85-14

Compression of the ECG Knowledge-base Using the AQ Inductive Learning Algorithm

85-14

Igor Mozetic

Department of Computer Science

University of Illinois at Urbana-Champaign

Urbana, Illinois

on leave from:

Jozef Stefan Institute

Ljubljana, Yugoslavia

File No. UIUCDCS-F-85-943

ISG REPORT 85-13

March, 1985

This research was supported in part by the Fulbright scholarship, the National Science Foundation under Grant No. DCR 84-06801, the Office of Naval Research under Grant No. N00014-82-K-0186, and by the Slovene Research Council.

Table of contents

Abstract

Medical expert system KARDIO contains, among other subsystems, qualitative model of the heart. The model simulates electrical activity of the heart and was used for automatic generation of the ECG knowledge-base. The knowledge-base relates all possible multiple heart disorders (cardiac arrhythmias) with their corresponding ECG descriptions. In principle, it can be used for diagnosing of cardiac arrhythmias. However, as the knowledge-base consists of more than 2400 rules, it requires too much computer resources to be usefull. The main motivation for this research was the need to transform the knowledge-base into computationally more efficient form, i.e., to compress it. The paper describes how can we transform rules that relate objects defined by attribute-values vectors and what information is lost during the transformations. Further we show how an inductive learning algorithm can be applied on the transformed rules to simplify them and how the induced rules can be used for inference. We transformed the ECG knowledge-base in various ways to compress it and to obtain human understandable descriptions of cardiac arrhythmias. Specifically, we produced diagnostic rules that require 200 times less storage than the original knowledge-base. We induced descriptions of cardiac arrhythmias (that can be simple or multiple heart disorders) which in many cases turned out to be much more detailed as we can find them in medical literature. And finally, we justified one of the motives for development of the heart's model by showing that construction of ECG descriptions for combined arrhythmias is not trivial and most probably cannot be done manually.

Key words:

knowledge representation, inductive learning

1. Introduction

We first give a short overview of the expert system KARDIO and discuss problems concerning construction of the knowledge-base for ECG diagnosis of heart disorders. The knowledge-base was constructed automatically through qualitative model of the heart. Complexity of the knowledge-base forced us to transform it into computationally more efficient form. We present a general method of compression using inductive learning algorithm. The method is applicable to any knowledge-base with rules in the form of implication and objects represented by attribute-values vectors if certain conditions are satisfied. The method was applied on the ECG knowledge-base to obtain human understandable descriptions of heart disorders and to produce efficient diagnostic rules.

1.1. Expert system KARDIO

KARDIO is a medical expert system for diagnosis and treatment of cardiac arrhythmias that consists of several subsystems. In the paper we concentrate on the problem of electrocardiographic (ECG) diagnosis of cardiac arrhythmias. Cardiac arrhythmias are disorders in the electrical activity of the heart. The ECG is a graphic recording of this electrical activity. To interpret a given ECG means to identify those disorders in the heart which caused this ECG.

In KARDIO each ECG is described in symbolic, *qualitative terms* by a set of attributes and corresponding values. So, instead of diagnosing directly from ECG curve, the problem for KARDIO is: for a given ECG description, find those arrhythmias that correspond to such an ECG. To do that, KARDIO uses an ECG knowledge-base which specifies the relations between possible arrhythmias and their corresponding ECG descriptions. Those relations are represented by a set of rules of the form:

$$\text{ARR} \Rightarrow \text{ECG}$$

where ARR is an arrhythmia and ECG is a disjunction of all corresponding ECG descriptions:

$$\text{ECG} \Leftrightarrow \text{Ecg}_1 \vee \text{Ecg}_2 \vee \ldots \vee \text{Ecg}_n$$

In the following example for "junctional_tachycardia with left_bundle_branch_block" we use Annotated Predicate Calculus notation [Michalski 83]. The arrhythmia has associated three disjunctive ECG descriptions. They are represented by two conjunctions and two disjunctive values of the attribute "regular_PR" in the first conjunction (internal disjunction):

{junctional_tachycardia, left_bundle_branch_block} =>

[rhythm = regular] &
[regular_P = abnormal] &
[rate_of_P = between_100_250] &
[relation_P_QRS = after_P_always_QRS] &
[regular_PR = shortened, after_QRS_is_P] &
[regular_QRS = wide_LBBB] &
[rate_of_QRS = between_100_250]

v

[rhythm = regular] &
[regular_P = absent] &
[rate_of_P = zero] &
[relation_P_QRS = meaningless] &
[regular_PR = meaningless] &
[regular_QRS = wide_LBBB] &
[rate_of_QRS = between_100_250]

The problem of specifying relations between simple arrhythmias and their corresponding ECG descriptions is relatively simple. However, the problem arise when several disorders are presented in the heart at the same time, as in the example above. This gives rise to a *multiple arrhythmia* which is in general a combination of one or more simple arrhythmias. As a combination of simple arrhythmias Arr_1, Arr_2, ... is a commutative and associative relation, we can denote it by $\{Arr_1, Arr_2, ...\}$. There are few thousand multiple arrhythmias, and therefore it was impossible to construct complete ECG knowledge-base by manual tabulation. Besides that, we could not find any simple combination function for computing an ECG description of a multiple arrhythmia, say $\{Arr_1, Arr_2\}$, given ECG descriptions ECG_1 and ECG_2 of simple arrhythmias Arr_1 and Arr_2, respectively:

$$\{Arr_1\} \Rightarrow ECG_1$$
$$\{Arr_2\} \Rightarrow ECG_2$$

$$\{Arr_1, Arr_2\} \Rightarrow f(ECG_1, ECG_2) = ??$$

These problems were the main motivation for developing the model of the heart that simulates its electrical activity. The model was used for automatic generation of all physiologically possible combinations of 30 simple arrhythmias and their corresponding ECG descriptions. The thus generated exhaustive ECG knowledge-base relates 2419 multiple arrhythmias to 140966 corresponding ECG descriptions.

Detailed description of the model, generated knowledge-base and its evaluation can be found in [Mozetic et al. 84].

The ECG knowledge-base generated by the model is supposed to be complete in two ways. First, it comprises all physiologically possible multiple arrhythmias. And second, each arrhythmia is associated with all possible ECG descriptions. In principle, the problem of diagnosing is simple now. As the rules in the knowledge-base are logical implications we can apply modus tollens rule of inference on them:

$$\neg \mathrm{ECG} \implies \neg \mathrm{ARR}$$

If a given ECG description does not match any ECG in the rule than the arrhythmia is rejected. All arrhythmias that cannot be rejected are possible regarding the ECG data. Any further discrimination between the so obtained set of possible arrhythmias can be done only on the basis of some other evidences (e.g. clinical data). Besides that, as the knowledge-base is complete, the empty set of possible arrhythmias would imply that a given ECG description is physiologically impossible.

The only problem is that the knowledge-base is too big and even if it fits in the memory, the diagnosing would be very inefficient. But as the knowledge-base is exhaustive we can try some transformations on it to obtain more simple diagnostic rules. The idea is to regard the knowledge-base as a set of events that characterize different classes, e.g. arrhythmias. By applying inductive learning methods we could possibly obtain more general and compact descriptions of arrhythmias by means of ECG features. If induced descriptions are sufficient for diagnosing they could be regarded as (relatively simple) diagnostic rules.

1.2. Transformations of the ECG knowledge-base

The ECG knowledge-base can be regarded into two ways. The first, already mentioned, is as a set of rules of the form:

$$ARR_1 \;=>\; Ecg_1 \vee Ecg_2 \vee Ecg_4 \vee \ldots$$
$$ARR_2 \;=>\; Ecg_2 \vee Ecg_3 \vee \ldots$$

Note that different arrhythmias may imply equal ECG descriptions. As the knowledge-base is complete it contains all physiologically possible ECG descriptions. Therefore we may invert it and collect all equal ECG descriptions together. With each ECG description we may associate a disjunction of all arrhythmias that can cause it, to obtain rules of the form:

$$Ecg_1 \;=>\; ARR_1$$
$$Ecg_2 \;=>\; ARR_1 \vee ARR_2 \vee \ldots$$
$$Ecg_3 \;=>\; ARR_2 \vee \ldots$$

Each multiple arrhythmia ARR can also be uniquely represented by a set of attributes and their values. So, we have similar situation in both cases. A set of objects (arrhythmias and ECG descriptions), where each object has associated a set of events, defined by attribute-value vectors (implied ECG descriptions and possible arrhythmias, respectively). Each object itself is defined in terms of an attribute-value vector. However, in both cases the number of different objects is very large, few 1000. Therefore, taking single objects as classes and applying an inductive learning program to obtain (more compact) descriptions of implied events would not lead to great improvement. If we want to achieve a substantial compression of the knowledge-base we must define classes comprising of sets of objects, instead of single objects.

The most natural way to divide objects into classes is according to their attribute-value definitions. Taking into account only one attribute and ignoring others, all objects may be divided into classes that correspond to different values of the selected attribute. If this is done for each constituent attribute then the total number of classes is relatively small, i.e., the total number of values of all objects' attributes.

Now, let us consider the general case, where we have rules of the form:

$$[Attr_1{=}Val1_1]\ \&\ [Attr_2{=}Val2_1]\ \&\ \ldots \Rightarrow Event_1 \vee Event_2 \vee \ldots \equiv Events1_1 2_1 \ldots N_1$$
$$[Attr_1{=}Val1_1]\ \&\ [Attr_2{=}Val2_2]\ \&\ \ldots \Rightarrow Event_1 \vee Event_3 \vee \ldots \equiv Events1_1 2_2 \ldots N_1$$
$$[Attr_1{=}Val1_2]\ \&\ [Attr_2{=}Val2_1]\ \&\ \ldots \Rightarrow Event_3 \vee Event_4 \vee \ldots \equiv Events1_2 2_1 \ldots N_1$$
$$[Attr_1{=}Val1_2]\ \&\ [Attr_2{=}Val2_2]\ \&\ \ldots \Rightarrow Event_4 \vee Event_5 \vee \ldots \equiv Events1_2 2_2 \ldots N_1$$

The left hand-side of a rule represents an object (e.g. an arrhythmia or an ECG description), and the right hand-side represents a disjunctive set of corresponding events (e.g. produced ECG descriptions or possible arrhythmias, respectively). Collecting together all events which correspond to a particular attribute-value pair would give us rules of the form:

$$[Attr_1{=}Val1_1] \Rightarrow Event_1 \vee Event_2 \vee Event_3 \vee \ldots \equiv Events1_1$$
$$[Attr_1{=}Val1_2] \Rightarrow Event_3 \vee Event_4 \vee Event_5 \vee \ldots \equiv Events1_2$$
$$[Attr_2{=}Val2_1] \Rightarrow Event_1 \vee Event_2 \vee Event_3 \vee Event_4 \vee \ldots \equiv Events2_1$$
$$[Attr_2{=}Val2_2] \Rightarrow Event_1 \vee Event_3 \vee Event_4 \vee Event_5 \vee \ldots \equiv Events2_2$$

However, these rules are not equivalent to the original ones. Following the transformation step by step, we show what additional conditions need to be satisfied to preserve equivalence and what information is lost.

A set of events for a particular attribute-value pair is obtained as an union of all events associated to objects that satisfy the attribute-value condition. E.g., for two attributes we have:

$$Events1_1 \Leftrightarrow Events1_1 2_1 \vee Events1_1 2_{i \neq 1}$$
$$Events2_1 \Leftrightarrow Events1_1 2_1 \vee Events1_{j \neq 1} 2_1$$

The intersection of these two sets of events would give us the original set of events plus some additional events. Therefore, if we replace the original set of an object's events with the intersection of events associated to its constituent attribute-values, we perform truth-preserving transformation, but introduce some ambiguity:

$$\mathrm{Events1}_1 \ \& \ \mathrm{Events2}_1 \iff \mathrm{Events1}_1 2_1 \vee (\mathrm{Events1}_1 2_{i \neq 1} \ \& \ \mathrm{Events1}_{j \neq 1} 2_1)$$

$$\mathrm{Events1}_1 2_1 \Rightarrow \mathrm{Events1}_1 \ \& \ \mathrm{Events2}_1$$

In general case, when objects are defined with more attributes, the ambiguity introduced is normally small. There should exist two objects that are defined with different values in at least two attributes and they must imply some equal events. Further, there must be another, different object, defined with a combination of the former values of the two attributes which does not imply any of the equal events from above. After the replacement of the original events, the equal events are added to the right hand-side of the rule with the latter object at the left hand-side. If there is no such a case among the original rules there is no ambiguity introduced and the implication below could be changed into the equivalence:

$$\mathrm{Events1}_1 2_1 \ldots \mathrm{N}_1 \Rightarrow \mathrm{Events1}_1 \ \& \ \mathrm{Events2}_1 \ \& \ \ldots \ \& \ \mathrm{EventsN}_1$$

By replacing the original sets of events with the intersections of events associated to objects' constituent attribute-values, we get the rules of the form:

$$[\mathrm{Attr}_1 = \mathrm{Val1}_1] \ \& \ [\mathrm{Attr}_2 = \mathrm{Val2}_1] \ \& \ \ldots \Rightarrow \mathrm{Events1}_1 \ \& \ \mathrm{Events2}_1 \ \& \ \ldots$$

$$[\mathrm{Attr}_1 = \mathrm{Val1}_1] \ \& \ [\mathrm{Attr}_2 = \mathrm{Val2}_2] \ \& \ \ldots \Rightarrow \mathrm{Events1}_1 \ \& \ \mathrm{Events2}_2 \ \& \ \ldots$$

$$[\mathrm{Attr}_1 = \mathrm{Val1}_2] \ \& \ [\mathrm{Attr}_2 = \mathrm{Val2}_1] \ \& \ \ldots \Rightarrow \mathrm{Events1}_2 \ \& \ \mathrm{Events2}_1 \ \& \ \ldots$$

$$[\mathrm{Attr}_1 = \mathrm{Val1}_2] \ \& \ [\mathrm{Attr}_2 = \mathrm{Val2}_2] \ \& \ \ldots \Rightarrow \mathrm{Events1}_2 \ \& \ \mathrm{Events2}_2 \ \& \ \ldots$$

From here on, we use just basic reformulaton rules of propositional logic. First, we take into account the distribution law:

$$(A \Rightarrow B \ \& \ C) \iff (A \Rightarrow B) \ \& \ (A \Rightarrow C)$$

that gives us:

