaneously. The ventricles therefore contract as a strong unit pumping blood out of the heart.

At this point, the action potential is exhausted and there is a pause for the heart muscle to relax before the next SA

node impulse starts another beat.

## 1.2 The Electrocardiogram

The ECG is a recording, from the surface of the body, of the electrical activity of the heart. This activity is the shift of electrical charge in the muscle as the action potential moves through it. To understand the features of an ECG waveform, it is first necessary to review the way this muscle electrical potential recording is made.

### 1.2.1 Electrical Potentials from Muscles

The propagation of an action potential along a muscle, and the electrical signal measured from it is shown in figure 1-3.

Due to the biochemistry and physiology of the muscle tissue, it is polarized in the relaxed state having a positive (+ve) charge on the outside and a negative (-ve) charge on the inside. If electrodes are placed near the muscle, as shown in figure 1-3, and connected to a device for recording the potential between them, no voltage will be registered in the relaxed state.

If a -ve action potential impulse is applied to the outside of the muscle at one end, it will depolarize at that point, shifting charge so that it becomes -ve outside and +ve inside. This -ve outside charge causes the muscle next to it to depolarize as well, and, in this way, the -ve action potential "flows" along the outside of the muscle. Due to the charge shift, the muscle contracts. During this contraction,

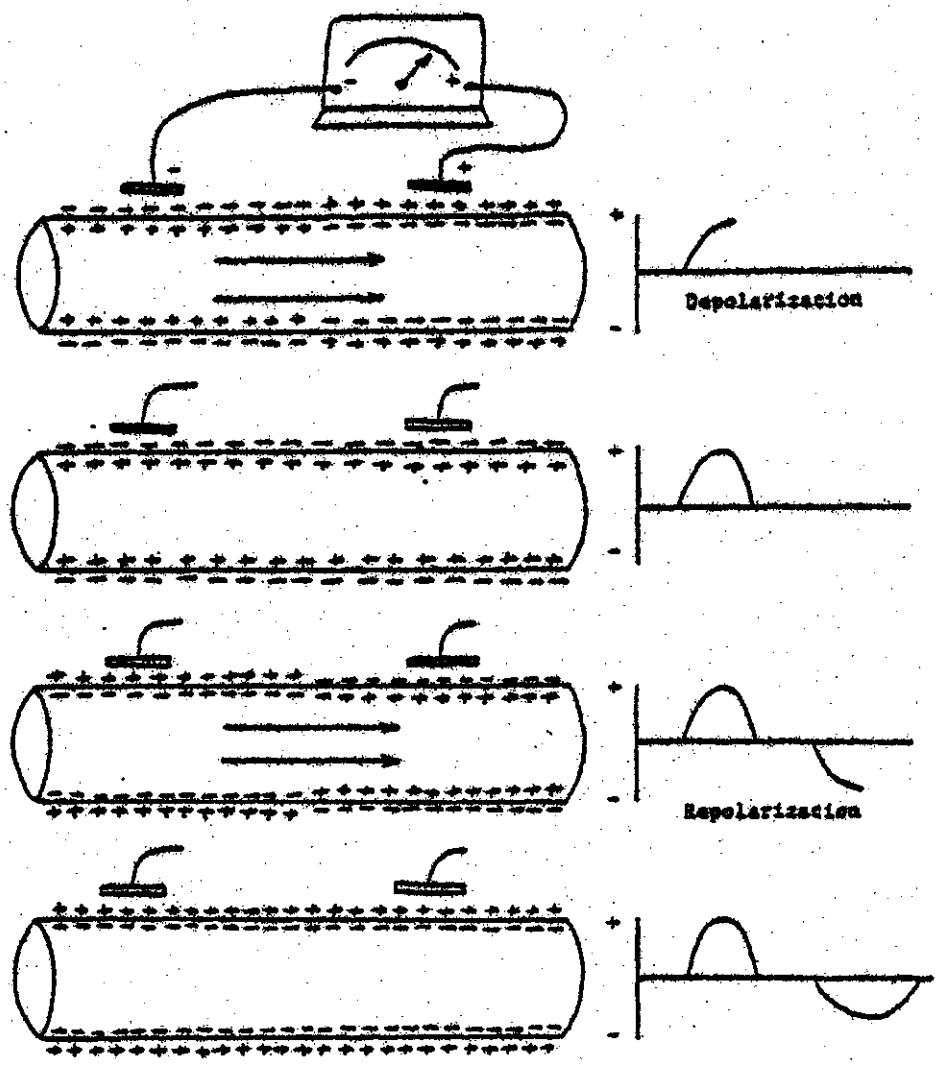


Figure 1-3: The Measured Electrical Potential of Muscle (from [24]).

the electrodes see a -ve charge flow towards the +ve electrode and a positive going wave is recorded.

A short time after the action potential passes, the physiology of the muscle causes it to repolarize again. During repolarization the muscle relaxes, and a +ve charge "flows" along the muscle. In this stage, therefore, a negative going

wave will be recorded at the electrodes.

These depolarization and repolarization waves can be measured when any muscle contracts, with the amplitude depending on the amount of muscle and the period depending on the speed of the action potential propagation.

### 1.2.2 ECG Lead Systems

The basic idea behind the ECG is that, since heart muscle behaves electrically as described in the last section, electrodes placed around the heart will record the depolarization and repolarization of the atria and ventricles. Since the electrical potential from heart depolarization spreads easily through the surrounding body tissue, it is possible to use leads attached to the skin rather than to the heart for this recording. Several systems for positioning ECG leads around the body exist, depending on exactly what information is to be recorded. The two most important systems are the Frank 3 lead orthogonal system used in vectorcardiography [48], and the 12 lead system used in most diagnostic electrocardiography. The latter system, being the one most commonly used by doctors, is described here.

The position of the electrodes used in the 12 lead system is shown in figure 1-4. These leads are usually considered in three sets.

The first set of leads is known as the bipolar limb leads. It consists of 3 leads, I,II, and III, measuring the potential between the electrodes marked L-R, R-R, and L-F respectively.

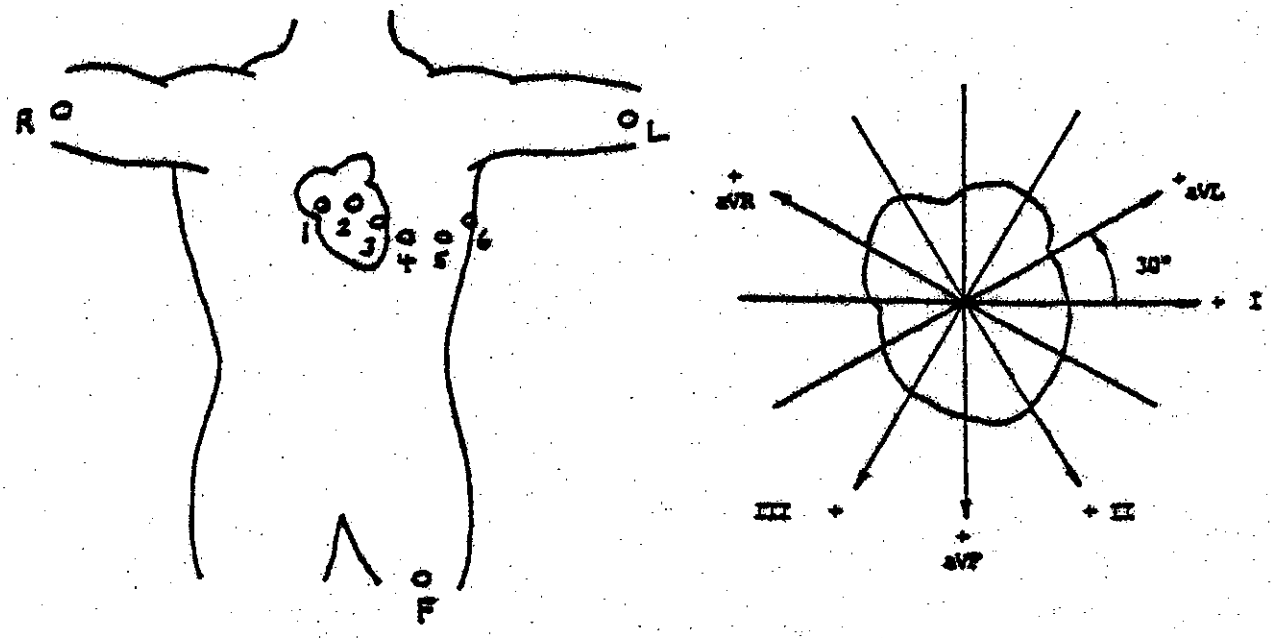


Figure I-4: The 12 Lead Diagnostic Electrode System.

Each lead will "see" that component of the heart depolarization along the line joining the electrodes. Thus, as shown in the diagram, this lead set records the heart action along three lines, 120 degrees apart in the body frontal plane.

The second set of leads is the unipolar limb leads: aVR, aVL, and aVR. These leads use the electrodes R, L, and F, respectively as the +ve terminal and the other two from the set R,L,F connected together as the -ve terminal. These three leads add information along another three lines in the frontal plane perpendicular to those of the bipolar limb leads.

The final set of leads is the precordial or chest leads. These six electrodes are placed at heart level on the chest approximately equally spaced from the sternum to the left side

under the arm, and are labelled 1 to 6 on the diagram. The leads are known as V1 to V6, and use the correspondingly numbered electrodes as the +ve terminal and the electrodes R,L and F connected together as the -ve terminal. A view of the heart is thus obtained along six lines in a plane through the body.

### 1.2.3 The Normal ECG

The 12 diagnostic leads give a fairly detailed three-dimensional view of the depolarization of the heart. To describe what is "seen", and therefore what information is contained in the ECG, it is enlightening to discuss in detail lead II, the lead which displays the most "classic" ECG waveforms. The effective positions of the lead II electrodes are given in figure 1-5, and it is useful to compare this arrangement to that in figure 1-3.

Before the heart beat begins, the electrical potential of the heart is unchanging and the lead II ECG records 0 volts. When the SA node produces an action potential, too weak to be recorded in its own right, a potential wave spreads over the atria and they contract. The depolarization accompanying the contraction is recorded as the P-wave shown in figure 1-5.

At the AV node, the action potential pauses and then rapidly moves down to the Purkinje fibres and the interior of the ventricular wall. Because the potential travels to the ventricles so quickly, their depolarization occurs in three very short stages. The left bundle branch allows the action

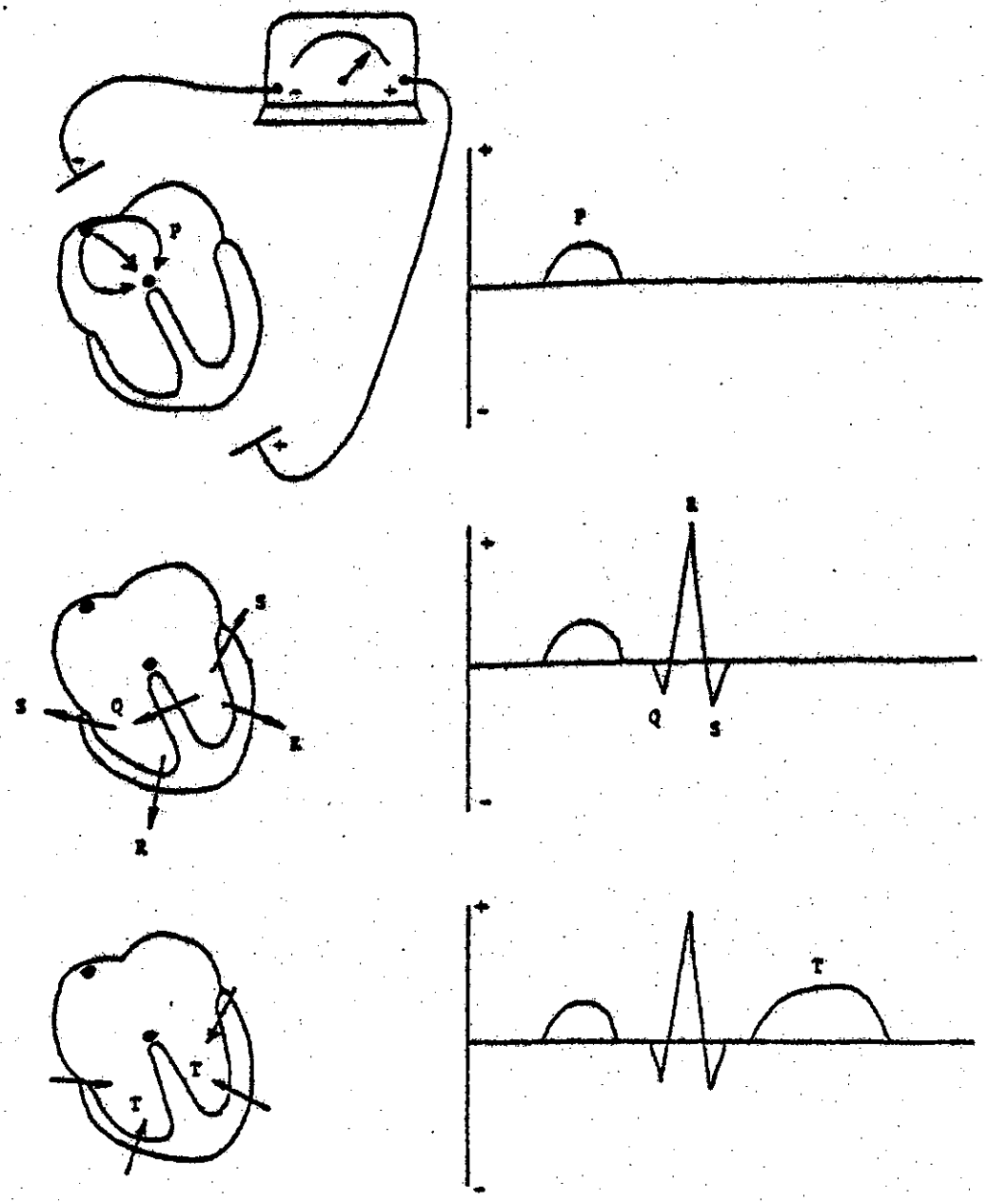


Figure 1-5: The Effective Lead II Electrocardiogram.

potential to reach the septum between the ventricles first, which depolarizes from the left to right ventricle, producing a small downward Q-wave in the ECG. The action potential then reaches the bottom portion of the ventricles simultaneously, and they polarize outward, producing a large positive R-wave,

due to the thick muscle involved. Finally, the upper portion of the ventricular wall contracts, producing a small downward S-wave.

Once the beat is complete, the ventricular muscle relaxes, repolarizing slowly from outside to inside, producing the T-wave in figure 1-5 (note this direction is opposite to that of the isolated muscle shown in figure 1-3). The atria also produce a repolarization wave, but it is usually small and falls on the QRS complex and is lost.

Therefore, a normally beating heart produces a lead II ECG like that illustrated in figure 1-6. Also shown in that figure is the same beat sequence as seen by the eleven other electrodes of the diagnostic ECG. The different deflection shapes and amplitudes are due to the different orientation of the leads with respect to the heart.

## 1.3 The Abnormal Electrocardiogram

From the last section, it can be seen that the ECG measures only the depolarization and hence the contraction of the heart muscles. However, by applying the knowledge of heart function from section 1.1, a great deal beyond mere muscle contraction can be deduced from the ECG. In particular, information is contained in both the position of each wave in the ECG (rhythm) and in the shape of each wave (contour).

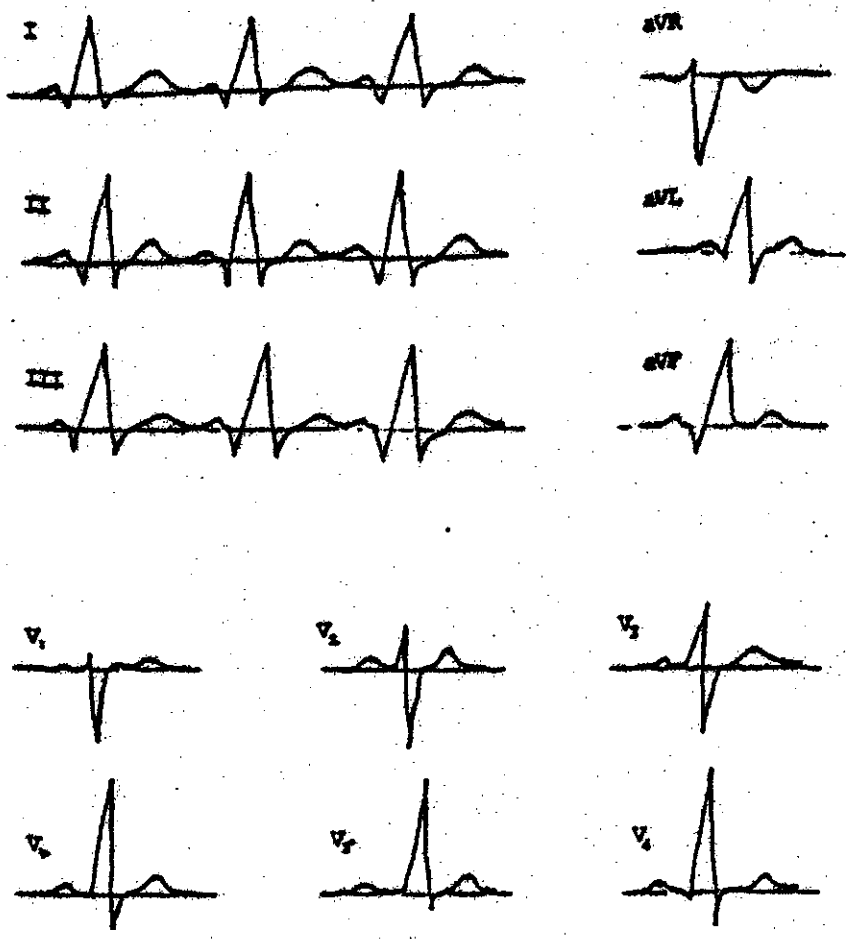


**Figure 1-6: The ECG of the 12 Diagnostic Leads.**

### 1.3.1 Contour

An example of an ECG with abnormal contour is given in figure 1-7. The clinical diagnosis for this ECG is right bundle branch block. This self-explanatory condition arises either congenitally or when some form of cardiac trauma or disease blocks the bundle branch in the right ventricle from conducting the action potential from the bundle of HIS. Its existence is deduced with the following model.

As the action potential leaves the AV node, it travels

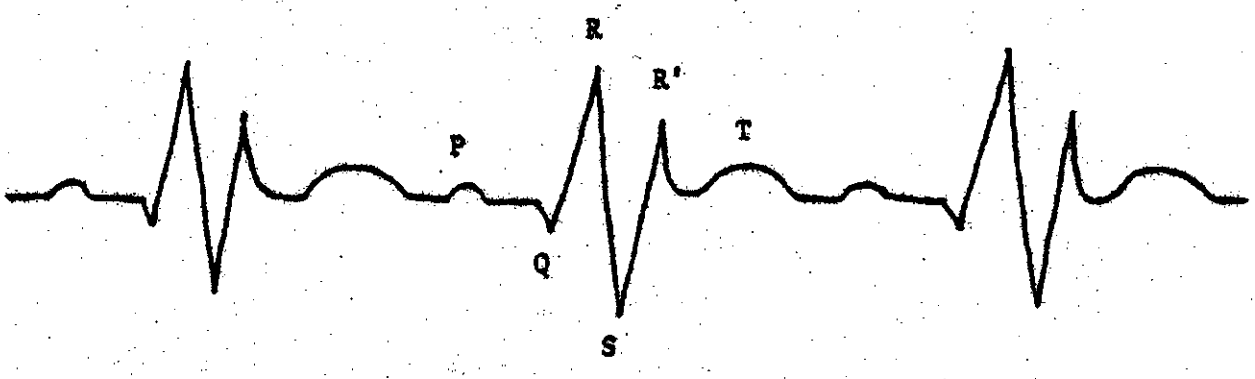


Figure 1-7: Right Bundle Branch Block.

along the bundle of HIS and down the left bundle branch to the left ventricle, but is stopped on the right side and cannot reach the right ventricle. Thus the septum and the left ventricle depolarize normally producing a Q and R-wave. The right ventricle, however, must wait until the action potential travels through the septum and reaches the Purkinje fibres. It then depolarizes, producing another, delayed R-wave, designated R'. The heart finally repolarizes, producing a T-wave and the ECG illustrated in the figure.

This right bundle branch block is only one condition which displays an ECG wave with abnormal contour. Other notable examples are Ventricular Hypertrophy and most forms of Infarct.

### 1.3.2 Rhythm and Arrhythmias

Figure 1-8 shows an ECG with abnormal rhythm. As with the contour example, this arrhythmia can be diagnosed as 2° AV Block, using deduction based on a knowledge of heart function. In 2° AV Block, the AV node intermittently fails to conduct an action potential from the atria to the ventricles. That this

is the correct diagnosis for the ECG in figure 1-8 is easy to see.

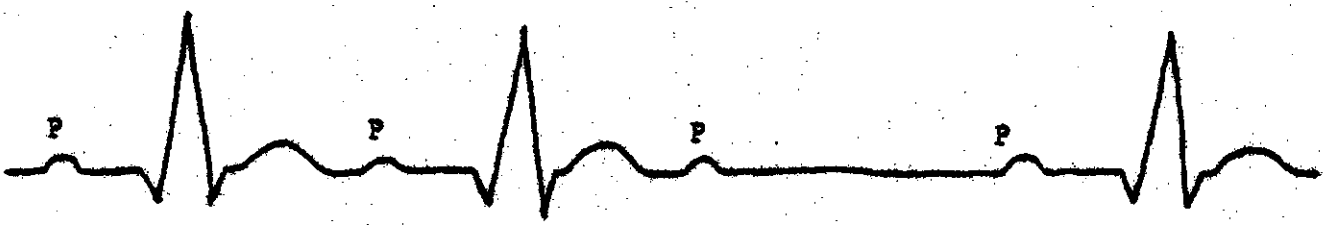


Figure 1-8: 2° AV Block.

Beats 1 and 2 in the figure are conducted normally and appear normal. In beat 3, however, the SA node triggers a depolarization of the atria, producing a normal P-wave. At the AV node, the action potential from the atria stops and no QRS-complex is elicited. The next beat, 4, is once again conducted normally.

This is a fairly simple example of literally thousands of arrhythmias described by cardiologists, some of which, like Wenkebach Syndrome and 3° AV Block appear very complex on the ECG.

### 1.3.3 Summary

In practice, the large number of patients involved ensures that, in a clinical environment, almost every conceivable problem with the heart will be encountered in some ECG trace. To make a proper diagnosis, the cardiologist must, in each case from normal to complex combinations of both contour and rhythm abnormalities, extract the useful information from the ECG trace and deduce exactly what is wrong with the heart.

## 1.4 The Automation of ECG Diagnosis

With the large amount of information in an ECG, and with the relative ease and inexpensiveness of obtaining one, it has become a very important and commonly used diagnostic aid. The increasing use of the ECG over the last century has brought pressure to automate the measurement and diagnosis of the waveform traces obtained.

This pressure comes for several reasons [11]. With the very common use of the ECG at the time of writing, up to one hundred are taken each day at the University of Saskatchewan Hospital. Even for a skilled cardiologist who averages a diagnosis of one ECG per minute, that means every day nearly two hours are taken up in this very mechanical task. Another problem is that, in cardiac care units (CCU) and emergency rooms, abnormalities in ECGs can be overlooked because doctors are not, and should not be, available for constant monitoring and diagnosis [55]. Burthermore, to increase the usefulness of ECGs, more and more measurements of wave duration, spacing, etc., must be made for each diagnosis, and cardiologists do not have the time to incorporate such complexities in routine reading.

Machines can overcome each of these problems. If a machine could be made to do most or all routine ECG diagnosis, the skilled cardiologist would be free for a considerable portion of each day to deal with more important tasks. In the CCU, the machine could be constantly monitoring for simple and

complex arrhythmias and contour changes, and could notify the attendant nurses immediately with a diagnosis, should they occur. A machine could also expand the usefulness of the diagnosis made, by incorporating more measurements with less human variability than a cardiologist can afford [39].

### 1.4.1 Outline of Thesis

The purpose of this thesis is to describe the beginnings of a system being constructed at the University of Saskatchewan as a research tool into computer diagnosis of ECGs. The research behind the thesis has three main goals:

1. To construct a useful qualitative model of human ECG diagnosis, so that it can be compared and contrasted with proposed machine algorithms.
2. To review the considerable literature existing on past attempts at computer ECG diagnosis.
3. To begin implementation of an actual computer program aimed at detailed ECG diagnosis.

The organisation of the thesis is as follows. Chapter 2 describes an original model of the procedure used by human doctors to diagnose an ECG. Chapter 3 gives a review of the literature on computer ECG diagnosis, with emphasis placed on similarities and differences to this human model, and in Chapter 4, the algorithms for the implementation of the first two stages of the ECG program this thesis is based on are described. Chapter 5 concludes the thesis with a summary of the work produced and a discussion of future extensions.

## Chapter 2

# A MODEL OF HUMAN ELECTROCARDIOGRAM DIAGNOSIS

*The eye sees only what the mind is prepared to comprehend.*
[Robertson Davies, Tempest-Tost, 1951]

When producing an automatic ECG diagnosis system, the first step is to have a theory or model of the way people make diagnoses. Examination of such a model gives insight into the way the automation may be accomplished by suggesting an algorithm of the most successful method of making a diagnosis: the human method. A model also gives a useful standard against which previous work on automation can be compared.

Any model of ECG diagnosis should, for several related reasons, be of the process most commonly used by doctors. The most important reason is that almost all currently available diagnostic information is based on long research into the most used procedures. Significant deviations from the human model will therefore require that a good deal of research be redone in order to derive diagnostic criteria for the new method. For almost the same reason, it is important that a human-like model be used, so that doctors can easily appreciate the system and

help it to gain acceptance. Finally, because a diagnosis program which closely follows human procedure can be expected to have human failings, when it does make mistakes they will be easy to find and correct.

This chapter presents the key aspects of an original model of human ECG analysis along with arguments for its plausibility from such diverse areas as medicine, psychology, education and computer science.

## 2.1 Introduction to the Model

One way to derive the essentials of the human ECG analysis process is by a consideration of those things which must occur if a diagnosis is to be made. This section discusses the aspects of the model which can be identified in this way.

The diagnosis problem can be expressed as a continuous abstraction from the original ECG trace, to a set of waves, to a set of clinical labels (P,QRS,T,etc), and finally to a diagnosis of the underlying heart problem. This view incorporates the human aspects of the procedure while introducing a useful mechanical quality. Any division of the continuous process of abstraction into separate pieces is somewhat artificial, but by taking cues from human behaviour and the types of knowledge required at each level, three independent stages emerge: reduction, labelling and diagnosis. Examples of the output of each stage are given in figure 2-1.

Before any other processing of an ECG trace can begin, it

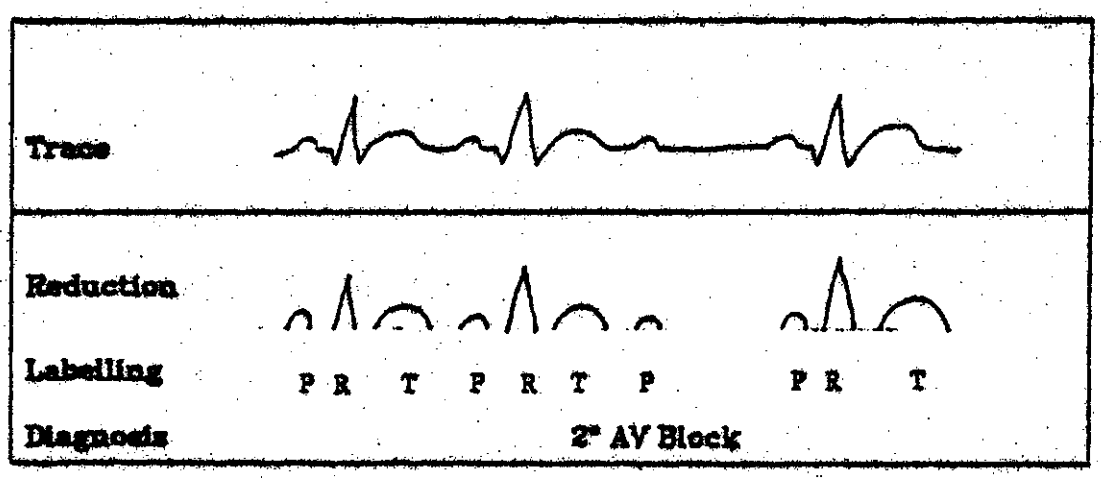


**Figure 2-1: An example of the stages of the human analysis model.**

must be broken down into some set of low level components, such as corners, peaks, waves, plateaus, etc. This leads to the first stage of the human analysis model in which the ECG trace is reduced from a continuous waveform to a collection of known primitive shapes or features. In the figure, this reduction is shown as a separation from the trace of only those features (i.e.,waves) which are necessary to make the diagnosis. In people, reduction is normally an "unconscious" process although it clearly must occur. Another important characteristic of this stage is that it requires knowledge of shape and size, but no direct knowledge of the functioning of the heart.

After simple features are extracted from the trace, they must be grouped together and a particular meaning, i.e. a specific heart action, associated with each feature or group. This assigning of meaning leads to a second diagnosis stage, matching exactly the human action of attaching clinical labels (i.e., P,QRS,T, etc.) to the waves in an ECG trace. As seen in

section 1.2.3, each label corresponds to a specific contraction in the heart muscles; therefore, this stage leads naturally to the type of output shown in the figure. For a doctor, the clinical labelling is always the lowest level of analysis which is consciously recognized.

The third and final stage of the analysis process is one of grouping the labels and abstracting the resulting string of heart actions into a diagnosis of any underlying abnormality. This is the area in human analysis which requires the most knowledge of the workings of the heart, and is therefore the subject of the greater part of all textbooks on ECG reading [13][18].

While the three stages of the human ECG analysis model are derived from a simple consideration of the kinds of processing and knowledge needed for each, it is compelling that almost identical stages have been discerned in several other areas of human expertise [73]. For example, the evidence Dale puts forth on the process of language learning in children [17] suggests the following model for language acquisition: discovering single words, identifying an object (meaning) with each word, and finally learning to use the words in abstract sentences. These models of ECG analysis and language acquisition are compared stage by stage in table 2-1.

