Syntactic Pattern Recognition of ECG for Diagnostic Justification

P. Flasiński*

IT Systems Department, Jagiellonian University, ul. St. Lojasiewicza 4, Cracow 30-384, Poland

Abstract. A novel hybrid structural-parametric model for ECG diagnostic justification is presented in the paper. In order to distinguish between specific subclasses of heart dysfunction phenomena both grammars and automata are enhanced with a formalism of dynamic programming. It allows one to construct a system, which is feasible for aiding a process of teaching and evaluating medical students' diagnostic reasoning in the area of electrocardiography.

Key words: syntactic pattern recognition, ECG analysis, diagnostic justification.

1. Introduction

Computers have been already aided ECG diagnosing for more than fifty years [1]. In the area of computer ECG analysis both decision-theoretic approach [10, 16, 20, 21, 23, 24, 26, 30, 32, 33] and syntactic pattern recognition methods [3, 6, 7, 11, 13, 15, 17, 25] have been commonly used. In syntactic pattern recognition a pattern is treated as a complex structure, which is decomposed into subpatterns that in turn are decomposed into simpler subpatterns, etc [4, 5, 8, 14]. In cardiology an ECG signal pattern is also treated as a linear structure, which consists of separable substructures describing the different phases of human heart's beating (e.g. P wave, T wave, ST segment, QRS complex). According to the syntactic pattern recognition paradigm a set of various structures is treated as a formal language. Words (structural patterns) of such a language can be analyzed by formal automata [4, 5, 8, 14], which not only are able to identify proper categories (diseases) for patterns, but also can characterize their structural features. Therefore, syntactic pattern recognition seems to be convenient, if a descriptive structural characterization is a goal of ECG analysis rather than only its classification (i.e. assigning an ECG signal to one of classes of heart dysfunction phenomena).

Providing an adequate *diagnostic justification* is a basic skill that is required during medical studies as well as at early stages of a physician professional development [12]. A diagnostic justification consists in explaining how key findings identified by a physician have allowed him/her to formulate initial hypotheses in order to achieve a final diagnosis [34]. Unfortunately, although this skill is crucial for improving diagnostic

*The author has conducted the research in a cooperation with IT Systems Department, Jagiellonian University.

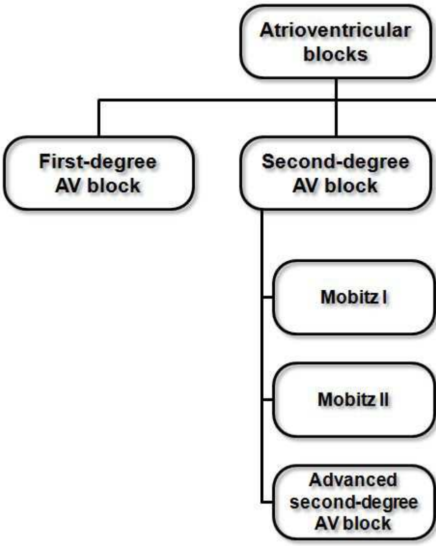

Fig. 1. A class of AV (atrioventricular) blocks and its subclasses.

competencies, medical students frequently exhibit a poor diagnostic justification performance [36]. The skill is especially important, if an interpretation of charts like ECG, EEG is concerned [18, 19]. Such an interpretation is made on the basis of structural features of the charts, as well as parametric values (frequencies of waves, lengths of segments, etc). Therefore, a research into constructing a syntactic pattern recognition-based system for teaching and evaluating students' diagnostic reasoning in the area of electrocardiography has been led since 2012 at the IT Systems Department, Jagiellonian University in Cracow.

Firstly, a set of structural primitives, which is feasible for the purpose of an ECG diagnostic justification has been identified in [35]. Then, a class of programmed attributed regular grammars, PARG has been defined as a tool for generating ECG patterns. System for Teaching ElectroCardioGraphy, STECG has been constructed as a syntax analyzer based on a class of programmed attributed finite-state automata, PAFSA [35]. During the use of the STECG system it has turned out that although the system distinguishes main classes of ECG abnormalities (e.g. various AV (atrioventricular) blocks, various branch blocks) it cannot differentiate between some of their specific subclasses in certain cases (e.g. between *Mobitz I* and *Mobitz II* subclasses of the *Second-degree AV (atrioventricular) block* class – cf. Fig. 1). It results from a too weak generative power of PARG grammars, and in consequence too weak discriminative power of corresponding

PAFSA automata. The results of a research into enhancing a generative/discriminative power of the model are presented in the paper.