$[Attr_1=Val1_1]$ & $[Attr_2=Val2_1]$ & ... => $Events1_1$

$[Attr_1=Val1_1]$ & $[Attr_2=Val2_1]$ & ... => $Events2_1$

$[Attr_1=Val1_1]$ & $[Attr_2=Val2_2]$ & ... => $Events1_1$

$[Attr_1=Val1_1]$ & $[Attr_2=Val2_2]$ & ... => $Events2_2$

$[Attr_1=Val1_2]$ & $[Attr_2=Val2_1]$ & ... => $Events1_2$

$[Attr_1=Val1_2]$ & $[Attr_2=Val2_1]$ & ... => $Events2_1$

$[Attr_1=Val1_2]$ & $[Attr_2=Val2_2]$ & ... => $Events1_2$

$[Attr_1=Val1_2]$ & $[Attr_2=Val2_2]$ & ... => $Events2_2$

Collecting together rules which have equal right hand-sides, by the law:

(A => C) & (B => C) <=> (A v B => C)

and exposing a primary attribute-value factor, gives us the following rules:

$[Attr_1=Val1_1]$ & $[Attr_2=Val2_1 \vee Val2_2 \vee ...]$ & ... => $Events1_1$

$[Attr_1=Val1_2]$ & $[Attr_2=Val2_1 \vee Val2_2 \vee ...]$ & ... => $Events1_2$

$[Attr_1=Val1_1 \vee Val1_2 \vee ...]$ & $[Attr_2=Val2_1]$ & ... => $Events2_1$

$[Attr_1=Val1_1 \vee Val1_2 \vee ...]$ & $[Attr_2=Val2_2]$ & ... => $Events2_2$

Here, left hand-side of a rule is a conjunctive-disjunctive expression and, in general, cannot be simplified into an expression that contains only internal disjunctions. With further applications of exposition, the expression can be eventually transformed into a tree structure, with the primary attribute-value as a root. Still, such structures can be rather complex and if we want to achieve a substantial compression of the rules we may simply omit all conditions but the primary attribute-value:

$[Attr_1=Val1_1]$ => $Events1_1$

$[Attr_1=Val1_2]$ => $Events1_2$

$[Attr_2=Val2_1]$ => $Events2_1$

$[Attr_2=Val2_2]$ => $Events2_2$

When the rules are used by an inference mechanism this simplification results in an unability to reject "impossible" objects, i.e. objects not occuring in the original knowledge-base. Besides that, when the rules are used in the backward direction, for a given event we may not be able to reject all objects that could be rejected using the original knowledge-base. If the rules are supposed to be used only in the forward direction, and if we have means to distinguish between "possible" and "impossible" objects, no usefull information is lost after this simplification.

Finally, the rules are in an appropriate form for the application of an inductive learning method. For each attribute, we regard its different values as different classes and take corresponding events as positive examples of the class. All other events represent negative examples (with exception of the intersecting (common) events of different classes). A sound inductive learning algorithm then generalizes positive examples as much as possible, without covering any negative example. Formally, induced descriptions must satisfy the *completness* and *consistency* conditons [Michalski 83]. Every positive example of some class must satisfy the induced description of the class:

$$\mathrm{Events}_i \;=>\; \mathrm{Desc}_i$$

and no event, that satisfies a description of some class can be a negative example of that class:

$$\mathrm{Desc}_i \;=>\; \neg\,\mathrm{Events}_j, \quad i \neq j$$

For each attribute, we divide the set of all possible events into sets corresponding to different values of the attribute. Inductive learning algorithm then iteratively considers events of each individual value as positive examples of the class and all others events as negative examples. Replacing the sets of events by corresponding induced descriptions, we finally obtain rules that are hopefully significantly simpler than the original ones:

$$[\mathrm{Attr}_1 = \mathrm{Val1}_1] \;=>\; \mathrm{Desc1}_1$$
$$[\mathrm{Attr}_1 = \mathrm{Val1}_2] \;=>\; \mathrm{Desc1}_2$$
$$[\mathrm{Attr}_2 = \mathrm{Val2}_1] \;=>\; \mathrm{Desc2}_1$$

$[Attr_2=Val2_2] \;=>\; Desc2_2$

The consistency condition guarantees that no "wrong" conclusions can be derived from the transformed rules. However, the generalization over events introduces another loss of information. Using only the induced descriptions, it is possible to construct "impossible" events, i.e., events that are not implied by any object in the original knowledge-base.

Summarizing, the proposed transformation of the knowledge-base has the following effects on the information content:

- The right hand-sides of the rules are weakened by adding some other events to the ones occuring in the original conclusions.
- The left hand-sides of the rules are weakened by dropping all conditions but the primary attribute-value factor.
- The generalization of the right hand-sides of the rules replaces events by their simplified descriptions, so covering some empty space in the attribute space of events.

The goal of the transformation is to compress the knowledge-base, so that it can be used for inference. From the transformed rules we can infer conclusions equivalent to those derived from the original rules if the following conditions are satisfied:

- There are no implied events in common between any two objects, which are defined with different values for at least two attributes. If there are two such objects and there is also another object defined with a combination of those values, the latter should also imply all events that are in common between the former two objects.
- The transformed rules are used only in the forward direction and are not supposed to reject objects that are not in the original knowledge-base.
- We are able to recognize "impossible" events, i.e., events not occurring in the original knowledge-base.

1.3. Summary of results

For compression of the ECG knowledge-base we used two programs that implement the basic AQ inductive learning algorithm [Michalski 83]. However, the original knowledge-base is too complex to be used by any implementation of AQ algorithm. Therefore we first generated a sufficient subset of the knowledge-base that retains enough information which enables us to deduce all interesting facts from it (Figure 1). The subset in its original form was further divided into two knowledge-bases (for regular and ectopic arrhythmias), regarding the number of attributes that are needed to describe corresponding ECG descriptions. The regular arrhythmias knowledge-base provided sets of input examples for the GEM inductive learning program. The program induced descriptions of regular arrhythmias that were in most cases satisfactory in terms of their complexity. When the same approach was applied on the ectopic arrhythmias knowledge-base, their descriptions tended to be considerably more complicated. To improve them, we used program VARSEL that selects the most representative attributes, and, when this did not help, we introduced some new attributes.

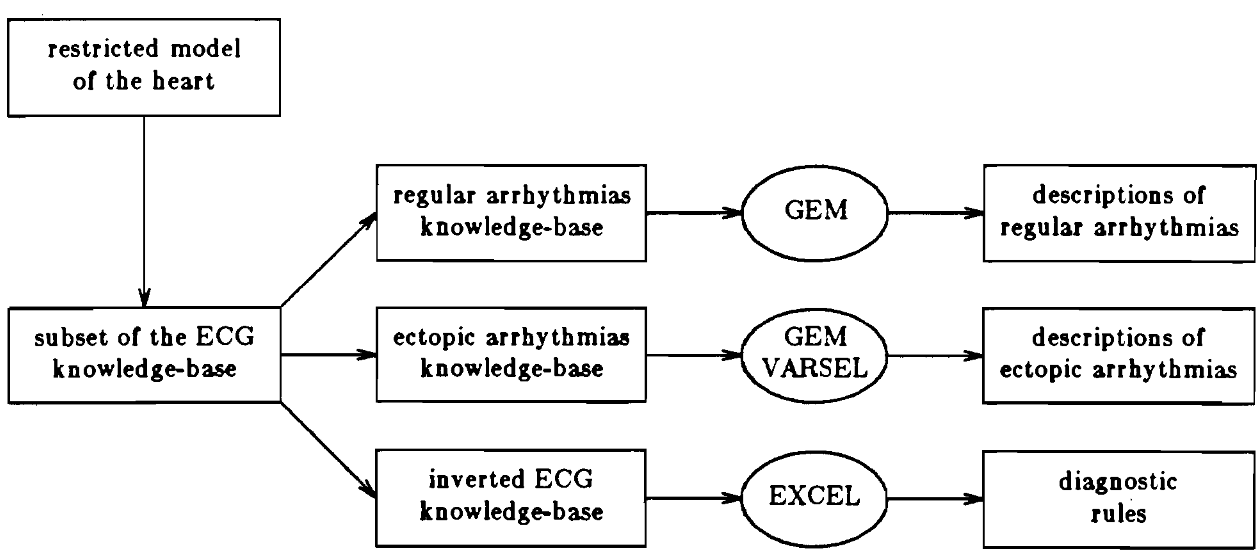

Figure 1: An overview of the experiments described in the paper.

The induced descriptions of regular and ectopic arrhythmias are interesting from medical point of view, but they are not sufficient for diagnosing. Therefore, we inverted the generated subset of the ECG knowledge-base and applied induction over definitions of combined arrhythmias. Learning program EXCEL was used to induce descriptions of each individual ECG feature in terms of possible arrhythmias. Resulting diagnostic rules are relatively complicated (from medical point of view), but compact enough to be efficiently used for diagnosig.

As already mentioned, the ECG knowledge-base in its original form (a multiple arrhythmia implying all possible ECG descriptions) was used to induce *descriptions of arrhythmias*, regardless if they occur alone or in a combination. Formally, for each attribute and its value (used to define a multiple arrhythmia), we considered two cases. Either the attribute takes the selected value (the arrhythmia occurs in all combinations), or it does not (the arrhythmia does not occur in any combination):

$$[\mathrm{ARRattr}=\mathrm{Arr}_1] \iff \mathrm{ARRcomb}_1 \iff \{\mathrm{Arr}_1\} \vee \{\mathrm{Arr}_1, \mathrm{Arr}_2\} \vee \ldots$$
$$[\mathrm{ARRattr}\neq\mathrm{Arr}_1] \iff \mathrm{ARRnot}_1 \iff \{\mathrm{Arr}_2\} \vee \{\mathrm{Arr}_2, \mathrm{Arr}_3, \mathrm{Arr}_4\} \vee \ldots$$

For the inductive learning program the two cases represent two classes and corresponding ECG descriptions are used as positive and negative examples. As a result we get both, positive and negative description for each arrhythmia, regarding its occurrence or absence in a combination, respectively:

$$\mathrm{ARRcomb}_1 \Rightarrow \mathrm{Ecg}_1 \vee \mathrm{Ecg}_2 \vee \ldots \Rightarrow \mathrm{POSdesc}_1$$
$$\mathrm{ARRnot}_1 \Rightarrow \mathrm{Ecg}_3 \vee \mathrm{Ecg}_4 \vee \ldots \Rightarrow \mathrm{NEGdesc}_1$$

For diagnosing, the induced descriptions should be used in the backward direction, which does not meet criteria for appropriate use of transformed rules, mentioned at the end of the previous section.

New rules can be used only as conditions that must be satisfied by any ECG description for a multiple arrhythmia. This gives us an answer to a slightly reformulated question of the combination function. Recall that one of the motivations for developing the model of the heart was inability to find a simple function for computing an ECG description of a multiple arrhythmia, given ECG descriptions of its constituent arrhythmias. Now, instead of dealing with simple arrhythmias alone, we deal with simple arrhythmias occuring alone or combined. A pair of such arrhythmias combined, say $ARRcomb_1$ and $ARRcomb_2$, is a set of all combinations containing the two arrhythmias, Arr_1 and Arr_2 :

$$\frac{\begin{array}{l} ARRcomb_1 \iff \{Arr_1\} \vee \{Arr_1, Arr_2\} \vee \{Arr_1, Arr_2, Arr_3\} \vee \ldots \\ ARRcomb_2 \iff \{Arr_2\} \vee \{Arr_1, Arr_2\} \vee \{Arr_2, Arr_3\} \vee \ldots \end{array}}{\begin{array}{l} ARRcomb_{1,2} \iff ARRcomb_1 \,\&\, ARRcomb_2 \\ \qquad\qquad\quad \iff \{Arr_1, Arr_2\} \vee \{Arr_1, Arr_2, Arr_3\} \vee \ldots \end{array}}$$

Given induced descriptions of simple arrhythmias combined, $ARRcomb_1$ and $ARRcomb_2$, we can easily construct description of their combination, $ARRcomb_{1,2}$:

$$\frac{\begin{array}{l} ARRcomb_1 \implies POSdesc_1 \\ ARRcomb_2 \implies POSdesc_2 \end{array}}{ARRcomb_{1,2} \implies POSdesc_1 \,\&\, POSdesc_2}$$

Induction of necessary conditions for each simple arrhythmia to occur alone or combined, enables us to construct necessary conditions for arrhythmias combined. Combinatorial function is simple - logical conjunction. However, descriptions of necessary conditions are not so simple that they could be acquired manually from medical literature. It is obvious that construction of complete ECG descriptions for multiple arrhythmias is even more complicated, what justifies the need for deeper, causal knowledge, i.e., model of the heart.

To produce usefull *diagnostic rules* we had to invert the ECG knowledge-base, so that each possible ECG description has associated a set of arrhythmias that can cause it. For each individual ECG feature (attribute-value pair) we induced a description in terms of possible arrhythmias, using EXCEL learning program:

$$[ECGattr_1{=}Val1_1] \implies ARRdesc1_1$$
$$[ECGattr_1{=}Val1_2] \implies ARRdesc1_2$$

$[ECGattr_2=Val2_1] \Rightarrow ARRdesc2_1$

$[ECGattr_2=Val2_2] \Rightarrow ARRdesc2_2$

The so obtained rules can be used for diagnosing, if we do not need to reject impossible ECG descriptions. An input ECG description is given as a conjunction of ECG features:

$$Ecg \Leftrightarrow [ECGattr_1=Val1_2] \; \& \; [ECGattr_2=Val2_2] \; \& \; ...$$

and application of diagnostic rules gives us an expression that must be satisfied by all possible arrhythmias:

$$Ecg \Rightarrow ARRdesc1_2 \; \& \; ARRdesc2_2 \; \& \; ...$$

Multiple arrhythmias that are not occuring in the original knowledge-base but still satisfy the expression are only physiologically impossible arrhythmias. They may be eliminated by using the same physiological constraints as are used by the model of the heart.

2. Subset of the ECG knowledge-base

As it was already mentioned a qualitative model of the heart was used to generate the complete knowledge-base for ECG diagnosis of cardiac arrhythmias. The knowledge-base relates all physiologically possible combinations of 30 simple arrhythmias to all ECG descriptions produced by these disorders. It consists of 2419 multiple arrhythmias and 140966 corresponding ECG descriptions.