While this correspondence between models of such apparently different human processes is not complete evidence for the correctness of the models, it is persuasive. In the next sections of this chapter, more evidence will be brought

| Model Stage | ECG Diagnosis | Language Learning |
|---|---|---|
| Sensory Input | ECG Trace | Spoken Language |
| Reduction | Singling out features and waves | Singling out sounds and words |
| Labelling | Associating waves with heart actions | Associating words with physical objects |
| Diagnosis | Abstracting from heart actions to final diagnoses | Abstracting from objects and meanings to form sentences |

Table 2-1: A comparison of human ECG analysis and language learning.

forward from similar sources to elaborate the three stages of the ECG model and to reinforce their plausibility.

## 2.2 The Details of the Human ECG Model

The three stage model of human ECG analysis introduced in the previous section has the interesting property that each stage is similar in function; a set of input data is processed into an abstracted set of output data. This similarity highlights several key concepts common to each stage.

The most direct procedure for generating output from input is the application of a set of rules which maps one into the other. Therefore, each stage in the human analysis model can be thought of as a set of procedures which go from trace to primitive features, features to clinical labels and labels to diagnosis of the heart condition. The output from each stage will often be refered to in this thesis as a set of tokens,

where a token represents, in an abstract form, some specific feature in the input. For example, the reduction stage may identify several pieces of ECG trace which together form a wave and will pass this information on as a single wave token representing all of the pieces at once.

A set of rules used in this way, to abstract from input to output, must have two important characteristics: it must be expressive, and it must be complete. In a qualitative way, expressiveness can be defined as the amount of information available in the input that the rules of the stage can actually extract. Therefore, for an ECG analysis process to be sufficiently expressive, it must be able to extract every important ECG feature from the input trace. For example, the first stage of the ECG model must be capable of identifying the characteristics of each type of wave likely to be found in the input. If it can only detect whether a wave is present at a particular point in a trace then, clearly, no diagnosis based on contour can be made by the later stages. Such an example is illustrated in figure 2-2.

This same figure shows that the rules for later stages in the model must also be sufficiently expressive. If the labelling stage is not capable of dealing with the shape of the R´, even an expressive reduction stage would allow an incorrect final diagnosis.

The other important characteristic of rules is their completeness. Completeness can be defined in an intuitive way as the amount of input data extracted which is actually passed

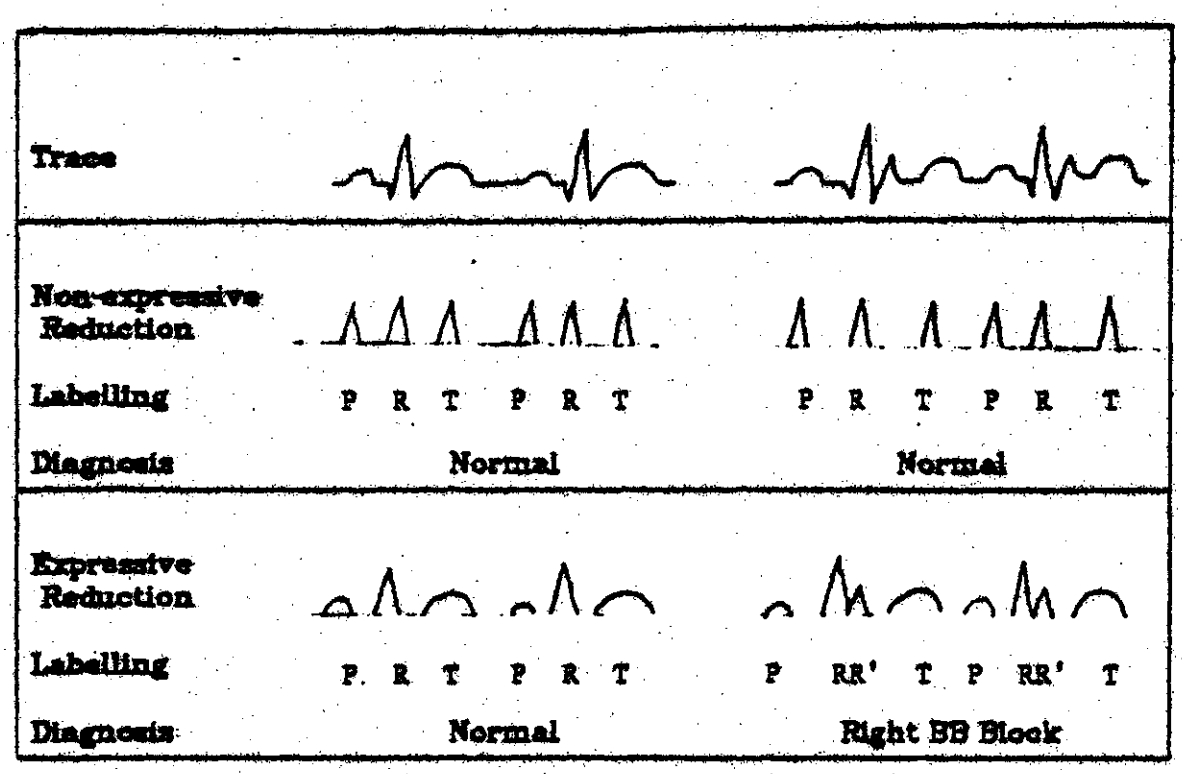


Figure 2-2: The problems of an insufficiently expressive rule set.

on to the output. Therefore, a perfectly complete ECG analysis system would not only perceive every important ECG trace feature, but would also report every relevant feature in its final diagnosis. For example, even for a sufficiently expressive set of rules, if all instances of an output choice are not passed on, an incorrect diagnosis can result. Figure 2-3 shows a clear example of this problem in the labelling stage.

As seen in figures 2-2 and 2-3, failure of a rule set to be either expressive or complete results in a loss of important information in a model stage and therefore an incorrect diagnosis based on incomplete information. The rest of this section explores the implications of these concepts --rule sets, expressiveness and completeness-- on the three stages of

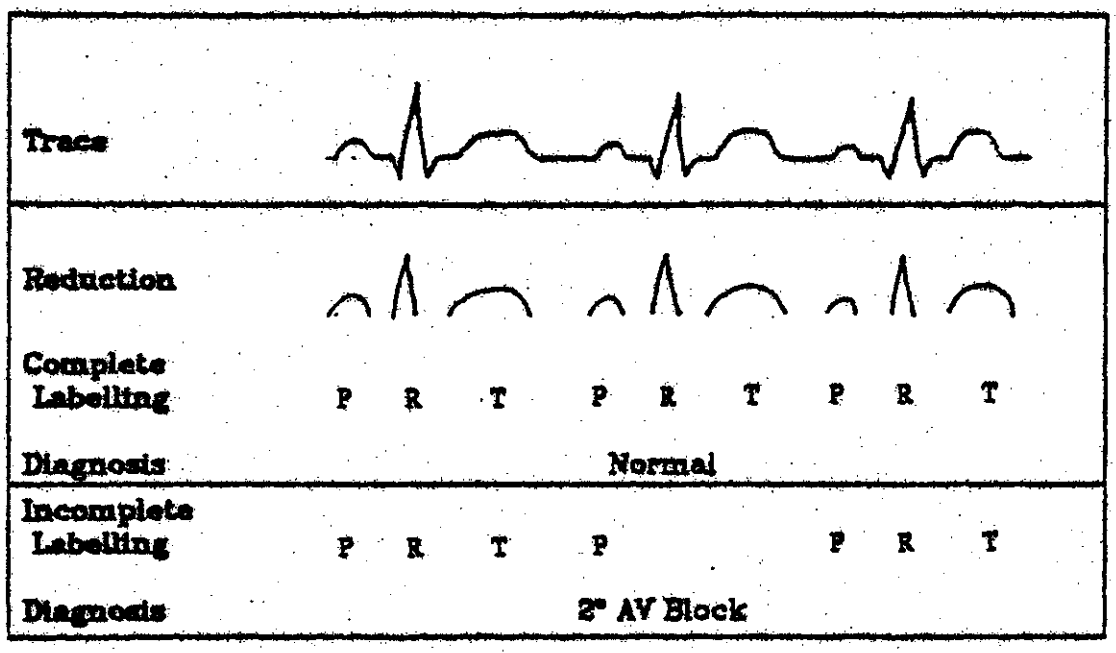


Figure 2-3: An example of the problems with an incomplete rule set.

the human ECG diagnosis model.

### 2.2.1 Reduction

The rule set for the first stage of the human ECG analysis model takes as input the type of ECG trace obtained using the normal measurement procedure outlined in section 1.2.2. As output, the stage must produce a set of tokens representing the clinically important waves in the trace. These output tokens must express both the position and the shape of each wave, since both of these properties are crucial to a correct diagnosis (see section 1.3.3). Every wave must also be found, as every heart action is important.

The type of knowledge necessary in the rules of the

reduction stage of the model is only to a small extent knowledge of the heart. While it is true that for the rules to be sufficiently expressive, all possible shapes of waves expected in the ECG must be known, the actual methods of discovering peaks, valleys and waves do not in any way have to relate to the heart. Once the expressiveness of the rules is defined, their application is essentially context-free, requiring no knowledge of what may be wrong with the ECG trace.

Direct evidence that ECG analysis begins with a reduction stage based on rules generating feature tokens from the ECG trace is scarce. However, evidence that general low level human perception works in this way is readily available.

Hubel and Wiesel dramatically demonstrated the existence of low level visual tokens with their work on cats in 1962 [32]. They investigated cells in the visual cortex with electrodes, and found that complex cortical cells exist. These are activated by a single well defined feature in a specific area of the visual field. For example, a particular cell might be activated by a single line at a specific orientation, but will remain inactive for the same line at a slightly different angle. In fact, similar work has shown that a complex layering of tokens is used in the abstraction of the visual field. The retinal ganglion cells in the eye are known to be grouped in such a way that each responds to a single circular spot of light. In the visual cortex, simple cortical cells respond to small line segments at specific orientation occurring anywhere in a larger area. There even exist "hypercomplex cortical

cells" which respond only when two lines meet at a particular angle [20][25].

In addition to this evidence for tokens, there are also experiments which demonstrate that the tokens are not hardwired at birth, but grow to be most expressive for the environment at hand. In Restak [52], several related studies in this area are described. In the most revealing, newborn kittens were placed in cages which contained only vertical or horizontal stripes. After several months, upon their release, it was found with electrodes that the "vertical cats" lacked cortical cells to respond to horizontal lines, while the "horizontal cats" lacked vertical cells. In the experimental cages, the cells which did exist were sufficiently expressive for the environment and the cats had no trouble, but in the outside world they lacked the low level expressiveness needed and had serious difficulties with their environment; for example, "horizontal cats" often walked into table legs because they could not see the vertical lines of the legs.

While the evidence for tokens and expressiveness exists for lower animals only, it can reasonably be inferred in man as well. In fact, another body of evidence exists which suggests that the building of tokens occurs in man even at cognitive levels too abstract to measure in single cells. In 1963, Segall et al. did a study using optical illusions drawn with straight lines [57]. The illusions were shown to members of "Western" and "Non-Western" cultural groups, and the number of times an individual was fooled was recorded. It was found that

Westerners were more often fooled by illusions containing acute and obtuse angles, while non-Westerners were fooled by right angle illusions. The conclusion reached was that Westerners interpreted the acute-obtuse angles as tokens representing right angles because of the large number of rectangles in our environment (i.e., buildings, windows, doors, etc.). The non-Westerners interpret vertical lines as tokens representing lines extending away in the horizontal plane, like a path in a field. There is also current research into theories of learning which suggest that this abstracting of tokens is crucial to any explanation of human reasoning abilities [54].

In summary, the first stage of human ECG analysis, reduction, consists of a set of rules to abstract the ECG trace into a set of waveshape tokens which are sufficiently expressive to support further diagnosis. Although evidence for their existence at higher cognitive levels is less conclusive, evidence for the existence of such tokens in low level human vision is quite strong. The evidence also explains an interesting phenomenon that occurs in classes in which ECG reading is being taught. Often a new student will make an incorrect diagnosis because an important wave is overlooked, and the wave cannot be seen until it is physically pointed out, and sometimes not even then. Perhaps this occurs because the rules for generating tokens that the student had before the class are not sufficiently expressive or complete for ECG reading, and therefore the wave in question can actually not be seen.

### 2.2.2 Labelling

The rule set for the second stage in the human ECG analysis model must take as input the token list from the first stage and produce as output a set of clinical labels associating a specific heart action to each token or set of tokens input. In this case, the expressiveness criterion requires that both the type of heart action and the particular contour of each type of wave, if more than one exists, be accounted for. Again, in this stage, it is crucial that every heart action be discovered, since each muscle depolarization, or lack of one, is important.

The nature of this labelling stage dictates that much more direct knowledge of the heart be known than in stage one. In particular, a knowledge of the way the heart produces an ECG is required so that each type of wave in the input tokens can be associated with the action producing it. Some knowledge of possible heart abnormalities is also necessary, as problems often produce irregularly shaped waves in unusual positions.

Psychological evidence supporting the activities in this stage is that given in section 2.2.1 for a continuous layering of more abstract tokens (heart actions) on less abstract ones (the input waveshape tokens). There is, however, more direct evidence that labelling occurs in ECG diagnosis: everyone does it. The fundamental teaching in ECG reading is to first label each wave in the trace a P,QRS,T,etc. [13][18][24]. Talking to a doctor about the content of an ECG trace will also invariably bring a discussion of the various waves displayed in it.

In summary, the second stage, labelling, in human ECG analysis consists of a set of rules to associate a heart action label to each token or set of tokens in the output from the reduction stage. Although the psychological evidence leaves the boundary between the first and second stages of the model somewhat arbitrary, the final labels the rule set produces at this stage are dictated by the actual practice of doctors. This existence of real clinical labels should help in machine analysis by supplying a readily available set of sufficiently expressive labels for use in further diagnosis.

### 2.2.3 Diagnosis

Like the previous two stages, the final stage, diagnosis, of the human ECG analysis model consists of a set of rules. These rules abstract an output diagnosis of the underlying heart conditions from the input set of clinical labels. In this stage, the output must first be able to signal every possible abnormality of the heart, and second, it must be able to express it in the correct accepted medical terminology. These two problems may at first appear to be the same, but it is entirely possible to create a set of rules that discovers specific problems in the heart, for example "a P-wave was not conducted through the AV-node", and then another set of rules to transform that statement of heart function into the diagnosis "2°AV Block". The completeness criterion also applies to this stage in which it is important to find everything the ECG trace holds, often even multiple unrelated diagnoses.

At this stage in the diagnosis, the knowledge required comes in two varieties. First, this is the stage which must contain a complete knowledge not only of the functioning of heart but also of all possible influences on it, like drugs, which can affect heart activity and hence the ECG produced. Second, knowledge is required about medical terminology and the proper way to report each specific abnormality.

The most direct evidence that this stage of diagnosis occurs separately as a set of rules in human analysis is the fact that the largest part of every ECG textbook is devoted to describing the rules used to generate a diagnosis from clinical labels [13][18]. In fact, at least one report has been expressly written by doctors as a detailed, rigid codification of hundreds of these rules for the use in machine analysis [40]. In addition to this direct confirmation, more general evidence exists in computer science work on expert systems, which suggests that tasks going from a well-known set of labels to a well-known set of diagnoses through a clearly defined problem space are most naturally expressed as a set of rules mapping the input to the output [14][30][33][45][60].

In summary, the third stage of the human ECG diagnosis model, diagnosis, is most clearly understood as a set of rules abstracting from clinical labels to a medical diagnosis. This understanding has even led to books on ECG diagnosis rules which should aid computer analysis a great deal.

### 2.2.4 Information Flow

The final important area of discussion about human ECG analysis has remained implicit to this point: the flow of information from one stage of the model to the next. With the previous description of the three stages of the model and the emphasis on completeness, it would be expected that the ECG trace be input to the reduction stage and completely broken down into tokens, passed to the labelling stage to be completely labelled and finally, given to the last stage for diagnosis. However, for real ECG traces this straight bottom-up information flow does not work because some features cannot be labelled, or even detected, until put in an unambiguous context with a diagnosis. For example, consider the two traces in figure 2-4.

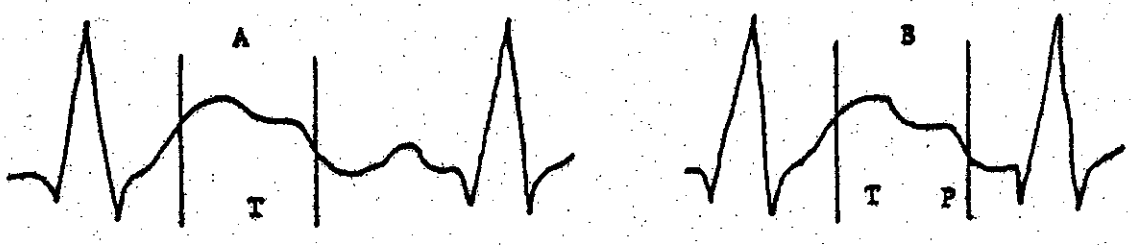


Figure 2-4: Labelling an ambiguous wave.

Both traces contain an identically shaped large wave with a notch in it. At the reduction stage, this shape would produce identical output tokens. However, at the labelling stage, it is important to make the distinction that the first is a T-wave with a notch, and the second is a T-wave with a P-wave overlapping.

To handle this ambiguous situation, and others like it, the human model adopts an information flow based on holding back ambiguous data at the lower stages of analysis, and then correcting for the incompleteness this introduces through the use of feedback queries once a context has been established. In the example above, for instance, a human would normally just assume both notched waves were T-waves at first. In the first case this would lead to a normal diagnosis because a normal P-wave can be identified separately. In the second case however, the diagnosis stage would note that no P-wave had been found between the T and the QRS, and would suggest possible AV Block. To confirm this, a feedback query would then be used to check the T-wave for a notch because in the context of AV Block such a feature would most probably really be a P-wave. In this example, finding the notch establishes the existence of the P-wave and the AV Block diagnosis can be rejected.

As another example of this information flow, consider figure 2-5. A quick look will show that the P-waves in this trace are difficult to unambiguously differentiate from the baseline and hence are easily missed. The usual human process would be to find the QRS- and T-waves and reach the conclusion that the diagnosis would be normal but for the missing P-waves. However, in order to be sure, the expected position for the P-waves derived from the labelling already done would be examined more closely through a feedback query, and the missing waves discovered, giving a correct diagnosis.

As illustrated in these two examples, feedback has the

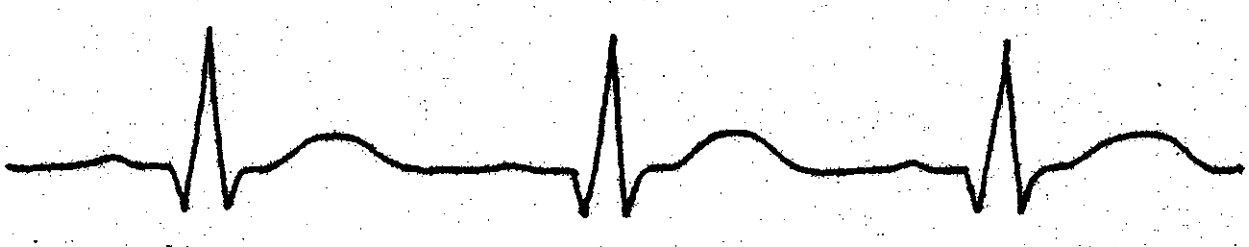

Figure 2-5: An example of the use of feedback queries.

general property of compensating for a lack of completeness in the lower stages of the analysis model. It should be noted, however, that feedback cannot correct for deficiencies in expressiveness. Clearly, if during the feedback query, the lowest perception level is incapable of detecting as small a wave as above, there is no other way to find it.

The use of feedback queries, therefore allows for a loosening of the restriction on completeness in the human analysis model. As long as the output of the reduction and labelling stages contain enough unambiguous data, they need not be 100% complete in their treatment of ambiguous data. However, this loosening cannot apply to the diagnosis stage as well, because there is no feedback into it.

This lack of feedback to the diagnosis stage leads to one final adjustment to the human ECG analysis model. A complete set of diagnosis rules is extremely unlikely, even among doctors, because of the thousands of heart problems, and therefore the excessively large size of any such set [40]. There is, however, a way to avoid this difficulty, and it is common practice in human analysis: the creation of artificial

feedback queries through learning not only a set of rules for common diagnoses, but also a detailed representation of the heart. The feedback is generated by deriving a trial diagnosis using the incomplete rule set, and then applying that diagnosis to the internal heart representation. If, under the assumptions of the trial diagnosis, the heart representation produces a hypothetical ECG trace with features that match those of the input ECG trace, the trial diagnosis is correct, and is reported. If the hypothetical trace does not match the real one, then the diagnosis is incorrect. Not only does this use of feedback thus discover inaccurate diagnoses, but by manipulating the heart representation until an identical hypothetical trace is produced, a report on what is actually wrong with the heart can also be made. In fact, this heart representation method of diagnosis is precisely that used in section 1.3.1, when the problems behind right bundle branch block and 2°AV block were discussed. It is evident as well that this is the way the accepted ECG diagnosis rules were worked out in the first place [13][18].

As an example consider the ECG trace in figure 2-6.

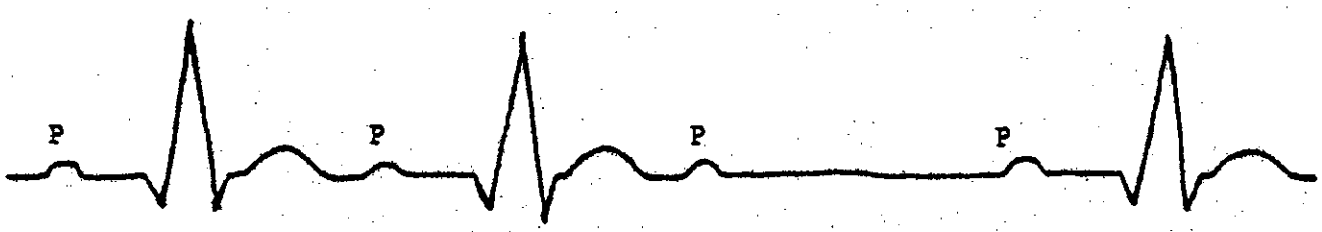


Figure 2-6: An example of feedback from a heart representation.

If the diagnosis stage lacked a rule for 2°AV block, the labels might point to a diagnosis of normal rhythm. With no feedback, this would have to be reported as the final diagnosis. However, if the diagnosis stage contains a facility for representing heart behaviour, this diagnosis could be used to generate a hypothetical set of normal clinical labels. The R and T-waves missing from the original trace would then be immediately evident through feedback queries, and a diagnosis of "a P-wave failed to pass through the AV node" could be reported.

Psychological literature carries a constant debate on the necessity and uses of feedback in human cognitive processes, and discussions which compliment that given here can be found in [8] and [46]. Computer science is also dealing with the problem of feedback queries [26], as well as with producing systems which allow representation of the functional aspects of processes like the heart [59].

In summary, the information flow in the human analysis model, illustrated in figure 2-7, allows feedback queries from the diagnosis stage to compensate for the lack of completeness that ambiguous information forces on the reduction and labelling stage. Feedback can also occur within the third stage, diagnosis, if it contains a method of producing hypothetical ECG traces using an internal representation of the heart. The completeness restriction on the diagnosis rule set is then relaxed as well. It must be remembered, however, that feedback cannot compensate for insufficient expressiveness at

any stage in the model because no amount of expectation will allow fundamental inadequacies in perception to be overcome.

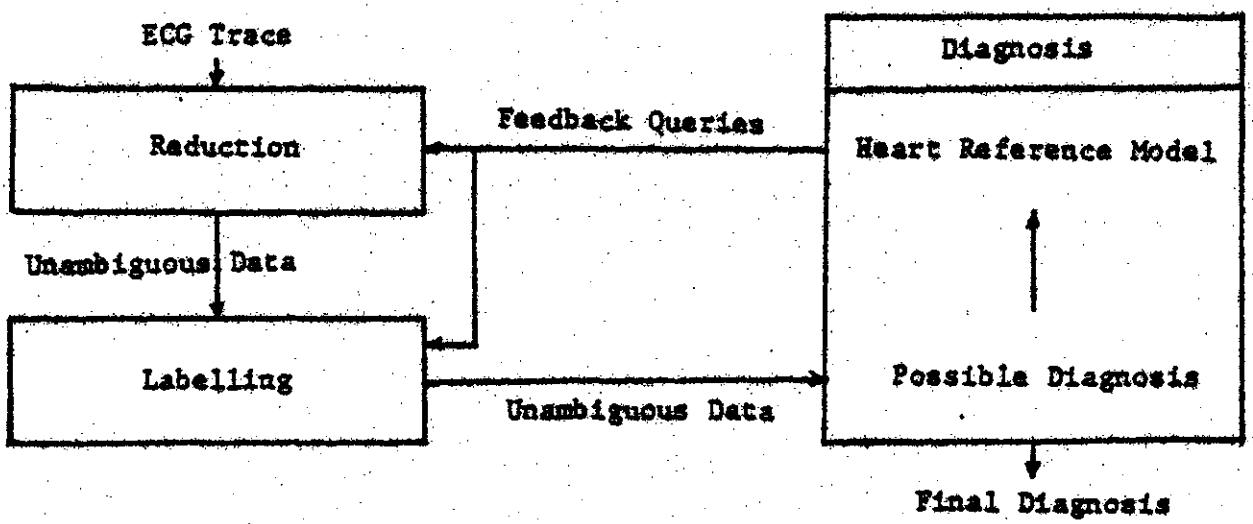


Figure 2-7: The human ECG analysis model information flow.

## 2.3 Summary of the Human ECG Model

In the previous sections of this chapter, the various aspects of the human ECG analysis problem were discussed, culminating in the three stage model: reduction, labelling and diagnosis. The reduction stage abstracts from the input ECG trace a set of tokens which must capture the shape and position of each wave in the trace. The labelling stage associates with each of the tokens, or set of tokens, a specific heart action. Finally, the diagnosis stage produces a final description of the underlying heart abnormalities using proper medical terminology.

Each stage consists of a set of rules which map the input into the output. These rules must be sufficiently expressive

for the output to include all the information required by the later stages of the model. To compensate for the difficulty in producing a perfectly complete rule set for ambiguous information, there are feedback queries from later stages to help eliminate the ambiguity. In order to bring this compensation to the third stage in the model, the diagnosis stage should contain, in addition to the rule set, a representation of the heart which can be manipulated to produce a trial ECG trace matching the input so a diagnosis can be either confirmed or worked out.

A summary of the model is shown in table 2-2.

| HUMAN ELECTROCARDIOGRAM ANALYSIS MODEL | | |
|---|---|---|
| **Model Stage** | **Output** | **Internal Structure** |
| **1) Reduction** | A set of tokens capturing the position and shape of each wave in the trace | A rule set to go from the input trace to the output tokens |
| **2) Labelling** | A set of labels (P,QRS,T) which associate a heart action to each token | A rule set to go from the input tokens to the clinical labels |
| **3) Diagnosis** | Diagnoses of the underlying heart problems in proper proper medical terminology | A rule set for a trial diagnosis.<br><br>A heart representation to compare the trial diagnosis to the input the ECG. |
| **4) Feedback** | Passes information about possible problems at one stage back to the previous stages for more complete elaboration | |

Table 2-2: Summary of the Human ECG Analysis Model

With the completion of the outline of the human analysis model, it remains, in order to produce a working automatic ECG

analysis system, to supply the actual rule sets in each stage. The rest of this thesis concentrates on that problem.

# Chapter 3

# A REVIEW OF EXISTING DIAGNOSTIC ECG PROGRAMS

*To remember the old is to add strength and background to the new.*
[Leslie M. Frost, Fighting Men, 1967]

The human ECG analysis model outlined in the previous chapter would lend itself easily to implementation as a computer automatic analysis system if the rule sets for its three stages were known. The first step in discovering what these rule sets should be is to review the algorithms behind existing diagnosis systems. However, the literature in the automatic ECG diagnosis field presents two problems. First, the field is very difficult to review in a comprehensive manner, because of the large number of programs which have been attempted, the differing goals of each, and the multitude of techniques used. A few attempts have been made in this direction, though, by considering the overall structure of the field rather than particular program details [4][12][34][37]. The second problem is that the successes of the various programs are extremely hard to compare, since their goals often differ and there is a lack of both meaningful standards and test results in the literature [3][27].

To overcome these problems, the literature reviewed in this chapter is grouped by general technique under the stages of the human analysis model. This approach was chosen with the realization that the philosophy behind any rule set produced from previous work would be governed generally by technique, while only the details would be influenced by particular implementation. The grouping under the human model has also been compressed from three stages to two because the importance of feedback between the first stages has meant that "all computer programs for analysis of ECG data are logically divided into two parts, pattern-recognition [reduction-labelling] and diagnostic evaluation [diagnosis]" [69].

## 3.1 Reduction-Labelling Algorithms

As outlined in sections 2.2.1 and 2.2.2, the reduction stage of the human ECG analysis model abstracts from the initial ECG trace to low-level tokens, and the labelling stage maps the tokens into a set of clinical labels. These tokens and labels must both retain the position of the relevant waves in the trace, and code their shape for later contour diagnosis. The artificial nature of the division between the stages and the importance of feedback from labelling to reduction, however, has meant that in most cases, existing analysis systems have developed the two as a single unit and therefore they will be reviewed together.

In the literature, there are literally dozens of

algorithms for these two stages, each differing slightly from the next. However, small details aside, they can for the most part be divided into two separate categories: slope and spatial velocity techniques, and token techniques.

This section reviews the important features of these two techniques and relates the characteristics of each to the requirements for the rule sets belonging to the reduction and labelling stages of the human ECG model.

### 3.1.1 Slope and Spatial Velocity Techniques

Slope and spatial velocity techniques for finding and labelling waves in the ECG trace are the most widely used and, although the implementation details vary from program to program, the same basic algorithm is used in each.* This section gives an outline of that algorithm.

The spatial velocity reduction-labelling technique breaks down easily into a set of rules based on the slope of the ECG trace:

- Calculate the slope or spatial velocity of the trace
- Find the position of the QRS-complex using a slope threshold
- Find the positions of the P and T-waves using another slope threshold

---

* For examples of the spatial velocity reduction-labelling techniques, see [10] [21] [28] [29] [38] [41] [50] [53] [56] [62] [64] [70].