In the section 2 preliminary definitions of programmed grammars and automata introduced in [35] are presented and reasons of a necessity of their enhancement are discussed. Enhanced dynamically programmed attributed regular grammars, DPARG are defined in section 3 for the purpose of generating patterns of specific subclasses of AV (atrioventricular) blocks. In section 4 a dynamically programmed attributed finite-state automaton, DPAFSA, which is an enhanced model of PAFSA automaton, is constructed. Its big discriminative power is discussed with the help of an example of recognizing *Mobitz I* and *Mobitz II* subclasses of the *Second-degree AV (atrioventricular) block* class. Concluding remarks concerning the second version of System for Teaching ElectroCardioGraphy, STECG v.2 and its role in teaching and verifying ECG diagnosis justification are contained in the section 5.

2. Preliminaries

As we have mentioned it above, the model proposed in [35] is based on the syntactic pattern recognition paradigm. It means that ECG charts are considered as defined with a set of structural primitives. For example, a subset of primitives used for a description of *Mobitz I* and *Mobitz II* subclasses of the *Second-degree AV block* class, which are discussed in this paper are shown in Fig. 2. On the other hand, during modeling and recognizing ECG charts for a purpose of a diagnostic justification parametric values of structural primitives (frequencies of waves, lengths of segments, etc) should be taken into account. Therefore, we have defined attributed grammars and automata and we have enhanced them by a programming formalism allowing us to make both production applications and transitions in automata conditional on values of primitive parameters.

In the succeeding two subsections (2.1, 2.2) we present definitions of programmed attributed regular grammar, PARG and programmed attributed finite-state automaton, PAFSA, which are characterized in a detailed way in [35].

2.1. Programmed attributed regular grammar

Let us introduce a definition of a programmed attributed regular grammar, PARG [35].

Definition 1. A programmed attributed regular grammar, PARG is a quadruple

$$G = (V, \Sigma, P, S), \quad \textit{where}$$

V *is a finite set of symbols,*
$\Sigma \subset V$ *is a set of terminal symbols,* $N = V \setminus \Sigma$ *is a set of nonterminal symbols,*
P *is a finite set of productions of the form:*

$$(\pi : \quad X \longrightarrow \alpha), \quad \textit{in which}$$

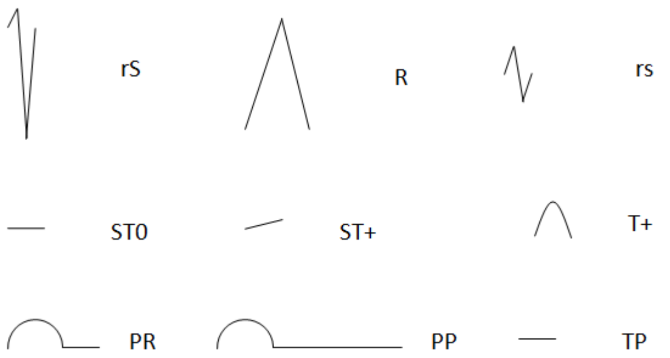

Fig. 2. A set of primitives used for a description of *Mobitz I* and *Mobitz II* subclasses of the *Second-degree AV block* class.

$\pi : \mathcal{A} \longrightarrow \{TRUE, FALSE\}$ *is the predicate of the production applicability,* $\mathcal{A}$ *is a finite set of attributes,* $X \in N$ *,* $\alpha \in \Sigma \cup \Sigma N$*,*
$S \in N$ *is the starting symbol.*

As we have already mentioned it, PARG is strong enough to model/describe structural patterns of general classes of phenomena observed in electrocardiography and their subclasses. It results from the use of the predicate of the production applicability, which tests whether primitive parameters fulfill certain *predefined* conditions.