All these ECG descriptions should be used as positive and negative examples characterizing various classes. However, such number of examples is far too large for all implementations of the AQ inductive learning algorithm. We had to reduce the number of events somehow, without affecting completness of the knowledge-base. Besides that, the language for ECG description varies in the number of attributes that are used to describe various arrhythmias. There is seven "regular" attributes, used to describe regular arrhythmias and additional triples of "ectopic" attributes, each for one type of ectopic arrhythmia.

Below we give a Prolog clause (stretching down several lines) that defines *regular attributes*, their types, and possible values. Values on the left hand side of a colon are used in the ECG knowledge-base, and on the right hand side in some experiments with the AQ algorithm:

```
regular_dict( [
   rhythm : nominal :
              [ regular : regular,
                irregular : irregular ],

   regular_P : nominal :
              [ normal : normal,
                abnormal : abnormal,
                changing : changing,
                absent : absent ],

   rate_of_P : linear :
              [ zero : zero,
                under_60 : under60,
                between_60_100 : in60to100,
                between_100_250 : in100to250,
                between_250_350 : in250to350,
                over_350 : over350 ],

   relation_P_QRS : nominal :
              [ meaningless : notApp,
```

```
              after_P_always_QRS : alwaysQRS,
              after_P_some_QRS_miss : missingQRS,
              independent_P_QRS : independ ],

   regular_PR : nominal :
             [ meaningless : notApp,
              normal : normal,
              prolonged : prolonged,
              shortened : shortened,
              changing : changing,
              after_QRS_is_P : aftQRSisP ],

   regular_QRS : nominal :
             [ normal : normal,
              wide_LBBB : wLBBB,
              wide_RBBB : wRBBB,
              wide_LBBB_RBBB : wLBBBaRBBB,
              delta_LBBB : deltaLBBB,
              delta_RBBB : deltaRBBB,
              absent : absent ],

   rate_of_QRS : linear :
             [ under_60 : under60,
              between_60_100 : in60to100,
              between_100_250 : in100to250,
              between_250_350 : in250to350,
              over_350 : over350 ]         ]).
```

An ECG description for each type of an ectopic arrhythmia consists of the three *ectopic attributes* below. As several ectopic arrhythmias may occur in a combination at the same time, each attribute is indexed by "I" to connect them in triples. In experiments with the AQ algorithm the index is omitted and the attributes are abbreviated to "ect_P", "ect_PR", and "ect_QRS" respectively.

```
ectopic_dict( I, [
   ectopic_P(I) : nominal :
             [ abnormal : abnormal,
              absent : absent ],

   ectopic_PR(I) : nominal :
             [ meaningless : notApp,
              normal : normal,
              prolonged : prolonged,
              shortened : shortened,
              after_QRS_is_P : aftQRSisP ],

   ectopic_QRS(I) : nominal :
             [ normal : normal,
              wide_LBBB : wLBBB,
```

```
wide_RBBB : wRBBB,
wide_LBBB_RBBB : wLBBBaRBBB,
delta_LBBB : deltaLBBB,
delta_RBBB : deltaRBBB,
absent : absent ]          ]).
```

Below we give a list of all simple arrhythmias that are known to the KARDIO expert system. Abbreviations are used in some examples and in the appendices. The following Prolog clauses define 26 *regular arrhythmias* :

```
arrhythmias( sr..sinus_rhythm ).
arrhythmias( sa..sinus_arrhythmia ).
arrhythmias( sb..sinus_bradycardia ).
arrhythmias( st..sinus_tachycardia ).
arrhythmias( sad..sa_node_disorders ).
arrhythmias( wp..wandering_pacemaker ).
arrhythmias( at..atrial_tachycardia ).
arrhythmias( mat..multi_atrial_tachycardia ).
arrhythmias( afl..atrial_flutter ).
arrhythmias( af..atrial_fibrillation ).
arrhythmias( avb1..av_block_1 ).
arrhythmias( wen..wenckebach ).
arrhythmias( mob2..mobitz_2 ).
arrhythmias( avb3..av_block_3 ).
arrhythmias( wpw..wpw_syndrome ).
arrhythmias( lgl..lgl_syndrome ).
arrhythmias( jb..junctional_bradycardia ).
arrhythmias( jr..junctional_rhythm ).
arrhythmias( jt..junctional_tachycardia ).
arrhythmias( lbbb..left_bundle_branch_block ).
arrhythmias( rbbb..right_bundle_branch_block ).
arrhythmias( vr..ventricular_rhythm ).
arrhythmias( avr..accelerated_ventricular_rhythm ).
arrhythmias( vt..ventricular_tachycardia ).
arrhythmias( vfl..ventricular_flutter ).
arrhythmias( vf..ventricular_fibrillation ).
```

And here are the remaining 4 *ectopic arrhythmias* :

```
arrhythmias( aeb..atrial_ectopic_beats ).
arrhythmias( jeb..junctional_ectopic_beats ).
arrhythmias( veb..ventricular_ectopic_beats ).
arrhythmias( mveb..multi_ventricular_ectopic_beats ).
```

All arrhythmias can be divided into seven groups, where in each group arrhythmias are mutually exclusive. Names of the groups are used as attributes and simple arrhythmias as their values. The following Prolog clause defines those seven attributes and corresponding values:

```
groups( [
      sinus : [none, sr, sa, sb, st, sad],
      atrial : [none, wp, at, mat, afl, af, aeb],
      av_cond : [none, avb1, wen, mob2, avb3, wpw, lgl],
      junction : [none, jb, jr, jt, jeb],
      bb_cond : [none, lbbb, rbbb],
      ventric : [none, vr, avr, vt, vfl, vf],
      ectvent : [none, veb, mveb]).
```

As an example we show how a multiple arrhythmia "sinus_bradycardia with wenckebach and ventricular_ectopic_beats" can be transformed into a vector of attributes and values:

```
{sinus_bradycardia, wenckebach, ventricular_ectopic_beats} <=>
      [sinus = sb] &
      [atrial = none] &
      [av_cond = wen] &
      [junction = none] &
      [bb_cond = none] &
      [ventric = none] &
      [ectvent = veb]
```

Fortunately, there are some simple relations between arrhythmias which allow us to generate substantially smaller subset of the ECG knowledge-base that is still sufficient for diagnosing. Namely, regular arrhythmias are independent of the ectopic ones in the sense that their ECGs are not affected with the presence or absence of any ectopic arrhythmia. In other words, the regular part of an ECG description (consisting of 7 regular attributes) does not change if an ectopic arrhythmia is present in a given multiple arrhythmia or not. The opposite is not true, as regular arrhythmias do affect ECG descriptions of ectopic arrhythmias (triples of ectopic attributes). The same independency holds between ectopic arrhythmias, i.e., they do not affect ectopic part of an ECG description of each other.

These facts provide justification for the generation of two subsets of the ECG knowledge-base. The first one contains only combinations of regular arrhythmias and corresponding ECG descriptions with only 7 regular attributes. The induced descriptions should be the same as descriptions of regular arrhythmias induced from the original knowledge-base. The second subset comprises all combinations of regular arrhythmias with exactly one ectopic arrhythmia in each combination. In this case the ECG language consists of 10 attributes: 7 regular and 3 ectopic. This subset should be sufficent to induce descriptions of ectopic arrhythmias.

When the actual subsets of the knowledge-base were generated we took into account some additional facts. For the generation of the first subset we selected 24 regular arrhythmias. Following arrhythmias were eliminated:

- "aeb", "jeb", "veb", "mveb" - all ectopic arrhythmias,
- "sa", as it has identical ECG description to "sad",
- "rbbb" which has symmetrical ECG description to "lbbb".

The generated subset of the ECG knowledge-base relates 175 combinations of 24 regular arrhythmias to 333 ECG descriptions. All constraints regarding possible combinations of arrhythmias remain the same as in the original program. Between 333 ECG descriptions there is 263 different events. This means that some ECG descriptions are equal for different arrhythmias. Below we give a detailed account on the number of generated ECG descriptions and events that are equal for 1, 2, 3 or 4 different arrhythmias:

equal arr.		events		ECG desc.
1	×	207	=	207
2	×	48	=	96
3	×	2	=	6
4	×	6	=	24
Σ		263		333

For the generation of the second subset of the knowledge-base we selected 27 arrhythmias; 24 regular and 3 ectopic. We have omitted:

- "sa" and "rbbb" for the same reasons as above,
- "mveb", as its ECG description can be deduced from the description of "veb".

The second generated subset of the knowledge-base relates 411 combinations of arrhythmias with 2072

ECG descriptions. Here, all ECG descriptions contain exactly 10 attributes (7 regular and 3 ectopic). Details about the number of equal events for different arrhythmias follows:

equal arr.		events		ECG desc.
1	×	1 043	=	1 043
2	×	369	=	738
3	×	47	=	141
4	×	26	=	104
5	×	2	=	10
6	×	6	=	36
Σ		1 493		2 072

Note, that this is not a complete set of combinations of all 27 arrhythmias. Rather, it is the first generated subset combined with the additional 3 ectopic arrhythmias: "aeb", "jeb", "veb".

It is somehow surprising that those small subsets of the complete ECG knowledge-base are (almost) sufficient for induction of all diagnostic rules. These experiments showed that the main reason for the high number of possible combinations of arrhythmias is primarily due to the unconstrained combinatorial nature of ectopic arrhythmias. Below we give a comparison between sizes of the original ECG knowledge-base and the generated subset. Each rule relates a multiple arrhythmia to a set of ECG descriptions, which are represented by several conjunctions (Prolog clauses).

total number of	original ECG knowledge-base	subset of the ECG knowledge-base
rules	2 419	586
conjunctions	8 314	957
ECG desc.	140 966	2 405

From here on we will refer to the first subset of the knowledge-base as the regular arrhythmias knowledge-base and to the second as the ectopic arrhythmias knowledge-base. Details of how to use the induced descriptions for description of omitted arrhythmias and even some new ones are given in the last part of the paper.

3. Induction of descriptions for arrhythmias combined

For the first experiment in compression, the ECG knowledge-base was used in its original form. A multiple arrhythmia on the left hand-side of a rule and a set of corresponding ECG descriptions on the right hand-side. Each multiple arrhythmia can be expressed with an attribute-values vector, where an individual attribute-value pair represents a simple arrhythmia combined, i.e., occuring alone or in a combination. The knowledge-base was first transformed so that on the left hand-sides of the rules only single attribute-value pairs occured. Then an inductive learning algorithm was applied over ECG descriptions on the right hand-sides of the rules. The result were simplified descriptions of arrhythmias combined, sufficient for discrimination between arrhythmias whenever this is possible. Because of efficiency, the experiment was run twice, for regular and ectopic arrhythmias separately.

3.1. Descriptions of regular arrhythmias

The inductive learning program GEM that was used for these experiments was not capable of dealing with intersecting classes. Therefore we divided the 263 events from the regular arrhythmias knowledge-base into three classes for each of 24 regular arrhythmias. Recall that for a given arrhythmia Arr, ARRcomb denotes all multiple arrhythmias containing Arr, and ARRnot all combinations without Arr. As some ECG descriptions may belong to arrhythmias from both sets we separated the intersection region into the third class. We named the three classes of events POS, INTER and NEG:

- class POS contains events from: ARRcomb – ARRnot,
- class INTER contains events from: ARRcomb ∩ ARRnot,
- class NEG contains events from: ARRnot – ARRcomb.

Let us consider an example. Given are three arrhythmias: "atrial_tachycardia", "atrial_tachycardia with lgl_syndrom", and "junctional_tachycardia" with their corresponding ECG descriptions:

{atrial_tachycardia} =>
[rhythm = regular] &
[regular_P = abnormal] &
[rate_of_P = between_100_250] &
[relation_P_QRS = after_P_always_QRS] &
[regular_PR = shortened, normal, prolonged] &
[regular_QRS = normal] &
[rate_of_QRS = between_100_250]

{atrial_tachycardia, lgl_syndrome} =>
[rhythm = regular] &
[regular_P = abnormal] &
[rate_of_P = between_100_250] &
[relation_P_QRS = after_P_always_QRS] &
[regular_PR = shortened] &
[regular_QRS = normal] &
[rate_of_QRS = between_100_250]

{junctional_tachycardia} =>
[rhythm = regular] &
[regular_P = abnormal] &
[rate_of_P = between_100_250] &
[relation_P_QRS = after_P_always_QRS] &
[regular_PR = shortened, after_QRS_is_P] &
[regular_QRS = normal] &
[rate_of_QRS = between_100_250]

Note that the first ECG description of all three arrhythmias is the same. If ECG descriptions are treated as events, we can divide them into three classes for each simple arrhythmia, in this case e.g. "atrial_tachycardia". As the simple arrhythmia "atrial_tachycardia" occurs in the first two multiple arrhythmias but does not in the last one, the event that is equal for all three arrhythmias belongs to the intersection class INTER. Below are events of the classes POSat, INTERat and NEGat. They are written in the form of relational tables that are required as the input data format for the inductive learning program used.

POSat-events

#	rhythm	regular_P	rate_of_P	relation_P_QRS	regular_PR	regular_QRS	rate_of_QRS
1	regular	abnormal	in100to250	alwaysQRS	normal	normal	in100to250
2	regular	abnormal	in100to250	alwaysQRS	prolonged	normal	in100to250

INTERat-events

#	rhythm	regular_P	rate_of_P	relation_P_QRS	regular_PR	regular_QRS	rate_of_QRS
1	regular	abnormal	in100to250	alwaysQRS	shortened	normal	in100to250

NEGat-events

#	rhythm	regular_P	rate_of_P	relation_P_QRS	regular_PR	regular_QRS	rate_of_QRS
1	regular	abnormal	in100to250	alwaysQRS	aftQRSisP	normal	in100to250

For each regular arrhythmia the descriptions for all three classes were induced. We used the GEM1.0 program [Reinke 84] which is a Pascal implementation of the AQ algorithm. Discriminant descriptions (the shortest ones) were produced in intersecting covers mode (descriptions may intersect over space where there are no learning events) and in valued logic mode (descriptions are order dependent). Characteristic descriptions (the longest ones) were produced only in intersecting covers mode.