- Code the shape of the QRS-complex

Only the details of each rule vary among programs.

The spatial velocity, V, is defined for the three lead Frank electrode system, with simultaneous traces x,y,z sampled at time interval $\Delta t$, as [62]:

$$V = \frac{1}{\Delta t}\sqrt{\Delta x^2+\Delta y^2+\Delta x^2} \qquad (3.1)$$

However, a single trace, x, from the 12 lead diagnostic system is often used instead, where equation 3.1 reduces to the absolute value of the slope of the trace.

The QRS-complex is found with this technique by comparing the spatial velocity to a threshold value; if it is over the threshold, a QRS is present (see figure 3-1). This simple method of extracting the QRS works because the rapid depolarization of the ventricles produces a steep slope in the ECG trace, even when the amplitude is very Low. QRS detection using a threshold is universal among programs using the spatial velocity, but the mechanism for determining the threshold varies. For example, it is fixed at 3.75 V/ms in [10], but is a floating 1/6 of the maximum trace slope in [50].

After the QRS-complexes have been found, the trace between adjacent QRS's is searched for P and T-waves. The usual method used here is to apply another, lower threshold to the spatial velocity which will catch all waves, as shown in figure 3-1. The T-wave is then labelled as the first wave found after each QRS, and the remaining waves between QRS's are labelled P

[38] [64] [70].

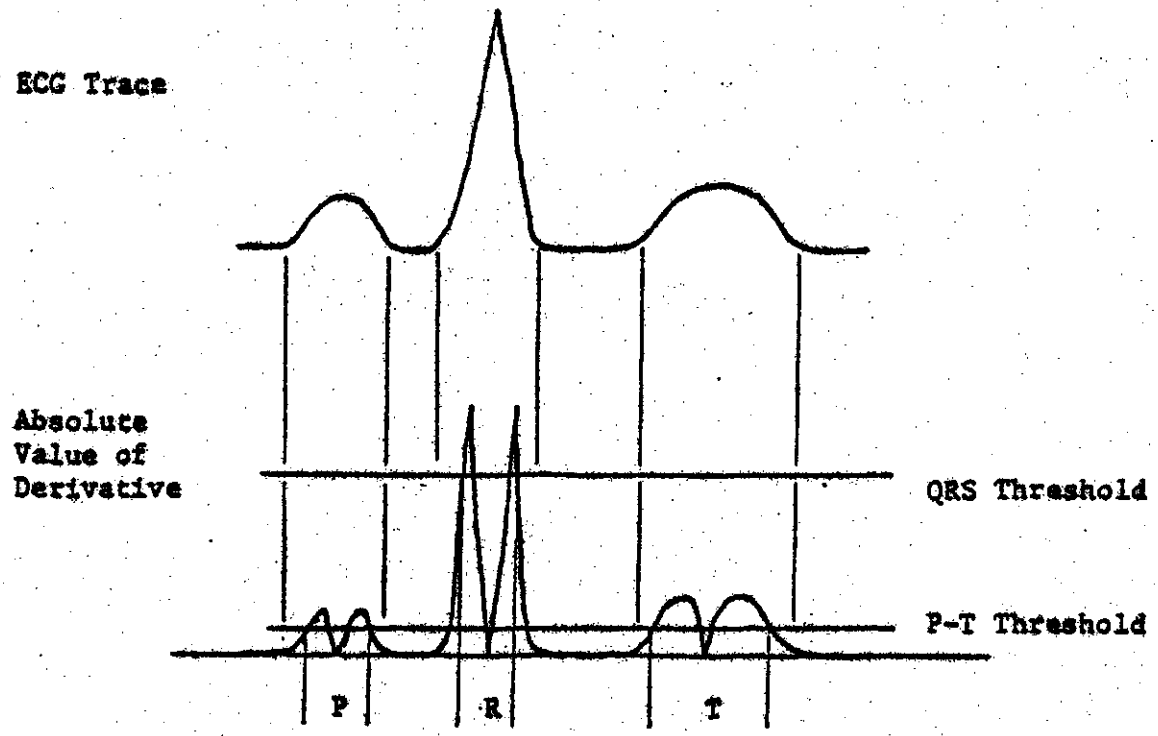


**Figure 3-1: Using spatial velocity and threshold techniques to identify the waves in an ECG trace.**

The final step in this reduction-labelling technique is to code the shape of the QRS-complex for later contour diagnosis. At this point, the many programs differ, each having a different coding scheme. Two representative examples, though, are [38] and [21]. In the former, shown in figure 3-2A, the shape of each QRS is defined by four values: $A_1$, the amplitude of the first +ve peak in the QRS; $A_2$, the amplitude of the following -ve peak; D, the time between these two peaks; and S, the average slope of the rising side of the -ve peak. In the other example, figure 3-2B, the shape is represented with a point by point coding scheme representing each sampled point within the QRS as a +ve, -ve, or zero slope. However, shape

coding schemes like these two fail to carry a true "feel" for the actual shape of the wave, and thus often fail to be useful in unusual QRS cases. Another problem is that these are not "diagnostically significant waveform parameters" [38]. The shape of the P and T-waves is seldom measured in these programs.

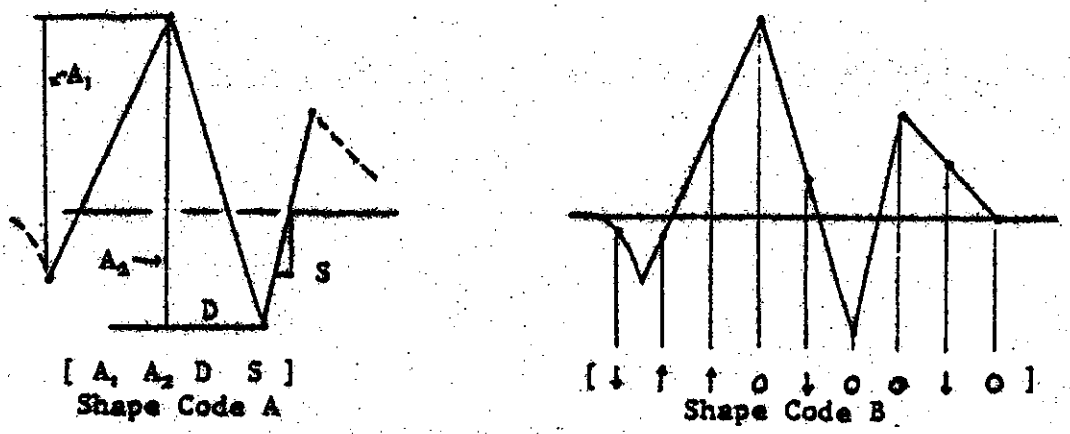


Figure 3-2: Examples of shape coding schemes. Diagram A is from [38] and B is from [21]. For details see text.

While the QRS threshold detection step of this algorithm is well tested and appears very successful [38][67], the rest of the technique has serious drawbacks which have been recognized in the literature. The first and most important problem is that since P and T-wave detection is based on thresholds, very small waves are difficult to detect unless the threshold is so low as to falsely detect noisy fluctuations of the trace. For the same reason, P-waves superimposed on T-waves, as shown in figure 3-3 are almost impossible to detect with simple thresholds, since the P-wave is unlikely to change the T-slope a significant amount. Furthermore, since P and T-waves must be extracted together (because of similar slopes)

and the first found is labelled T, a P-wave which occurs in an unexpected place, such as between the QRS and T, will often be missed or mislabelled.

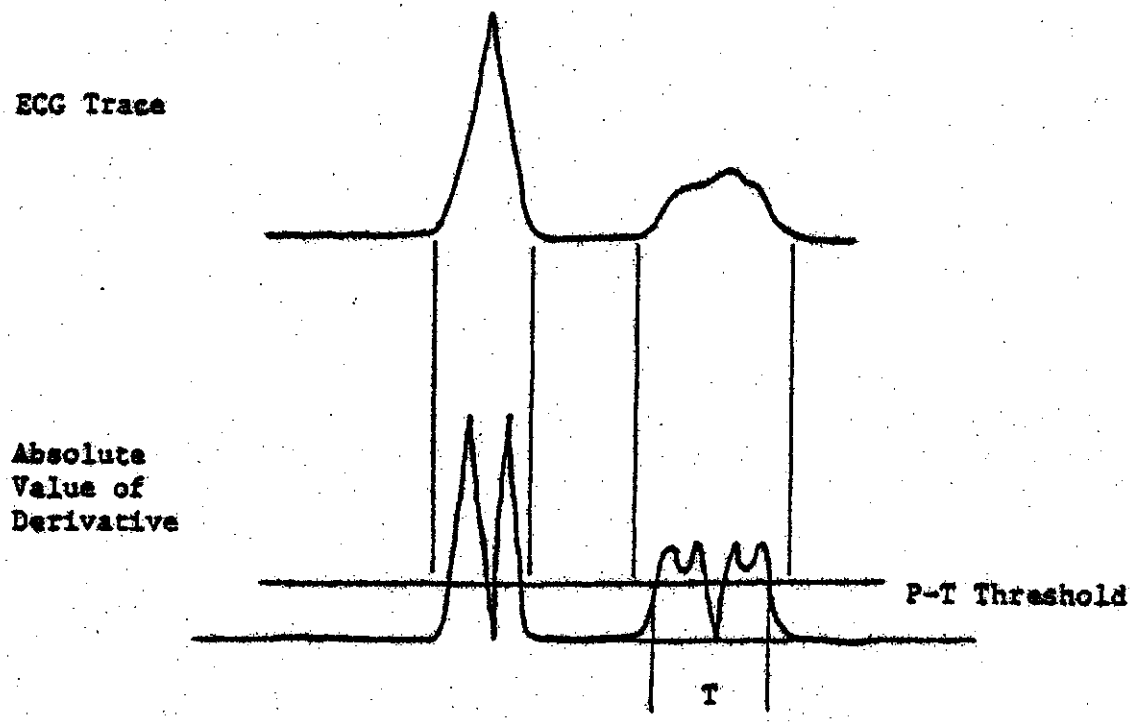


**Figure 3-3: A missed P-wave superimposed on a T-wave.**

Even with these problems the spatial velocity algorithm does fit very well into the human ECG analysis model, however. The slope and threshold criteria can be viewed as defining a set of low-level tokens which can immediately be labelled as clinically important waves. Feedback queries are also evident in the technique, since the T-wave is labelled as the first lower threshold wave found after the QRS, which must therefore be previously labelled. The problems with this technique can be phrased, in terms of the human model too. Since P-waves can often slip undetected below the thresholds, it is clear that the thresholds do not define a sufficiently expressive set of

tokens. This is also true of the shape coding examples which cannot deal with unusual QRS-complexes. This lack of expression of detail means that all the relevant information for a diagnosis is not available for later stages of processing.

### 3.1.2 Token Techniques

While not in the majority, a few ECG program designers have explicitly recognized the separate existence of both the reduction and labelling stages in ECG diagnosis [2][5][61][66]. Of these systems, the most human-like, and therefore most relevant in this context, is a refinement of the spatial velocity technique which uses the slope of the trace along with several thresholds to divide the ECG into a set of slope section tokens [2][5]. The slope tokens are grouped into segments which are then labelled as P, QRS, and T-waves.

The slope tokens are generated using the following steps:

- Calculate the derivative of the ECG trace
- Establish two +ve and two -ve clipping levels well separated from the noise (see figure 3-4A)
- Establish one +ve and one -ve clipping level to float just above the noise level of the derivative
- Flag a point on the ECG trace as significant if the derivative at that point crosses a clipping level in any direction (see figure 3-4B)

Once this method has been used to flag significant points on the ECG trace, the points are joined together by straight lines, each line being a slope token. The rest of the

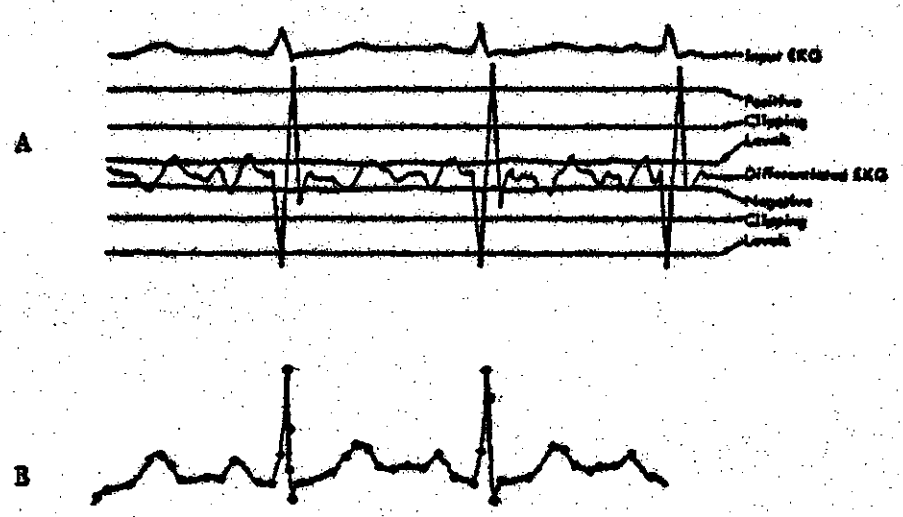


Figure 3-4: Flagging the end points of slope tokens, from [2]. Diagram A shows the clipping levels for the ECG derivative and B shows the points flagged by those clipping levels.

processing then sees an ECG divided into slope tokens, as in figure 3-5.

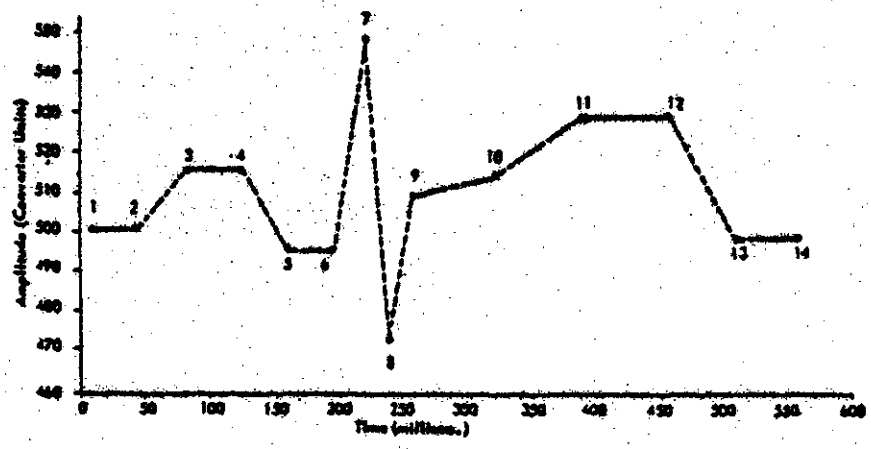

Figure 3-5: An ECG trace divided into slope tokens, from [5].

The next step in this technique is to group the slope tokens into segments or "bumps", each having slope sections with opposite sign at either end, as shown in figure 3-6. The segments that represent QRS-complexes are identified by using a dynamic derivative threshold, like those examined in the spatial velocity algorithm in section 3.1.1, and matching the

position of detected QRS-complexes with the position of a segment. Each QRS segment is then examined with a complex set of rules to see whether the segment next to it also belongs to the QRS. In this way, QRS-complexes are identified and "grown" to their proper extent. The rules governing the growth consist of several tests to see if the next segment to add is too small, too large, a P-wave, etc. an example of part of the logic involved is shown in figure 3-7.

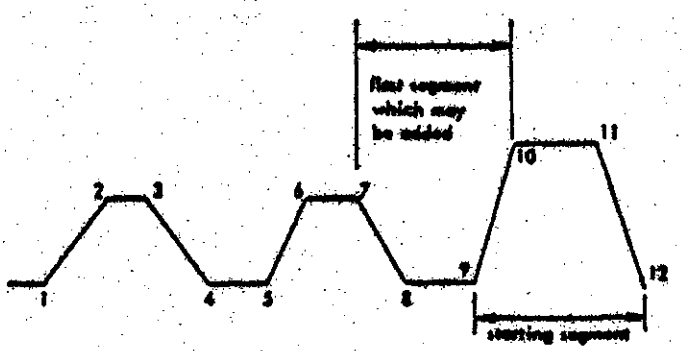


Figure 3-6: Grouping slope tokens into segments or bumps, from [5].

After all QRS-complexes have been identified and their onset and offset positions determined, a search is made for P and T-waves. This search consists of examining each segment between adjacent QRS-complexes and using a set of rules to decide whether it represents a P or T-wave. The final result of the complete procedure is a set of clinical labels describing the trace, as shown in figure 3-8.

This slope token algorithm appears very similar to the spatial velocity technique discussed in section 3.1.1 and, while it does retain some of the problems of that method, it incorporates several important improvements. The low-level

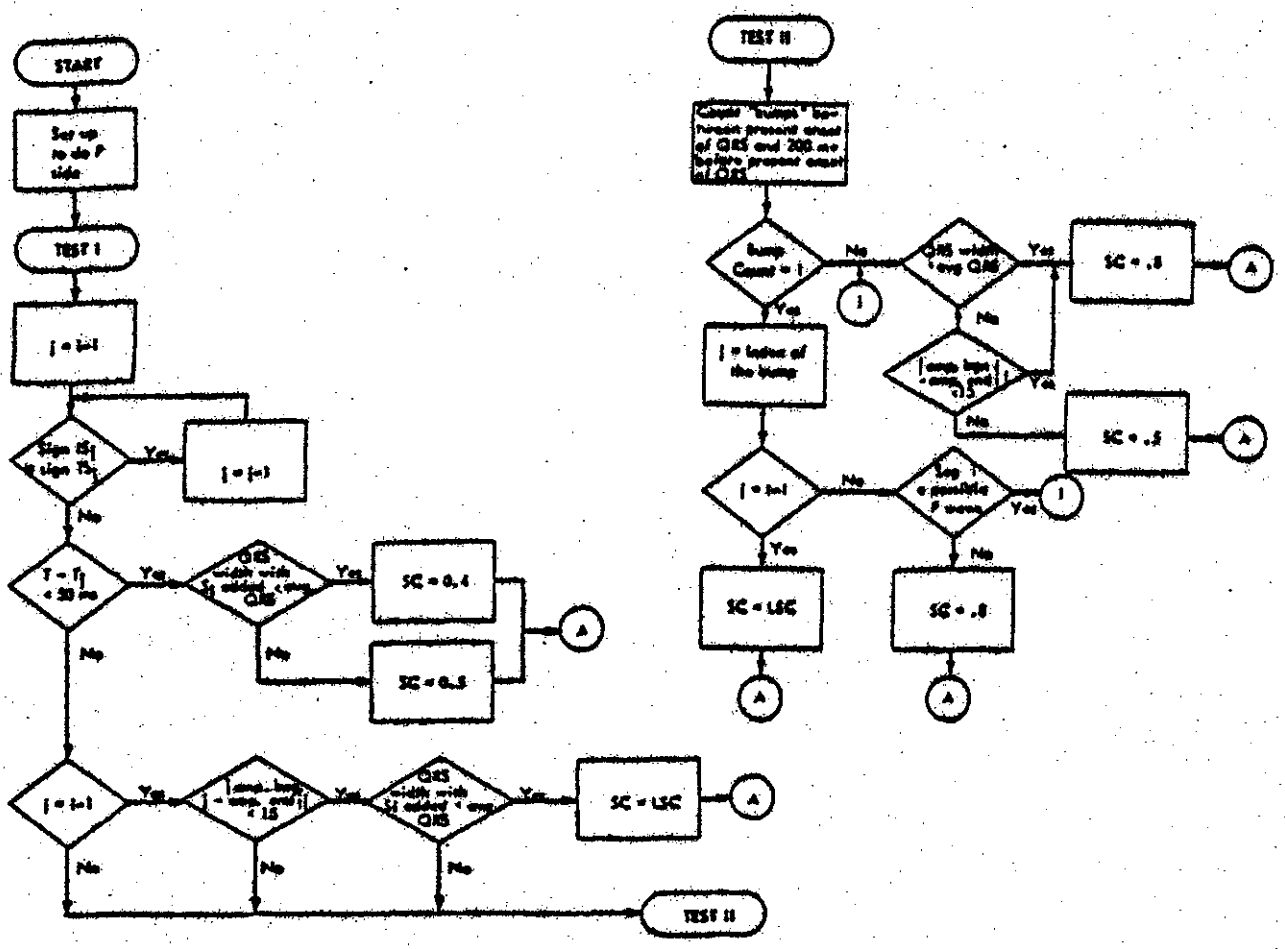


Figure 3-7: Some of the logic for adding a segment to the P side of a QRS-complex, from [5].

Figure 3-8: The final set of clinical labels for the slope token technique, from [5].

slope sections are similar to the spatial velocity tokens, since thresholds are used, but the labelling rules use much more of the information available in the ECG trace and thus are far more expressive. Another important distinction is that the slope segments or "bumps", by their nature, immediately code

the shape of the QRS in an intuitive manner. Finally, however, the use of thresholds to delineate the slope tokens means there still remains the problem of small R-waves slipping past undetected. This loss of information about the trace is evident in the "flatness" of the processed ECG trace in figure 3-5.

This slope token to clinical labelling algorithm also fits in well with the human ECG analysis model. The trace-to-slope-section and slope-section-to-segment steps in the procedure are an example of layered abstraction from one set of tokens to another according to a fixed set of rules. The method used to grow QRS-complexes and test P-and T-waves are sets of rules to abstract the clinical labelling from the final token set. The complexity of the method simply reflects the effort used to make the labelling rules as complete and expressive as possible.

Overall, then, this token technique is more expressive and more complete than the spatial velocity technique and therefore is a better set of rules to use as a simulation of the human ECG analysis procedure. This intuition is supported with the fact that the little comparative testing of the two techniques which exists shows the token technique to be superior [3][12][72]. However, the fact that small P-waves are missed at the lowest level suggests that, while this set of slope tokens is better than the spatial velocity set, it is still not sufficiently expressive for truly general ECG diagnosis.

## 3.2 Diagnosis Algorithms

In the human ECG analysis model, after the trace is reduced to a set of clinical labels, the third diagnosis stage abstracts from the labels a medical diagnosis of the underlying heart disorder. This diagnosis stage is universally recognized as a separate problem in the automatic ECG analysis literature.

As with the reduction-labelling algorithms, there exist many diagnosis computer programs, each slightly different from the others, and once again the literature divides easily into two major areas: decision trees and statistical diagnosis methods.

This section discusses the basis of each of these diagnosis systems, and relates their characteristics to the diagnosis stage of the human ECG analysis model.

### 3.2.1 Decision Trees

Of the diagnostic algorithms which have appeared in the literature, the most popular by far is the decision tree.* The decision tree algorithm is very simple in concept, and works the same way in all programs.

To build a decision tree, heart knowledge required for all the different diagnoses possible is assembled into a set of

---

* For examples of programs using decision trees, see [6] [7] [23] [31] [43] [44] [47] [50] [63] [68] [69] [70] and [71].

rules. The rules are then programmed into a fixed form which resembles in character that of a tree in graph theory. A diagnosis is made by travelling down the tree and, at each branch, choosing a direction based on the information contained in the input clinical label set. The only place the decision tree algorithms differ is in the exact specifications of their rules. These differences are necessary because each algorithm has a different shape coding scheme and therefore different input labels and decisions to be made. Two excellent examples of decision trees are shown in figures 3-10 and 3-9.

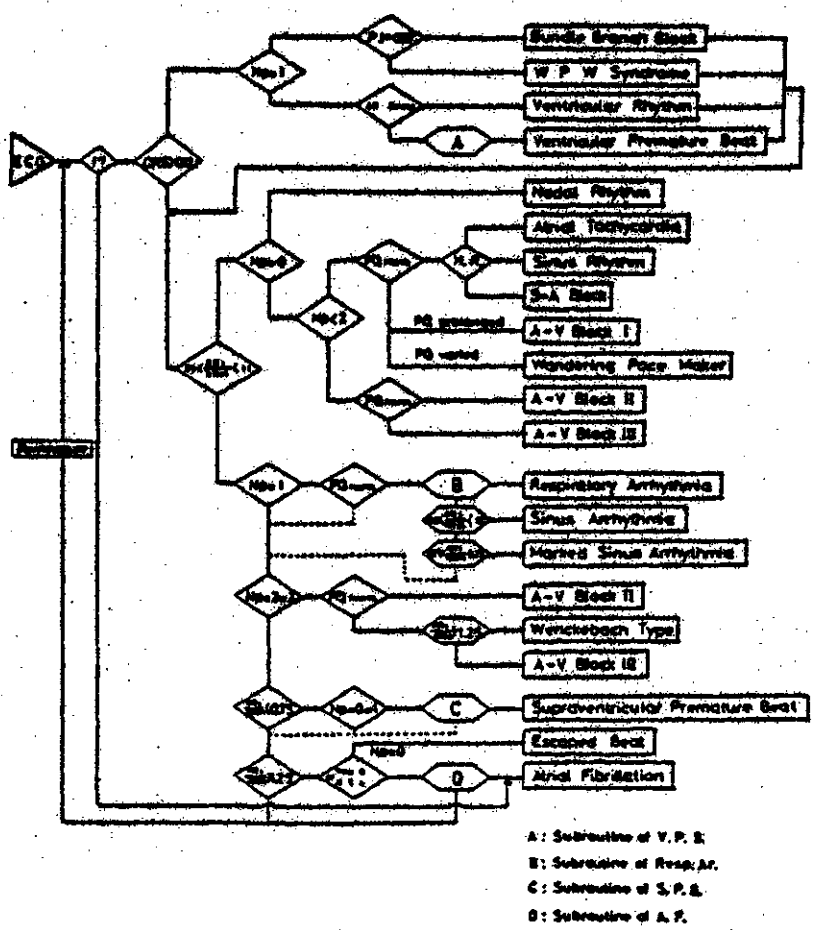


Figure 3-9: The decision tree used in the diagnosis section of the program in [70].

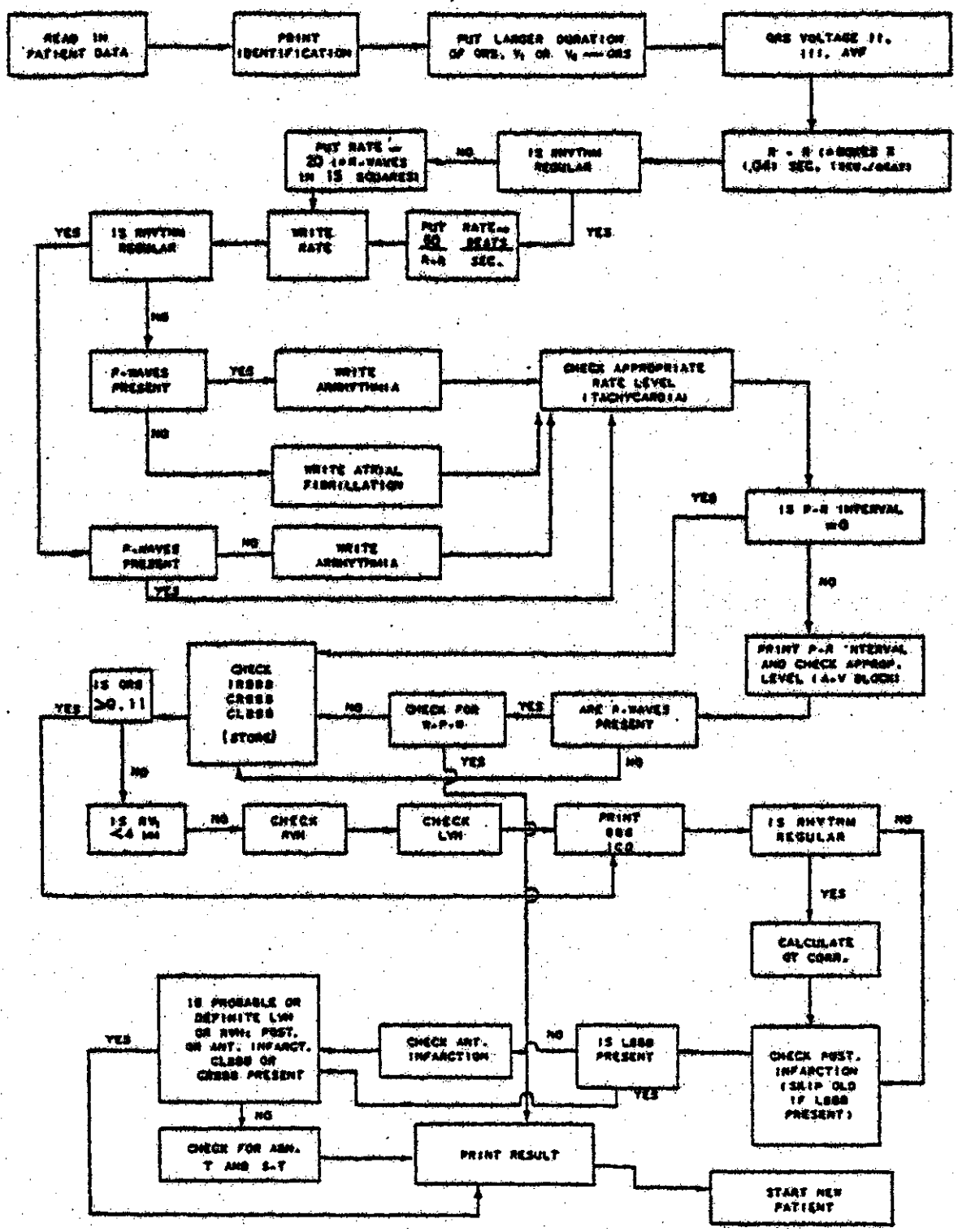


Figure 3-10: The decision tree used in the diagnosis program of [63].

The concept of a decision tree again fits smoothly into

the diagnosis stage of the human ECG analysis model. In particular, it is a set of rules to map the clinical label input set into a diagnosis. However, all the disadvantages of using only a rule set at this stage also apply, as discussed in sections 2.2.3 and 2.2.4. The first of these is the problem of producing a complete, expressive set of rules, with the large number of possible heart problems, and hence diagnoses involved. An example of this problem is shown in figure 3-10, in which the only arrhythmia diagnosis available is "arrhythmia". The other major problem is that of producing diagnoses from incomplete information and not being able to verify them before reporting. If a decision in the tree is based on a missing clinical label or a borderline threshold, the nature of the algorithm is such that the rest of the diagnosis carries on with little or no ability to recognize or evaluate the error. For this reason, the decision tree algorithm cannot compensate for a lack of expressiveness in earlier stages of analysis.