However, during the use of the STECG system (System for Teaching Electrocardiography), it has turned out that in order to distinguish between specific classes of ECG phenomena, like *Mobitz I* and *Mobitz II* (see Fig. 1), in some cases parameters of primitives should be compared not with predefined constants, but with certain parameters of primitives analyzed previously. Therefore, in section 3 we will define an enhanced PARG with a programming formalism, which enables such comparisons.

2.2. Programmed attributed finite-state automaton

For a language of ECG patterns generated with the help of a programmed attributed regular grammar, a programmed attributed finite-state automaton, PAFSA as a tool for a syntax analysis has been defined in [35].

Definition 2. A programmed attributed finite-state automaton, PAFSA is a quintuple

$$A = (Q, I, \delta, q_0, F), \quad where$$

Q *is a finite nonempty set of states,*
I *is a finite set of input symbols,*
δ *is the transition function of the form:*

$$\delta : Q \times I \times \Pi \longrightarrow Q \ , \ in\ which$$

$\Pi : \mathcal{A} \longrightarrow \{TRUE, FALSE\}$ *is the predicate of the transition permission,* $\mathcal{A}$ *is a set of attributes,*
$q_0 \in Q$ *is the initial state,*
$F \subseteq Q$ *is a set of final states.*

Similarly as in case of a programmed attributed regular grammar, the predicate of the transition permission in PAFSA should be predefined. It means that only a static parametrization of a transition is possible. Therefore, in section 4 we will define an enhanced PAFSA with a programming formalism, which allows one to make transitions in the automaton conditional on comparisons between (changing) parameters of various structural primitives processed till a current step of the automaton analysis.

3. Dynamically programmed attributed regular grammar

In order to show a necessity of introducing a dynamic programming mechanism to our PARG grammar/PAFSA automaton model, let us consider the following example of two specific subclasses of the *Second-degree AV blocks* class, namely: *Mobitz I* and *Mobitz II* (see Fig. 1). Their structural patterns are shown in Figure 3 (a) and (b), respectively. In both classes an occasional lack of a QRS complex occurs as it is shown in Fig. 3. However, whereas in *Mobitz II* a PR segment is of a constant length, in *Mobitz I* the first PR segment after the QRS complex lack is shorter than the last one before this lack. It means that in order to identify *Mobitz I* we should compare both PR segments. As a result, both a grammar and an automaton should remember the length of a previous PR segment and in case of the QRS complex lack the lengths of the corresponding PR segments should be compared.

Let us define a dynamically programmed attributed regular grammar, DPARG in the following way.

Definition 3. A dynamically programmed attributed regular grammar, DPARG is a quadruple

$$G = (V, \Sigma, P, S), \quad \textit{where}$$

V *is a finite set of symbols,*
$\Sigma \subset V$ *is a set of terminal symbols,* $N = V \setminus \Sigma$ *is a set of nonterminal symbols,*
P *is a finite set of productions of the form:*

$$(\pi : \quad X \longrightarrow \alpha, \quad CM), \quad \textit{in which}$$

$\pi : \mathcal{A}_\Sigma \cup \mathcal{V}_A \longrightarrow \{TRUE, FALSE\}$ *is the predicate of the production applicability,* $\mathcal{A}_\Sigma$ *is a finite set of attributes of terminal symbols,* $\mathcal{V}_A$ *is a finite set of auxiliary variables,*
$X \longrightarrow \alpha,\ X \in N\ , \alpha \in \Sigma \cup \Sigma N$ *is called the* core,
$CM : \mathcal{V}_A \longrightarrow VAL_A$ *is the control mapping ascribing values to auxiliary variables,* VAL_A *is a set of admissible values,*
$S \in N$ *is the starting symbol.*