In the Appendix A we give only discriminant descriptions produced in intersecting covers mode. Descriptions in valued logic mode are shorter, but they are less comprehensive because of their order dependancy. It turned out that characteristic descriptions are not the most specific ones as it was expected, therefore they are of no particular interest. Here we give an example of induced discriminant descriptions (intersecting covers mode) for "atrial_tachycardia":

```
POSat =>
    [regular_P=abnormal] & [rate_of_P=in100to250] & [regular_PR=notApp,normal,prolonged] v
    [regular_P=abnormal] & [regular_PR=normal,prolonged,shortened]
    & [regular_QRS=wLBBBaRBBB,deltaLBBB,deltaRBBB]

INTERat =>
    [regular_P=abnormal] & [rate_of_P=in100to250] & [regular_PR=shortened]
    & [regular_QRS=normal,wLBBB]

NEGat =>
    [regular_P=normal,changing,absent] v
    [rate_of_P=in250to350..over350] v
    [rate_of_P=zero..in60to100] v
    [regular_PR=changing,aftQRSisP]
```

We divided all regular arrhythmias into three categories, with regard to the complexity of the produced descriptions. We took into account only classes POS and INTER (discriminant descriptions, intersecting covers mode) as the class NEG does not positively characterize an arrhythmia. Number of events in each class and a triple of the form Conj-Attr-Val are associated with each arrhythmia. The triple contains the cummulative information for both classes, POS and INTER:

- Conj is the total number of conjunctions,
- Attr is the total number of attributes,
- Val is the total number of values.

Arr	POS	INTER	NEG	Conj-Attr-Val	Complexity
st	25	0	238	1-2-2	SIMPLE
wp	27	0	236	1-2-2	
mat	27	0	236	1-2-2	
afl	23	0	240	1-1-1	
wen	22	0	241	1-2-2	
mob2	28	0	235	1-2-2	
avb3	81	0	182	1-2-2	
vfl	2	0	261	1-2-3	
vf	0	1	262	1-2-2	
sr	12	13	238	2-8-12	MODERATE
sb	10	11	242	2-7-11	
sad	22	24	217	3-10-14	
at	23	2	238	3-10-17	
af	43	1	219	4-8-10	
avb1	14	2	247	2-6-7	
wpw	19	1	243	2-4-8	
lgl	14	2	247	2-7-10	
vr	11	11	241	2-7-10	
avr	11	11	241	2-7-10	
vt	11	11	241	2-7-10	
jb	13	11	239	4-17-29	COMPLICATED
jr	13	11	239	4-17-26	
jt	11	13	239	3-14-23	
lbbb	71	33	159	3-11-22	
Σ				49-157-235	

In the table above it is interesting to notice the correlation between the number of events in the intersecting class INTER and the complexity of descriptions. It makes sense to say that arrhythmias that are difficult to distinguish from others, have more complicated descriptions. From the medical point of view it is known that junctional arrhythmias (jb, jr, jt) are sometimes indistinguishable from atrial arrhythmias, and when combined with bundle branch block (lbbb) also from ventricular arrhythmias. As it is evident from the table, this four arrhythmias have the most complicated descriptions.

Another point of interest is to compare the complexity of the regular arrhythmias knowledge-base with the descriptions induced. We can compare the number of conjunctions (Prolog clauses) in the knowledge-base with the total number of conjunctions in the induced descriptions. In the case of the knowledge-base the total number of attributes used is seven times the number of conjunctions, as each conjunction is described with exactly seven attributes. The total number of values used is seven times the number of ECG descriptions.

total number of	regular arrhythmias knowledge-base	descriptions of regular arrhythmias
rules	175	39
conjunctions	216	49
attributes	1 512	157
values	2 331	235

Using the inductive learning algorithm for compression of the knowledge-base we gain almost factor 5 in the number of conjunctions and factor 10 in the total number of attributes and values needed to represent the knowledge. The results would be even more favorable for the induced descriptions if we considered discriminant descriptions produced in the valued logic mode which are shorter than those in the intersecting covers mode.

3.2. Descriptions of ectopic arrhythmias

The same procedure as for regular arrhythmias was applied for the three ectopic arrhythmias. For each of them we diveded the set of 1493 events from the ectopic arrhythmias knowledge-base into three classes: POS, INTER and NEG. Again, the program GEM was used for induction of various types of descriptions for all classes. This time the induced descriptions tend to be considerably more complicated then in the case of regular arrhythmias. In the following table we use the same notation as in the previous section:

Arr	POS	INTER	NEG	Conj-Attr-Val
aeb	185	46	1 262	8-30-61
jeb	396	294	803	12-42-91
veb	618	248	627	6-19-42
Σ				26-91-42

Complete results of the induced descriptions are in the Appendix B. In the following table we compare the complexity of the ectopic arrhythmias knowledge-base with the descriptions induced. Here we gain almost factor 30 in the number of conjunctions, factor 60 in the number of attributes, and factor 75 in the number of values used to represent the knowledge. However, this comparison is not adequate as the ectopic arrhythmias knowledge-base comprises also combinations of regular arrhythmias. Results of the overall compression are discussed in the last section of the paper.

total number of	ectopic arrhythmias knowledge-base	descriptions of ectopic arrhythmias
rules	411	6
conjunctions	741	26
attributes	5 187	91
values	14 504	194

The induced descriptions were too complex to be transparent to an expert. Therefore we tried to further transform the knowledge-base before applying induction over it to obtain better descriptions. The first idea was to reduce the number of attributes in the knowledge-base as it seemed that some "regular" attributes are irrelevant for describing ectopic arrhythmias. For this purpose the attribute selection program VARSEL was used.

The program VARSEL [Baim 84] performs the task of attribute selection based on a measure of attribute relevancy for class discrimination. It evaluates each attribute individually and then compiles a subset of attributes which completely differentiates each class from the others. One of two ways in which the compilation is to proceed may be selected. The first method uses the principle of random adaptive search. Small subsets of attributes are evaluated and the individual relevancy measure of each constituent attribute is improved or degraded based on the performance of the selected subset as a whole. The second method involves a greedy search scheme in which attributes are added to a subset of attributes until a sufficiently discriminatory attribute set has been found. The attribute relevancy measure (Pvalue) is based on classical information theory and has value 0 for worst-case attributes and value 1 for perfectly discriminant attributes.

VARSEL was applied to the ectopic arrhythmias knowledge-base to select a set of the best discriminatory attributes between the original 10 ones, for each arrhythmia. The output of VARSEL was then used as the input for the GEM learning program:

VARSEL			GEM
Arr	#Attr	Pvalue	Conj-Attr-Val
aeb	8	1.0000	8-30-67
jeb	8	1.0000	12-42-92
veb	4	0.9968	5-14-27

In the first two cases VARSEL selected a subset of 8 attributes that were perfectly disciminant. In the last case it selected a subset of only 4 attributes, but they were not sufficient to discriminate between the three classes any more. This is an obvious error of the VARSEL program (probably due to rounding errors of relevancy measure) as classes POS, INTER and NEG can be disciminated by definition. Results from VARSEL were equal for both greedy and random adaptive search. The list of selected attributes for each ectopic arrhythmia follows:

ECG attribute	aeb	jeb	veb
rhythm	*	*	
regular_P	*	*	
rate_of_P	*	*	*
relation_P_QRS	*	*	
regular_PR	*	*	
regular_QRS	*	*	*
rate_of_QRS			
ectopic_P			
ectopic_PR	*	*	*
ectopic_QRS	*	*	*

Regardless of the irrelevant case of "veb" (because of the error in VARSEL), the descriptions induced by GEM were a little bit more complicated then in the first experiment. This should be expected. Elimination of some attributes means some loss of information which must be somehow compensated. Therefore, if we want to obtain more compact descriptions we must provide some additional information.

Next idea was to use regular arrhythmias as additional attributes for descriptions of ectopic arrhythmias. It was assumed that for a given ECG description, at first all possible combinations of regular arrhythmias are found, which can then be used for description of ectopic arrhythmias. In that way we can augment each ECG description from the ectopic arrhythmias knowledge-base with additional six attributes (the attribute "ectvent" may be omitted as its value is always "none"). The set of 1880 events so obtained must be divided again into three classes for each arrhythmia and than used as the input for the GEM program. GEM was used to induce the same types of descriptions as in the first case. Complete results are in the Appendix C, here we give only the summary:

Arr	POS	INTER	NEG	Conj-Attr-Val
aeb	244	44	1 592	4-13-26
jeb	552	192	1 136	7-26-54
veb	892	148	840	4-11-17
Σ				15-50-97

Comparing to the first experiment with the ectopic arrhythmias knowledge-base the number of events in the class INTER considerably decreased (note also that the total number of events is greater then in the first experiment). The induced descriptions were almost two times more simple. This supports the hypothesis about the correlation between number of events in the class INTER and the complexity of induced descriptions.

Here we give an example of discriminant descriptions in intersecting covers mode for "ventricular_ectopic_beats":

```
POSveb =>
    [ect_QRS=wLBBB,wLBBBaRBBB] & [bb_cond=none] v
    [ect_QRS=wLBBBaRBBB] v
    [ect_PR=notApp,aftQRSisP] & [ect_QRS=wLBBB,wLBBBaRBBB] & [junction=jb,jr,jt]

INTERveb =>
    [ect_PR=notApp,aftQRSisP] & [ect_QRS=wLBBB] & [junction=none] & [ventric=none]
    & [regular_QRS=wLBBB]

NEGveb =>
    [ect_QRS=normal,deltaLBBB,deltaRBBB,absent] v
    [ect_PR=normal,prolonged,shortened] v
    [atrial=none,wp,mat,afl] & [av_cond=none,wen,mob2,avb3,wpw] & [junction=none]
    & [bb_cond=lbbb] v
    [relation_P_QRS=notApp,alwaysQRS,independ]
    & [regular_PR=notApp,prolonged,shortened,aftQRSisP] v
    [av_cond=avb3,wpw] & [junction=none] & [bb_cond=lbbb]
```

The complexity of the induced descriptions is comparable to that of the complicated regular arrhythmias. Again, as in the case of regular arrhythmias, junctional arrhythmia (jeb) has the most complicated description.

4. Induction of diagnostic rules

In the second experiment we first had to invert the ECG knowledge-base and transform it to the form appropriate for induction of diagnostic rules. For each ECG description on the left hand side of an inverted rule we had to find all possible multiple arrhythmias. We expressed them as a disjunction of attribute-value vectors which then represented the right hand side of an inverted rule. As an example we can take an ECG description from section 3.1 that may be produced by any of the three arrhythmias: "atrial_tachycardia", "atrial_tachycardia with lgl_syndrome", and "junctional_tachycardia":

```
[rhythm = regular] &
[regular_P = abnormal] &
[rate_of_P = between_100_250] &
[relation_P_QRS = after_P_always_QRS] &
[regular_PR = shortened] &
[regular_QRS = normal] &
[rate_of_QRS = between_100_250]     =>

        [sinus = none] &          [sinus = none] &          [sinus = none] &
        [atrial = at] &           [atrial = at] &           [atrial = none] &
        [av_cond = none] &        [av_cond = lgl] &         [av_cond = none] &
        [junction = none] &   v   [junction = none] &   v   [junction = jt] &
        [bb_cond = none] &        [bb_cond = none] &        [bb_cond = none] &
        [ventric = none] &        [ventric = none] &        [ventric = none] &
        [ectvent = none]          [ectvent = none]          [ectvent = none]
```

The inverted knowledge-base was further transformed so that on the left hand sides of each rule only a single ECG attribute-value (ECG feature) occured. Then an inductive learning algorithm was applied over arrhythmias descriptions on the right hand sides of the rules. The result were descriptions of individual ECG features expressed in terms of possible arrhythmias.

For the second experiment we used an implementation of an algorithm similar to the AQ, the program EXCEL [Becker 85]. The program is less efficient than GEM due to its LISP implementation, but is capable of dealing with intersecting classes, i.e., some events occurring in more than one class. In this experiment this feature is particularly important, as separating all possible intersections of different values of an ECG attribute into new classes would be very repulsive effort.

Regular and ectopic arrhythmias knowledge-bases were used together. ECG descriptions in the first one are defined with 7 regular attributes only, while it is implicitly understood that the remaining 3 ectopic attributes are not applicable. To ensure uniformity of representation, we added the triple of ectopic attributes (ectopic_P, ectopic_PR, ectopic_QRS) with a new value (none) to all ECG descriptions in the regular arrhythmias knowledge-base. This resulted in more complicated descriptions of ectopic ECG features, but contributed to efficiency of diagnosing in cases when ectopic arrhythmias are presented.

Complete results of induced diagnostic rules can be found in the Appendix D. In the following table we give only complexity of descriptions for all ECG attributes:

ECG attribute	rules	conjunctions	attributes	values
rhythm	2	13	32	101
regular_P	4	4	9	23
rate_of_P	6	19	48	92
relation_P_QRS	4	11	31	56
regular_PR	6	17	38	83
regular_QRS	6	8	16	42
rate_of_QRS	5	27	59	127
ectopic_P	3	11	28	86
ectopic_PR	6	19	55	146
ectopic_QRS	7	15	55	135
Σ	49	144	371	891

As an example we give three rules that are slightly less complex then a typical rule induced:

[regular_P = abnormal] =>
    [sinus = none] & [atrial = none, at, afl, af, aeb] & [ventric = none, vr, avr, vt]

[regular_PR = shortened] =>
    [atrial = none, wp, at, mat, aeb] & [av_cond = wpw, lgl] v
    [av_cond = none] & [junction = jb, jr, jt] v
    [atrial = at] & [av_cond = none, wpw, lgl]

[regular_QRS = normal] =>
    [av_cond = none, avb1, wen, mob2, avb3, lgl] & [bb_cond = none] & [ventric = none]

Now, let us assume that a partially specified ECG description is given, for which we would like to find a set of possible arrhythmias:

Ecg <=> [regular_P = abnormal] & [regular_PR = shortened] & [regular_QRS = normal]

First, we have to compute the "intersection" of the right hand-side expressions of diagnostic rules:

Ecg =>
[sinus = none] & [atrial = none, **at**, aeb] & [av_cond = **lgl**] & [bb_cond = none]
& [ventric = none] v
[sinus = none] & [atrial = none, at, afl, af, aeb] & [av_cond = none] & [junction = jb, jr, **jt**]
& [bb_cond = none] & [ventric = none] v
[sinus = none] & [atrial = **at**] & [av_cond = none, lgl] & [bb_cond = none] & [ventric = none]

The expression is then translated from attribute-values vectors to multiple arrhythmias. At last a set of logically possible arrhythmias is further reduced by application of physiological constraints over multiple arrhythmias, which gives us the final result:

Ecg =>
{**a**trial_tachycardia} v
{**a**trial_tachycardia, **lgl**_syndrome} v
{**j**unctional_tachycardia}

An ECG diagnostic program that uses compressed diagnostic rules is implemented in UNSW Prolog, running on a SUN workstation. To diagnose a completely specified ECG, it takes between 1 to 5 minutes. However, in the case of partially specified ECG descriptions it may take as much as half an hour or even run out of memory.