### 3.2.2 Statistical Methods

Statistical methods of abstracting a diagnosis from a set of clinical labels are used by fewer system designers at present than is the technique of decision trees, but they appear to be gaining popularity in the literature [22][23][36][49]. Of the few programs which do exist, the most influential uses the theory of Bayesian classifiers to make QRS-complex contour diagnoses [49].

The first step in the Bayesian classifier technique is to choose a reference point in each QRS-complex that appears in the clinical label input. This point is usually chosen as the point of maximum -ve slope, as that appears to be the most consistently reproducible [10]. The ECG trace is sampled at fixed positions on each side of this reference to produce a so-called "feature vector", representing the QRS-complex as shown in figure 3-11.

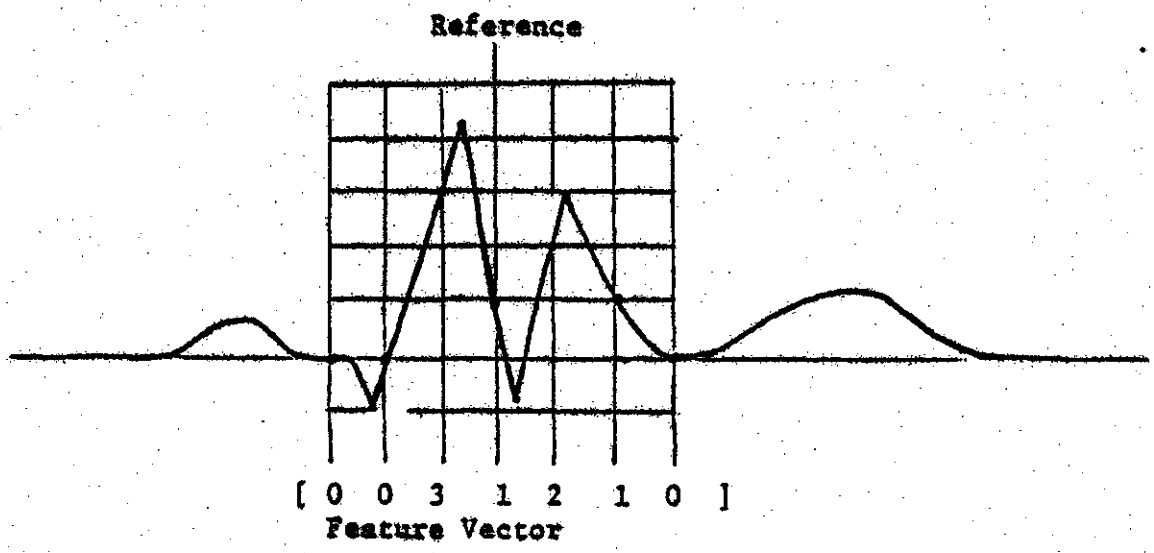


Figure 3-11: A feature vector for Bayesian classification of the QRS contour.

This feature vector is compared to previously determined diagnosis feature vectors using Bayesian classification methods [19], and the closest match becomes the diagnosis.

This diagnosis method has been shown to be very successful in limited cases [34], but in general, shows severe limitations. Bayesian classification, where the feature vectors for many diseases are quite similar, is very sensitive to the prior probabilities of each known disease vector, i.e. the probability of the disease occurring in the general ECG

population. Assembling these data accurately, along with producing the known diagnosis feature vectors for hundreds or even thousands of contour and arrhythmia diagnoses is a very difficult and, at present, unfinished task. Statistical methods of diagnosis are therefore generally limited to only QRS-complex contour diagnoses.

Although statistical Bayesian classification does not at first appear to be particularly human-like, it is possible that a similar type of processing does occur at cognitive levels above the most basic ones discussed in section 2.2.1. The diagnostic feature vectors, along with the Bayesian classification rules, can also be considered as forming an implicit rule set for diagnosis similar in nature to decision trees. This similarity is supported by the fact that the statistical method shares the same problems of completeness and expressiveness as the decision tree: difficulty in producing a set of rules (diagnosis vectors) for all possible heart abnormalities, and sensitivity of the diagnosis to missing or inexpressive lower level processing (i.e., an insufficiently detailed feature vector).

### 3.3 Summary

The human ECG analysis model divides ECG processing into three stages but, as seen above, the literature intertwines the first two, forming the reduction-labelling stage and the diagnosis stage. The reduction-labelling algorithms break the ECG into tokens and then abstract from the results a set of

clinical labels. The diagnosis stage abstracts further from the clinical labels to an understanding of the underlying heart disorder.

A review of the literature shows that the important algorithms used for the reduction-labelling stage fit very well into the framework of the human model. First, a fixed set of rules is applied to break the trace up into tokens, i.e. spatial velocity thresholds and slope sections; then another set of rules is used to abstract the tokens into a list of clinical labels. The literature also concludes that as the rules sets become more complex, and therefore more complete, the results of the labelling also get better [3][67]. However, as discussed in section 3.1.2, the use of thresholds still leaves even the best algorithms insufficiently expressive at the lowest token level.

The diagnosis algorithms presented in the literature share the same human-like qualities of the reduction-labelling ones. The concepts of both decision trees and statistical classification can be thought of as sets of rules to map the clinical labels into the final medical diagnosis. Both diagnosis algorithms also share the defect of being unwieldly, hard to assemble rule sets for any truly realistic number of diagnoses.

In general, then, it can be concluded that the best combinations of algorithms in the literature improve by using more complex and complete labelling and diagnostic rule sets

[34][72]. However, all the algorithms are limited in general by a lack of expressiveness at the lowest stages of analysis, which allows information to slip by unprocessed. This defect is noted by one researcher who states in a review article that one of the major "difficulties encountered included lack of reliability of P-wave recognition" [12]. This lack of P-wave recognition leaves the information at the labelling and diagnostic levels incomplete, so that even the most complete rule sets cannot make proper rhythm diagnoses.

A final major deficiency in the ECG analysis literature is that no system has yet been put forward for simulating a heart representation, allowing verification of a suspected diagnosis or even developing one from the representation's characteristics. As discussed in section 2.2.3, this aspect of diagnosis is crucial for the production of a complete, reliable analysis of all possible ECG traces. It does appear that work has begun in this area; however, it is not at present complete [59].

The summary of the literature in table 3-1 suggests that to improve on the algorithms that have already appeared, and to adhere more closely to the human ECG analysis model, the first work on an automatic diagnosis system should be on increasing the expressiveness of the reduction stage tokens. The next chapter describes new algorithms for the reduction and labelling stages, which put emphasis on increasing the expressiveness of these lowest level tokens, allowing more information to be available for the later diagnosis stage.

| ELECTROCARDIOGRAM ANALYSIS TECHNIQUES | | |
|---|---|---|
| Model Stage | Existing Technique | Characteristics |
| 1) Reduction<br>2) Labelling | Spatial Velocity | Insufficiently expressive with incomplete labelling rules |
| | Shape Tokens | More complete labelling rules but still lacks expression |
| 3) Diagnosis rule based | Diagnostic Trees<br><br>Statistical Classification | Both get better as the rules get more complete but suffer from difficulty in assembling a complete rule set |
| causal model | | No work yet complete in this area |

Table 3-1: A summary of the major ECG analysis algorithms from the literature, and their important characteristics.

# Chapter 4

# IMPROVED REDUCTION AND LABELLING ALGORITHMS

*Ce n'est que le premier pas qui coûte.*
[Mme. du Deffand, 1763]

The ultimate purpose of the work behind this thesis is to develop an automatic computer-based system for ECG interpretation which improves on the earlier systems discussed in the literature review of chapter 3. As stated there, this improvement must start by increasing the expressiveness of the earliest stages of processing and must keep the detailed information so obtained available for later stages. To achieve this improvement, the algorithms for automatic ECG analysis given in this chapter are designed to capture the essential features of the human model put forward in chapter 2.

Using a system design which incorporates the human ECG analysis model has resulted in two important design decisions. First, the algorithms are both logically and, in implementation, physically divided into the three stages of the human model: reduction, labelling and diagnosis. Second, basing the algorithms on the human model has influenced the

type and level of information flow between stages. In the discussion of the model in section 2.3, it was suggested that initially only clear, unambiguous information is passed from one stage to the next, with a more detailed picture of the ECG trace being solicited later through feedback queries once some initial diagnosis possibilities have been formulated. Using this information flow has resulted in the decision to have the reduction and labelling stages first output only those waves which have a reasonable probability of being correct, holding back ambiguous data until feedback queries from the diagnosis stage requests them. The influence of these design decisions will be seen throughout this chapter.

Achieving a complete diagnosis system based on the design goals of increased expressiveness and adherence to the important features of the human model is a very large task. Therefore, the work behind this thesis implements only the reduction and labelling stage algorithms. This limited implementation cannot, of course, make diagnoses, but its improvements in expressiveness can be illustrated , forming a firm basis for future expansion.

This chapter discusses a novel reduction stage algorithm based on three layers of tokens with fixed rules for deriving them from the ECG trace. The new approach described is more expressive than the systems in the literature, and supplies information to the later stages at three levels of abstraction, corresponding to the three layers of tokens.

The labelling algorithm described is less novel in concept, drawing heavily from the literature, but it is nonetheless unique, since it must work on the different tokens generated by the reduction stage. This algorithm makes available only the first output waves to the diagnosis stage at present because, without a diagnosis algorithm implementation, there is as yet no need for it to deal with feedback queries.

Also given in this chapter are the results of applying the implementation of the reduction and labelling stages to several sample ECG traces. Unfortunately, while adequate, this testing has not been extensive. The present difficulties in obtaining "interesting" (i.e., abnormal) ECG data in machine readable form, however,are being rectified, and further testing is an area of future extension.

Only outline diacussions of the algorithms used are given in this chapter, with the details of the actual implementation on the University of Saskatchewan College of Engineering VAX 11-780 in Appendix C.

## 4.1 The Reduction Algorithm

When designing a machine algorithm for the reduction stage of automatic ECG analysis, there are several important design considerations.

First, on a general level, the algorithm must be both expressive and complete. As discussed in section 2.2.1, this means that the algorithm must identify every whole wave, or

part of wave, in the case of overlap, which appears in the ECG trace. Both the position and the shape of each of these waves must be coded for further processing. From this expressive set, the reduction stage must pass to the labelling stage all identified whole waves for initial analysis, holding back the partial waves until feedback queries request them.

To achieve the identification and shape coding of the wave in the trace, the human model of ECG analysis suggests breaking the trace into tokens, each representing a wave or partial wave. This is done by using a set of fixed rules designed for the ECG context, with the literature review of chapter .3 suggesting some further restraints on the form of these rules. The most important restraint is that the rules should avoid application of thresholds whenever possible. This is due to two related phenomena. First, the variability in the amplitude and shape of ECG waves from person to person makes useful thresholds very difficult to set, and second, the variation of waves from beat to beat in a single trace means that some feature in a beat very near a threshold may produce different tokens and diagnoses as it varies slightly above and below the threshold.

Another important restraint on the rules for producing tokens, and on the tokens themselves, is that there should be no artificial coding schemes involved such as those for shape discussed in section 3.1.1. That is, all rules and coding schemes should produce tokens which are intuitively meaningful to human observers. This trait insures that the algorithm

will be more easily understood and accepted, and that there will exist proper standards for their interpretation.

Deciding on a particular algorithm to incorporate all of these design goals is difficult, but the algorithm presented in this section achieves the goals using three layers of tokens, each derived from the one below:

1. The ECG trace and its derivative in digitized form
2. Partial Wave Shape Tokens
3. Whole Wave Shape Tokens

The first layer of tokens, being the ECG trace, derives from the need to retain the trace in the computer as a final resource for feedback queries, to insure expressiveness. The second layer of tokens maintains expressiveness, but isolates each wave and partial wave in the trace for later processing, and the third layer of tokens extracts from the partial tokens only whole waves to pass on to the labelling stage.

The rest of this section describes the properties of each layer of tokens in detail, and shows examples of their use.

### 4.1.1 ECG Trace and Derivative

The first step in implementing an automatic ECG interpretation system is to put the ECG trace into a form which can be understood by the machine. In past systems, several analog data compression techniques have been used to aid data input [1][15][16][62], but because a computer is used in this system and computer memory and speed are no longer serious

limitations, straight digitization of the trace is used.

An ECG trace is digitized by instantaneously sampling it at a fixed time interval $\Delta t$, and in the computer the trace is a string of numbers representing the trace amplitude at each sample time. No actual method of digitization is described here because any reliable technique will do [58]. The data used as sample traces in this thesis were received, already in digitized form, from the Hemodynamics Laboratory of the University of Saskatchewan Hospital.

The entire derivative of the trace is also calculated and stored in the computer for use in defining the later tokens in the reduction algorithm and to be available for feedback queries. The derivative, f', is calculated at each sample point, n, assuming the trace, f, was sampled at interval $\Delta t$, using equation 4.1:

$$f'_n = \frac{f_{n-2} - 8f_{n-1} + 8f_{n+1} - f_{n+2}}{12\Delta t} \qquad (4.1)$$

This derivative formula is similar to the standard central difference formula, but is slightly more accurate [9] (for a more detailed discussion of the signal processing aspects of the reduction algorithm see Appendix A). It may also be recalled that the derivative is often used as a one lead spatial velocity in the systems discussed in 3.1.1.

Retaining the whole ECG trace and derivative in the computer as the first layer of reduction stage tokens has the

advantage of maintaining expressiveness by retaining all information in the trace. However, digitization has the disadvantage that certain fixed threshold values (i.e., sampling interval $\Delta t$, amplitude measurement resolution, etc) are integral to the nature of the process and can, if not properly set up, limit the basic expressiveness of the entire diagnostic system.

Examples of the digitized ECG traces used to test this system are shown in figure 4-1, along with their calculated derivatives.

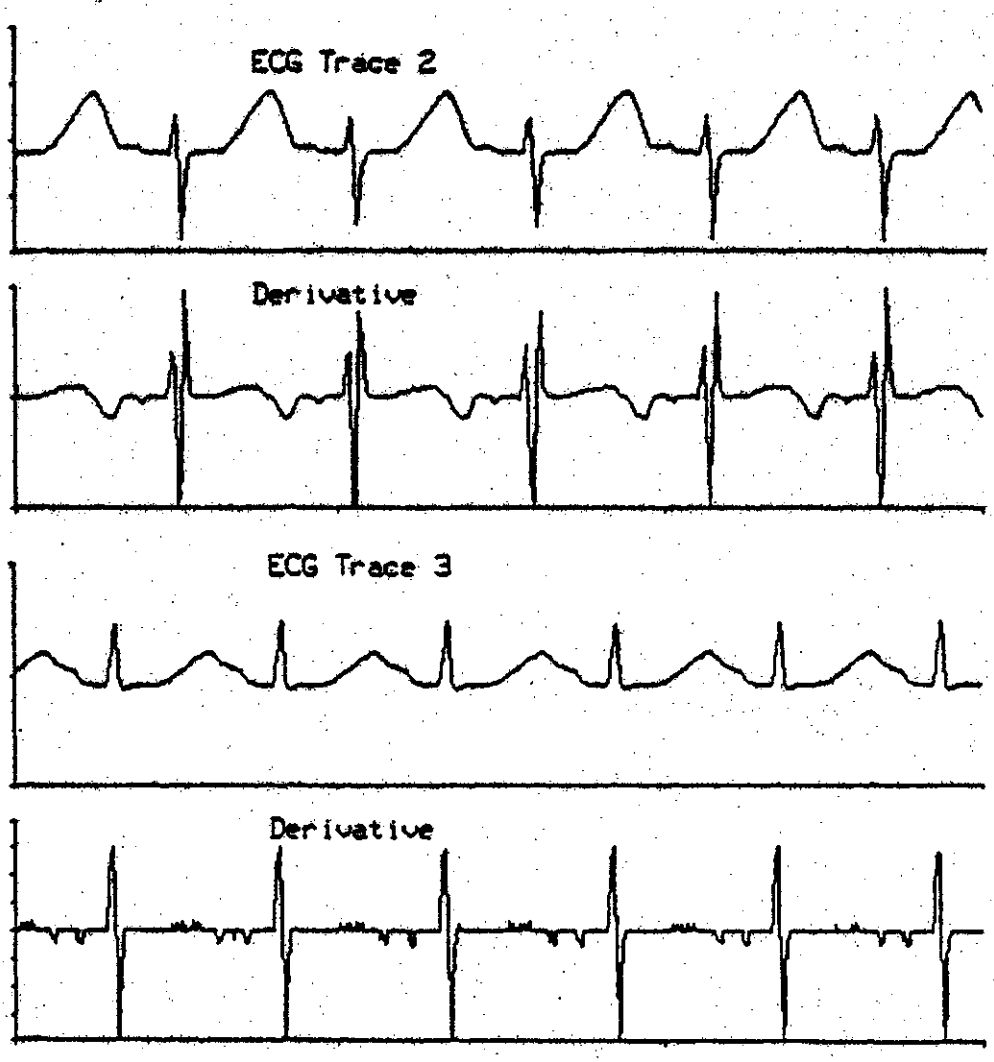


Figure 4-1: Examples of digitized ECG traces and their derivatives.

### 4.1.2 Partial Wave Shape Tokens

The second layer of reduction stage tokens is designed to overcome the problem that each wave in the digitized ECG trace is spread over a large number of sample points. To allow the later diagnosis stages to deal only with waves or parts of waves, it is advantageous to reduce each to a single token rather than a variable series of amplitude samples. The reduction, to remain expressive, must find every partial wave in the trace and also retain information about each wave shape.

The idea of a set of tokens at this stage to represent each ECG wave is almost unknown in the ECG literature. The nearest approach is the token technique discussed in section 3.1.2, which introduces the concept of breaking the trace into segments based on slope thresholds. These segments, however, do not directly try to code wave pieces and, because of the fixed thresholds, are not particularly expressive [5].

The tokens used in this implementation of the reduction algorithm, however, have been chosen to code directly partial waves as one of the seven possible shapes illustrated in figure 4-2.

Consideration of these seven tokens will show that no partial wave can appear in an ECG trace without causing at least one of the tokens to appear. For example, consider the three cases in figure 4-3. In the first case, two waves are quite close together and are separated by a rounded valley; however, each generates a type 3 token. In the other two cases, a short wave overlaps a larger one on either the left or

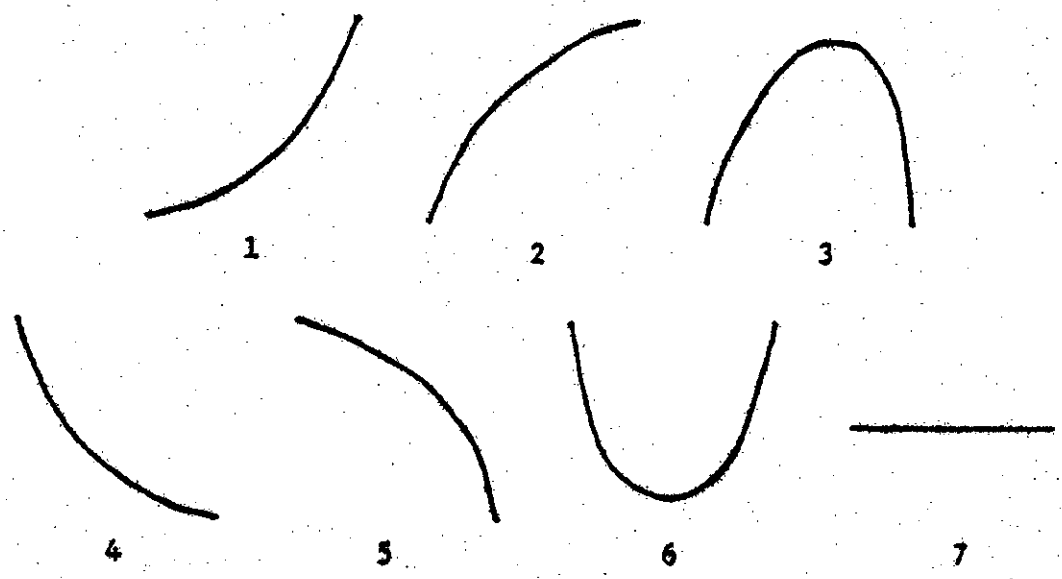


**Figure 4-2: The seven partial wave shape tokens.**

the right. The larger wave generates a type 3 token, while the smaller wave, if visible at all, must generate either a type 2 or type 5 token.

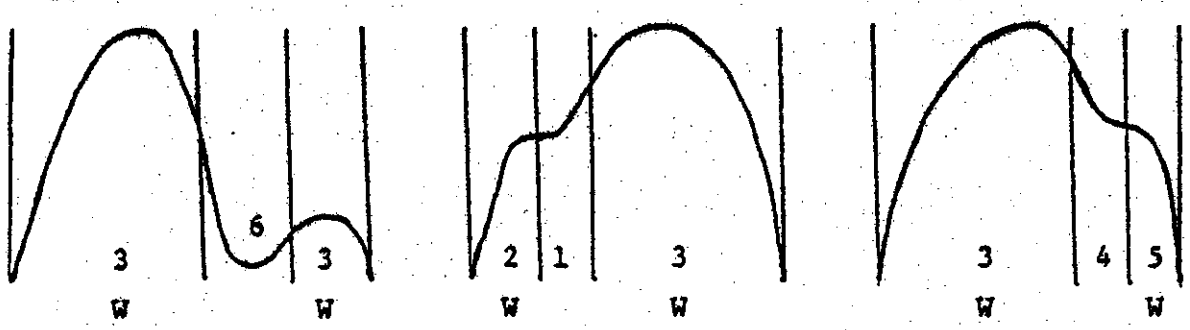


**Figure 4-3: Overlapping partial waves and the resulting wave shape tokens.**

This set of seven partial wave shape tokens is quite easily derived from a digitized trace and its derivative. The reason for this is the interesting property that each token begins and ends at a maxima or minima of either the trace or the derivative, and the token type is uniquely defined by the properties of these endpoints. Therefore, the algorithm for generating the tokens can be briefly stated in two steps:

- Find all local maxima and minima in both the trace and derivative (i.e., find the endpoints of each token)

- To each pair of endpoints found, apply one of 13 rules to identify which token type they define.

The token type rules are given in detail in Appendix C, but as an example, there are two rules for type 5 tokens:

1. If the first point of the token is a maximum in the derivative where the derivative is negative and the endpoint of the token is a minimum in the derivative where the derivative is negative, then the ECG trace is sloping downward with an increasingly negative slope for the duration of the token, and hence it is a type 5 (see the third case in figure 4-3).

2. If the first point in the token is a maximum in the trace and the endpoint is a minimum in the derivative, then the ECG trace slopes downward with increasingly negative slope, and hence is a type 5.

The advantages of this method for generating partial wave shape tokens are many. First, there are no thresholds involved, so no waves in the trace can slip by unnoticed (for real traces, however, the electrical noise ultimately limits the expressiveness possible at this stage. See Appendix A for a discussion of this). Second, because every wave in the trace must generate at least one token that also codes its end points and orientation, this layer of tokens remains both sufficiently expressive for further diagnosis and complete. Finally, and perhaps most importantly, each of the seven wave shape tokens is intuitively meaningful to human observers. Therefore, the different combinations of tokens that might occur in any ECG trace can be accommodated in later diagnostic stages using easily understood criteria.

There are also two minor disadvantages which are closely related. The generation of the tokens depends on "local" properties of the ECG trace, such as maxima and minima, and therefore, the actual tokens which a wave generates may vary from beat to beat in the same trace, as shown in figure 4-4.

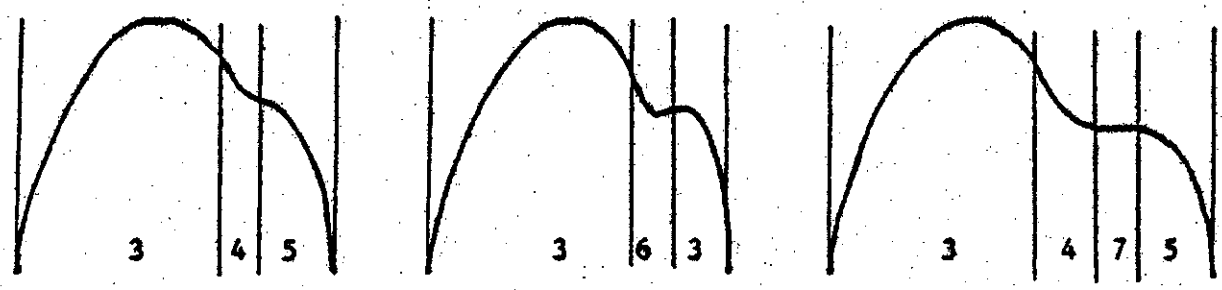


**Figure 4-4:** Various partial wave shape tokens for very similar ECG traces.

This variation in labelling is not as serious as that which may occur with thresholds, however, as no waves are actually missed, but it aggravates the other minor disadvantage of the procedure: many tokens are generated, such as the 4,6,7 in the figure, which do not necessarily represent waves or partial waves. Unfortunately, this disadvantage cannot be overcome, since it is possible that in the first example in figure 4-4, the overlapping partial wave is actually a downward one, and is in fact properly represented by the type 4 token rather than the 5.

The details of the implementation of this token generation algorithm are given in Appendix C, and examples of actual output are shown in figure 4-5.

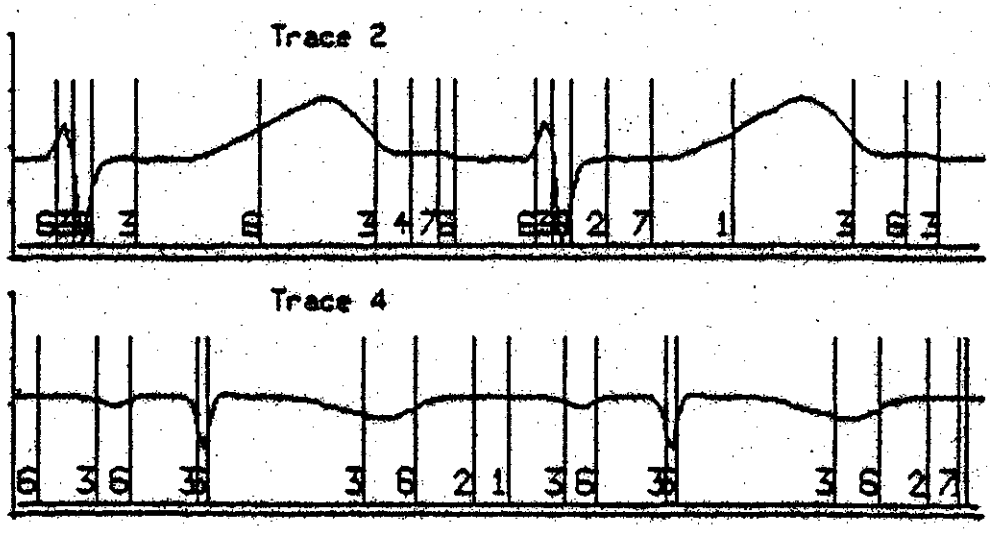


**Figure 4-5: The partial wave shape tokens generated by actual ECG traces.**

### 4.1.3 Whole Wave Group Tokens

The third and final layer of reduction algorithm tokens is designed to eliminate the ambiguous extra tokens from the partial wave shape token list. In this way, only whole waves will be retained for output to the labelling algorithm in accordance with the initial design decision to use the human ECG analysis model information flow.

Consideration of the seven partial wave shape tokens shows that the only ones which unambiguously define whole waves are types 3 and 6. Therefore, generating a new list of tokens for the ECG trace which contains only whole waves reduces to the task of keeping all type 3 and type 6 partial wave tokens. In reality, it is more complicated than this, because there are two other cases similar to type 3 and 6 which define peaks that may be whole waves. These two cases are illustrated in figure

4-6.

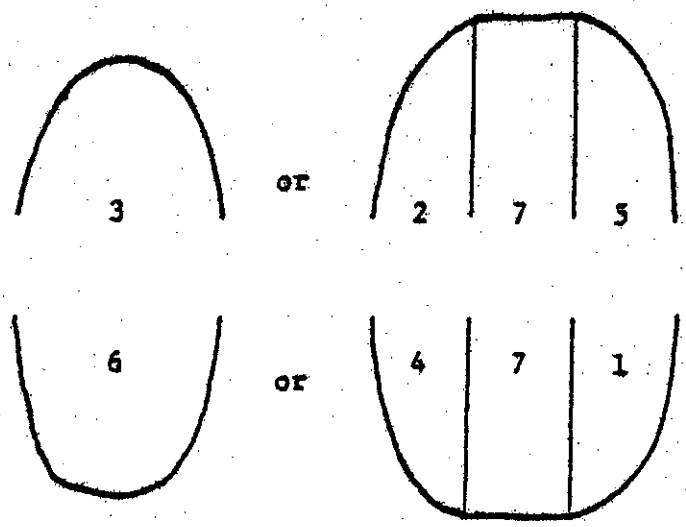


Figure 4-6: Cases of partial wave shape tokens which define whole wave group tokens.

The details of the algorithm for generating whole wave shape tokens is given in Appendix C, but in general it finds all cases of the four defining sets of figure 4-6 in the partial wave shape tokens, and expands the endpoints of each token outward until they meet (see figure 4-7).

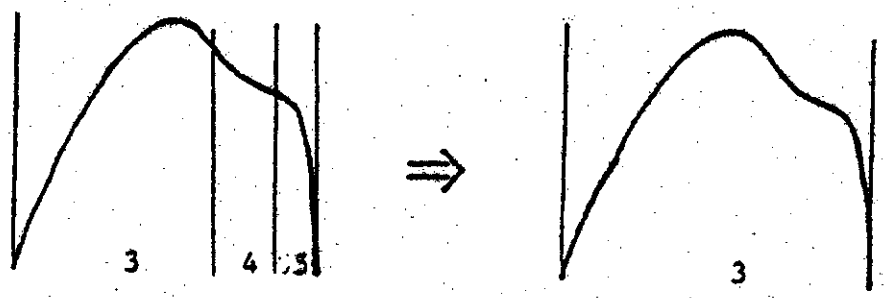


Figure 4-7: Removing ambiguous information with whole wave group tokens.