Now, we define DPARG productions for the the subclass *Mobitz I*. We will use

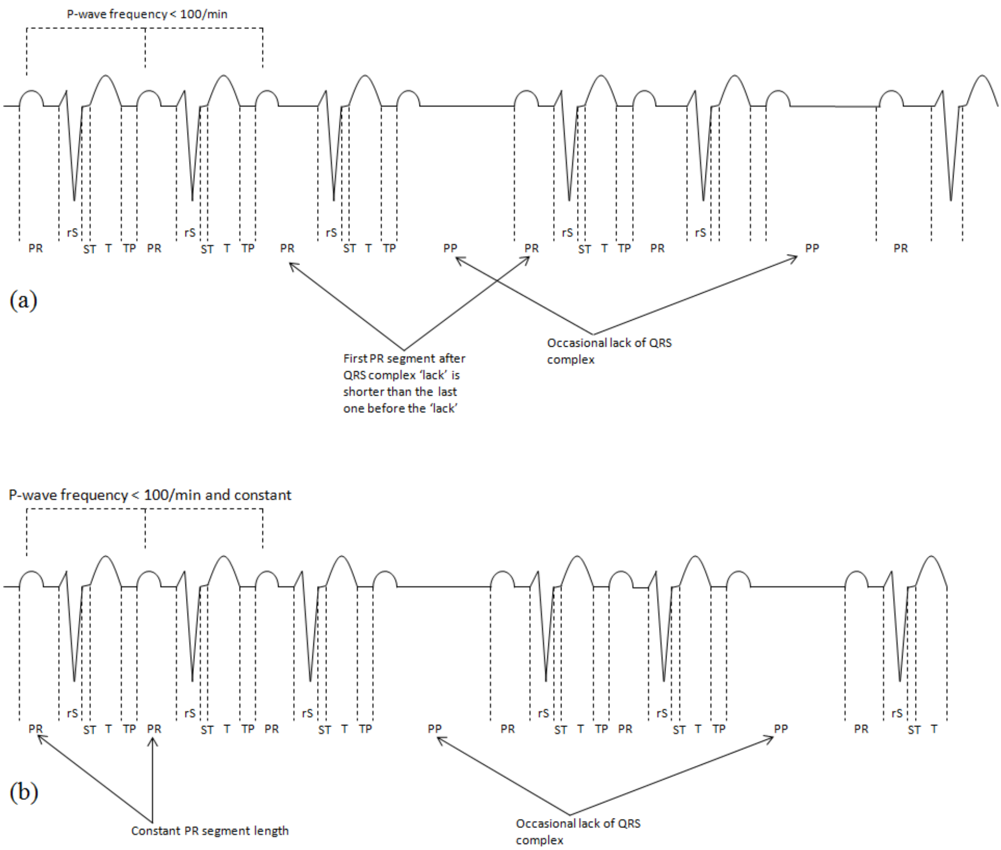

Fig. 3. Second-degree AV blocks of: (a) *Mobitz I* type, (b) *Mobitz II* type.

two auxiliary variables: the Boolean qrs_lack that will be set to $TRUE$ each time a lack of a QRS complex occurs, and the real $potent\_last\_before\_qrs\_lack$ used for storing the length of the last PR segment. We denote the length of a PR segment with l_{PR}. The structural primitives used for defining *Mobitz I* are denoted with: $\mathbf{PR}, \mathbf{rs}, \mathbf{ST+}, \mathbf{T+}, \mathbf{TP}, \mathbf{PP}$ (cf. Figures 2 and 3).

Firstly, the auxiliary variables are set to their initial values:
$qrs\_lack := FALSE$,
$potent\_last\_before\_qrs\_lack := 0$.
The DPARG productions are defined in the following way.

1. $\pi = (qrs\_lack \implies l_{PR} < potent\_last\_before\_qrs\_lack):$
$X^{(0)} \longrightarrow \mathbf{PR}\, X^{(1)}$,
$CM: potent\_last\_before\_qrs\_lack := l_{PR};\ qrs\_lack := FALSE$,
2. $\pi = TRUE: \quad X^{(1)} \longrightarrow \mathbf{rs}\, X^{(2)}, \quad CM: none$,
3. $\pi = TRUE: \quad X^{(2)} \longrightarrow \mathbf{ST+}\, X^{(3)}, \quad CM: none$,
4. $\pi = TRUE: \quad X^{(3)} \longrightarrow \mathbf{T+}\, X^{(4)}, \quad CM: none$,
5. $\pi = TRUE: \quad X^{(4)} \longrightarrow \mathbf{TP}\, X^{(5)}, \quad CM: none$,
6. $\pi = TRUE: \quad X^{(5)} \longrightarrow \mathbf{PP}\, X^{(6)}, \quad CM: qrs\_lack := TRUE$,
7. $\pi = TRUE:$
$X^{(5)} \longrightarrow \mathbf{PR}\, X^{(7)}$,
$CM: potent\_last\_before\_qrs\_lack := l_{PR};\ qrs\_lack := FALSE$,
8. $\pi = (qrs\_lack \implies l_{PR} < potent\_last\_before\_qrs\_lack):$
$X^{(6)} \longrightarrow \mathbf{PR}\, X^{(7)}$,
$CM: potent\_last\_before\_qrs\_lack := l_{PR};\ qrs\_lack := FALSE$,
9. $\pi = TRUE: \quad X^{(7)} \longrightarrow \mathbf{rs}\, X^{(2)}, \quad CM: none$.