The induced diagnostic rules are well suited to be used for diagnosing as we argue that the conditions concerning transformations of the knowledge-base (stated at the end of section 1.2) are satisfied. For diagnosing, the rules are used only in forward direction (from ECG features to possible arrhythmias) and we assume that impossible ECG descriptions are not given as an input. We have procedure to eliminate physiologically impossible arrhythmias from the set of logically possible arrhythmias, i.e., physiological constraints from the model of the heart. And at last we argue that there is no rule in which a new arrhythmia was added to the set of original conclusions during the transformations performed. This statement is difficult to prove because of the complexity of the knowledge-base. But so far no example was found to reject our belief. To be more specific, let us construct a condition that must be satisfied by such a conterexample.

There should exist an arrhythmia causing few ECG descriptions that have different values in at least two attributes, and some combinations of values are not possible. For example:

```
{sad} =>
     [rate_of_P = between_60_100] & [rate_of_QRS = between_60_100] v
     [rate_of_P = under_60] & [rate_of_QRS = under_60]
```

Suppose there is another arrhythmia, causing an ECG description consisting of a combination of values not possible in the previous case:

```
{sad, wen} =>
     [rate_of_P = between_60_100] & [rate_of_QRS = under_60]
```

After inversion of the knowledge-base and the transformations, the rules would look like:

```
[rate_of_P = between_60_100]      =>   {sad} v {sad, wen}
[rate_of_P = under_60]            =>   {sad}

[rate_of_QRS = between_60_100]    =>   {sad}
[rate_of_QRS = under_60]          =>   {sad} v {sad, wen}
```

There application for diagnosing obviously leads to conclusions not equivalent to ones that can be infered from the original rules. If the following ECG description is given:

```
[rate_of_P = between_60_100] &
[rate_of_QRS = under_60]          =>   {sad} v {sad, wen}
```

we get "sad" as possible arrythmia although it can be rejected using the original rules. Note that in the case of additional attributes, all their values must be equal for both arrhythmias if such a weaker conclusion is possible to derive.

In the case of the ECG knowledge-base no such example of a pair of arrhythmias satisfying the above construction was found. Therefore we argue that if an input ECG description is completely specified (with values for all attributes) a set of possible arrhythmias derived from the diagnostic rules is equivalent to the set derived from the original knowledge-base.

5. Conclusion

The following table gives the summary of results from the viewpoint of the inductive learning programs applied to the domain of ECG diagnosis of cardiac arrhythmias:

descriptions of:	number of runs	numbers for each run of classes	attributes	examples	max width of best-first search	total time spent
regular arrhythmias	24	3	7	263	10	9 min.
ectopic arrhythmias	3	3	10	1493	10	40 min.
ectopic arrhythmias with new attributes	3	3	16	1880	30	2 hours
diagnostic rules	10	2-7	7	950-1300	100	72 hours

Time in the last column of the table is real time, i.e., time that would be spend by a program running on a single-user machine. No input initialization and output time is included. All experiments were run on a SUN 170 workstation. In the first three cases the GEM program was used (a Pascal implementation of the AQ algorithm) and in the last case we used program EXCEL (a LISP implementation of the modified AQ algorithm).

As expected, GEM turned out to be more efficient, but has other disadvantages. It is much more difficult to change some of its features as its code is less transparent to the programmer and recompilation may take few hours. However, its greatest disadvantage is that it cannot handle intersecting classes. If it can, we would be able to induce descriptions of both, regular and ectopic arrhythmias combined in only 7 runs. Instead of dealing with "binary" attribute-values for each arrhythmia (which makes it easy to separate the intersecting events into the third class):

$[ARRattr=Arr_1]$ $[ARRattr\neq Arr_1]$

$[ARRattr=Arr_2]$ $[ARRattr\neq Arr_2]$

$[ARRattr=Arr_3]$ $[ARRattr\neq Arr_3]$

we could simply use each attribute-value pair as a class as in the case of diagnostic rules:

[ARRattr=Arr_1] [ARRattr=Arr_3]

[ARRattr=Arr_2] [ARRattr=none]

Beside simplification of the experiments it is also very likely that induced descriptions would be less complex. The argument for this assertion is that now each arrhythmia is characterizied with two rules (for classes POS and INTER) instead of only one. One rule can be constructed by simply taking a disjunction of the right hand-sides of the two rules, but no further simplification of the so obtained expression can be done simply.

For the end let us summarize the overall compression of the ECG knowledge-base in the following table:

total number of	original ECG knowledge-base	subset of the ECG knowledge-base	descriptions of arrhythmias combined	diagnostic rules
rules	2 419	586	45	49
conjunctions	8 314	957	75	144
attributes	58 197	6 699	248	371
values	986 762	16 835	429	891
Kbytes	5 100	400	10	13

Complexity of descriptions for arrhythmias combined is counted for regular and ectopic arrhythmias together, taking into account the case where no additional attributes were used. Only complexity of the classes POS and INTER is included in the numbers.

From the viewpoint of knowledge compression, it must not be forgotten that the compressed rules alone are not sufficient for diagnosing or construction of the ECG combinatorial function. First, we have to add descriptions of arrhythmias and ECG features that were eliminated when generating the subset of the ECG knowledge-base. And second, rules or procedures that can distinguish between "possible" and "impossible" events (e.g. physiological constraints over arrhythmias) must be included. After adding all these to the compressed diagnostic rules, the size of memory required to store all knowledge needed for diagnosing increases to 25 Kbytes. So, we may conclude that the overall compression of the ECG knowledge-base is approximately 200 times. Factor 10 is due to the generation of the subset, and approximately factor 20 was obtained by transformations and an application of inductive learning methods.

6. References

[Baim 84]

P.W. Baim : Automated Acquisition of Decision Rules: The Problem of Attribute Construction and Selection. M.S. Thesis, ISG 84-5, UIUCDCS-F-84-922, Dept. of Computer Science, University of Illinois at Urbana-Champaign, July, 1984.

[Becker 85]

J. Becker : Inductive Learning of Decision Rules with Exceptions: Methodology and Experimentation. M.S. Thesis in preparation, Dept. of Computer Science, University of Illinois at Urbana-Champaign, 1985.

[Michalski 83]

R.S. Michalski : A Theory and Methodology of Inductive Learning. Artificial Intelligence, pp. 111-161, 1983.

[Mozetic et al. 84]

I. Mozetic, I. Bratko, N. Lavrac : The Derivation of Medical Knowledge from a Qualitative Model of the Heart. ISSEK Workshop '84, Bled, Yugoslavia, August 22-24, 1984.

[Reinke 84]

R.E. Reinke : Knowledge Acquisition and Refinement Tools for the ADVISE META-EXPERT System. M.S. Thesis, ISG 84-4, UIUCDCS-F-84-921, Dept. of Computer Science, University of Illinois at Urbana-Champaign, July, 1984.

Appendices

A. Descriptions of regular arrhythmias

Numbers at the end of descriptions (:Unique :Total) have the following meaning:

- Unique - percentage of events that are uniquely covered by a conjunction,
- Total - percentage of all events covered by a conjunction.

Sinus rhythm

POSsr-outhypo
 # cpx
 1 [rhythm=regular] [regular_P=normal] [rate_of_P=in60to100] [regular_PR=normal,prolonged,shortened] :1.00 :1.00

INTERsr-outhypo
 # cpx
 1 [regular_P=normal] [rate_of_P=in60to100] [relation_P_QRS=missingQRS,independ] [regular_PR=notApp,prolonged] :1.00 :1.00

NEGsr-outhypo
 # cpx
 1 [rate_of_P=zero..under60] :0.09 :0.26
 2 [rate_of_P=in100to250..over350] :0.11 :0.55
 3 [rhythm=irregular] [regular_PR=notApp,normal,shortened,changing,aftQRSisP] :0.02 :0.31
 4 [regular_P=abnormal,changing,absent] :0.08 :0.71
 5 [rhythm=irregular] [relation_P_QRS=notApp,alwaysQRS,independ] :0.01 :0.24

This run used (milliseconds of CPU time):
 System time: 3483
 User time : 14033

Sinus bradycardia

POSsb-outhypo
 # cpx
 1 [rhythm=regular] [regular_P=normal] [rate_of_P=under60] [regular_PR=normal,prolonged,shortened] :1.00 :1.00

INTERsb-outhypo
 # cpx
 1 [rate_of_P=under60] [relation_P_QRS=missingQRS,independ] [regular_PR=notApp,prolonged] :1.00 :1.00

NEGsb-outhypo
 # cpx
 1 [rate_of_P=in60to100..over350] :0.21 :0.83
 2 [rhythm=irregular] [relation_P_QRS=notApp,alwaysQRS,independ] :0.01 :0.23
 3 [regular_P=abnormal,changing,absent] :0.10 :0.70
 4 [rhythm=irregular] [regular_PR=notApp,normal,shortened,changing,aftQRSisP] :0.01 :0.31

This run used (milliseconds of CPU time):
 System time: 4067
 User time : 12900

Sinus tachycardia

POSst-outhypo
 # cpx
 1 [regular_P=normal] [rate_of_P=in100to250] :1.00 :1.00

INTERst-outhypo
 # cpx

NEGst-outhypo
 # cpx
 1 [rate_of_P=zero..in60to100] :0.29 :0.55
 2 [regular_P=abnormal,changing,absent] :0.45 :0.71

This run used (milliseconds of CPU time):
 System time: 2484
 User time : 8500

Sinus node disorders

POSsad-outhypo
cpx
1 [rhythm=irregular] [regular_P=normal] [regular_PR=normal,shortened] :0.27 :0.82
2 [rhythm=irregular] [regular_P=normal] [relation_P_QRS=alwaysQRS] :0.18 :0.73

INTERsad-outhypo
cpx
1 [regular_P=normal] [rate_of_P=under60..in60to100] [relation_P_QRS=missingQRS,independ] [regular_PR=notApp,prolonged] :1.00 :1.00

NEGsad-outhypo
cpx
1 [rhythm=regular] [relation_P_QRS=notApp,alwaysQRS,missingQRS] :0.10 :0.36
2 [regular_P=abnormal,changing,absent] :0.20 :0.78
3 [rate_of_P=in100to250..over350] :0.06 :0.60

This run used (milliseconds of CPU time):
System time: 6750
User time : 19017

Wandering pacemaker

POSwp-outhypo
cpx
1 [regular_P=changing] [rate_of_P=in60to100] :1.00 :1.00

INTERwp-outhypo
cpx

NEGwp-outhypo
cpx
1 [regular_P=normal,abnormal,absent] :0.44 :0.89
2 [rate_of_P=in100to250..over350] :0.11 :0.56

This run used (milliseconds of CPU time):
System time: 1300
User time : 8834

Atrial tachycardia

POSat-outhypo
cpx
1 [regular_P=abnormal] [rate_of_P=in100to250] [regular_PR=notApp,normal,prolonged] :0.91 :0.91
2 [regular_P=abnormal] [regular_PR=normal,prolonged,shortened]
[regular_QRS=wLBBBaRBBB,deltaLBBB,deltaRBBB] :0.09 :0.09

INTERat-outhypo
cpx
1 [regular_P=abnormal] [rate_of_P=in100to250] [regular_PR=shortened] [regular_QRS=normal,wLBBB]
:1.00 :1.00

NEGat-outhypo
cpx
1 [regular_P=normal,changing,absent] :0.21 :0.73
2 [rate_of_P=in250to350..over350] :0.21 :0.21
3 [rate_of_P=zero..in60to100] :0.02 :0.55
4 [regular_PR=changing,aftQRSisP] :0.01 :0.05

This run used (milliseconds of CPU time):
System time: 12667
User time : 26367

Multifocal atrial tachycardia

POSmat-outhypo
cpx
1 [regular_P=changing] [rate_of_P=in100to250] :1.00 :1.00

INTERmat-outhypo
cpx

NEGmat-outhypo
cpx
1 [regular_P=normal,abnormal,absent] :0.44 :0.89
2 [rate_of_P=zero..in60to100] :0.11 :0.56

This run used (milliseconds of CPU time):
System time: 9800
User time : 11200

Atrial flutter

POSafl-outhypo
```
  #  cpx
  1  [rate_of_P=in250to350]  :1.00 :1.00
```

INTERafl-outhypo
```
  #  cpx
```

NEGafl-outhypo
```
  #  cpx
  1  [rate_of_P=zero..in100to250]  :0.88 :0.88
  2  [rate_of_P=over350]  :0.12 :0.12
```

This run used (milliseconds of CPU time):
System time: 8783
User time : 17083

Atrial fibrillation

POSaf-outhypo
```
  #  cpx
  1  [rate_of_P=over350]  :0.65 :0.65
  2  [rhythm=irregular] [regular_P=absent] [rate_of_QRS=under60..in100to250]  :0.14 :0.14
  3  [regular_P=absent] [relation_P_QRS=missingQRS,independ]  :0.21 :0.21
```

INTERaf-outhypo
```
  #  cpx
  1  [regular_P=absent] [rate_of_QRS=over350]  :1.00 :1.00
```

NEGaf-outhypo
```
  #  cpx
  1  [rate_of_P=under60..in250to350]  :0.64 :0.95
  2  [rhythm=regular] [relation_P_QRS=notApp,alwaysQRS,missingQRS]  :0.05 :0.36
```

This run used (milliseconds of CPU time):
System time: 26950
User time : 48917

Atrio-ventricular block 1

POSavb1-outhypo
  # cpx
  1 [regular_P=normal,changing] [relation_P_QRS=alwaysQRS] [regular_PR=prolonged] :1.00 :1.00

INTERavb1-outhypo
  # cpx
  1 [rhythm=regular] [regular_P=abnormal] [regular_PR=prolonged] :1.00 :1.00