The advantages of generating this layer of tokens are twofold. First, it codes directly only those waves in the ECG trace which have a reasonably unambiguous labelling, immediately enforcing the design goal of initially passing along only good data. Second, as with the partial wave shape

tokens, these tokens have an easily understood intrinsic meaning which allows later interpretation to be based on well-known criteria.

The major disadvantage of this algorithm is that it is incomplete, since partial waves are not represented. However, if only whole waves are to be passed on to the labelling stage, this disadvantage cannot be avoided. In addition, the expressiveness of the whole reduction algorithm is not compromised, because the partial wave shape tokens and the input ECG trace are both available to feedback queries.

Actual whole wave group labellings of sample ECG traces are shown in figure 4-8.

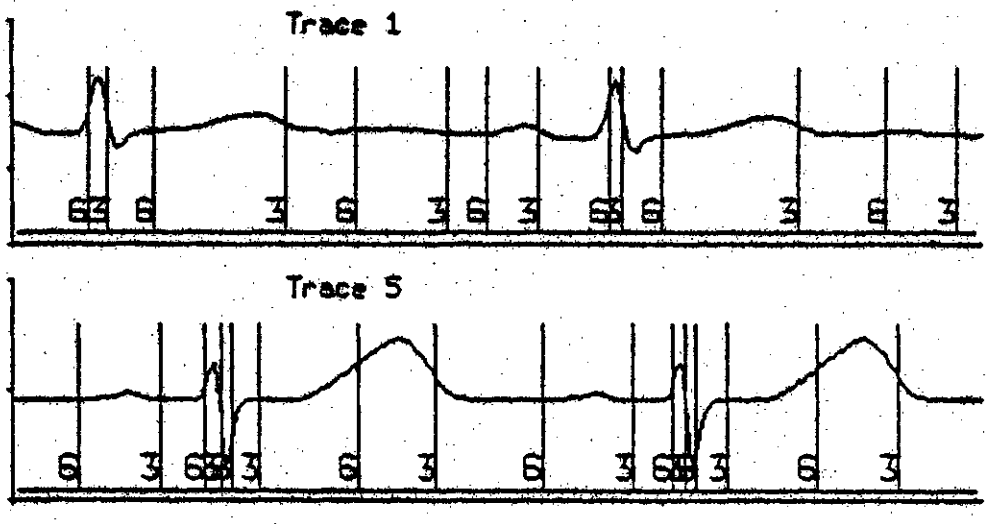


Figure 4-8: Examples of whole wave group tokens.

### 4.1.4 Summary

In summary, the automatic reduction algorithm developed for the work behind this thesis consists of rules to generate

three layers of tokens: the ECG trace and derivative, partial wave shape tokens, and whole wave group tokens.

The ECG trace and its derivative form an expressive and complete token set, and they are kept available as the lowest level of information for feedback queries from the labelling and diagnosis stages.

The second layer of partial wave shape tokens is again an expressive, complete set, but in it each token represents a single wave or partial wave. The seven tokens used also code every possible ECG wave in a way which is intuitively meaningful to human observers. However, some ambiguous tokens are generated at each wave boundary.

To eliminate the ambiguous tokens, the final whole wave group algorithm searches through the partial tokens to find only those which represent whole waves. This set of tokens thus codes only those features in the trace which are sure to be individual waves.

The labeling and diagnosis algorithms should work initially with this incomplete whole wave token layer to formulate possible diagnoses for the ECG trace. Details to confirm or reject these possible diagnoses can then be obtained from the lower level tokens through feedback queries.

## 4.2 The Labelling Algorithm

The overall design decision to pattern the automatic interpretation algorithm on the human ECG analysis model means

the responsibility of the labelling algorithm is initially to assign a specific heart action (P,QRS,T,etc) only to unambiguous waves in the ECG trace. Further consideration of the human model also shows that to allow proper diagnosis later, the labelling algorithm must be both expressive and complete. By basing the algorithm on the three layers of tokens generated at the reduction stage, expressiveness is maintained, but by initially labelling only unambiguous waves, completeness is not. However, this lack of initial completeness is designed to be overcome with feedback queries from the diagnosis stage. Only the initial labelling algorithm is described in this section, because no diagnosis algorithm has yet beeen implemented to supply the feedback queries.

To perform the task of assigning initial heart actions, the labelling algorithm first accepts as input the whole wave group tokens from the reduction stage. These tokens represent the most unambiguous whole waves in the ECG trace and therefore, the labelling should be able to assign a specific heart action to each. However, careful consideration of the derivation of these tokens will show that each represents either a wave or a space between waves, as in figure 4-9.

One way to overcome this problem is to establish a reference of either a wave or a space, to which the other tokens can be compared. It happens that the easiest reference to establish is the positions of the QRS-complexes. Therefore, the labelling algorithm implemented in this work breaks down naturally into four steps:

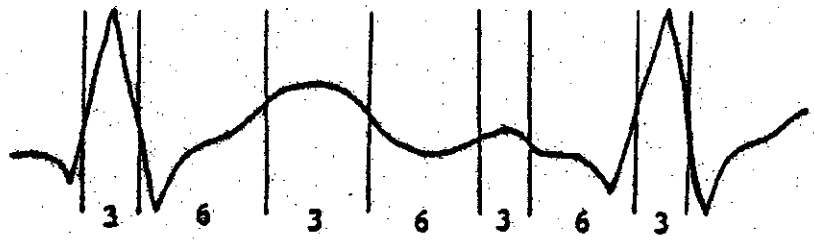


Figure 4-9: An example to show that some whole wave tokens represent waves while others represent the space between waves.

1. Find the approximate location of each QRS-complex.
2. Use a fixed set of rules to find the extent of each QRS.
3. Classify the QRS-complexes into groups according to shape.
4. Use the extended QRS-complexes as a reference and find whole P and T-waves between.

Each of these four steps is discussed in this section, along with preliminary test results. The details of the implemented algorithms can be found in Appendix C.

### 4.2.1 QRS Position

The first step in the labelling algorithm is to unambiguously label some of the whole wave group tokens for reference. A human being looking at an ECG trace is regularly drawn initially to the QRS-complexes, and uses these as a reference for further analysis; therefore, it would be most useful if the computer did the same. In fact, previous work on ECG diagnosis, as discussed in section 3.1, has shown that QRS-complexes can be easily and reliably located by examining the derivative of the ECG trace [21][38].

The algorithm used in this work follows much of the literature by using a dynamic threshold technique to examine the derivative. First, the entire trace is searched for the largest absolute value of the derivative amplitude and a threshold is set at 50% of that value. A QRS-complex is then suspected at every point in the trace where the absolute value of the derivative crosses the threshold.

This algorithm has many advantages, with the most important being its ease of implementation and its high reliability in most cases [67]. The steps involved in the process can also be thought of as a feedback query to the derivative level of reduction tokens, bypassing the whole wave group tokens.

Unfortunately, the algorithm relies on a threshold and therefore cannot be considered to be perfectly complete; some abnormal QRS-complexes may slip under the threshold. For the same reason, the algorithm is not completely unambiguous, as some large T-waves may be falsely detected. However, these problems occur only in unusual cases, and can be corrected by feedback queries from the diagnosis stage.

The details of the algorithm are given in Appendix C, and the results of applying the algorithm to actual traces are shown in figure 4-10.

To test the reliability, five 10-second long ECG traces were analysed and 100% of the QRS-complexes were correctly identified with no false indications recorded.

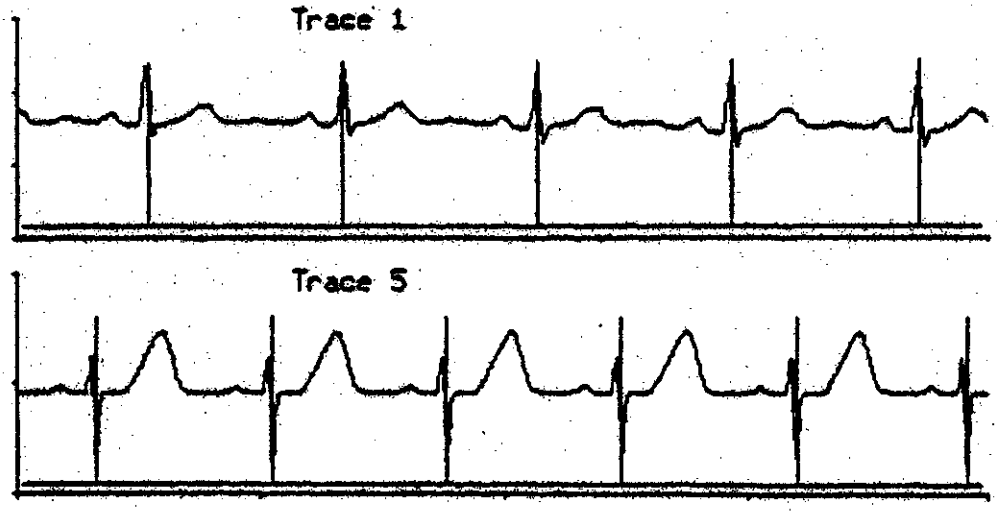


**Figure 4-10: Detecting QRS-complexes.**

### 4.2.2 QRS Extent

Once the positions of the QRS-complexes have been identified in the trace, it remains to discover precisely which whole wave group tokens are members of each QRS. This identification not only gives a reference for further processing, but also gives the contour of the QRS by defining the width and orientation of each wave within it.

The algorithm used to find the extent of the QRS-complexes within the whole wave group tokens is similar to the QRS "growing" technique discussed in section 3.1.2 [5]. However, this implementation can use a much smaller rule set as the whole wave group tokens hold more natural meaning. The essential features of the technique are:

- Find the first whole wave group token in the QRS, i.e., the one which contains the QRS detection point.

- Apply a set of rules to the tokens on either side of this initial QRS-wave to see whether they too should be part of the QRS.

- When the whole wave group tokens on both sides should no longer be added, the QRS is complete.

Only an outline of the implementation of this algorithm is given here; the details can be found in Appendix C.

After the first wave in the QRS-complex is identified by matching it to a QRS location, it is examined with a feedback query to determine whether it should be expanded. This query is called the group-property query, and it consists of looking in the trace for the following features within the wave (see figure 4-11):

1. The location of the peak of the wave.
2. Either a flat plateau of 25 ms duration or a corner where the slope changes by more than 30% within 25 ms to the left of the peak; called the left stop.
3. A similar plateau or corner on the right side of the peak; called the right stop.

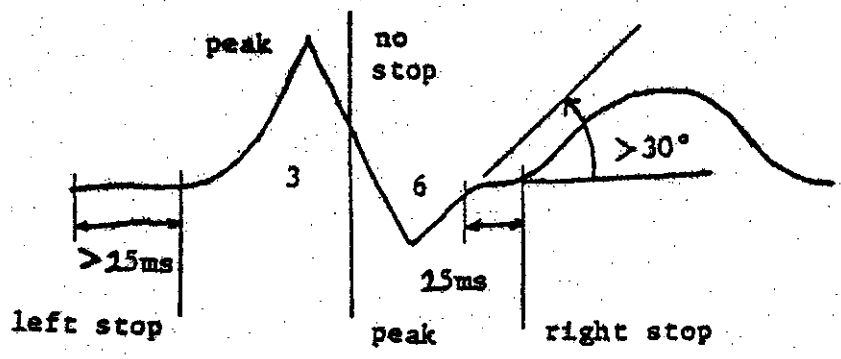


**Figure 4-11:** Features sought by the group-property feedback query.

If either a left or right stop is found within the initial QRS whole wave group token, the QRS is assumed to end at the

stop. If one side or the other is not stopped, the wave next to it is examined to see whether it should be added to the QRS. If both sides are free for addition, the smallest wave on the right or left is examined first. The decision of whether the wave should be added is based on a set of rules which differ slightly on each side. The rules for adding a wave to the left side are given below. Note that the sixth step uses another feedback query.

- Use the group-property feedback query on the wave being considered.
- Do not add it if it has a right hand stop. None of it belongs; the QRS ends at the stop.
- Flag it as the end if adding it will make the total QRS width greater than 220ms.
- Flag it as the end if it is wider than 115 ms.
- Always add it if it is narrower than 50 ms.
- If the average slope of the wave under consideration is less than 40% of the average for the steepest wave of the trace, flag it as the end. This wave is too shallow to be part of the QRS.
- If the wave is flagged as the end, keep it only to its peak. This token represents a space between waves.
- If the wave is to be added, add only to the left stop if there is one. Otherwise add the whole wave and consider adding the next one over.

When no more wave tokens can be added to either side, the extent of the QRS-complex has been determined.

This QRS growth algorithm has two important properties. First, only complete waves are considered for addition, so the QRS must end in reasonable places in the trace. Second, each of

the wave addition rules has a meaningful interpretation from an outside point of view. However, the algorithm also has the disadvantage that it is based on several fixed threshold values. These thresholds occasionally result in similar QRS-complexes having different extents, if the significant plateaux or corners are very near the borderline. Examples of this problem can be seen in figure 4-12.

The implementation of this algorithm has been tested with eight 10-second ECG traces and, while most QRS-complexes are extended correctly, as shown in figure 4-12, some are not. These results do not, however, mean that the labelling algorithm is unreliable, since the algorithm described in the next section corrects for most extension errors.

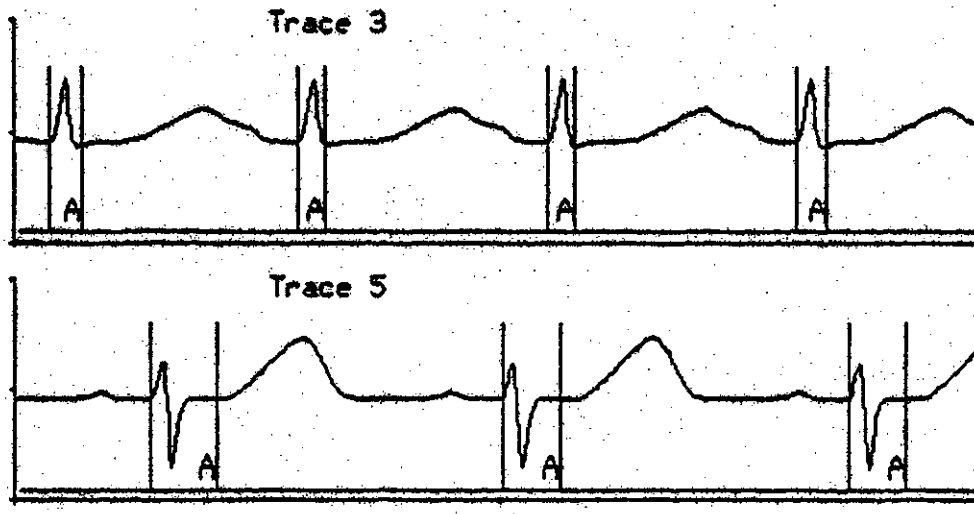


Figure 4-12: Results of the QRS-complex extension algorithm.

### 4.2.3 QRS Classification

The next step in the labelling algorithm, classification, is necessary for two reasons. First, it often occurs that in abnormal ECG traces, there are several differently shaped QRS-complexes; one representing normal ventricular depolarization, and the others representing abnormal depolarizations. This is an important diagnostic distinction, and since it is based primarily on shape, it can be determined without a diagnostic context. Second, due to the many thresholds involved in the previous extension algorithm, errors are often made in the endpoints of the QRS-complexes. These errors can also be detected at this stage by examining QRS shape.

The shape of a QRS-complex at this stage in the diagnosis system can be defined by the number and type (i.e., 3 or 5) of whole wave group tokens within it, the width of each token, and the approximate "peakedness" of each represented by average slope over the token. Therefore, the solutions to the problems of labelling and extension both start with grouping the QRS-complexes in terms of these variables. The group, or class, with the most members is then assumed to be of correct extent and to represent the dominant QRS type in the trace. Each other class of QRS is compared to the dominant class. If it is sufficiently similar, it is assumed to be the same class, and its extension is corrected accordingly. Any classes which are not similar are assumed to be abnormal beats, and each is labelled separately. The details of the algorithm outlined

below are given in Appendix C.

The first step in the algorithm is to group the different QRS-complexes into classes. This classification is accomplished with the following algorithm (see figure 4-13):

- Choose the next QRS-complex. Calculate the average slope of each wave in the QRS and designate those waves with an average greater than 40% of the maximum average as essential.
- Compare this QRS-complex to each class already defined using the following criteria:
    * If the number and type of waves do not match, it is in a different class.
    * If essential waves do not match, it is in a different class.
    * If, with this QRS added to the class, the variation of the average width of any wave in the class is greater than 25 ms, this QRS is in a different class.
- If this QRS does not match any previous class, it defines a new class.

Once the classification of the QRS-complexes is complete, it remains to look for "similar" classes and correct their extensions. The similarity of two classes is based on the width and shape of the essential waves within the classes, as specified below (see figure 4-14):

- Choose the next QRS class to examine.
- Compare this QRS class to each class with a greater number of members according to the following criteria:
    * If the type of essential waves does not match, they are not similar classes. They have

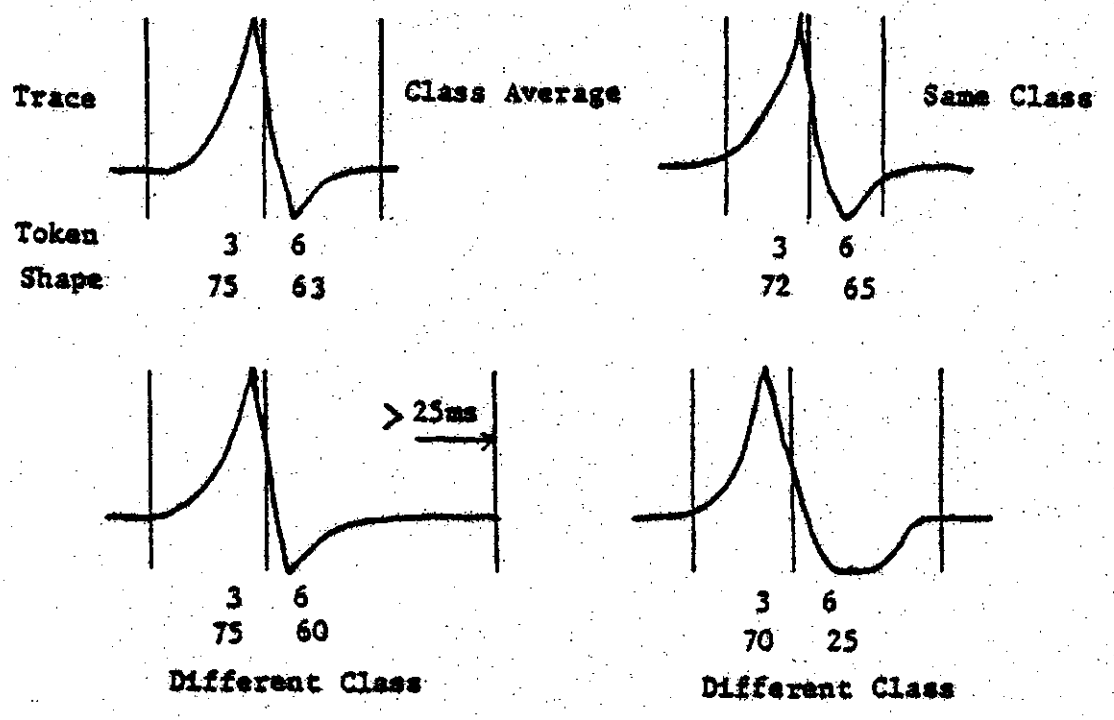


Figure 4-13: Classifying QRS-complexes.

basically different shapes.

* If the variations of the average width of any essential wave except the outside two do not overlap, they are not similar classes. The outside two waves are not included as they may have incorrect extensions.

* If the variation in average slope of the two outside waves does not match, they are not similar classes. Even with extension errors, the average slopes should be similar.

- If this QRS class is similar to the class being compared, correct the extension of each of its members to the average extension of the larger class and assign them to that class.

- If this QRS class is not similar to any larger class, it forms a class by itself.

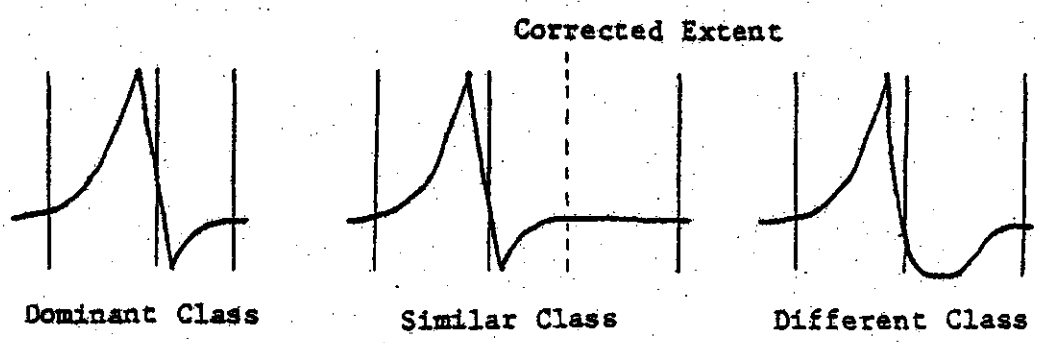


Figure 4-14: Comparing QRS classes.

With completion of this class comparison and correction, the algorithm is left with one or more classes of QRS-complexes, and each QRS should be corrected to its proper extent. These classes are numbered according to the number of QRS complexes in each. Hence, the dominant ventricular depolarization events will be labelled 1, the next most common 2, etc.

Besides labelling each QRS-complex with a class number related to the frequency of its contour in the ECG trace, this QRS classification algorithm has two other advantages. First, it corrects extension errors made in the previous labelling step for similar QRS wave forms, but leaves very different QRS types alone. Second, the algorithm uses only whole wave group tokens, so it is tolerant to minor variations in constituent wave shapes.

There are still a few disadvantages with the process, however. This algorithm, like the QRS extent, relies on some thresholds; hence, it may still have problems in some unusual cases and cannot be considered complete. In addition, by assuming the most common class is correct, a common extension error may be forced onto all QRS types in the trace. An example of this is the retrograde P-wave shown in figure 4-15. This problem be corrected with feedback queries from the diagnosis stage.

In the small sample of five 10-second normal ECG traces available for testing, this algorithm properly classified and corrected 100% of the QRS-complexes. Examples are shown in

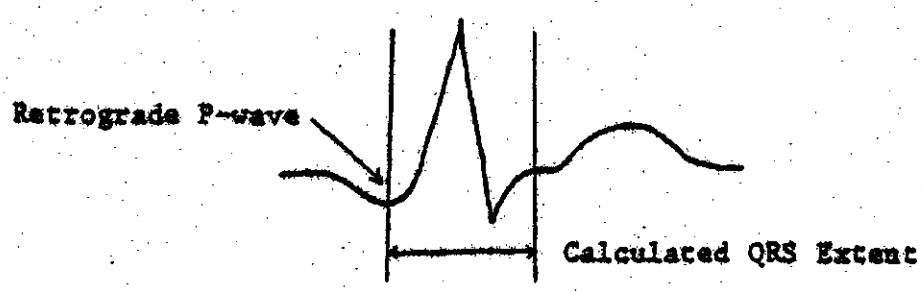


Figure 4-15: A possible QRS extension error.

figure 4-16.

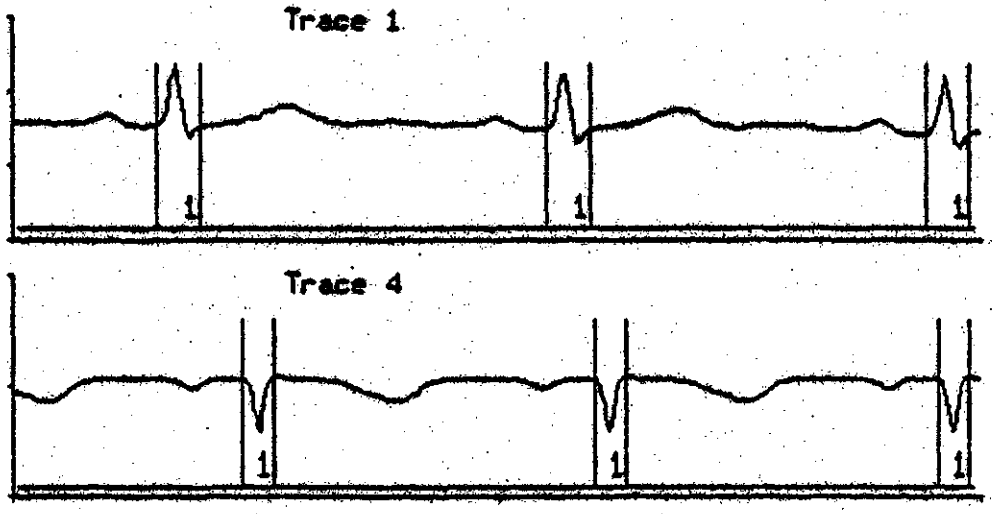


Figure 4-16: Examples of actual QRS-complex identification.

### 4.2.4 R-T Wave Detection

With the QRS-complexes to use as a reference, the final step in the initial labelling of the ECG trace is to identify any P or T-waves. The basis of the solution to this problem is very simple, and is used repeatedly in the literature discussed in section 3.1: find each significant wave between the QRS-complexes, then label the first found T and the rest B.

Implementation of this step would be straightforward if each whole wave group token between the QRS-complexes actually represented a single wave, but, as previously stated, some of the tokens represent spaces. To overcome this problem in most cases, the following algorithm has been developed (for details see Appendix C):

- Between each two QRS-complexes, examine each complete whole wave group token.
- If the absolute value of the peak amplitude of the wave is larger than that of the wave on each side of it, it is significant. Other waves represent spaces. (see figure 4-17)
- The first significant wave to the right of a QRS-complex is a T-wave, and all other significant waves are P-waves.

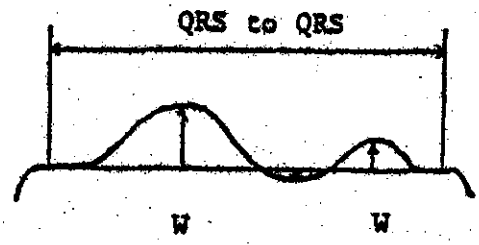


Figure 4-17: Significant whole wave group tokens.

Although this algorithm is very simple, it has one important feature. It passes on only large complete waves for initial diagnosis, so that stage can be confident that each wave received actually represents a heart action. This same feature is also a disadvantage, however, because all partial waves and some low amplitude whole waves are not passed on (see figure 4-18). This incompleteness cannot be unambiguously overcome however, so the diagnosis stage must correct for it by using feedback queries. Another disadvantage is that, in some

abnormalities, a P-wave may be found between a QRS and the following T-wave. Again, this problem cannot be corrected without an idea of what may be wrong with the heart, and must therefore be compensated for in the diagnosis stage.

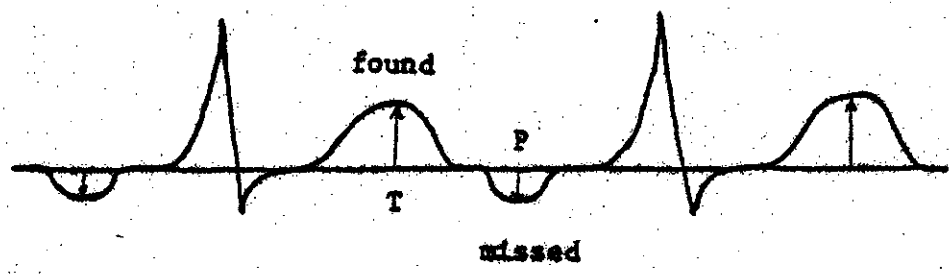


Figure 4-18: A missed P-wave.

Testing this algorithm on actual ECG traces reveals the expected; only well-separated complete waves were detected. Examples are shown in figure 4-19 below.

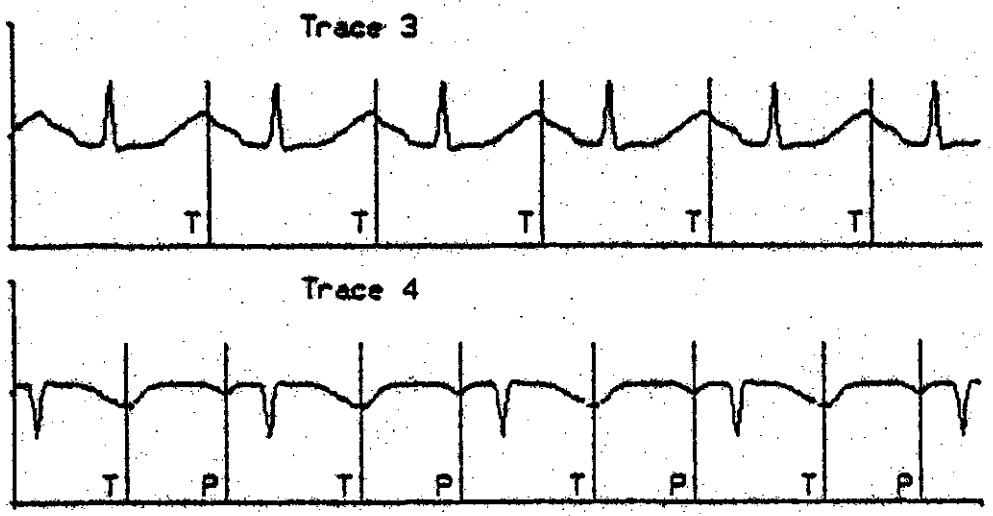


Figure 4-19: Examples of actual initial P and T-wave detection.

### 4.2.5 Summary

To summarize, the labelling algorithm used in this

automatic ECG interpretation system consists of four steps: finding QRS-complexes, discovering the extent of each QRS, classifying the QRS types according to contour, and finally locating any significant T and P-waves which fall between the QRS's. Therefore, the information passed on to the next diagnosis stage consists of a list of all QRS-complexes in the ECG trace, each having a class number to identify it as a dominant or abnormal beat, and a list of all easily identified P and T-wave candidates.

A major property of these output lists is that they are incomplete; that is, they do not necessarily contain every QRS, P, and T-wave in the trace, and those that are included may be mislabelled. However, the rules of each step are designed so that the majority of the information passes on to the diagnosis stage is reliable; therefore, the diagnosis algorithm should be able to formulate reasonable possible diagnoses, even with a few errors. Once these possible diagnoses are generated, the incompleteness of the labelling algorithm can be corrected with feedback queries.