As one can easily see, the productions defined above allow us to model *Second-degree AV block* of the *Mobitz I* type (see Fig. 3a).

4. Dynamically programmed attributed finite-state automaton

After defining a dynamically programmed grammar, we can construct a dynamically programmed attributed finite-state automaton, DPAFSA, which is based on the PAFSA [35] presented in section 2.2. In a DPAFSA automaton a mechanism of a transition control is strengthened by adding a set of working memory objects allowing the automaton to store the values of those attributes, which occur in one of the predicates of the transition permission.

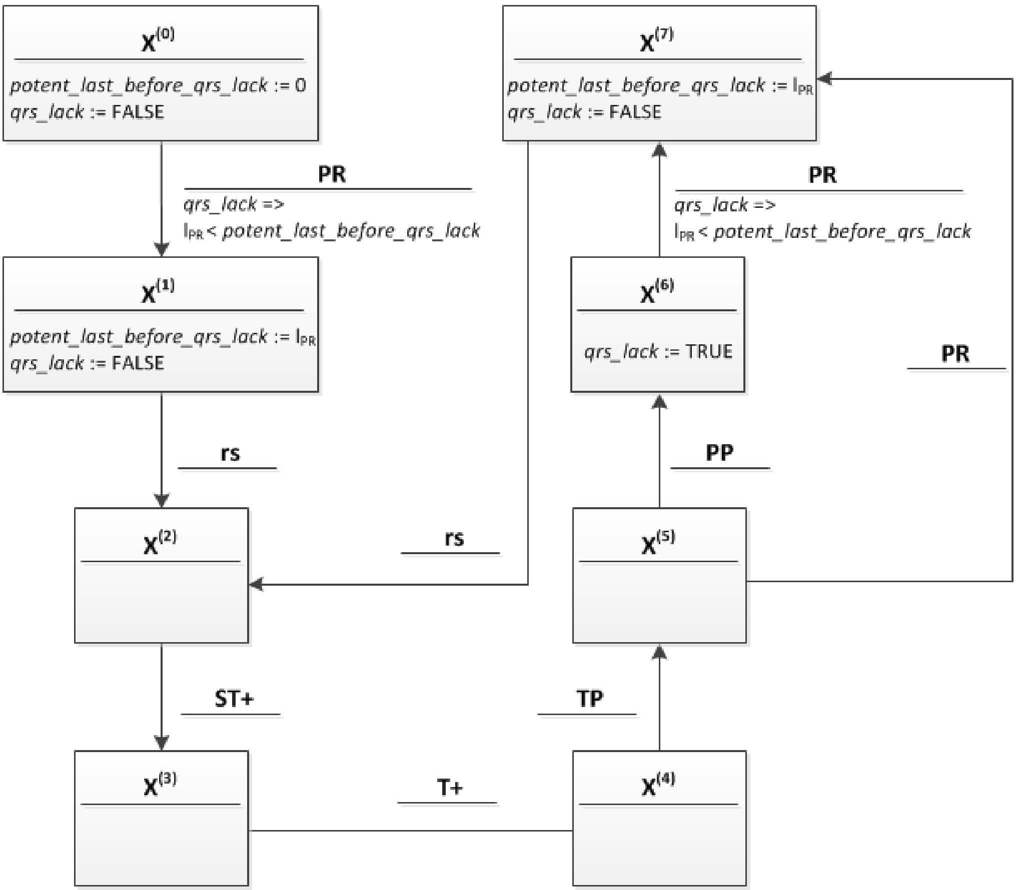

Fig. 4. A part of a dynamically programmed attributed finite-state automaton for a class *Second-degree AV block* of the *Mobitz I* type.