NEGavb1-outhypo
  # cpx
  1 [regular_PR=notApp,normal,shortened,changing,aftQRSisP] :0.26 :0.91
  2 [relation_P_QRS=notApp,missingQRS,independ] :0.09 :0.74

This run used (milliseconds of CPU time):
 System time: 9166
 User time : 19300

Wenckebach

POSwen-outhypo
  # cpx
  1 [relation_P_QRS=missingQRS] [regular_PR=prolonged] :1.00 :1.00

INTERwen-outhypo
  # cpx

NEGwen-outhypo
  # cpx
  1 [regular_PR=notApp,normal,shortened,changing,aftQRSisP] :0.19 :0.93
  2 [relation_P_QRS=notApp,alwaysQRS,independ] :0.07 :0.81

This run used (milliseconds of CPU time):
 System time: 5100
 User time : 11116

Mobitz 2

POSmob2-outhypo
 # cpx
 1 [relation_P_QRS=missingQRS] [regular_PR=normal] :1.00 :1.00

INTERmob2-outhypo
 # cpx

NEGmob2-outhypo
 # cpx
 1 [relation_P_QRS=notApp,alwaysQRS,independ] :0.07 :0.83
 2 [regular_PR=notApp,prolonged,shortened,changing,aftQRSisP] :0.17 :0.93

This run used (milliseconds of CPU time):
 System time: 5600
 User time : 9816

Atrio-ventricular block 3

POSavb3-outhypo
 # cpx
 1 [rhythm=regular] [relation_P_QRS=independ] :1.00 :1.00

INTERavb3-outhypo
 # cpx

NEGavb3-outhypo
 # cpx
 1 [relation_P_QRS=notApp,alwaysQRS,missingQRS] :0.43 :0.97
 2 [rhythm=irregular] :0.03 :0.57

This run used (milliseconds of CPU time):
 System time: 14533
 User time : 30083

WPW syndrome

POSwpw-outhypo
 # cpx
 1 [regular_P=normal,abnormal,changing] [regular_QRS=deltaLBBB,deltaRBBB,absent] :1.00 :1.00

INTERwpw-outhypo
 # cpx
 1 [regular_P=absent] [regular_QRS=absent] :1.00 :1.00

NEGwpw-outhypo
 # cpx
 1 [regular_QRS=normal,wLBBB,wLBBBaRBBB] :1.00 :1.00

This run used (milliseconds of CPU time):
System time: 12100
User time : 28600

LGL syndrome

POSlgl-outhypo
 # cpx
 1 [regular_P=normal,changing] [regular_PR=shortened] [regular_QRS=normal,wLBBB] :1.00 :1.00

INTERlgl-outhypo
 # cpx
 1 [regular_P=abnormal] [rate_of_P=in100to250] [regular_PR=shortened] [regular_QRS=normal,wLBBB] :1.00 :1.00

NEGlgl-outhypo
 # cpx
 1 [regular_PR=notApp,normal,prolonged,changing,aftQRSisP] :0.55 :0.92
 2 [regular_P=abnormal,absent] [rate_of_QRS=under60..in60to100] :0.02 :0.28
 3 [regular_QRS=wLBBBaRBBB,deltaLBBB,deltaRBBB,absent] :0.06 :0.22

This run used (milliseconds of CPU time):
System time: 8067
User time : 17800

Junctional bradycardia

POSjb-outhypo

#	cpx
1	[rhythm=regular] [regular_P=abnormal,changing,absent] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_QRS=normal] [rate_of_QRS=under60] :0.46 :0.69
2	[regular_P=abnormal,changing,absent] [rate_of_P=zero..under60] [regular_PR=shortened] :0.08 :0.15
3	[regular_P=normal,changing] [regular_PR=notApp,aftQRSisP] [regular_QRS=normal] [rate_of_QRS=under60] :0.23 :0.38

INTERjb-outhypo

#	cpx
1	[rhythm=regular] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_PR=notApp,aftQRSisP] [regular_QRS=wLBBB] [rate_of_QRS=under60] :1.00 :1.00

NEGjb-outhypo

#	cpx
1	[rate_of_QRS=in60to100..over350] :0.22 :0.77
2	[regular_P=normal,changing,absent] [regular_PR=normal,prolonged,shortened,changing,aftQRSisP] :0.03 :0.43
3	[rhythm=irregular] :0.03 :0.44
4	[relation_P_QRS=missingQRS] :0.01 :0.28
5	[regular_QRS=wLBBBaRBBB,deltaLBBB,deltaRBBB,absent] :0.05 :0.23

This run used (milliseconds of CPU time):

System time: 22050

User time : 45383

Junctional rhythm

POSjr-outhypo

#	cpx
1	[rhythm=regular] [regular_P=abnormal,changing,absent] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_QRS=normal] [rate_of_QRS=in60to100] :0.08 :0.69
2	[regular_P=abnormal,absent] [regular_PR=shortened] [rate_of_QRS=in60to100] :0.08 :0.15
3	[rhythm=regular] [relation_P_QRS=notApp,independ] [regular_QRS=normal] [rate_of_QRS=in60to100] :0.23 :0.77

INTERjr-outhypo

#	cpx
1	[rhythm=regular] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_PR=notApp,aftQRSisP] [regular_QRS=wLBBB] [rate_of_QRS=in60to100] :1.00 :1.00

NEGjr-outhypo

#	cpx
1	[regular_P=normal,abnormal,changing] [relation_P_QRS=notApp,alwaysQRS,missingQRS] [regular_PR=notApp,normal,prolonged,changing] :0.02 :0.47
2	[rate_of_QRS=under60] :0.10 :0.33
3	[rhythm=irregular] :0.02 :0.44
4	[rate_of_QRS=in100to250..over350] :0.10 :0.37
5	[regular_QRS=wLBBBaRBBB,deltaLBBB,deltaRBBB,absent] :0.05 :0.23
6	[regular_P=normal,changing,absent] [regular_PR=normal,prolonged,shortened,changing,aftQRSisP] :0.01 :0.43

This run used (milliseconds of CPU time):
System time: 22734
User time : 49067

Junctional tachycardia

POSjt-outhypo

cpx
1 [rhythm=regular] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_PR=notApp,aftQRSisP] [regular_QRS=normal] [rate_of_QRS=in100to250] :1.00 :1.00

INTERjt-outhypo

cpx
1 [regular_P=abnormal,absent] [rate_of_P=in100to250..over350] [regular_PR=shortened] [regular_QRS=normal,wLBBB] :0.15 :0.15
2 [rhythm=regular] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_PR=notApp,aftQRSisP] [regular_QRS=wLBBB] [rate_of_QRS=in100to250] :0.85 :0.85

NEGjt-outhypo

cpx
1 [rate_of_QRS=under60..in60to100] :0.20 :0.73
2 [regular_P=normal,abnormal,changing] [relation_P_QRS=notApp,alwaysQRS,missingQRS] [regular_PR=notApp,normal,prolonged,changing] :0.03 :0.47
3 [rhythm=irregular] :0.02 :0.44
4 [regular_QRS=wLBBBaRBBB,deltaLBBB,deltaRBBB,absent] :0.05 :0.23
5 [rate_of_QRS=in250to350..over350] :0.00 :0.03
6 [regular_P=normal,changing,absent] [regular_PR=normal,prolonged,shortened,changing,aftQRSisP] :0.01 :0.43

This run used (milliseconds of CPU time):
System time: 12317
User time : 33867

Left bundle branch block

POSlbbb-outhypo
cpx
1 [rate_of_P=under60..over350] [relation_P_QRS=notApp,alwaysQRS,missingQRS] [regular_PR=notApp,normal,prolonged,shortened,changing] [regular_QRS=wLBBB] :0.34 :0.92
2 [rhythm=irregular] [regular_QRS=wLBBB] :0.08 :0.66

INTERlbbb-outhypo
cpx
1 [rhythm=regular] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_PR=notApp,aftQRSisP] [regular_QRS=wLBBB] [rate_of_QRS=under60..in100to250] :1.00 :1.00

NEGlbbb-outhypo
cpx
1 [regular_QRS=normal,wLBBBaRBBB,deltaLBBB,deltaRBBB,absent] :0.96 :0.99
2 [rate_of_QRS=in250to350..over350] :0.01 :0.04

This run used (milliseconds of CPU time):
System time: 8684
User time : 25950

Ventricular rhythm

POSvr-outhypo
cpx
1 [regular_QRS=wLBBBaRBBB] [rate_of_QRS=under60] :1.00 :1.00

INTERvr-outhypo
cpx
1 [rhythm=regular] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_PR=notApp,aftQRSisP] [regular_QRS=wLBBB] [rate_of_QRS=under60] :1.00 :1.00

NEGvr-outhypo
cpx
1 [rate_of_QRS=in60to100..over350] :0.19 :0.76
2 [rate_of_P=under60..over350] [relation_P_QRS=notApp,alwaysQRS,missingQRS] [regular_PR=notApp,normal,prolonged,shortened,changing] :0.03 :0.62
3 [regular_QRS=normal,deltaLBBB,deltaRBBB,absent] :0.05 :0.51
4 [rhythm=irregular] :0.01 :0.43

This run used (milliseconds of CPU time):
System time: 2383
User time : 18634

Accelerated ventricular rhythm

POSavr-outhypo
 # cpx
 1 [regular_QRS=wLBBBaRBBB] [rate_of_QRS=in60to100] :1.00 :1.00

INTERavr-outhypo
 # cpx
 1 [rhythm=regular] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_PR=notApp,aftQRSisP] [regular_QRS=wLBBB] [rate_of_QRS=in60to100] :1.00 :1.00

NEGavr-outhypo
 # cpx
 1 [rate_of_P=under60..over350] [relation_P_QRS=notApp,alwaysQRS,missingQRS] [regular_PR=notApp,normal,prolonged,shortened,changing] :0.03 :0.62
 2 [regular_QRS=normal,deltaLBBB,deltaRBBB,absent] :0.05 :0.51
 3 [rate_of_QRS=under60] :0.09 :0.33
 4 [rate_of_QRS=in100to250..over350] :0.10 :0.37
 5 [rhythm=irregular] :0.01 :0.43

This run used (milliseconds of CPU time):
System time: 3016
User time : 20317

Ventricular tachycardia

POSvt-outhypo
 # cpx
 1 [regular_QRS=wLBBBaRBBB] [rate_of_QRS=in100to250] :1.00 :1.00

INTERvt-outhypo
 # cpx
 1 [rhythm=regular] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_PR=notApp,aftQRSisP] [regular_QRS=wLBBB] [rate_of_QRS=in100to250] :1.00 :1.00

NEGvt-outhypo
 # cpx
 1 [rate_of_QRS=under60..in60to100] :0.18 :0.72
 2 [rate_of_P=under60..over350] [relation_P_QRS=notApp,alwaysQRS,missingQRS] [regular_PR=notApp,normal,prolonged,shortened,changing] :0.04 :0.62
 3 [regular_QRS=normal,deltaLBBB,deltaRBBB,absent] :0.05 :0.51
 4 [rate_of_QRS=in250to350..over350] :0.01 :0.02
 5 [rhythm=irregular] :0.01 :0.43

This run used (milliseconds of CPU time):
System time: 2217
User time : 19000

Ventricular flutter

POSvfl-outhypo
\# cpx
1 [regular_QRS=wLBBB,wLBBBaRBBB] [rate_of_QRS=in250to350] :1.00 :1.00

INTERvfl-outhypo
\# cpx

NEGvfl-outhypo
\# cpx
1 [rate_of_QRS=under60..in100to250] :0.52 :0.98
2 [regular_QRS=normal,deltaLBBB,deltaRBBB,absent] :0.02 :0.48

This run used (milliseconds of CPU time):
System time: 1700
User time : 9917

Ventricular fibrillation

POSvf-outhypo
\# cpx

INTERvf-outhypo
\# cpx
1 [regular_P=absent] [rate_of_QRS=over350] :1.00 :1.00

NEGvf-outhypo
\# cpx
1 [regular_QRS=normal,wLBBB,wLBBBaRBBB,deltaLBBB,deltaRBBB] :0.10 :1.00
2 [regular_P=normal,abnormal,changing] :0.00 :0.90

This run used (milliseconds of CPU time):
System time: 1550
User time : 14183

B. Descriptions of ectopic arrhythmias

Atrial ectopic beats

POSaeb-outhypo

#	cpx
1	[ectopic_P=abnormal] [ectopic_PR=notApp,normal,prolonged]
2	[rhythm=regular] [regular_P=abnormal,absent] [rate_of_P=zero..in60to100] [regular_QRS=normal,deltaLBBB,deltaRBBB] [ectopic_PR=normal,prolonged,shortened]
3	[regular_PR=aftQRSisP] [regular_QRS=normal,deltaLBBB,deltaRBBB] [ectopic_PR=normal,prolonged,shortened]
4	[ectopic_QRS=deltaLBBB,deltaRBBB,absent]
5	[regular_P=abnormal,absent] [regular_PR=shortened] [rate_of_QRS=under60..in60to100] [ectopic_PR=normal,prolonged,shortened]

INTERaeb-outhypo

#	cpx
1	[regular_P=normal] [relation_P_QRS=notApp,alwaysQRS] [regular_PR=notApp,normal,shortened,aftQRSisP] [regular_QRS=normal,wLBBB,wLBBBaRBBB] [ectopic_PR=shortened]
2	[rhythm=regular] [rate_of_P=zero..in100to250] [regular_PR=notApp,aftQRSisP] [regular_QRS=wLBBB,wLBBBaRBBB] [ectopic_PR=shortened]
3	[regular_P=normal,abnormal,absent] [rate_of_P=in100to250] [regular_PR=shortened] [regular_QRS=normal,wLBBB,wLBBBaRBBB] [ectopic_PR=shortened]

NEGaeb-outhypo

#	cpx
1	[ectopic_PR=notApp,aftQRSisP] [ectopic_QRS=normal,wLBBB,wLBBBaRBBB]
2	[rhythm=irregular] [regular_P=abnormal,changing,absent]
3	[regular_P=abnormal,changing] [regular_PR=notApp,normal,prolonged,changing]
4	[regular_PR=prolonged,changing] [ectopic_PR=notApp,shortened,aftQRSisP]
5	[relation_P_QRS=missingQRS] [ectopic_PR=notApp,shortened,aftQRSisP]
6	[regular_QRS=deltaLBBB,deltaRBBB] [ectopic_QRS=normal,wLBBB,wLBBBaRBBB]