To illustrate the way this feedback correction can work, a single diagnosis query is included in the system implemented in this work. This feedback query is:

- If only one significant wave is found ahead of a dominant QRS-complex, then assume it is a T-wave and look for a partial wave shape token between its peak and the QRS. This, if found, should be a P-wave. (see Appendix C for details)

By employing this simple diagnosis rule, the five

10-second ECG test traces can all be correctly labelled, including the traces with P-waves which overlap T-waves and do not appear as whole wave group tokens. For an example, see figure 4-20.

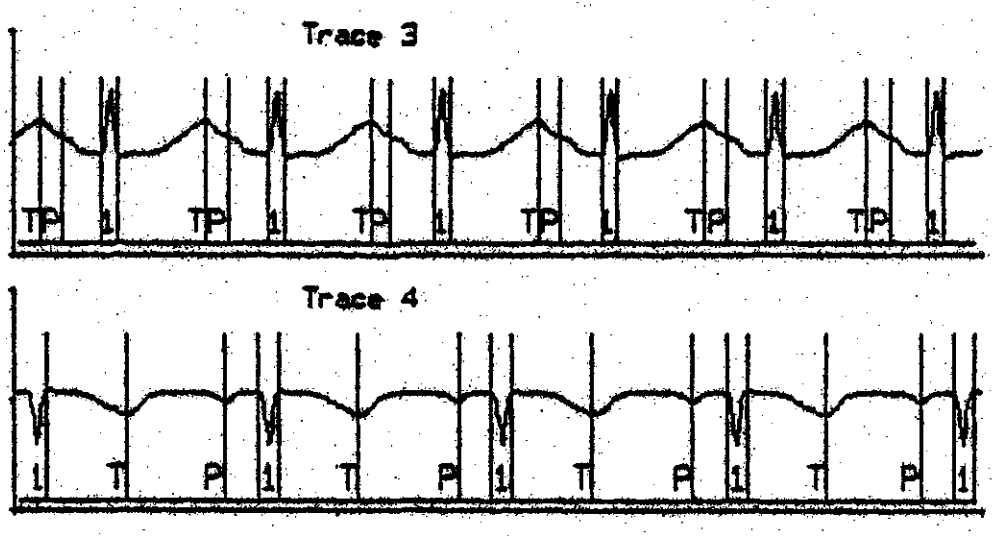


Figure 4-20: Actual ECG traces labelled after a single diagnosis feedback query.

## 4.3 Summary of Reduction and Labelling Algorithms

The reduction and labelling algorithms implemented in the automatic ECG analysis system described in this thesis are modelled after the human analysis model put forth in chapter 2. Therefore, both algorithms consist of rule sets that extract the significant information from the input ECG trace.

The reduction algorithm has three steps, each of which generates a more abstract layer of output tokens. The first layer consists of the digitized ECG trace and its derivative, while the second layer is made up of partial wave shape tokens,

each representing an extended segment of the trace. Both of these token layers are expressive and complete, retaining all of the information originally in the ECG trace. The final set of whole wave group tokens representing only the most significant partial wave shape tokens, absorbing ambiguous partial waves. This causes a loss of completeness, but diminishes the amount of extraneous detail while retaining the most important waves for later analysis.

The labelling algorithm works to assign a specific heart action to each of the whole wave group tokens. This labelling is done in four steps: finding each QRS-complex in the trace, discovering the extent of each QRS, classifying the QRS types according to contour, and then locating potential P and T-wave candidates between QRS's. Each of these steps is initially incomplete, since they are based primarily on the incomplete list of whole wave group tokens. This incompleteness is the result of a conscious design decision to follow the human ECG analysis model of information flow. In order to increase the ease and reliability of the initial formulation of possible diagnoses, the diagnosis algorithm should begin with only those ECG waves which can unambiguously be assigned a heart action. Once these possible diagnoses have been worked out, the incompleteness of the input label set is eliminated through feedback queries to the lower token levels. Therefore, the final output of the labelling algorithm consists of a list of QRS-complexes grouped according to contour and a list of the unambiguous T and P-waves between them.

The final important area of discussion on the implementation of the reductions and labelling algorithms is of the actual testing of their performance and reliability. Reliability testing of systems such as the one implemented in this work comes in two phases. First, the actual logic of each algorithm must be tested to see whether it performs as planned. The examples of program output included in this chapter are from this type of testing and, in the limited five trace sample, the programs perform perfectly. Further complete examples of the analysis of these five traces are given in Appendix B.

The second type of reliability testing required is one of determining the ultimate usefulness of the tokens and initial labelling algorithm in formulating first possible and then final ECG diagnoses. This testing requires both a large sample of abnormal data and a diagnosis algorithm to supply feedback queries. As the implementation of the system has not progressed to that extent, such testing has not, as yet, been possible.

One final measure of algorithm performance which can be measured, however, is execution time, or the time to produce the final output from the labelling algorithm. Unfortunately, while it can be measured for each trace, the results are only a guideline, because the execution time depends on machine related factors such as internal speed, real memory available, etc, and also on input trace factors such as data sampling rate, heart rate, length of trace, etc. However, as a

reference, the average execution times on the VAX 11-780 for the five 10-second test ECG traces are given in table 4-1.

| Program Execution Times | | | | |
|---|---|---|---|---|
| Sample ECG Trace | Sampling Rate (Hz) | Length (sec) | Heart Rate (bpm) | Analysis Time (CPU sec) |
| 1 | 100 | 10 | 72 | 11.5 |
| 2 | 100 | 10 | 114 | 8.2 |
| 3 | 250 | 10 | 120 | 26.9 |
| 4 | 250 | 10 | 84 | 42.5 |
| 5 | 250 | 10 | 84 | 44.0 |

Table 4-1: Sample algorithm execution times.

With this description of the reduction and labelling algorithms, the work behind this thesis is complete. The next chapter contains a discussion of the work thus far, and includes recommendations for the directions future work on this automatic ECG interpretation system might take.

# Chapter 5

# DISCUSSION AND RECOMMENDATIONS FOR FUTURE WORK

*What's past is prologue.*
[Shakespeare, The Tempest]

With the work behind this thesis explained in detail in the previous chapter, it is traditional at this point to summarize its key aspects. Perhaps the best way to summarize in this case is to outline the experience which led to the algorithms and implementation described.

When initially given a problem such as producing an automatic ECG analysis system, the first step toward a solution is to read the relevant literature. Unfortunately, the literature on ECG analysis is filled with dozens of systems, each slightly different, with very little consensus as to which systems are better and why. To make sense of this multiplicity of views, a standard of comparison is required and, for this work, the human ECG analysis procedure was chosen to be that standard. This choice was based on the grounds that human doctors indisputably have the best rate of success in interpreting diagnostic ECG's.

In chapter 2, a novel model of the human ECG analysis system is suggested, with arguments supporting its plausibility coming from such seemingly diverse areas as experimental psychology, computer science, education, medicine, and personal experience. Several features of this human model play central roles in the work described in chapters 3 and 4. The most striking aspect of the model is that it is comprised of three logically separate stages: reduction, labelling, and diagnosis. In the reduction stage, the ECG trace is broken down into pieces, or tokens, each representing a meaningful wave in the trace. The labelling stage matches each of these waves with a specific heart action, and finally from this string of heart actions, the diagnosis stage formulates a diagnosis of the underlying heart problem.

The second major feature of the human ECG analysis model is the way information is passed between stages. The simple division of the process into three steps would suggest that the ECG trace first be accepted into the reduction algorithm and all waves in it identified. Then, every wave would be assigned a heart action and finally, the diagnosis stage would need only those actions to produce a diagnosis. In the case of actual ECG traces, however, this simple bottom-up data flow is not possible because, until put into context by a reasonable diagnosis, the labelling of many waves, and even the existence of some, is ambiguous. Therefore, in the human model, most of the ambiguous information is held back at the reduction and labelling stages to permit possible initial diagnoses to be

worked out using only the most reliable data. These potential diagnoses can then be confirmed by extracting and placing the ambiguous information into context by means of feedback queries.

The third, and most important feature of the human analysis model is that it suggests the concepts of expressiveness and completeness. Each stage of the model must be expressive in the sense that it extract all significant information from its input; it must also be complete in passing that information on to the next stage. A lack of completeness can be overcome by using feedback queries to retrieve more information and, in fact the information flow model above requires that the reduction and labelling algorithms be initially incomplete. However, a lack of expressiveness at any stage implies a basic inability to perceive important pieces of information, and will unavoidably lead to incorrect diagnoses in some cases.

With the adoption of this model as a reference, reading the literature on automatic ECG analysis discussed in chapter 3 produces several important observations. First, while no real effort seems to have been made to attempt an understanding of the human analysis process, virtually all programs developed include reduction-labelling and diagnosis sections. Second, the trend toward improving systems has been merely to add more rules for labelling and diagnosis, hence increasing their completeness, but leaving the basic reduction stage unchanged from program to program. As can be seen in section 3.1, this

reduction algorithm is not sufficiently expressive to permit identification of all P-waves in all traces, and therefore, some incorrect diagnoses must be expected. While the improved completeness of the later stages does allow more correct diagnoses, this lack of expressiveness creates the impression in the literature of a fixed level of accuracy which cannot be improved, implying that certain P-waves are simply impossible to find [12][38]. The human ECG analysis model suggests that this barrier to P-wave detection is artificial, and can be removed with a more expressive reduction algorithm.

Hence a new automatic ECG interpretation system was designed based heavily on the human ECG analysis model, with particular emphasis placed on the reduction stage. As described in chapter 4, the reduction algorithm is designed to produce as output three layers of tokens. The first layer consists of the digitized ECG trace and its derivative, thus ultimately retaining expressiveness by keeping all the data available. The next layer of partial wave shape tokens is carefully chosen to represent intuitively meaningful pieces of ECG trace. Special care was taken to ensure that the rules which form the tokens find every wave in the trace, so that the output set remains both expressive and complete. The third and final layer of whole wave group tokens represents unambiguous waves only, and thus passes on only reliable waves for initial labelling. The lack of completeness in this layer is offset by the ability to make feedback queries to the more detailed partial wave shape tokens below.

The initial labelling algorithm designed for this system has four steps: find the location of each QRS-complex in the trace, discover the extent of each QRS, classify the QRS types according to contour, and identify definite T and P-wave candidates between each pair of QRS's. With these steps operating primarily on the whole wave group tokens of the reduction algorithm, the resulting output from the labelling stage is a reliable list of well-defined waves for initial diagnosis, as required by the human ECG analysis model. At the time of writing, no diagnosis algorithm has been implemented.

The reduction and labelling algorithms have been tested on a VAX 11-780 using a limited sample of ECG data, as explained in chapter 4. These test results show that the algorithms perform as expected. This success of a more expressive algorithm indicates that the design goal of an improved automatic ECG interpretation system is nearing achievement, although this cannot be confirmed until a diagnosis algorithm is implemented to make full use of the power of the added expression. The rest of this chapter discusses the future work needed to accomplish this task.

## 5.1 The Diagnosis Algorithm

Clearly, the most important area of future work is the implementation of a diagnosis algorithm. The actual form of the algorithm applied may vary, but several guidelines should be followed in order to match it with the existing reduction and labelling stages. The first and most important guideline

is that the algorithm should be patterned after the human ECG analysis model. It must therefore have two steps: first, the formation of several possible diagnoses using the incomplete but unambiguous output from the initial labelling algorithm, and second the confirmation or rejection of each possible diagnosis through the use of feedback queries to the two lower layers of reduction stage tokens.

One way to implement this type of process is the decision tree, as discussed in section 3.2.1. The possible diagnoses would be achieved by traversing the tree as far as possible on the initial data, and then the final path would be decided using feedback queries at each further branch. Although this method would be the simplest to implement at first, and may be worthwhile in a preliminary investigation, it would have all the problems outlined in section 3.2.1, and would be very difficult to update when incorporating new diagnoses.

A better choice for the diagnosis algorithm is the more intuitive approach of formulating one set of rules to make the initial diagnoses, and a further set of rules to confirm or reject them. In fact, it would be best if some of the rules actually constructed a heart reference model for simulating the final diagnosis and then compared its output to the input trace through feedback queries, as suggested in section 2.2.4. It would be advantageous for the rules to be explicitly coded as separate rules rather than linked together in the form of a tree. A great deal of work is being done on this type of algorithm in the computer science field of expert medical

systems [30][35][59][60][65]. This readily available experience, the intuitive nature of rules, and the relative ease of adding new diagnosis rules makes this type of algorithm the most attractive.

## 5.2 Improvements to the Reduction and Labelling Algorithms

Another avenue of future research is the improvement of the reduction and labelling algorithms outlined in chapter 4. While they perform adequately in the present implementation, there are several minor adjustments which could make them more generally applicable.

The first possible improvement is the inclusion of a preprocessor to remove from the trace certain extraneous data, such as sudden shifts of baseline, bursts of high amplitude noise and pacemaker spikes. In the diagnostic environment, the first two problems are not severe, but the occurrence of pacemakers is fairly common. Pacemakers, in an ECG trace, usually produce very narrow high amplitude spikes which have very steep slopes. The resulting spikes in the derivative will confuse the QRS detection algorithm used in the labelling stage. The solution is the addition of a preprocessor which contains an algorithm to filter out the pacemaker spikes while retaining their positions [56].

Another modification which might be made is in the area of noise measurement and elimination. As outlined in Appendix A, the ultimate expressiveness of the partial wave shape token

algorithm is dependent on an accurate measurement of the amplitude of the noise in the ECG trace. The algorithm for noise measurement used in this implementation is described in detail in Appendix C, and is adequate for diagnostic traces of relatively low noise. In harsher environments, such as a CCU, however, a more general noise measurement procedure will be required.

Finally, a third area of improvement is increasing the efficiency of the algorithm used to find the local maxima and minima in the ECG trace and its derivative. On the VAX 11-780, processor speed is no barrier, so this algorithm was written to be easily understood rather than maximally fast. If a smaller, slower computer is used, a major decrease in processing time might be effected by finding a more efficient algorithm for this operation.

## 5.3 System Testing

A final suggestion for future work on this, or on any ECG interpretation system, is the building or acquisition of a database of sample ECG traces for program testing and comparison at the University of Saskatchewan. Such a database would ideally consist of several hundred normal and abnormal ECG traces, each trace having previously been diagnosed by a physician. While this type and amount of data is not immediately necessary to test the algorithms in this thesis, it is crucial for the testing of a diagnosis algorithm.

## 5.4 Conclusion

The ability of a machine to interpret automatically electrocardiograms in a reliable human-like fashion will open up new avenues of heart research and free heart specialists to conduct that research. This is already evident in the area of routine computer comparison of new ECG's with old ones from the same patient [5][42][51]. It is hoped that the insights into the ECG diagnosis process reported in this thesis will help advance automatic interpretation a step closer to that goal.

**References**

[1] Abenstein J.P. and Tomkins W.J.
A New Data-Reduction Algorithm for Real-Time ECG Analysis.
IEEE Transactions on Biomedical Engineering BME-29:43-48, 1982.

[2] Alexander D.C. and Wortzmann D.
Computer Diagnosis of Electrocardiograms. I. Equipment.
Computers and Biomedical Research 1:348-365, 1968.

[3] Bailey J.J., Horton M., and Itscoitz S.B.
A Method for Evaluating Computer Programs for Electrocardiographic Interpretation III. Reproducibility Testing and the Sources of Program Errors.
Circulation 50:88-93, 1974.

[4] Bethesda Conference Report.
Optimal Electrocardiography.
The American Journal of Cardiology 41:111-191, 1978.

[5] Bonner R.E. and Schwetman H.D.
Computer Diagnosis of Electrocardiograms. II. A Computer Program for EKG Measurements.
Computers and Biomedical Research 1:366-386, 1968.

[6] Bonner R.E. and Schwetman H.D.
Computer Diagnosis of Electrocardiograms. III. A Computer Program for Arrhythmia Diagnosis.
Computers and Biomedical Research 1:387-407, 1968.

[7] Bonner R.E., Crevasse L., Ferrer M.I., and Greenfield J.C.
A New Computer program for Analysis of Scalar Electrocardiograms.
Computers and Biomedical Research 5:629-653, 1972.

[8] Bransford J.D.
Human Cognition: Learning, Understanding, and Remembering.
Wadsworth Publishing Company, 1979, pages 128-153.

[9] Burden R.L., Faires J.D., and Reynolds A.C.
Numerical Analysis.
Prindle, Weber and Schmidt, 1978.

[10] Caceres C.A., Steinberg C.A., Abraham S., Carbery W.J., McBride J.M., Tolles W.E., and Rikli A.E.
Computer Extraction of Electrocardiographic Parameters.
Circulation 25:356-362, 1962.

[11] Caceres C.A.
The Case Against Electrocardiographic Automation.
Computer 6(7):15-21, 1973.

[12] Caceres C.A.
Limitations of the Computer in Electrocardiographic Interpretation.
The American Journal of Cardiology 38:362-376, 1976.

[13] Chung E.K.
Electrocardiography: Practical Application with Vectorial Principles, 2nd edition.
Harper and Row Publishers Inc., 1980.

[14] Clancey W.J. and Letsinger R.
NEOMYCIN: Reconfiguring a Rule-Based Expert System for Application to Teaching.
Proceedings of 7th IJCAI II:829, 1981.

[15] Cox J.R., Fozzard H.A., Nolle F.M. and Oliver G.C.
Transformations Useful in Electrocardiography.
In Computers in Biomedical Research, Vol. III, chapter 7. Academic Press, 1965.

[16] Cox J.R., Nolle F.M., Fozzard H.A., and Oliver G.C.
AZTEC, A Preprocessing Program for Real-Time ECG Rhythm Analysis.
IEEE Transactions on Biomedical Engineering BME-15:128-129, 1968.

[17] Dale P.S.
Language Development: Structure and Function, 2nd edition.
Holt, Rinehart and Winston, 1976.

[18] Dubin D.
Rapid Interpretation of EKGs, 3rd edition.
COVER Publishing Co., 1980.

[19] Duda R.O. and Hart P.E.
Pattern Classification and Scene Analysis.
John Wiley and Sons, Inc., 1973.

[20] Eyzaguirre C. and Fidore S.J.
Physiology of the Nervous System, 2nd edition.
Year Book Medical Publishers Inc., 1977.

[21] Geddes J.S. and Warner H.R.
A PVC Detection Program.
Computers and Biomedical Research 4:493-508, 1971.

[22] Gustafson D.E., Willsky A.S., Wang J., Lancaster M.C., and Triebwasser J.H.
A Statistical Approach to Rhythm Diagnosis of Cardiograms.
Proceedings of the IEEE 65:802-804, 1977.

[23] Gustafson D.E., Wang J.Y., and Willsky A.S.
Cardiac Rhythm Interpretation Using Statistical P and R Wave Analysis.
In Frontiers of Engineering in Health Care. IEEE Engineering in Medicine and Biology Society Conference, 1981.

[24] Guyton A.C.
Textbook of Medical Physiology, 6th edition.
W.B.Saunders Co., 1981, pages 150-203.

[25] Haber R.N. and Hershenson M.
The Psychology of Visual Perception.
Holt, Rinehart and Winston Inc., 1973, pages 52-59.

[26] Havens W.S.
A Procedural Model of Recognition for Machine Perception.
PhD thesis, Department of Computer Science U.B.C., 1978.

[27] Helppi R.P., Unite V., and Wolf H.K.
Suggested Minimal Performance Requirements and Methods of Performance Evaluation for Computer ECG Analysis Programs.
Canadian Medical Association Journal 108:1251-1259, 1973.

[28] Hengeveld S.J. and van Bemmel J.H.
Computer Detection of P-Waves.
Computers and Biomedical Research 9:125-132, 1976.

[29] Holsinger W.B., Kempner K.M., and Miller M.H.
A QRS Preprocessor Based on Digital Differentiation.
IEEE Transactions on Biomedical Engineering BME-18:212-217, 1971.

[30] Ho-Tai S.A.
A Prototype Knowledge-Based System for Computer-Assisted Medical Diagnosis.
PhD thesis, Computer Science Research Group, U. of T., 1981.

[31] Hu K., Francis D.B., Gau G.T., and Smith R.E.
Development and Performance of Mayo-IBM Electrocardiographic Computer Analysis Programs (V70).
Mayo Clinic Proceedings 48:260-268, 1973.

[32] Hubel D.H. and Wiesel T.N.
Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex.
Journal of Physiology 160:106-154, 1962.

[33] Ishuzuka M., Fu K.S. and Yao J.T.P.
Inexact Inference for Rule-Based Damage Assessment of Existing Structures.
Proceeding of 7th IJCAI II, 1981.

[34] Jenkins J.M.
Computerized Electrocardiography.
CRC Critical Reviews in Bioengineering 6:307-350, 1981.

[35] Kastner J.K. and Weiss S.M.
A Precedence Scheme for Selection and Explanation of Therapies.
Proceedings of the 7th IJCAI II:908, 1981.

[36] Klingeman J. and Pipberger H.V.
Computer Classification of Electrocardiograms.
Computers and Biomedical Research 1:1-17, 1967.

[37] LeBlanc A.R. and Roberge B.A.
Present State of Arrhythmia Analysis by Computer.
Canadian Medical Association Journal 108:1239-1251, 1973.

[38] LeBlanc A.R., Roberge F.A. and Nadeau R.A.
Evaluation of a New ECG Measurement Program for the Detection of Rhythm Disturbances.
Medical and Biological Engineering 13:370-381, 1975.

[39] Levine H.D.
Clinical Interpretation of Electrocardiogram by Means of Electronic Computers.
American Heart Journal 69:147-149, 1965.

[40] Lopez J.F., Covvey D., Gregor R.D., LeBlanc A.R., Nadeau R.A. and Waxman M.
Criteria for the Classification and Diagnosis of Cardiac Arrhythmias.
Report by the Arrhythmia Group of the Commitee on Computer Assisted ECG Interpretation, Canadian Heart Association, 1977.

[41] Macfarlane P.W.
ECG Waveform Identification by Digital Computer.
Cardiovascular Research 5:141-146, 1971.

[42] Macfarlane B.W., Cawood H.T., and Lawrie T.D.V.
A Basis for Computer Interpretation of Serial Electrocardiograms.
Computers and Biomedical Research 8:189-200, 1975.

[43] Miyahara H., Whipple G.H., Teager H.M., Webb T.W., Theophilis C.A. and Dohi Y.
Cardiac Arrhythmia Diagnosis by Digital Computer.
Computers and Biomedical Research 1:277-300, 1968.

[44] Murray T.M., Jones W.T., and Sanders C.
A Real-Time Microprocessor-Based ECG Contour Analysis System Utilizing Decision Tables.
IEEE Transactions on Biomedical Engineering BME-27:358-363, 1980.

[45] Newell A. and Simon H.A.
Programs as Theories of Higher Mental Processes.
In Computers in Biomedical Research, Vol. II, chapter 6. Academic Press, 1965.

[46] Norman D.A.
Memory and Attention: An Introduction to Human Information Processing, 2nd edition.
John Wiley and Sons, Inc., 1976, pages 38-60.

[47] Pipberger H.V., Stallman F.W., and Berson A.S.
Automatic Analysis of the R-QRS-T Complex of the Electrocardiogram by Digital Computer.
Annals of Internal Medicine 57:776-787, 1962.

[48] Pipberger H.V.
Computer Analysis of the Electrocardiogram.
In Computers in Biomedical Research, Vol. I, chapter 16. Academic Press, 1965.

[49] Pipberger H.V., McCaughan D., Littmann D., Pipberger H.A., Cornfield J., Dunn R.A., Batchlor C.D., and Berson A.S.
Clinical Application of a Second Generation Electrocardiographic Computer Program.
The American Journal of Cardiology 35:597-608, 1975.

[50] Pryor T.A., Russel R., Budkin A., and Price W.G.
Electrocardiographic Interpretation by Computer.
Computers and Biomedical Research 2:537-548, 1969.

[51] Pryor T.A., Lindsay A.E., and England R.W.
Computer Analysis of Serial Electrocardiograms.
Computers and Biomedical Research 5:709-714, 1972.

[52] Restak R.M.
The Brain, The Last Frontier.
Warner Books Inc, New York, 1979, pages 94-95.

[53] Rey W., Laird J.D., and Hugenholtz P.G.
R-Wave Detection by Digital Computer.
Computers and Biomedical Research 4:509-522, 1971.

[54] Robertson P.
Non-Temporal Prediction - A Distributed System for Concept Aquisition.
Proceedings CSCSI / SCEIO Conference :144-149, 1982.

[55] Romhilt D.W., Bloomfield S.S., Chou T., and Fowler N.O.
Unreliability of Conventional Electrocardiographic Care Units.
The American Journal of Cardiology 31:457-461, 1973.

[56] Sajet M., Decambre Y., Wurzburger J., and Demeester M.
A New Rhythm Classification Program Applied to Routine Electro- and Vector- Cardiograms.
Advances in Cardiology 16:256-259, 1976.

[57] Segall M.H., Campbell D.T. and Herskovits M.J.
Cultural Differences in the Perception of Geometric Illusions.
Science 139:767-771, 1963.

[58] Sherwood J.D.
Suggested Standards for Instrumentation in Computer Processing of Electrocardiograms.
The Canadian Medical Association Journal 108:1259-1261, 1973.

[59] Shibhara T., Mylopolous J, Tsotsos J.K. and Covvey H.D.
CAA: A Knowledge Based System with Causal Knowledge to Diagnose Rhythm Disorders of the Heart.
Proceedings CSCSI / SCEIO Conference :71-78, 1982.

[60] Shortliffe B.H.
Computer-Based Medical Consultaion: MYCIN.
American Elsvier, New York, 1976.

[61] Sornmo L., Borjesson P.O., Nygards M., and Pahlm O.
A Method of Evaluation of QRS Shape Features Using a Mathematical Model for the ECG.
IEEE Transactions on Biomedical Engineering BME-28:713-717, 1981.

[62] Stallman F.W. and Pipberger H.V.
Automatic Recognition of Electrocardiographic Waves by Digital Computer.
Circulation Research 9:1138-1143, 1961.

[63] Staples I.F., Gustafson J.E., Balm G.J., and Tate W.A.
Computer Interpretation of Electrocardiograms.
American Heart Journal 72:351-358, 1966.

[64] Steinberg C.A., Abraham S., and Caceres C.A.
Pattern Recognition in the Clinical Electrocardiogram.
IEEE Transactions on Biomedical Engineering BME-9:22-30, 1962.

[65] Tsuji S. and Nakano H.
Knowledge-Based Identification of Artery Branches in Cine-Cardigrams.
Proceedings of the 7th IJCAI II:710, 1981.

[66] Upuda J.K. and Murthy I.S.N.
Syntactic Approach to ECG Rhythm Analysis.
IEEE Transactions on Biomedical Engineering BME-27:370-375, 1980.

[67] Vincent G.M., Webb G.N., Ross R.S., and Guier W.H.
QRS-wave Detector Evaluation.
American Heart Journal 83:475-480, 1972.

[68] Wartak J.
A Practical Approach to Automated Diagnosis.
IEEE Transactions on Biomedical Engineering BME-17:37-43, 1970.

[69] Wartak J., Milliken J.A., and Karchmar J.
Computer Program for Diagnostic Evaluation of Electrocardiograms.
Computers and Biomedical Research 4:225-238, 1971.

[70] Watanabe Y.
Automated Diagnosis of Arrhythmia by Small Scale Digital Computer.
Japanese Heart Journal 11:223-242, 1970.

[71] Willems J.L. and Pipberger H.V.
Arrhythmia Detection by Digital Computer.
Computers and Biomedical Research 5:262-278, 1972.

[72] Willems J.L.
A Plea for Common Standards in Computer Aided ECG Analysis.
Computers and Biomedical Research 13:120-131, 1980.

[73] Woolfolk A.E. and Nicolich L.M.
Educational Psychology for Teachers.
Prentice-Hall, 1980, pages 45-50.

## Appendix A. SIGNAL PROCESSING ASPECTS OF THE SYSTEM

In any system which deals with signals generated by real physical sources, there are two low level engineering aspects which must be considered. The first aspect is the calculation of the theoretical effect the system will have on an ideal noise free signal. This theory defines the ultimate limit of system performance. Second, the effect noise in the input signal has on the performance of the system must also be calculated. This calculation shows how well the system will deal with the real world.

This appendix discusses the signal processing aspects of the automatic interpretation system described in chapter 4 under the following headings:

- Initial sampling of the ECG trace.
- Calculating the derivative of the trace.
- Measuring the noise in the trace to aid identification of the significant extrema in the trace and derivative.
- Smoothing the trace to eliminate some noise.

### A.1 Sampling the ECG Trace

The theoretical limits of the ECG interpretation system are very much dependent on the resolution of the digitizing method used to sample the ECG trace for input to the computer. For example, several of the algorithms in the labelling stage search for features in the ECG trace with a width of 25 msec. For these searches to find meaningful results, the initial

trace should be sampled at a rate high enough to generate several data points within the 25 msec time window. The number of points actually necessary depends on the specific degree of accuracy needed in the measurement. The sample ECG traces in this thesis could be properly processed with a sampling rate as low as 100 Hz, or 2.5 samples in 25 msec, but a standard suggested in Sherwood [58] claims a sampling rate of 500 Hz is necessary to diagnose some abnormalities properly. This standard also suggests the amplitude resolution needed by the digitizer as well as discussing the types of analog filters necessary to properly eliminate electrical noise above the Nyquist frequency. Future ECG work should follow this standard.

## A.2 Calculating the Derivative

As discussed at the beginning of this appendix, the derivative calculation used in this system must both properly take the derivative and perform well in the presence of noise. If the derivative works well for all signals, the problem of noise is unimportant; the noise will simply be differentiated as well.