Let us introduce the following definition of a dynamically programmed attributed finite-state automaton, DPAFSA.

Definition 4. A dynamically programmed attributed finite-state automaton, DPAFSA is a sextuple

$$A = (Q, I, M, \delta, q_0, F), \quad \textit{where}$$

Q is a finite nonempty set of states,
I is a finite set of input symbols,
M is a finite set of working memory objects,
δ is the transition function of the form:

$$\delta : Q \times I \times \Pi \longrightarrow Q \times CM, \textit{ in which}$$

$\Pi : \mathcal{A}_I \cup \mathcal{A}_M \longrightarrow \{TRUE, FALSE\}$ is the predicate of the transition permission, $\mathcal{A}_I$ is a set of attributes of input symbols, $\mathcal{A}_M$ is a set of attributes of working memory objects, $CM = \{cm_{q_i,q_j} : \mathcal{A}_M \longrightarrow VAL_M, q_i, q_j \in Q\}$ is a set of control mappings ascribing values to attributes of working memory objects, VAL_M is a set of admissible values,
$q_0 \in Q$ is the initial state,
$F \subseteq Q$ is a set of final states.

An example of a part of a dynamically programmed attributed finite-state automaton for a class *Second-degree AV block* of the *Mobitz I* type is shown in Fig. 4. As we can see there is a one-to-one correspondence between transitions of the DPAFSA automaton and the productions of the DPARG grammar defined in section 3.

5. Conclusions

As we have mentioned it in the introduction, providing an adequate *diagnostic justification* is a basic skill that is required during medical studies and first years of a physician professional development. On the other hand, medical students and novice physicians frequently exhibit a poor performance in this area. There are two types of information processing in medicine: analytic and automatic [31]. During automatic processing a diagnostic justification is made with the help of a pattern recognition-like mechanism, i.e. an unknown case to be diagnosed is unconsciously compared with known cases from the past [31]. Thus, this pattern recognition-like mechanism is analogous to (standard) pattern recognition scheme used in computer science. This kind of achieving a diagnosis is typical for experienced physicians. On the other hand, analytic processing consists in analyzing, synthesizing and interpreting the case itself on the basis on a biomedical knowledge (not the clinical one) [31]. This kind of diagnosing is, thus, based on general models learnt during a medical education and it is typical for students and novice physicians. In case of an interpretation of ECG, a structure of a chart and its generic parametrization is analyzed during a diagnosing process. It in turn is analogous to syntactic pattern recognition scheme in computer science. Therefore, this approach seems