This run used (milliseconds of CPU time):
System time: 227734
User time : 779216

Junctional ectopic beats

POSjeb-outhypo

#	cpx
1	[ectopic_PR=notApp,aftQRSisP] [ectopic_QRS=normal]
2	[rhythm=irregular] [regular_P=abnormal,changing,absent] [ectopic_PR=shortened]
3	[regular_P=abnormal,changing] [regular_PR=notApp,normal,prolonged,changing] [ectopic_PR=shortened]
4	[regular_PR=prolonged,changing] [ectopic_PR=shortened]
5	[relation_P_QRS=missingQRS,independ] [ectopic_PR=shortened]
6	[regular_QRS=deltaLBBB,deltaRBBB] [ectopic_PR=shortened] [ectopic_QRS=normal,wLBBB]

INTERjeb-outhypo

#	cpx
1	[regular_P=normal] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_PR=notApp,normal,shortened,changing,aftQRSisP] [regular_QRS=normal,wLBBB,wLBBBaRBBB] [ectopic_PR=shortened]
2	[rhythm=regular] [rate_of_P=zero..in100to250] [regular_PR=notApp,changing,aftQRSisP] [regular_QRS=wLBBB,wLBBBaRBBB,deltaLBBB,deltaRBBB] [ectopic_PR=shortened]
3	[regular_PR=notApp,normal,prolonged,changing,aftQRSisP] [regular_QRS=wLBBB,wLBBBaRBBB,deltaLBBB,deltaRBBB] [ectopic_PR=notApp,aftQRSisP] [ectopic_QRS=wLBBB]
4	[regular_P=normal,abnormal,absent] [rate_of_P=in100to250..over350] [regular_PR=shortened,changing] [regular_QRS=normal,wLBBB,wLBBBaRBBB] [ectopic_PR=shortened]
5	[regular_P=normal,changing,absent] [regular_QRS=wLBBB,wLBBBaRBBB,deltaLBBB,deltaRBBB] [ectopic_PR=notApp,aftQRSisP] [ectopic_QRS=wLBBB]
6	[regular_QRS=wLBBB,wLBBBaRBBB,deltaLBBB,deltaRBBB] [rate_of_QRS=in100to250] [ectopic_PR=notApp,aftQRSisP] [ectopic_QRS=wLBBB]

NEGjeb-outhypo

#	cpx
1	[ectopic_P=abnormal] [ectopic_PR=notApp,normal,prolonged]
2	[regular_QRS=normal] [ectopic_QRS=wLBBB,wLBBBaRBBB,deltaLBBB,deltaRBBB,absent]
3	[rhythm=regular] [regular_P=abnormal,changing,absent] [rate_of_P=under60..in60to100] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_PR=normal,prolonged,shortened,changing] [rate_of_QRS=under60..in60to100]
4	[rhythm=regular] [rate_of_P=zero..in100to250] [regular_PR=notApp,changing,aftQRSisP] [regular_QRS=normal,deltaLBBB,deltaRBBB]
5	[ectopic_QRS=wLBBBaRBBB,deltaLBBB,deltaRBBB,absent]

This run used (milliseconds of CPU time):
System time: 319050
User time : 1110783

Ventricular ectopic beats

POSveb-outhypo

```
#  cpx
1  [regular_QRS=normal] [ectopic_QRS=wLBBB,wLBBBaRBBB]
2  [ectopic_QRS=wLBBBaRBBB]
3  [regular_P=abnormal,absent] [regular_PR=shortened,changing] [rate_of_QRS=under60..in60to100]
   [ectopic_PR=notApp,aftQRSisP]
```

INTERveb-outhypo

```
#  cpx
1  [regular_PR=notApp,normal,prolonged,changing,aftQRSisP]
   [regular_QRS=wLBBB,wLBBBaRBBB,deltaLBBB,deltaRBBB]
   [ectopic_PR=notApp,aftQRSisP] [ectopic_QRS=wLBBB]
2  [regular_P=normal,changing,absent] [regular_QRS=wLBBB,wLBBBaRBBB,deltaLBBB,deltaRBBB]
   [ectopic_PR=notApp,aftQRSisP] [ectopic_QRS=wLBBB]
3  [regular_QRS=wLBBB,wLBBBaRBBB,deltaLBBB,deltaRBBB] [rate_of_QRS=in100to250]
   [ectopic_PR=notApp,aftQRSisP] [ectopic_QRS=wLBBB]
```

NEGveb-outhypo

```
#  cpx
1  [ectopic_QRS=normal,deltaLBBB,deltaRBBB,absent]
2  [ectopic_PR=normal,prolonged,shortened]
```

This run used (milliseconds of CPU time):

```
System time:   191233
User time  :   492683
```

C. Descriptions of ectopic arrhythmias with new attributes

Numbers at the end of descriptions (:Unique :Total) have the following meaning:

- Unique - percentage of events that are uniquely covered by a conjunction,
- Total - percentage of all events covered by a conjunction.

Atrial ectopic beats

POSaeb-outhypo
cpx
1 [ect_P=abnormal] [ect_PR=notApp,normal,prolonged] :0.45 :0.84
2 [ect_PR=normal,prolonged,shortened] [sinus=none,sr] [atrial=none] [av_cond=none,mob2,avb3] [ventric=none] [regular_PR=notApp,prolonged,shortened,aftQRSisP] :0.07 :0.22
3 [ect_QRS=deltaLBBB,deltaRBBB,absent] :0.08 :0.33

INTERaeb-outhypo
cpx
1 [ect_PR=shortened] [atrial=none] [av_cond=none,lgl] [junction=none] :1.00 :1.00

NEGaeb-outhypo
cpx
1 [ect_PR=notApp,aftQRSisP] [ect_QRS=normal,wLBBB,wLBBBaRBBB] :0.37 :0.90
2 [atrial=wp,at,mat,afl,af] :0.04 :0.50
3 [ect_PR=shortened,aftQRSisP] [ect_QRS=normal,wLBBB,wLBBBaRBBB] [av_cond=avb1,wen,mob2,avb3,wpw] :0.04 :0.22

This run used (milliseconds of CPU time):
System time: 73700
User time : 1609266

Junctional ectopic beats

POSjeb-outhypo

#	cpx
1	[ect_PR=notApp,aftQRSisP] [ect_QRS=normal] :0.49 :0.49
2	[ect_PR=shortened] [atrial=wp,at,mat,afl,af] :0.10 :0.18
3	[ect_PR=shortened] [ect_QRS=normal,wLBBB] [av_cond=avb1,wen,mob2,avb3,wpw] :0.09 :0.19
4	[ect_PR=notApp,aftQRSisP] [atrial=none,wp,mat,afl] [av_cond=none,wen,mob2,avb3,wpw] [junction=none] [bb_cond=lbbb] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_PR=notApp,prolonged,shortened,aftQRSisP] :0.04 :0.18
5	[av_cond=avb3,wpw] [junction=none] [bb_cond=lbbb] :0.04 :0.21

INTERjeb-outhypo

#	cpx
1	[ect_PR=shortened] [atrial=none] [av_cond=none,lgl] [junction=none] :0.23 :0.23
2	[ect_PR=notApp,aftQRSisP] [ect_QRS=normal,wLBBB] [junction=none] [ventric=none] [regular_QRS=wLBBB,wLBBBaRBBB] :0.77 :0.77

NEGjeb-outhypo

#	cpx
1	[ect_P=abnormal] [ect_PR=notApp,normal,prolonged] :0.10 :0.18
2	[ect_QRS=wLBBB,wLBBBaRBBB,deltaLBBB,deltaRBBB,absent] [bb_cond=none] :0.24 :0.62
3	[atrial=none,wp,mat,af] [av_cond=none,wen,mob2,avb3] [ventric=none] [rhythm=regular] [rate_of_P=zero..in250to350] [regular_PR=notApp,prolonged,shortened,changing,aftQRSisP] :0.00 :0.22
4	[ect_QRS=wLBBBaRBBB,deltaLBBB,deltaRBBB,absent] :0.13 :0.53
5	[junction=jb,jr,jt] :0.01 :0.25

This run used (milliseconds of CPU time):
System time: 2401567
User time : 3493316

Ventricular ectopic beats

POSveb-outhypo

#	cpx
1	[ect_QRS=wLBBB,wLBBBaRBBB] [bb_cond=none] :0.30 :0.72
2	[ect_QRS=wLBBBaRBBB] :0.17 :0.58
3	[ect_PR=notApp,aftQRSisP] [ect_QRS=wLBBB,wLBBBaRBBB] [junction=jb,jr,jt] :0.06 :0.23

INTERveb-outhypo

#	cpx
1	[ect_PR=notApp,aftQRSisP] [ect_QRS=wLBBB] [junction=none] [ventric=none] [regular_QRS=wLBBB] :1.00 :1.00

NEGveb-outhypo

#	cpx
1	[ect_QRS=normal,deltaLBBB,deltaRBBB,absent] :0.39 :0.64
2	[ect_PR=normal,prolonged,shortened] :0.16 :0.46
3	[atrial=none,wp,mat,afl] [av_cond=none,wen,mob2,avb3,wpw] [junction=none] [bb_cond=lbbb] [relation_P_QRS=notApp,alwaysQRS,independ] [regular_PR=notApp,prolonged,shortened,aftQRSisP] :0.03 :0.18
4	[av_cond=avb3,wpw] [junction=none] [bb_cond=lbbb] :0.03 :0.14

This run used (milliseconds of CPU time):
System time: 2408617
User time : 1908233

D. Diagnostic rules

Numbers at the beginning of each conjunction (New, Total, Except) have the following meaning:

- New - number of events that are newly covered by a conjunction, i.e., are not covered by any previous conjunction,
- Total - total number of events covered by a conjunction,
- Except - number of exceptions (should be always 0).

```
[rhythm = regular]  =>
   (210,210,  0)  [junction = jb v jr v jt] v
   (202,202,  0)  [ventric = vr v avr v vt v vfl] v
   (168,267,  0)  [av_cond = none v avb1 v mob2 v avb3 v wpw v lgl] & [sinus = sr v sb v st] v
   ( 22, 40,  0)  [atrial = at v afl] & [av_cond = none v avb1 v mob2 v avb3 v wpw v lgl]
                   & [ectvent = veb] v
   ( 24,102,  0)  [atrial = at v afl v aeb] & [av_cond = none v avb1 v mob2 v avb3 v wpw v lgl]
                   & [junction = jb v jr v jt v jeb] v
   ( 12, 30,  0)  [atrial = at v afl] & [av_cond = none v avb1 v mob2 v avb3 v wpw v lgl]
                   & [ectvent = none] & [junction = none v jb v jr v jt].

[rhythm = irregular]  =>
   (  1,  1,  0)  [sinus = none] & [ventric = vf] v
   (208,208,  0)  [atrial = none v wp v mat v af v aeb] & [av_cond = avb1 v wen v mob2 v wpw v lgl]
                   & [sinus = none v sad] v
   ( 46, 90,  0)  [av_cond = wen] v
   ( 20, 60,  0)  [atrial = wp v mat v afl v af] & [junction = jeb] & [ventric = none] v
   ( 20, 56,  0)  [atrial = wp v mat v afl v af] & [av_cond = none v avb1 v wen v mob2 v wpw v lgl]
                   & [ectvent = veb] v
   ( 10, 26,  0)  [atrial = wp v mat v afl v af] & [av_cond = none v avb1 v wen v mob2 v lgl]
                   & [ectvent = none] & [junction = none] v
   ( 24,136,  0)  [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [sinus = sad].
```

Time spent:
  1:25:40.617 Sec. CPU
  39:20.083 Sec. GC
  5:14:34.000 Sec. REAL

```
[rate_of_P = zero]  =>
   ( 88, 88,  0)   [atrial = none v af v aeb] & [sinus = none].

[rate_of_P = under_60]  =>
   (202,202,  0)   [sinus = sb v sad] v
   ( 13,143,  0)   [atrial = none] & [junction = none v jb v jeb] & [sinus = none v sb v sad]
                    & [ventric = none v vr] v
   (  2,  6,  0)   [atrial = aeb] & [junction = jb] & [sinus = none v sb v sad] v
   (  2,  9,  0)   [av_cond = none] & [ventric = vr].

[rate_of_P = between_60_100]  =>
   (202,202,  0)   [sinus = sr v sad] v
   ( 81, 81,  0)   [atrial = wp] v
   (  2, 26,  0)   [atrial = wp v aeb] & [av_cond = none v avb1 v wen v mob2 v wpw v lgl]
                    & [junction = jr v jeb] v
   (  4, 10,  0)   [atrial = none v wp] & [ectvent = veb] & [junction = jr] & [sinus = none v sr v sad] v
   (  9,  9,  0)   [av_cond = none] & [ventric = avr] v
   (  2,  8,  0)   [atrial = none v wp] & [ectvent = none] & [junction = jr] & [sinus = none v sr v sad].

[rate_of_P = between_100_250]  =>
   (162,162,  0)   [atrial = at v mat] v
   (101,101,  0)   [sinus = st] v
   (  6, 81,  0)   [atrial = none v at v mat] & [ectvent = veb] & [junction = none v jt]
                    & [sinus = none v st] & [ventric = none v vt] v
   (  2, 50,  0)   [atrial = at v mat v aeb] & [av_cond = none v avb1 v wen v mob2 v wpw v lgl]
                    & [junction = jt v jeb] v
   (  7,127,  0)   [atrial = none v at v mat] & [ectvent = none] & [junction = none v jt v jeb]
                    & [sinus = none v st] & [ventric = none v vt] v
   (  2,  9,  0)   [av_cond = none] & [ventric = vt].

[rate_of_P = between_250_350]  =>
   ( 35, 35,  0)   [atrial = afl] & [sinus = none].

[rate_of_P = over_350]  =>
   ( 35, 35,  0)   [atrial = af] & [sinus = none].

Time spent:
  1:7:35.433 Sec. CPU
  32:17.567 Sec. GC
  4:42:3.000 Sec. REAL
```

[regular_P = normal] =>
(505,505, 0) [atrial = none v aeb] & [sinus = sr v sb v st v sad].

[regular_P = abnormal] =>
(202,202, 0) [atrial = none v at v afl v af v aeb] & [sinus = none] & [ventric = none v vr v avr v vt].