The formula used to take the derivative is given in chapter 4 as:

$$f'_n = \frac{f_{n-2} - 8f_{n-1} + 8f_{n+1} - f_{n+2}}{12\Delta t}$$

- f is the input trace
- f′ is the derivative

Considering this as a digital filter with input V and output W, the amplitude transfer characteristic can be calculated for the general sinusoid:

$$V_n = Ae^{i\omega nT}$$

- T is the sampling interval
- $\omega$ is the sinusoid frequency
- A is the amplitude

Since no frequencies will occur in the input with $\omega$ greater than 1/2T (the Nyquist frequency), the quantity $\omega T$ will always be less than one, and the calculated transfer characteristic can be written approximately as:

$$\frac{W_n}{V_n} = i\omega\left(1 - \frac{\omega^4 T^4}{30}\right)$$

The ideal derivative of V would be a transfer characteristic of just $i\omega$, but over the frequency range 0→1/2T the equation given differs from $i\omega$ by less than 0.2%. Therefore, this formula is quite good and need not be improved upon unless measurements need be made in the diagnosis stage with accuracy better than 0.2%.

## A.3 Measuring the ECG Trace Noise

Noise in the ECG trace comes from several sources, such as muscles in the body other than the heart and loose electrodes, and it results in a low amplitude random signal superimposed on the trace. This noise shows up in the same bandwidth with the

ECG signals as extra peaks and valleys in the trace and thus affects the algorithm which finds the extrema in the trace.

One method of minimizing the extra minima and maxima generated in this way is to have the algorithm ignore those extrema separated from their neighbours by less than the amplitude of the noise in the trace. To accomplish this selectivity, however, the algorithm needs a good measurement of the noise amplitude. The accuracy of the measurement is crucial to the expressiveness of the token algorithms; if the measurement is too large, tokens will be missed, and if it is too small, many tokens that do not represent heart activities will be generated.

While the importance of the noise measurement can be simply stated, it is very difficult to make accurately. In the algorithm used in this system the noise is estimated as having an amplitude equal to the most common change found in the trace over all 10 msec segments. That is, a 10 msec window is moved over the ECG trace and the largest change in f within the window is calculated at each point. Since the digitizer has a finite resolution, the measured changes will be of finite number and the most common change is taken to be g, the approximate noise amplitude.

This is a very simple algorithm which can be best justified through a qualitative argument. For low noise amplitude, the slowly varying ECG trace will give changes of only a few digitizer units and a low g, but a high noise

amplitude will superimpose many high amplitude short duration changes on the slow trace and give a larger g.

A similar value, dg, is calculated for the derivative as:

$$dg = \frac{3}{4}\frac{g}{\Delta t}$$

Examining the equation given for the derivative in the last section will show dg to be slightly more than the maximum change in the derivative if one point in the trace over the interval n-2 to n+2 is incorrect by an amount g.

This simple method for measuring g works adequately for the sample ECG traces used for testing the algorithms in this thesis. However, because the accuracy of the measurement is crucial for the sufficient expressiveness of the entire system, future work should include a more sophisticated method for measuring it, if possible.

## A.4 Smoothing to Eliminate Noise

Another way to minimize the effects of noise of the algorithm for finding extrema in the trace is to eliminate as much of the noise as possible from the trace. Eliminating the noise is difficult since the noise resides in the same frequency band as the ECG signals, but, since all the high frequency content of the trace is concentrated in the QRS complexes, some noise can be reduced by selectively filtering the trace between QRS's.

The QRS complexes contain frequency components from zero

to several tens of Hz, but the P and T-waves are mainly constructed of components with frequency less than 10 Hz [29]. Therefore, digitally filtering the P and T-waves down to 10 Hz will eliminate the higher frequency noise superimposed upon them. To accomplish this task a two step process is used.

First, the entire ECG trace is filtered using a flat moving average filter of the form:

$$f_{out}(t) = \frac{1}{2\alpha} \int_{x=t-\alpha}^{x=t+\alpha} f_{in}(x)dx$$

- t is the time
- $2\alpha$ is the width of the smoothing window

For the general sinusoid,

$$f_{in}(t) = Ae^{i\omega t}$$

This filter has amplitude transfer characteristic:

$$\frac{f_{out}(t)}{f_{in}(t)} = \frac{\sin\alpha\omega}{\alpha\omega}$$

This characteristic has no phase shift and for $\alpha$=.050 sec, as used in the programs in this thesis, has a frequency cut off, i.e. 3db down point, of approximately 9 Hz.

Applying this filter to the whole trace eliminates not only the noise in the P and T-waves, but most of the QRS complexes as well. Therefore, to return the QRS complexes, the second step of the algorithm checks each filtered point to see whether the original trace anywhere within + or - $\alpha$ differs

from the filtered curve by more than 2g (from section A.3). If so, the filtered point is replaced with the original point. This comparison and replacement ensures that changes in the ECG trace larger than noise are put back into the filtered trace, returning the QRS complexes, and the + or - $\alpha$ window minimizes the discontinuities that can occur in a straight point-for-point comparison.

The width of the comparison window, however, can interact with an inaccuracy in measuring g to eliminate from the filtered output small waves very near QRS complexes. For example, consider figure A-1.

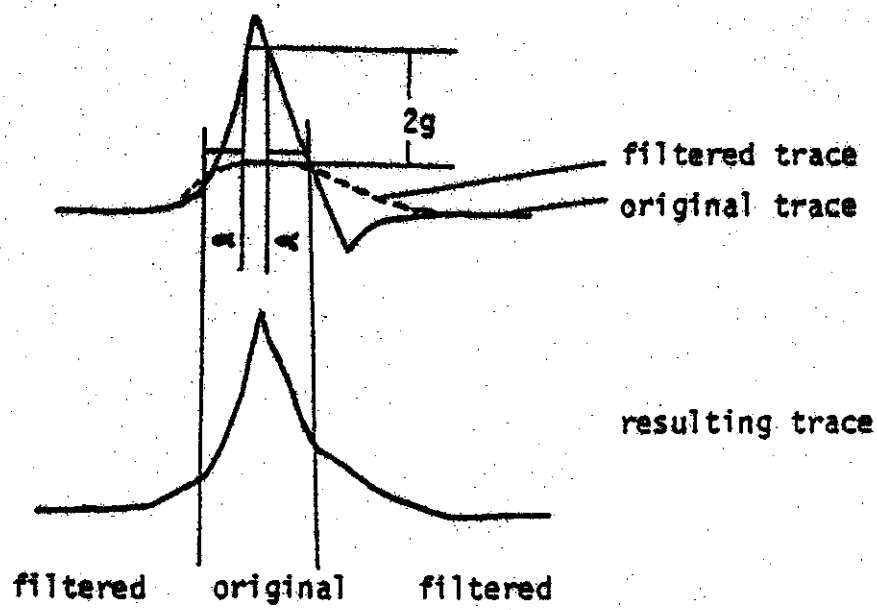


Figure A-1: The elimination of some small waves by the smoothing algorithm

In the figure, the filtered trace has smoothed away the small wave completely, as would be expected. However, in the comparison of the original to filtered traces, the two traces do not differ over the small wave by greater than 2g. Therefore, it cannot be put back in. This error can also be seen in figure B-3.

This problem of smoothing out small waves near the QRS complexes can be overcome in two ways. First, a different smoothing technique could be combined with a more accurate measurement of g to retain the smaller waves. Second, by acknowledging that the loss occurs and knowing it will happen only near QRS complexes, it can be compensated for later in the diagnosis stage through feedback queries that check for small waves near the QRS's. In practice, a compromise between the ability to retain all waves and the amount of extra work necessary in the diagnosis stage will have to be reached.

## A.5 Summary

In summary, the various signal processing aspects discussed with respect to the algorithms used in this thesis suggest three points to consider in future work:

1. A standard for ECG trace data input like that given in [58] should be adhered to.
2. The derivative calculation used should be adequate unless diagnosis measurements more accurate than 0.2% are needed.
3. The noise measurement and smoothing algorithms should be made more robust.

# Appendix B. EXAMPLES OF THE PROCESSING OF FIVE ECG TRACES

To aid in the understanding of the material contained in the previous chapters of this thesis, and to supply examples of the output from the different algorithms described in detail in Appendix C, five completely processed sample ECG traces are included in this appendix.

With each sample trace are shown the partial wave group tokens and whole wave group tokens that it breaks into, along with the QRS- complexes and P and T-waves found by the labelling algorithm and single diagnosis feedback query described in chapter 4.

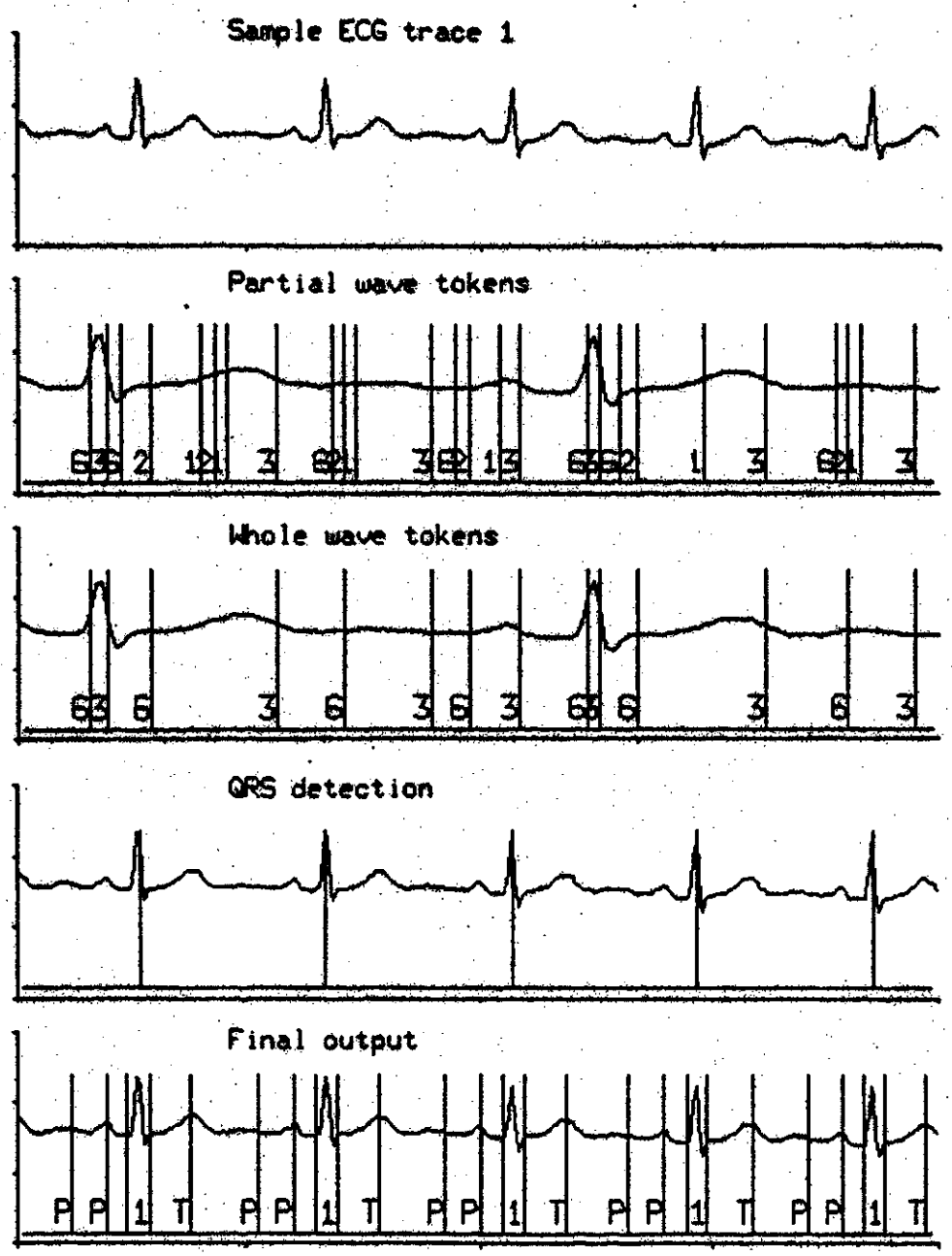


**Figure B-1:** The first sample trace. Sampling rate is 100Hz, and the noise value is 6 digitizer units.

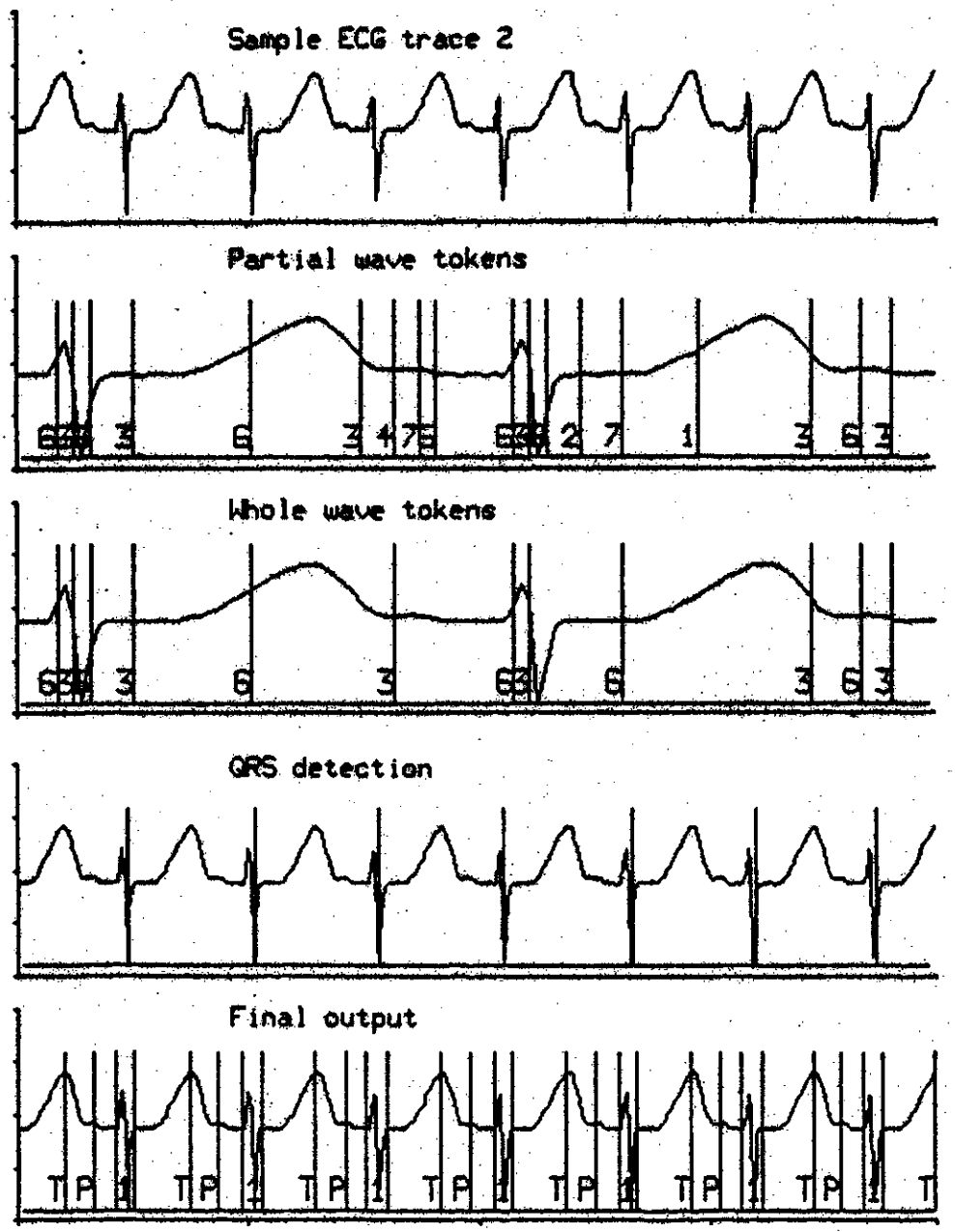


Figure B-2: The second sample trace. Sampling rate is 100Hz, and the noise value is 15 digitizer units.

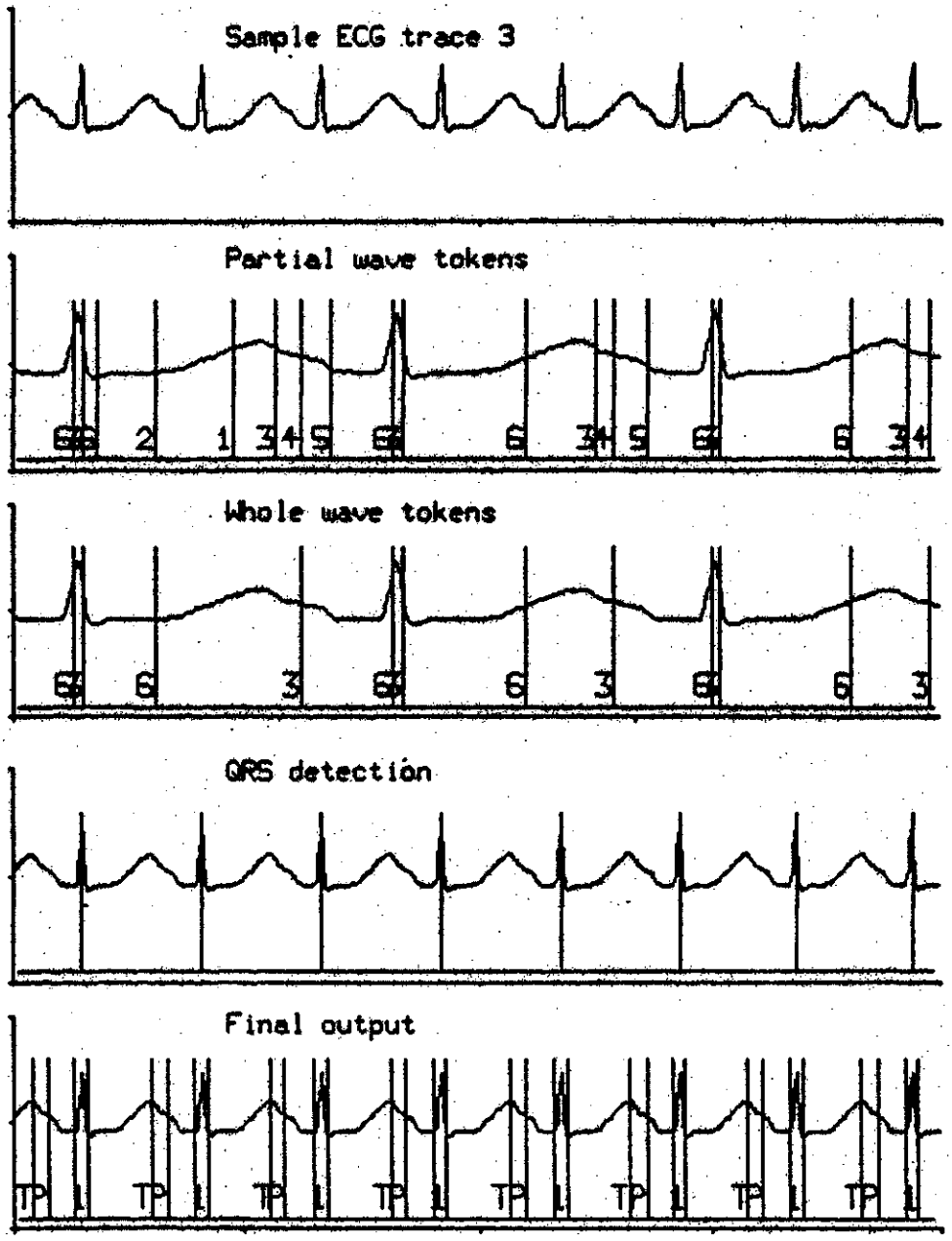


**Figure B-3:** The third sample trace. Sampling rate is 250Hz, and the noise value is 10 digitizer units.

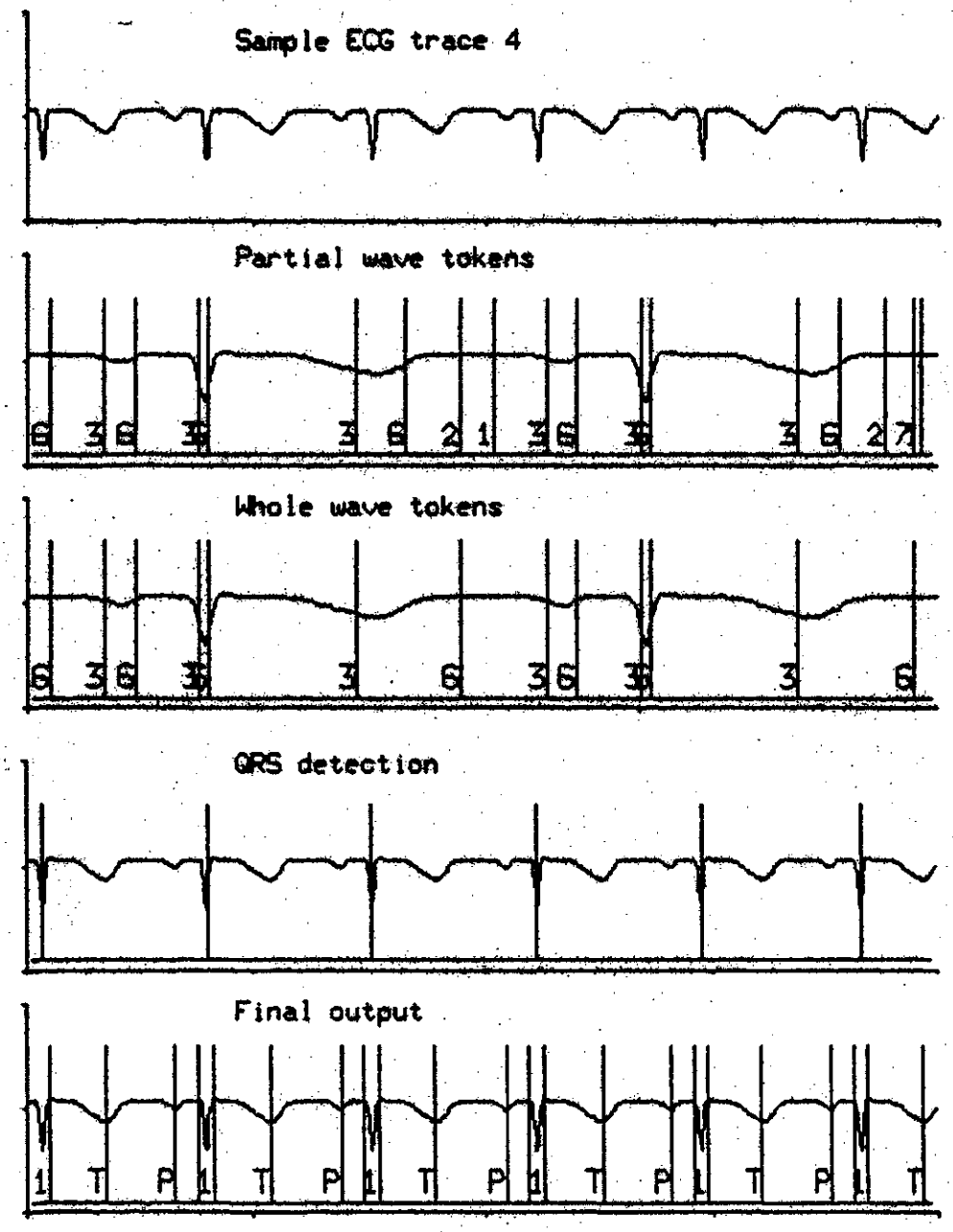


**Figure B-4:** The fourth sample trace. Sampling rate is 250Hz, and the noise value is 2 digitizer units.

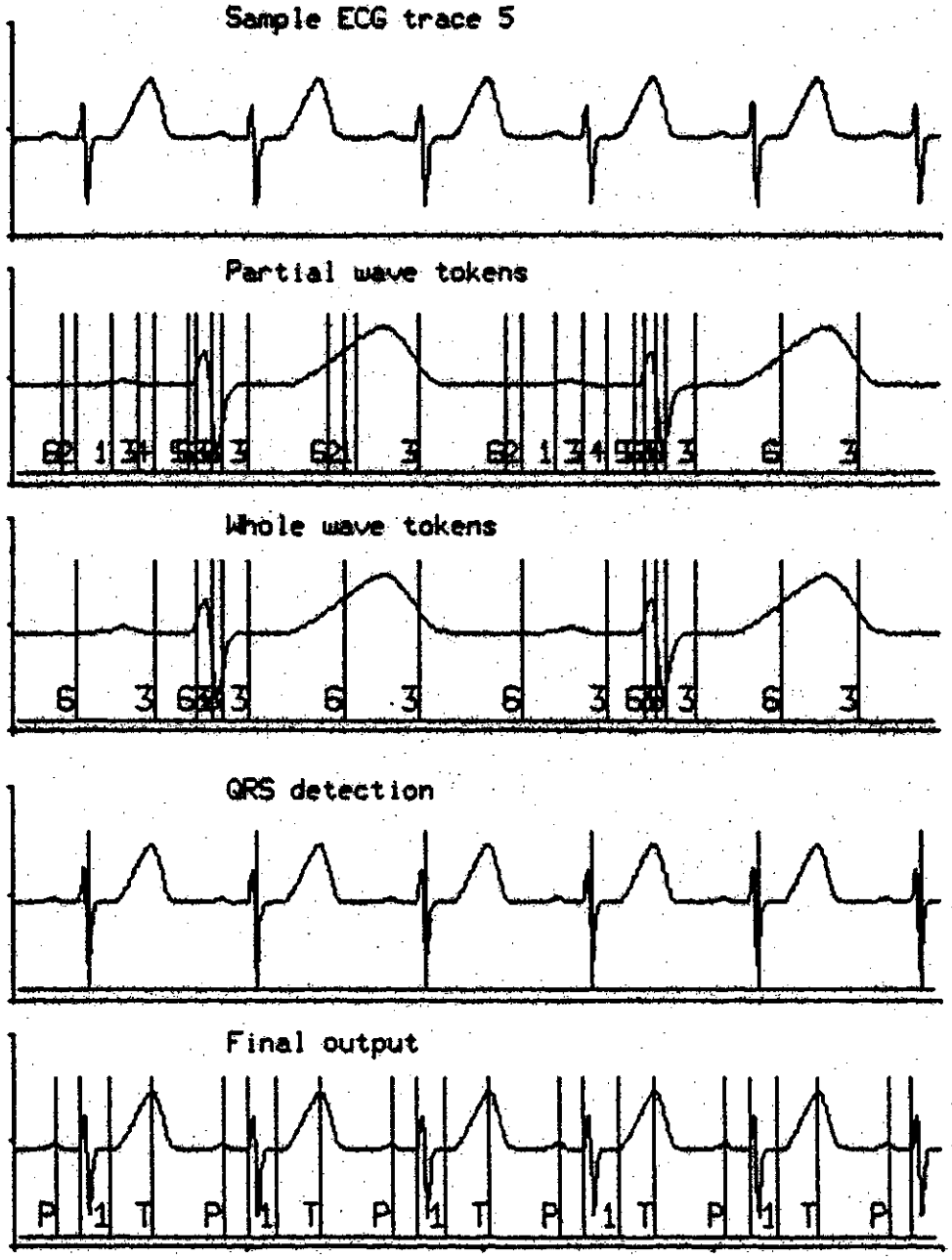


**Figure B-5:** The fifth sample trace. Sampling rate is 250Hz, and the noise value is 1 digitizer unit.

## Appendix C. ALGORITHMS FOR THE ECG INTERPRETATION SYSTEM

The algorithms for the reduction and labelling stages of the automatic ECG interpretation system outlined in chapter 4 are included in more detail in this appendix. The presentation format has been kept deliberately informal, as many of the details of general procedures, such as searches and data stuctures, are unimportant and can be implemented in many different ways. These algorithms have been implemented in Fortran-77 on the University of Saskatchewan College of Engineering VAX 11-780 and if more details are desired, the program sources may be obtained through the address on page ii of this thesis.

This appendix is organized in three sections: first, algorithms for the main program and non-specific subroutines; second, the reduction algorithms; and third, the labelling algorithms. All variables quoted in the algorithms are those in the Fortran sources. Examples of the results of each stage of processing using these algorithms are given in appendix B.

## C.1 Algorithm of the Main ECG Interpretation Program

This section contains the algorithm of the main program used in the ECG interpretation system, along with several routines used to take care of details not discussed in the main text of the thesis.

The routines included are:

- The main program
- ECG$READ: to read the trace into the program
- ECG$NOISE: to measure the noise amplitude in the trace
- ECG$GROSS_DRIFT: to remove severe artifacts from the trace
- ECG$BASELINE_DRIFT: to remove baseline drift from the trace

The algorithm for the main program of the

AUTOMATIC ELECTROCARDIOGRAM INTERPRETATION SYSTEM

Purpose: This is the main program used to process an ECG trace and generate the initial clinical labels.

Method: The program works by calling several routines to actually do the work.

Algorithm:

i) Call ECG$READ to read the ECG trace off the disk (already digitized) and save it in array 'in'. Set 'h' to be the time between samples of 'in'.

ii) Call ECG$NOISE to measure the noise in 'in' in mV. Save the noise amplitude in 'q'. Set the derivative noise amplitude to .75*'q'/'h'. A 10 msec window is used.

iii) Call ECG$GROSS_DRIFT to remove the worst baseline drift from 'in'. 500 msec is used as the smoothing width.

iv) Call ECG$SMOOTH_DIFF to smooth 'in' to remove some of the noise with the result saved in 'f'. The derivative 'df' is also calculated. 50 msec is used as the smoothing window.

v) Call ECG$FIND_QRS to locate the approximate positions of the QRS's in the trace and save them in 'rlist'. These are used later in ECG$GROW_QRS and also give the 'space' between the QRS's to use as the initial width in searching for local extrema in the trace.

vi) Call ECG$POINTS to find the max, min and zero-crossings in the derivative 'df'. These are saved in 'list' and define the start and end positions of the partial wave shape tokens. The intial window width used is one third the average time between QRS complexes.

vii) Call ECG$TOKENIZE to divide the trace into partial wave shape tokens, saved in

'tokens'. The endpoints come from 'list'.

viii) Call ECG$GROUP to find only the unambiguous whole wave group tokens within 'tokens'and save them in 'groups'.

ix) Call ECG$GROW_QRS to calculate the initial extent for each QRS in the trace and save them in 'qrs'.

x) Call ECG$QRS_CLASSIFY to group the QRS's into similar classes and correct those QRS's which had incorrect extents. The correct QRS extents are saved in 'qrs' and the class of each QRS is saved in 'qstat'.

xi) Call ECG$BASELINE_DRIFT to eliminate all baseline drift in the trace 'f'.

xii) Call ECG$FIND_SUBWAVES to find the unambiguous P and T waves represented by whole wave group tokens and save them in 'wave'.