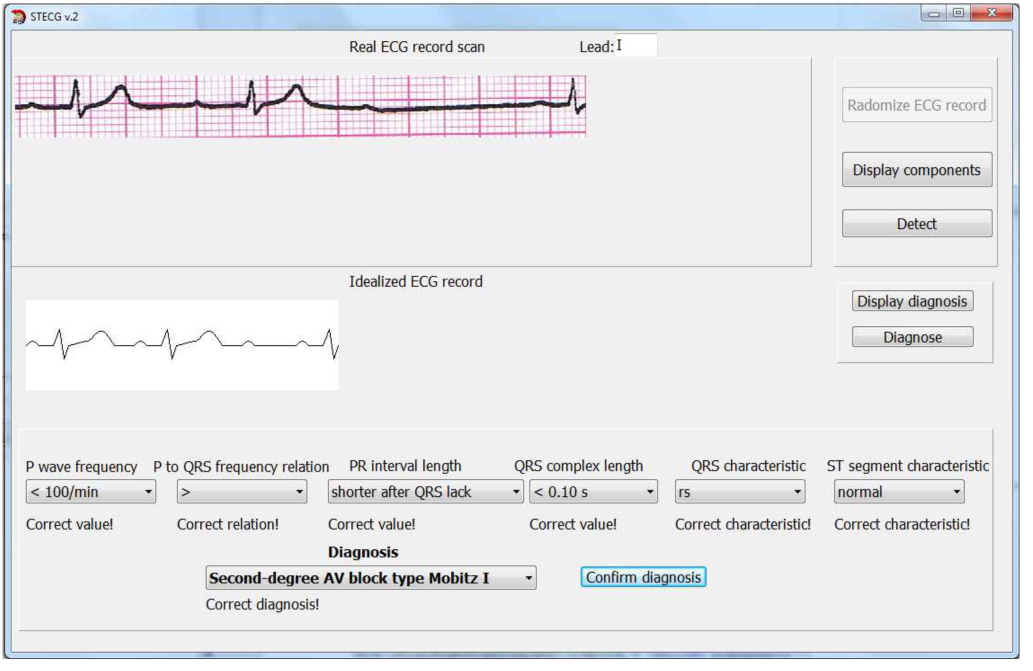

Fig. 5. An example of testing a diagnostic justification of *Second-degree AV (atrioventricular) block* of the *Mobitz I* type by the STECG v.2 system.

to be more feasible for designing systems for aiding medical students in achieving the ECG diagnostic skill *via* testing their diagnostic justification.

The System for Teaching ElectroCardioGraphy, STECG has been implemented on the basis of programmed attributed regular grammars and programmed attributed finite-state automata [35]. During its use it has turned out that although STECG distinguishes main classes of ECG abnormalities, some specific ECG types, like *Mobitz I* and *Mobitz II* are hardly distinguishable, because of a too weak discriminative power of the automaton mentioned above and used as a syntax analyzer. Therefore, the formal model introduced in [35] had to be enhanced. The dynamically programmed grammars and automata presented in this paper have helped to solve this problem.

The STECG v.2 system allows us to analyze ECG signals, in which current and previous values of primitives' parameters are to be compared. This way the system is able to differentiate between *Mobitz I* and *Mobitz II* types of *Second-degree atrioventricular blocks*. An example of testing a medical student diagnostic justification for *Mobitz I* is shown in Fig. 5.

References

1961

[1] Stallmann F. W., Pipberger H. V.: Automatic recognition of electrocardiographic waves by digital computer. Circ. Res., 9:1138-1143.

1969

[2] Rosenkrantz D. J.: Programmed grammars and classes of formal languages. Journal of the Association for Computing Machinery, 16:107-131.

1975

[3] Horowitz S. L.: A syntactic algorithm for peak detection in waveforms with applications to cardiography. Comm. ACM, 18:281-285.

1977

[4] Pavlidis T.: Structural Pattern Recognition. Springer, New York.

1978

[5] Gonzales R. C., Thomason M. G.: Syntactic Pattern Recognition: An Introduction. Addison-Wesley, Reading.

1979

[6] Belferte G., De Mori R., Ferraris F.: A contribution to the automatic processing of electrocardiograms using syntactic methods. IEEE Trans. Biomed. Eng., 26:125-136.

1980

[7] Udupa J., Murthy I. S. N.: Syntactic approach to ECG rhythm analysis. IEEE Trans. Biomed. Eng., 27:370-375.

1982

[8] Fu K. S.: Syntactic Pattern Recognition and Applications. Prentice Hall, Englewood Cliffs.

1983

[9] Bunke H. O.: Graph grammars as a generative tool in image understanding. Lecture Notes in Computer Science, 153:8-19.

1985

[10] Pan J., Tompkins W.J.: A real-time QRS detection algorithm. IEEE Trans. Biomed. Eng., 32:230-236.

1986

[11] Papakonstantinou G., Skordalakis E., Gritzali F.: An attribute grammar for QRS detection. Pattern Recognition, 19:297-303.

[12] Patel V., Groen G.: Knowledge based solution strategies in medical reasoning. Cognitive Science, 10:91-116.

[13] Skordalakis E.: Syntactic ECG processing: a review. Pattern Recognition, 19:305-313.

1990

[14] Bunke H. O., Sanfeliu A. (eds.) : Syntactic and Structural Pattern Recognition - Theory and Applications. World Scientific, Singapore.