[regular_P = changing] =>
(162,162, 0) [atrial = wp v mat] & [sinus = none].

[regular_P = absent] =>
(88, 88, 0) [atrial = none v af v aeb] & [sinus = none].

Time spent:
7:56.617 Sec. CPU
3:22.433 Sec. GC
25:49.000 Sec. REAL

[relation_P_QRS = meaningless] =>
(75, 75, 0) [atrial = none v af v aeb] & [av_cond = none v wpw] & [sinus = none] v
(1, 3, 0) [atrial = afl v af] & [av_cond = wpw] & [sinus = none].

[relation_P_QRS = after_P_always_QRS] =>
(382,382, 0) [atrial = none v wp v at v mat v aeb] & [av_cond = none v avb1 v wpw v lgl]
& [ventric = none v vr v avr v vt].

[relation_P_QRS = after_P_some_QRS_miss] =>
(180,180, 0) [av_cond = wen v mob2] v
(8, 8, 0) [atrial = afl v af] & [junction = jeb] & [ventric = none] v
(8, 8, 0) [atrial = afl v af] & [av_cond = none] & [ectvent = veb] v
(4, 4, 0) [atrial = afl v af] & [av_cond = none] & [ectvent = none] & [junction = none].

[relation_P_QRS = independent_P_QRS] =>
(309,309, 0) [av_cond = avb3] v
(4, 22, 0) [atrial = af] & [ectvent = veb] & [sinus = none] v
(4, 40, 0) [atrial = af] & [junction = jb v jr v jt v jeb] & [sinus = none] v
(2, 20, 0) [atrial = af] & [av_cond = none v avb3] & [ectvent = none]
& [junction = none v jb v jr v jt].

Time spent:
1:19:47.983 Sec. CPU
38:10.433 Sec. GC
6:4:41.000 Sec. REAL

[regular_PR = meaningless] =>
(309,309, 0) [av_cond = avb3] v
(29,161, 0) [sinus = none v sb v st v sad] & [ventric = vr v avr v vt v vfl v vf] v
(24,186, 0) [junction = jb v jr v jt] v
(12, 39, 0) [atrial = afl v af] & [ectvent = veb] v
(12,102, 0) [atrial = afl v af v aeb] & [junction = jb v jr v jt v jeb] v
(9, 36, 0) [atrial = afl v af] & [ectvent = none] & [junction = none v jb v jr v jt].

[regular_PR = normal] =>
(120,120, 0) [av_cond = none v mob2] & [sinus = sr v sb v st v sad] v
(60, 60, 0) [atrial = wp v at v mat] & [av_cond = none v mob2].

[regular_PR = prolonged] =>
(180,180, 0) [av_cond = avb1 v wen] v
(4, 12, 0) [atrial = at] & [av_cond = none v avb1 v wen] & [junction = jeb] v
(4, 12, 0) [atrial = at] & [av_cond = none v avb1 v wen] & [ectvent = veb] v
(2, 6, 0) [atrial = at] & [av_cond = none v avb1 v wen] & [ectvent = none] & [junction = none].

[regular_PR = shortened] =>
(151,151, 0) [atrial = none v wp v at v mat v aeb] & [av_cond = wpw v lgl] v
(24, 24, 0) [av_cond = none] & [junction = jb v jr v jt] v
(10, 27, 0) [atrial = at] & [av_cond = none v wpw v lgl].

[regular_PR = changing] =>
(20, 20, 0) [atrial = wp v mat] & [av_cond = none] & [sinus = none].

[regular_PR = after_QRS_is_P] =>
(51, 51, 0) [atrial = none v aeb] & [sinus = none] & [ventric = none v vr v avr v vt].

Time spent:
1:51:16.300 Sec. CPU
52:43.750 Sec. GC
7:8:8.000 Sec. REAL

```
[regular_QRS = normal] =>
   (345,345,  0)  [av_cond = none v avb1 v wen v mob2 v avb3 v lgl] & [bb_cond = none]
                   & [ventric = none].

[regular_QRS = wide_LBBB] =>
   (396,396,  0)  [av_cond = none v avb1 v wen v mob2 v avb3 v lgl] & [bb_cond = lbbb] v
   (151,202,  0)  [ventric = vr v avr v vt v vfl].

[regular_QRS = wide_LBBB_RBBB] =>
   (202,202,  0)  [ventric = vr v avr v vt v vfl].

[regular_QRS = delta_LBBB] =>
   ( 62, 62,  0)  [atrial = none v wp v at v mat v afl v aeb] & [av_cond = wpw].

[regular_QRS = delta_RBBB] =>
   ( 62, 62,  0)  [atrial = none v wp v at v mat v afl v aeb] & [av_cond = wpw].

[regular_QRS = absent] =>
   (  1,  1,  0)  [sinus = none] & [ventric = vf] v
   (  2,  2,  0)  [atrial = af] & [av_cond = wpw] & [sinus = none].

Time spent:
  12:36.867 Sec. CPU
  6:11.767 Sec. GC
  50:59.000 Sec. REAL
```

```
[rate_of_QRS = under_60]  =>
   ( 70, 70,  0)   [junction = jb] v
   ( 67, 67,  0)   [ventric = vr] v
   (160,160,  0)   [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [sinus = sb v sad] v
   ( 12, 16,  0)   [atrial = afl v af v aeb] & [av_cond = none] & [junction = jb v jeb] v
   (  6,  6,  0)   [atrial = afl v af] & [av_cond = none] & [ectvent = none] & [junction = none] v
   ( 12, 12,  0)   [atrial = afl v af] & [av_cond = none] & [ectvent = veb] v
   (  4,  4,  0)   [atrial = wp] & [av_cond = wen v mob2] & [ectvent = none] & [junction = none] v
   (  8,  8,  0)   [atrial = wp] & [av_cond = wen v mob2] & [junction = jeb] v
   (  8,  8,  0)   [atrial = wp] & [av_cond = wen v mob2] & [ectvent = veb] v
   ( 24, 96,  0)   [av_cond = wen v mob2] & [sinus = sr v sb v sad].

[rate_of_QRS = between_60_100]  =>
   ( 70, 70,  0)   [junction = jr] v
   ( 67, 67,  0)   [ventric = avr] v
   (136,136,  0)   [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [sinus = sr v sad] v
   ( 36, 74,  0)   [atrial = wp v afl v af v aeb] & [junction = jr v jeb] & [ventric = none v avr] v
   ( 48, 48,  0)   [atrial = wp v afl v af] & [av_cond = none v avb1 v wen v mob2 v lgl]
                    & [junction = none] v
   (  2, 34,  0)   [atrial = wp v afl v af] & [av_cond = none v avb1 v wen v mob2 v wpw v lgl]
                    & [ectvent = veb] v
   (  1, 11,  0)   [atrial = wp] & [av_cond = none v avb1 v wen v mob2 v wpw v lgl]
                    & [ectvent = none] & [junction = none] v
   ( 64,132,  0)   [av_cond = wen v mob2] & [sinus = none v sr v st v sad].

[rate_of_QRS = between_100_250]  =>
   ( 70, 70,  0)   [junction = jt] v
   ( 67, 67,  0)   [ventric = vt] v
   (130,130,  0)   [atrial = at v mat v afl v af] & [av_cond = none v avb1 v wen v mob2 v lgl] v
   ( 14,114,  0)   [atrial = at v mat] & [av_cond = none v avb1 v wen v mob2 v wpw v lgl] v
   ( 68, 68,  0)   [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [sinus = st].

[rate_of_QRS = between_250_350]  =>
   (  1,  1,  0)   [ventric = vfl] v
   (  1,  1,  0)   [atrial = afl] & [av_cond = wpw].

[rate_of_QRS = over_350]  =>
   (  1,  1,  0)   [ventric = vf] v
   (  2,  2,  0)   [atrial = af] & [av_cond = wpw].

Time spent:
  7:42:1.350 Sec. CPU
  3:10:18.850 Sec. GC
  35:42:45.000 Sec. REAL
```

```
[ectopic_P = none] =>
   (216,216, 0)  [atrial = none v wp v at v mat v afl v af] & [ectvent = none]
                 & [junction = none v jb v jr v jt].

[ectopic_P = abnormal] =>
   (112,112, 0)  [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [ectvent = veb] v
   (100,100, 0)  [atrial = aeb] & [sinus = sr v sb v st v sad] v
   ( 54, 54, 0)  [atrial = wp v at v mat v afl v af v aeb]
                 & [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [ectvent = none]
                 & [junction = jb v jr v jt v jeb] v
   ( 12, 12, 0)  [av_cond = none] & [junction = jeb] & [sinus = none] & [ventric = vr v avr v vt] v
   ( 12, 12, 0)  [atrial = aeb] & [av_cond = none] & [ventric = vr v avr v vt] v
   ( 60, 60, 0)  [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [junction = jeb]
                 & [sinus = sr v sb v st v sad].

[ectopic_P = absent] =>
   (211,211, 0)  [atrial = none v wp v at v mat v afl v af] & [ectvent = veb] v
   ( 78, 78, 0)  [junction = jeb] & [ventric = vr v avr v vt] v
   ( 60, 60, 0)  [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [junction = jeb]
                 & [sinus = sr v sb v st v sad] v
   ( 42, 42, 0)  [atrial = wp v at v mat v afl v af]
                 & [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [junction = jeb].

Time spent:
  23:44.333 Sec. CPU
  10:21.500 Sec. GC
  1:19:30.000 Sec. REAL
```

[ectopic_PR = none] =>
(216,216, 0) [atrial = none v wp v at v mat v afl v af] & [ectvent = none]
& [junction = none v jb v jr v jt].

[ectopic_PR = meaningless] =>
(211,211, 0) [atrial = none v wp v at v mat v afl v af] & [ectvent = veb] v
(78, 78, 0) [junction = jeb] & [ventric = vr v avr v vt] v
(30, 30, 0) [atrial = aeb] & [junction = jb v jr v jt] & [sinus = sr v sb v st v sad] v
(15, 15, 0) [atrial = aeb] & [sinus = sr v sb v st v sad] & [ventric = vr v avr v vt] v
(60, 60, 0) [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [junction = jeb]
& [sinus = sr v sb v st v sad] v
(42, 42, 0) [atrial = wp v at v mat v afl v af]
& [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [junction = jeb].

[ectopic_PR = normal] =>
(44, 44, 0) [atrial = aeb] & [av_cond = none v mob2].

[ectopic_PR = prolonged] =>
(54, 54, 0) [atrial = aeb] & [av_cond = none v avb1 v wen].

[ectopic_PR = shortened] =>
(80, 80, 0) [av_cond = avb1 v wen v mob2 v wpw v lgl] & [junction = jeb] v
(24, 42, 0) [atrial = wp v at v mat v afl v af v aeb] & [av_cond = none v mob2 v wpw v lgl]
& [ectvent = none] & [junction = jb v jr v jt v jeb] v
(12, 12, 0) [av_cond = none] & [ectvent = none] & [junction = jeb] & [ventric = vr v avr v vt] v
(12, 12, 0) [atrial = aeb] & [av_cond = none] & [ectvent = none] & [ventric = vr v avr v vt] v
(25, 25, 0) [atrial = aeb] & [av_cond = none v wpw v lgl] & [sinus = sr v sb v st v sad] v
(10, 60, 0) [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [junction = jeb]
& [sinus = sr v sb v st v sad].

[ectopic_PR = after_QRS_is_P] =>
(112,112, 0) [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [ectvent = veb] v
(12, 12, 0) [atrial = none] & [av_cond = none] & [junction = jeb] & [ventric = vr v avr v vt] v
(42, 42, 0) [atrial = wp v at v mat v afl v af]
& [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [junction = jeb] v
(60, 60, 0) [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [junction = jeb]
& [sinus = sr v sb v st v sad].

Time spent:
41:49.717 Sec. CPU
19:40.050 Sec. GC
4:41:13.000 Sec. REAL

[ectopic_QRS = none] =>
(216,216, 0) [atrial = none v wp v at v mat v afl v af] & [ectvent = none]
& [junction = none v jb v jr v jt].

[ectopic_QRS = normal] =>
(45, 45, 0) [bb_cond = none] & [ectvent = none] & [junction = jeb] & [ventric = vr v avr v vt] v
(25, 25, 0) [atrial = aeb] & [av_cond = none v avb1 v wen v mob2 v lgl] & [bb_cond = none]
& [ectvent = none] & [junction = none] & [sinus = sr v sb v st v sad] v
(48, 48, 0) [atrial = wp v at v mat v afl v af v aeb]
& [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [bb_cond = none]
& [ectvent = none] & [junction = jb v jr v jt v jeb] v
(60, 60, 0) [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [bb_cond = none]
& [ectvent = none] & [junction = jeb] & [sinus = sr v sb v st v sad] v
(6, 6, 0) [atrial = aeb] & [av_cond = none] & [bb_cond = none] & [ectvent = none]
& [ventric = vr v avr v vt].

[ectopic_QRS = wide_LBBB] =>
(323,323, 0) [atrial = none v wp v at v mat v afl v af] & [ectvent = veb] v
(51, 51, 0) [bb_cond = lbbb] & [ventric = vr v avr v vt] v
(25, 25, 0) [atrial = aeb] & [av_cond = none v avb1 v wen v mob2 v lgl]
& [bb_cond = lbbb] & [sinus = sr v sb v st v sad] v
(48, 48, 0) [atrial = wp v at v mat v afl v af v aeb]
& [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [bb_cond = lbbb]
& [junction = jb v jr v jt v jeb] v
(60, 60, 0) [av_cond = none v avb1 v wen v mob2 v wpw v lgl] & [bb_cond = lbbb]
& [junction = jeb] & [sinus = sr v sb v st v sad].

[ectopic_QRS = wide_LBBB_RBBB] =>
(323,323, 0) [atrial = none v wp v at v mat v afl v af] & [ectvent = veb].

[ectopic_QRS = delta_LBBB] =>
(5, 5, 0) [atrial = aeb] & [av_cond = wpw] & [junction = none].

[ectopic_QRS = delta_RBBB] =>
(5, 5, 0) [atrial = aeb] & [av_cond = wpw] & [junction = none].

[ectopic_QRS = absent] =>
(45, 45, 0) [atrial = aeb] & [av_cond = avb3] & [ectvent = none].

Time spent:
41:5.433 Sec. CPU
20:29.583 Sec. GC
5:42:25.000 Sec. REAL