Extension: Once step xii is complete, the arrays 'wave', 'qrs', and 'qstat' hold all the information which is necessary for figuring out initial possible diagnoses. The arrays 'f', 'df', 'tokens', 'groups' contain all the information required by feedback queries from the diagnosis stage.

```
 Subroutine ECG$READ
a           (fname, srate,calib, trace,np,maxt, ierr)
```

Variables:

```
fname  = the file name of the ECG trace to read
srate  = the sampling rate in Hz (returned)
calib  = the calibration in mV/pixel (returned)
         this is a constant that can be multiplied
         with each sampled value to convert it to mV
trace  = the trace in mV (returned)
np     = the number of points read into trace (returned)
maxt   = the maximum size of trace
ierr   = 0 for okay (returned)
```

Purpose: The purpose of this routine is to get the ECG trace off the disk add ready for processing. This routine is all that need be changed if the format of the ECG trace data on disk is changed.

Method: The data is just read off the disk and converted to mV.

Subroutine ECG$NOISE (f,np, calib,g, width,h, ierr)

Variables:

f = the ecg trace (in mV)
np = the length of f
calib = the step size of f (in mV/pixel)
g = the approximate noise (in mV) (returned)
width = the time window for noise check (in msec)
h = the between points in f (in sec)
ierr = 0 for okay, 1 for noise > 60 steps (returned)

Purpose: The purpose of this routine is to estimate the amplitude of the electrical noise in the ECG trace. The value obtained, 'g', is crucial to the correct working of routine ECG$POINTS.

Method: To estimate the noise, a histogram is formed measuring the number of times the change in the trace over 'width' is 1-60 steps. The most common change is taken as the amplitude of the noise.

The histogram has 21 slots. Slots 1-10 are one step wide for changes of 1-10, 11-12 are 5 steps wide for chages of 11-60 (ie. 11 is for 11-15, 12 for 16-20, etc.) and slot 21 is for step changes greater than 60.

Algorithm:

i) For each point 'k' in 'f' find the maximum change in 'f' over the interval 'width' centered on 'f(k)'. The change in steps is then the change in 'f' divided by 'calib'. If this change in steps is less than 10, increment the histogram value which equals it. If this change is greater than 10 but less than 60, increment the histogram value equal to the change divided by 5 + 9. If the change is greater then 60, increment the value for 60 or greater.

ii) Find the largest histogram value; it is the most common step change.

iii) Examine the histogram values for steps larger than the most common one and look for the first one .lt. a percentage, 'percnt' of the most common; this is the noise value.

iv) Convert the noise step size back to mV and return it.

Subroutine ECG$GROSS_DRIFT (in,np, wk, h, width)

Variables:

in = the input trace (input and returned)
np = the length of in,wk
wk = a work array of length np
h = the time between dat points (in sec)
width = the smoothing width (in msec)

Purpose: The purpose of this routine is to remove gross disturbances from the ECG trace. In particular, gross baseline drift. (This is also the place to add a pacemaker spike detector)

Method: To remove the baseline drift, the trace is first smoothed using a very wide, flat moving average filter to remove all ECG waves. This smoothed trace is then subtracted from the input removing most of the baseline drift.

Algorithm:

i) Calculate the smoothed trace, 'wk', by assigning each point, 'k', a value equal to the average of all points of 'in' within 'width' centered on point 'k'.

ii) Remove the drift by subtracting 'wk' from 'in'.

Subroutine ECG$BASELINE_DRIFT (f,np, wk, qrs,nq)

Variables:

f = the ECG trace (in mV) (input and returned)
np = the length of f
wk = a work array of length np
qrs = the QRS complex onset's and offset's
qrs(2,n) = the onset (in f)
nq = the number of QRS's (ie. length of qrs)

Purpose: The purpose of this routine is to correct the ECG trace for baseline drift once the QRS complexes have been found. This allows for the correct comparison of sub wave amplitudes in routine ECG$FIND_SUBWAVES.

Method: This routine works by calculating a linear interpolation between the onset of successive QRS complexes and subtracting it from the ECG trace.

Algorithm:

i) For each pair of QRS complexes, 'qrs(2,n)' and 'qrs(2,n+1)' calculate the slope and intercept of the line joining their onset points in 'f' ('f(qrs(2,n))' and 'f(qrs(2,n+1))').

ii) Subtract the line defined by the slope and intercept from 'f' between the QRS complexes.

iii) Go to i to do the next pair of QRS complexes.

## C.2 Reduction Algorithms

This section presents the reduction algorithms used in the ECG interpretation system. For a more general discussion see chapter 4.

The routines given are:

- ECG$SMOOTH_DIFF: to smooth some noise and calculate the derivative
- ECG$POINTS: to find the minima, maxima and zero-crossings in the derivative which define the end points of the partial wave shape tokens
- ECG$MIN_MAX: to find the extrema in a trace
- ECG$TOKENIZE: to find the partial wave shape tokens in the trace
- ECG$GROUP: to find the whole wave group tokens in the trace

```
Subroutine ECG$SMOOTH_DIFF
a                (trace, f,df,wk, np,h,dy,noise, width)
```

Variables:

```
trace = the trace to be smoothed (in mV)
f     = the smoothed output trace (in mV) (returned)
df    = the derivative of f (in mV/sec) (returned)
wk    = a work array
np    = the number of data points in trace,f,df,wk
h     = the time between data points (in sec)
dy    = the zero threshold for the derivative
        (in mV/sec)
noise = the smoothing threshold for the trace (in mV)
width = the moving average filter width (in msec)
```

Purpose: This program has two purposes. The first is to smooth the ECG trace to eliminate most of the influence of electrical noise and the second is to calculate the derivative of the trace.

Method: The trace is first smoothed be passing it through a flat, moving average filter. Then, the output trace is taken point by point as either the smoothed trace if the difference between the input and the smoothed trace is less than a threshold, 'noise', or as the original trace if the difference is greater. The derivative is then calculated point by point, using either the smoothed trace or the original trace, depending on which trace the output point was chosen from.

Algorithm:

i) Calculate the smoothed trace, 'wk', by assigning each point, 'k', a value equal to the average of all points of 'trace' within 'width' centered on point 'k'.

ii) For each point, 'k', compare 'wk(k)' and 'trace' over 'width'. If, at any point in the interval, the difference between 'wk' and 'trace' is greater than 'noise' then:

```
    f(k) = trace(k)
    df(k) is found using 'trace'
else
    f(k) = wk(k)
    df(k) is found using 'wk'
```

iii) Calculate the derivative of 'trace' using the formula below where 'g' is either 'wk'

or 'trace' as determined for each point above:

$$df(k) = (g(k-2) - 8g(k-1) + 8g(k+1) - g(k+2)) / 12'h'$$

To further reduce the influence of noise, set 'df(k)' to zero for every point 'k' where the absolute value of 'df(k)' is less than 'dy'.

```
 Subroutine ECG$POINTS
a    (f,df,np, work,work2, list,nl,maxl, g,dg,h, width)
```

Variables:

```
f     = the ECG curve (in mV)
df    = the derivative of f (in mV/sec)
np    = the number of points in f,df,work,work2
work  = a work array
work2 = another work array
list  = the list of significant points (returned)
        list(1,n) = the position of the point in f
        list(2,n) = 1 for max in derivative
                    2 for min in derivative
                    4 for zerocrossing in derivative
nl    = the number of points found (returned)
maxl  = the maximum size of list
g     = the noise amplitude in f (in mV)
dg    = the noise amplitude in df (in mV/sec)
h     = the time between sample points in f (in sec)
width = the initial min_max window width (in msec)
```

Purpose: This routine finds the list of points in the trace which define the endpoints of the partial wave shape tokens used in ECG$TOKENIZE.

Method: First the mins and maxs are found in 'f' (ie. these are the zero crossings) and then the mins and maxs are found in 'df'.

Algorithm:

i) Call ECG$MIN_MAX to find the mins and maxs in 'f' with the results in 'work'.

ii) Force 'df' to have only these zero crossings by setting all points in 'df' between a 1 and 2 which are +ve to zero and all points between a 2 and 1 which are -ve to zero.

iii) Call ECG$MIN_MAX to find the mins and maxs in 'df' with the results in 'work2'.

iv) Add all 1's and 2's in 'work' which do not fall on 1's or 2's in 'work2' to 'work2' as 4's.

v) Search through 'work2' and set any 1's where the derivative is -ve that follow a 4 to zero and the same with +ve 2's that follow a 4. (This ensures only physically

possible sets of points are generated.)

vi) Form 'list' by looking through 'work2' and for each 1,2,4 found, setting 'list(1,k)' to its position and 'list(2,k)' to the number.

Subroutine ECG$MIN_MAX (y,np, minmax, dy,dx, width)

Variables:

y = the input curve (in mV)
np = the length of y
minmax = the positions of the minimums and maximums (returned)
minmax(n) = 1 for max, 2 for min, 0 for neither
dy = the maximum spread for two equal points (in mV)
dx = the minimum closeness of two points (in pixels)
width = the initial width of the min max window (in pixels)

Purpose: This routine finds the local maxima and minima in the input trace.

Method: A window of initial width 'width' is moved over 'y' and at each point, the maximum and minimum points of 'y' within window are flagged in 'minmax' (if there are any). Then the window is made 10% smaller and moved over 'y' again. Once all mins and maxs are found this way, those which are closer together than 'dx' points are eliminated. Note that several constraints are used to insure that only physically real sets of extrema are produced.

Algorithm:

i) Move the window over the entire input trace and at the center point 'k':
   1) See if this point is more than 'dy' smaller than every point in the window (a min) 'minmax(k)'=2
   2) See if this point is more than 'dy' larger than every point in the window (a max) 'minmax(k)'=1

ii) Make the window 10% smaller. If it is larger than 'dx' then go to i.

iii) Search through minmax for two 1's or 2's in a row. If found, find the min or max between them. When this step is complete all important mins and maxs have been found.

iv) Look for mins and maxs that fall in an area of 'y' where the amplitude does not vary more than 'dy'. Find the centre of

this plateau. Then:
1) If the number of mins and maxs in the plateau is odd remove them all from 'minmax' and replace them with a single minmax (the same as the outside ones) in the centre.
2) If the number of mins and maxs is odd, leave the one farthest from the centre and replace the rest with a single minmax at the centre.

iv) Search through 'minmax' to find strings of minmax's that are closer to each other than 'dx' with amplitudes differing by less than 10*dy (empirical). Then:
1) If the end minmax's are the same, replace then with a single minmax at the most extreme point between.
2) If the end minmax's are different, remove them all.

v) Search through 'minmax' for two 1's or 2's in a row. Replace then with a single minmax at the most extreme point between.

```
      Subroutine ECG$TOKENIZE
     a                  (df,np, list,nl, tokens,nt,maxt, h)
```

Variables:

```
df      = the derivative of the ECG trace (in mV/sec)
np      = the number of points in df
list    = the min_max list from routine ECG$POINTS
          list(1,n) = position of the min_max in df
          list(2,n) = a code for the min_max
                      1 = a maximum in df
                      2 = a minimum in df
                      4 = a zero crossing in df
nl      = the length of the min_max list
tokens  = list of ECG partial wave tokens (returned)
          tokens(1,n) = a code for the token (ie. 1-6)
          tokens(2,n) = start position of the token in f
          tokens(3,n) = end position of the token
          tokens(4,n) = width of token in mV
nt      = the length of tokens (returned)
maxt    = the maximum length of tokens
h       = the time between sample points in df (in sec)
```

Purpose: The purpose of this routine is to divide the input ECG trace up into partial wave shape tokens.

Method: This routine works by examining the start and end positions of each token which are defined by successive min_max's (ie. 'list(1,n)' and 'list(1,n+1)'), and applying a set of rules based on the endpoints to decide on the type of token in between.

Once this division into tokens is complete, the routine applies two rules to remove tokens generated by noise. First, some single sample point wide tokens are removed as there is a high probability they are caused by noise (for any reasonably high sample rate, ie. >100Hz). Second, duplicate tokens are compressed into a single one (ie. two 7's in a row become one 7, etc.).

Algorithm:

i) For each min_max, 'list(1,k)', examine the min_max before 'list(1,K-1)' (and perhaps the one after 'list(1,k+1)') to see which type of token this min_max ends. Use the following rules and put the result in 'tokens':

1) If this min_max is a max and the one before is a min and the derivative at this point is -ve then this min_max ends a type 4 token.

2) If this is a max and the one before is a min and the derivative at the one before is -ve then this is a 6.

3) If this is a max and the one before is a min and the derivative at the one before is +ve then this is a 1.

4) If this is a max and the one before is a zero crossing then this is a 1.

5) If this is a min and the one before is a max and the derivative at this one is +ve then this is a 2.

6) If this is a min and the one before is a max and the derivative at the one before is +ve then this is a 3.

7) If this is a min and the one before is a max and the derivative at the one before is -ve then this is a 5.

8) If this is a min and the one before is a zero crossing then this is a 5.

9) If this is a zero crossing and the one before is a zero crossing then this is a 7.

10) If this is a zero crossing and the one before is a max and the one after is a min then this is a 3.

11) If this is a zero crossing and the one before is a max then this is a 2.

12) If this is a zero crossing and the one before is a min and the one after is a max then this is a 6.

13) If this is a zero crossing and the one before ia a min then this is a 4.

ii) After all tokens have been found, search 'tokens' for tokens with width 'h'. If any are found, check to see if they fall in one of the following sequences (where the token in the middle is the single sample wide one) and replace the sequence

with the one shown:

1) 2,1,2 => 2
2) 2,1,3 => 2,3
3) 1,2,1 => 1
4) 6,2,1 => 6,1
5) 5,4,5 => 5
6) 3,4,5 => 3,5
7) 4,5,4 => 4
8) 4,5,6 => 4,6
9) 4,7,1 => 6
10) 2,7,5 => 3

iii) Search through tokens for the same token appearing twice in a row and merge them if found.

iv) For each token, calculate its width in msec.

```
 Subroutine ECG$GROUP
a                (f,np, tokens,nt, groups,ng,maxg, g,h)
```

Variables:

```
f       = the ECG trace (in mV)
np      = the number of points in f
tokens  = the input set of ECG tokens
             (see ECG$TOKENIZE)
nt      = the number of tokens
groups  = the set of output whole wave group tokens
             (returned)
          group(1,n) = the group code (ie. 3 or 6)
          group(2,n) = start position of group in f
          group(3,n) = end position of group in f
          group(4,n) = width of the group token (in mV)
ng      = the number of groups found (returned)
maxg    = the maximum size of groups
g       = the noise amplitude in f (in mV)
h       = the time between sample points (in sec)
```

Purpose: This routine merges the partial wave shape tokens into unambiguous whole wave group tokens.

Method: The tokens which define an unambiguous wave are found. Then, these tokens are expanded outward until they touch.

Algorithm:

i) Find the next set of tokens defining an unambiguous whole wave group, ie. one of the following:
3 or 6 or 2,7,5 or 4,7,1

ii) Apply these rules to see where this group begins, ie. meets the last one found:

1) If this group touches the previous one then it begins where they touch.

2) If this group is separated from the previous one by 1 token and that token is a 7, then this group starts at the end of the token, else it starts at the beginning.

3) If this group is separated from the previous one by 2 tokens then this group starts between the tokens.

4) If this group is separated from the

previous one by 3 or more tokens then:

a) Find the midpoint in 'f' between the end of the last group and the start of this one.

b) Find the token containing this point.

c) If the token is next to the last group, this group begins where the token ends.

d) If the token is next to this group, this group begins where the token begins.

e) If the token is not next to this or the last group then:
   - if the token is a 7, this group begins at the end of the token closest to the midpoint.
   - if this group is downward. (ie. 6) and the token is 5, this group begins at the atart of the token, else the end of the token.
   - if this group is upward (ie. 3) and the token is 1, this group begins at the start of the token, else at the end.

iii) This group group ends at the place the next group begins (ie. go to i).

iv) Calculate the width of each group in msec.

## C.3 Labelling Algorithms

This section presents the algorithms used in the labelling stage of the automatic ECG interpretation system. For a further discussion of the algorithms, see chapter 4.

The routines given in this section are:

- ECG$FIND_QRS: to find the approximate positions of the QRS complexes in the trace
- ECG$GROW_QRS: to grow the QRS complexes to their proper extent
- ECG$GROUP_PROPERTIES: to find the relevant properties of whole wave group token
- ECG$WAVE_SHAPE: to calculate the shape of a whole wave group token
- ECG$QRS_CLASSIFY: to classify and correct the QRS complexes into types
- ECG$FIND_SUBWAVES: to find the unambiguous waves between the QRS's
- ECG$TOKEN_SUBWAVE: to find overlapping waves within the partial wave shape token set

Subroutine ECG$FIND_QRS (dx,lx, list,ls,nf, peak,period)

Variables:

dx = the derivative of the ECG trace
lx = the length of dx
list = the position of the QRS complexes detected (returned)
ls = the maximum length of list
nf = the number of QRS complexes found (returned)
peak = the threshold for QRS detection as a fraction of the peak derivative amplitude found.
period = the average period between QRS's (in pixels)

Purpose: This routine detects the approximate position of the QRS complexes in the trace as reference points for ECG$GROW_QRS.

Method: The QRS complexes are detected at points where the derivative crosses a threshold defined by 'peak' times the maximum value of the derivative amplitude.

Algorithm:

i) Search through the derivative for the largest absolute value of the derivative, 'dxmax'.

ii) Set the threshold at 'dxmax'*'peak'.

iii) Search through the derivative from the beginning for a point where the absolute value of the derivative is greater than the threshold. Save this as a QRS in 'list' and resume the search at the point where the derivative falls below the threshold again.

iv) Calculate 'period' as the average number of sample points between QRS's.

```
 Subroutine ECG$GROW_QRS
a  (f,df,np, rlist,nrl, groups,ng, qrs,nq,maxq, g,dg,h,ds)

 Variables:

 f       = the ECG trace (in mV)
 df      = the derivative  of f (in mV/sec)
 np      = the length of f and df
 rlist   = the approximate qrs positions (in pixels)
             from ECG$FIND_QRS
 nrl     = the number of qrs positions
 groups  = the whole wave group tokens from ECG$GROUP
           groups(1,n) = the group (ie. 3 or 6)
           groups(2,n) = the start position in f
           groups(3,n) = the end position in f
 ng      = the number of groups
 qrs     = the qrs complexes (returned)
           qrs(1,n) = onset of qrs in f
           qrs(2,n) = offset of qrs in f
 nq      = the number of qrs's (returned)
 maxq    = the maximum size of qrs
 g,dg,h  = the ecg trace parameters (noise, sample interval)
 ds      = the division between 'pointy' and 'shallow'
           wave shapes

 Purpose: This routine "grows" the QRS complexes found
          in ECG$FIND_QRS to their proper extent.

 Method:  The whole wave group token that the qrs
          position from ECG$FIND_QRS falls in is
          identified and then the whole waves on each
          side are checked to see if they should be
          added to it to form the QRS.

 Algorithm:

          i) For each qrs marker (ie. 'rlist(knrl)')
             find the whole wave group token that it
             falls in.

         ii) Call ECG$GROUP_PROPERTIES to find the
             properties of the first qrs group token
             found above.  'Left(1)'=1 for a left stop
             and 'right(1)'=1 for a right stop.

        iii) If a left stop is found stop the qrs
             growing on the left and set 'qrs(1,ng)' to
             the position of the stop ('left(2)').  Do
             the same on the right.

         iv) If either the left or right side of the
             qrs still needs to be added to (ie. no
             stops) then call ECG$GROUP_PROPERTIES for
```

the whole wave group token on that side.

v) If the width of the group on the left is less than the group on the right and the group on the left should be added then:

1) If the left group has a right stop, end addition to the left side at the stop.
2) If adding the group on the left would bring the total qrs width to >220ms then add the group only to its peak.
3) If the left group is wider than 150ms then add it only to its peak.
4) If the left group has width <50ms then add the whole thing and consider the next one over.
5) If the left group has width >50ms and <150ms check its shape to see if it is "peaked" enough to add. Ie., if the average slope of the left group is greater than 'ds' times the average slope of every wave added to this QRS so far, add the whole group and consider the next, else add the group only to its peak.

vi) If the right group width is less than the left and the right side is free to be added to, check the right side with the same rules as above but make the following changes:
1) 115ms => 150ms
2) 50ms => 60ms
3) right => left and left => right

vii) If there is a still a side to be added to go to iv.

viii) If there are more qrs's to do go to i.

ix) Once all qrs's are extended check their end points to see if any overlap. If two QRS's overlap, move their end points to the middle of the overlap so they only just touch.

```
 Subroutine ECG$GROUP_PROPERTIES
a                (f,df,np, groups,kg, peak,time,
b                 left,right,shape, g,dg,h, idir)
```

Variables:

```
f,df,np = the ecg trace, its derivative, and its
          length
groups  = the list of ecg groups
kg      = the group for which the properties are to
          be found
peak    = the location of the group peak (returned) (in f)
time    = the various important widths in the group
            (returned) (in msec)
          time(1) = time to the left plateau or corner
          time(2) = time between plateaus or corners
          time(3) = time to the right plateau or corner
left    = a left stop (returned)
          left(1) = 1 a left corner
          left(2) =  the position of the corner in f
right   = a right stop (returned)
          right(1) = 1 a right corner
          right(2) = the position of the stop if f
shape   = the average slope over the group (returned)
g,dg,h  = the ecg trace parameters (noise, interval)
idir    = the type of group properties wanted
          (1=left, 2=middle, 3=right)
```

Purpose: This routine looks for the peak, shape, and stops of a group as required for ECG$GROW_QRS and ECG$QRS_CLASSIFY.

Method: First, the peak of the group is found, and then the left side and right side are searched for a stop which could be either a corner or a plateau.

Algorithm:

i) Search 'f' between the ends of the group to find the 'peak' of the group. (ie. the highest point for a type 3 group or the lowest point for a type 6 group)

ii) Look for a left stop using a 25ms window and the following rules:
   1) If 'idir' is 1 or 2 start the window at 'peak' and move it to the left, else start at the left edge, 'groups(2,kg)', and move right.
   2) If 'df' over the whole window is < dg then a plateau stop has been found. (at the right edge of the window for 'idir'

1 or 2 and the left edge for 'idir' 3)

3) If the change in 'df' from one endpoint of the window to the other is more than 30 degrees then a corner stop has been found. (at the left edge of the window for 'idir' 1 or 2 and the right edge for 'idir' 3)

4) Move the window over one point and go to 2. Stop at the left edge for 'idir' 1 or 2 and at 'peak' for 'idir' 3.

iii) Look for a right stop using a 25ms window and the above rules, but for 'idir' 1 search right and for 'idir' 2 or 3 search left. (also left=>right and right=>left)

iv) Call ECG$WAVE_SHAPE to find the 'shape' of the group.

v) Calculate the 'time' values in msec.
   time(1) = time from left edge to left stop.
   time(2) = left stop to right stop.
   time(3) = right stop to right edge.

Subroutine ECG$WAVE_SHAPE (df,np, L,R, shape)

Variables:

df = the derivative of the ECG trace (in mV/sec)
np = the length of df
L,R = the left and right boundaries of the wave in df
shape = the shape (returned)

Purpose: This routine calculates a value to represent the 'peakedness' of an ECG wave.

Method: The 'shape' is calculated simply as the average of 'df' from 'L' to 'R'.

```
 Subroutine ECG$QRS_CLASSIFY
a                (f,df,np, groups,ng,qrs,nq, qstat,
b                                    g,dg,h,ds, ierr)
```

Variables:

```
f,df    = the ecg trace and derivative
np      = the length of f,df
groups  = the ecg trace group tokens from ECG$GROUP
ng      = the number of groups
qrs     = the qrs boundaries in f from ECG$GROW_QRS
nq      = the number of qrs complexes
qstat   = the qrs classifications (returned)
          qstat(1,n) = first group in the qrs
          qstat(2,n) = the initial class of the qrs
          qstat(3,n) = the final class of the qrs
g,dg,h  = the parameters of the trace (noise, interval)
ds      = the fraction of a 'shape' to consider the
          same
ierr    = 0 for okay
```

Purpose: This routine has two purposes. The first is to classify the QRS complexes into classes of similar shape, thus separating the normal beats from the abnormal ones. The second is to check the classifications and if two are found to be very similar, assume the smallest has had its extent incorrectly measured and correct it.

Method: The QRS complexes are initially classified by comparing the width of each whole wave token within the QRS to those in the other QRS's. If they are within +or- 25ms of each other, the QRS's are assumed to be in the same class.

Once all QRS's are classified, the class averages are compared in the same way, and those classes with widths that overlap are considered to be similar, and the smaller class is corrected to have the same average extent as the larger.

Algorithm:

i) For each QRS, calculate the 'shape' of each whole wave token within it using ECG$WAVE_SHAPE. Then, flag each whole wave token within it as significant if it has a shape greater than 'ds' times the maximum shape found in the QRS. Compare this QRS to every QRS class already defined.

1) If the QRS contains a different number of whole wave tokens then it is not in this class.

2) If the significant waves in the QRS do not line up with those in the class then it is not in this class.

3) If with this QRS added into the average, the maximum deviation from the average of any whole wave token within the QRS is more than 25ms then the QRS is not in this class.

4) If the QRS passes the above tests it is in the class it is being compared to.

5) If the QRS is not in any already defined class it defines a new class.

ii) For every QRS class calculate:
- the average width of each wave in the class
- the maximum +ve and -ve deviation of any wave in the class from the average
- the maximum and minimum 'shape' for each wave in the class

iii) Once all QRS's have been classified above compare all classes to see if any were formed due to an error in calculating the QRS extent. Use the following rules:

1) Find the class with the most members that hasn't been checked and compare all other unchecked classes to it.

2) When comparing two classes compare only the significant waves (as others may be in error). "Slide" the classes back and forth trying every orientation where at least one significant wave matches in each class.

3) Compare the widths of all significant waves (except the outside two which may be the measurement error). If the maximum deviations from the average do not overlap for even one wave, these classes are not similar.

4) Compare the shapes of the outside significant waves. If the maximum deviations do not overlap, these are

not similar classes.

5) If these two classes are similar, mark the the smaller one as due to measuement error which should be corrected to the width of the larger one.

iv) For every class that is marked for correction set up the corrected extent using the following rules:

1) Lining up this class with the correct class, find the last whole waves on the left and right which match.

2) Calculate the time to add or subtract from the boundary of this last correct token, to correct the extent of this class to match that of the correct class.

v) For every class that needs to be corrected, correct each QRS in the class using the data calculated in iv.

```
 Subroutine ECG$FIND_SUBWAVES
a           (f,df,np, tokens,nt, groups,ng,
b            qrs,nq,qstat, wave,maxw,nw, g,dg,h)
```

Variables:

```
f,df   = the ECG trace and its derivative
np     = the length of f,df
groups = the whole wave group tokens from ECG$GROUP
ng     = the number of group tokens
qrs,nq = the positions and extents of the QRS complexes
           from ECG$QRS_CLASSIFY
qstat  = statistics about the QRSs from
         ECG$QRS_CLASSIFY
         qstat(2,n) = the class of each QRS
wave   = the subwaves found between QRS's (returned)
         wave(1,n) = the position of the wave peak in f
         wave(2,n) = the label of the wave (1 T, 2 P)
maxw   = the maximum size of wave
nw     = the number of subwaves found
g,dg,h = the parameters of the trace (noise, interval)
```

Purpose: This routine finds the unambiguous waves in the ECG trace which fall between the QRS complexes. These waves are to be used as initial input for the diagnostic section.

This routine at present also simulates a single diagnostic rule which finds some partial (ie. overlapping P waves). This part of the routine should be removed in future work.

Method: To find the initial sub waves, the whole wave group tokens are searched between the offset of one QRS ('qrs(3,n)') and the onset of the next ('qrs(2,n)') for a token with a peak larger than that of the tokens on either side. These sub waves are returned in 'wave'.

The diagnosis rule simulation checks the sub-waves already found, and if only one is found between QRS's, then a partial wave token of the appropriate type is searched for between the peak of that wave and the next QRS onset.

Algorithm:

i) For each QRS, find the first whole wave group token after it, 'isg', and the last whole wave group token before the next QRS, 'ieg'.

ii) For each group from 'isg' to 'ieg', find the position of its 'peak' and the peaks of the groups to its right 'peakR' and left 'peakL'. If the absolute value of 'f(peak)' is larger than that of 'f(peakL)' and 'f(peakR)' then this wave group token represents a sub-wave. Save it in 'wave'.

iii) If there are more QRS's go to i.

iv) Label the first sub-wave found between each pair of QRS's a T wave and any others found P waves.

Algorithm for the diagnosis rule:

i) If no sub-waves were found between two QRS's then call ECG$TOKEN_SUBWAVE to search for a partial wave shape token that may be a P or T overlapping a QRS.

ii) If only one sub wave was found between QRS's then call ECG$TOKEN_SUBWAVE to search for a partial wave shape token between the peak of the wave found and the onset of the next QRS that has the same orientation as the sub-wave and may be a P overlapping a T or QRS.

```
 Subroutine ECG$TOKEN_SUBWAVE
a                  (f,df,np, tokens,nt, is,ie, peak,
b                                width, g,dg,h, idir)
```

Variables:

```
f,df,np = the ECG trace, derivative, and length
tokens  = the partial wave shape tokens from ECG$TOKENIZE
nt      = the number of tokens
is,ie   = the start and end points in f to search
peak    = the peak point in the token found (returned)
g,dg,h  = the parameters of the ECG trace
idir    = the type of subwave to look for
          (1=up,3=down,2=both)
```

Purpose: This routine is a feedback query to the partial wave shape tokens to pick out those tokens which may represent overlapping waves.

Method: The partial wave tokens, 'tokens', are searched from 'is' to 'ie' and the token found with the greatest amplitude in direction 'idir' is returned.

Algorithm:

i) Find the partial wave shape token containing 'is', 'ist', and the token containing 'ie', 'iet'.

ii) If 'idir' is 1 then search from 'ist' to 'iet' for the token of type 2,5,3 with the largest amplitude.

iii) If 'idir' is 2 then search from 'ist' to 'iet' for the token of type 1,4,6 with the largest amplitude.

iv) If 'idir' is 3 search for the largest amplitude type 1,2,3,4,5,6 token.

v) Return the location of the 'peak' of the token found.