[15] Trahanias P., Skordalakis E. : Syntactic pattern recognition of the ECG. IEEE Trans. Patt. Analysis Mach. Intell., 12:648-657.

1991

[16] Piętka E. : Feature extraction in computerized approach to the ECG analysis. Pattern Recognition, 24:139-146.

1995

[17] Koski A., Juhola M., Meriste M. : Syntactic recognition of ECG signals by attributed finite automata. Pattern Recognition, 28:1927-1940.

1997

[18] Gilhooly K. J., McGeorge P., Hunter J. et al. : Biomedical knowledge in diagnostic thinking: the case of electrocardiogram (ECG) interpretation. European Journal of Cognitive Psychology, 9:199-223.

[19] Kolykhalov I. V., Iznak, A. F., Chayanov N. V. et al. : EEG mapping in diagnostic assessment of patients with mild dementia. European Neuropsychopharmacology, 7:248.

1998

[20] Barro S., Fernandez-Delgado M., Villa-Sobrino J. A., Regueiro C. V., Sanchez E. : Classifying multichannel ECG patterns with an adaptive neural network. IEEE Eng. Med. Biol. Mag., 17:45-55.

[21] Maglaveras N., Stamkopoulos T., Diamantaras K., Pappas C., Strintzis M. : ECG pattern recognition and classification using non-linear transforms and neural networks: a review. Int. J. Med. Inform., 52:191-208.

1999

[22] Flasiński M., Jurek J. : Dynamically programmed automata for quasi context sensitive languages as a tool for inference support in pattern recognition-based real-time control expert systems. Pattern Recognition, 32:671-690.

2001

[23] Osowski S., Linh T.H. : ECG beat recognition using fuzzy hybrid neural network. IEEE Trans. Biomed. Eng., 48:1265-1271.

2002

[24] Sternickel K. : Automatic pattern recognition in ECG time series. Comp. Meth. Programs in Biomedicine, 68:109-115.

2003

[25] Tumer M. B., Belfore L. A., Ropella K. M. : A syntactic methodology for automatic diagnosis by analysis of continuous time measurements using hierarchical signal representations. IEEE Trans. on Syst. Man Cybern., 33:951-965.

2004

[26] Engin M. : ECG beat classification using neuro-fuzzy network. Patt. Rec. Lett., 25:1715-1722.

[27] Flasiński M., Reroń E., Wójtowicz P., Atlasiewicz K. : Mathematical linguistics model for medical diagnostics of organ hearing in neonates, Lecture Notes in Computer Science (LNAI), 3019:746-753.

[28] Tadeusiewicz R., Ogiela M. R. : Medical Image Understanding Technology. Springer, Berlin-Heidelberg-New York.

2006

[29] Flasiński M., Jurek J. : On the analysis of fuzzy string patterns with the help of extended and stochastic GDPLL(k) grammars. Fundamenta Informaticae, 71:1-14.

2007

[30] Dong J., Xu S., Zhan C. : ECG recognition and classification: approaches, problems and new method. J. Biomed. Eng., 24:1224-1229.

[31] McLaughlin K. J. : The Contribution of Analytic Information Processing to Diagnostic Performance in Medicine, Ph.D. Thesis, Erasmus University Rotterdam.

2009

[32] Martis R. J., Chakraborty C., Ray A. K. : A two-stage mechanism for registration and classification of ECG using Gaussian mixture model. Pattern Recognition, 42:2979-2988.

[33] Noponen K., Kortelainen J., Seppanen T. : Invariant trajectory classification of dynamical systems with a case study on ECG. Pattern Recognition, 42:1832-1844.

2013

[34] Cianciolo A. T., Williams R. G., Klamen D. L., Roberts N.K. : Biomedical knowledge, clinical cognition and diagnostic justification: a structural equation model. Medical Education, 47:309-316.

[35] Flasiński M., Flasiński P., Konduracka E. : On the use of programmed automata for verification of ECG diagnoses. Advances in Intelligent Systems and Computing 226, Springer, Berlin-Heidelberg-New York, 591-599.

2014

[36] Williams R. G., Klamen D. L., Markwell S. J. et al : Variations in senior medical student diagnostic justification ability. Academic Medicine, 89:790-798.