ARTICLE IN PRESS

Applied Mathematics and Computation xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A portable knowledge-based system for car breakdown evaluation

Eugenio Roanes-Lozano [a], José Luis Galán-García [b,*], Gabriel Aguilera-Venegas [b]

[a] Instituto de Matemática Interdisciplinar & Depto. de Algebra, Universidad Complutense de Madrid, Spain
[b] Depto. de Matemática Aplicada, Universidad de Málaga, Spain

ARTICLE INFO

Keywords:
Knowledge-based systems
Groebner bases
Computer algebra
Breakdown diagnosis

ABSTRACT

Modern cars have many dashboard lights and not all drivers recognize or know the importance of all of them. Red symbols usually indicate a safety issue or a serious problem, meanwhile yellow symbols use to indicate a not so urgent problem. Green and blue symbols usually provide information about the systems connected. But not all the red icons require of the same action, and the *user manuals* of most modern cars, with their sophisticated electronic systems, have hundreds of pages. Meanwhile, smart devices have become popular and have an outstanding computing power plus Internet connection. Consequently, the conditions for developing a knowledge-based system that helped the unaware driver in case a dashboard light went on, exist. The application should evaluate the situation and recommend the best actions to be carried out by the driver (regarding the possible repair on site or at a workshop, its urgency or even the need to immobilize the vehicle immediately). We have designed such a knowledge-based system and have developed a simplified one as example. A friendly Graphical User Interface has been developed in order to ease the communication with the application and its use in remote using any smart device with Internet connection.

1. Introduction

This work arises from a real fact: some time ago an aunt of one of the authors kept on driving to the next gas station (to ask for advice) when the *oil pressure* dashboard light of her car went on. As a consequence of this decision, the engine was ruined. The importance of this dashboard light was not correctly evaluated!

Modern cars have many dashboard lights and not all drivers recognize or know the importance of all of them. Applications for smartphones, that help in recognizing the icons are already available [1].

Red symbols usually indicate a safety issue or a serious problem, meanwhile yellow symbols use to indicate a not so urgent problem. Green and blue symbols usually provide information about the systems connected. But not all the red icons require of the same action (for instance, to *stop the car immediately* is not always required, although it can be the best option sometimes). We should take into account that the *user manuals* of most modern cars, with their sophisticated electronic systems, have hundreds of pages, and their terminology is not always mastered by the driver.

* Corresponding author.
E-mail addresses: eroanes@mat.ucm.es (E. Roanes-Lozano), jlgalan@uma.es (J.L. Galán-García), gabri@ctima.uma.es (G. Aguilera-Venegas).

http://dx.doi.org/10.1016/j.amc.2014.12.001
0096-3003/

Please cite this article in press as: E. Roanes-Lozano et al., A portable knowledge-based system for car breakdown evaluation, Appl. Math. Comput. (2014), http://dx.doi.org/10.1016/j.amc.2014.12.001

We had previously developed knowledge-based systems (KBS), whose underlying logic was Boolean or many-valued modal, for different tasks such as managing medical appropriateness criteria [2] and the early detection of illnesses [3–5]. In all cases we used computational techniques borrowed from computer algebra (Groebner bases) [6,7].

Meanwhile, smart devices have become popular and have an outstanding computing power plus Internet connection. Consequently, the conditions for developing a KBS that helped the unaware driver in case a dashboard light went on, exist.

The application should evaluate the situation and recommend the best actions to be carried out (regarding the possible repair on site or at a workshop, its urgency or even the need to immobilize the vehicle immediately).

We have designed and developed such a KBS. It returns explanations about the repairs to be made such as *substitute brake pads*, and recommendations such as

$$drive\ carefully\ and\ smoothly \wedge increase\ safety\ distance\ (breaking\ distance\ possibly\ increased)$$

and suggestions about where to have the car to be repaired such as have the car repaired at a workshop $\vee$ DIY repair at home

The underlying idea of this work, together with others' cars diagnosis research, are stated in Section 2. The KBS has been structured in three subsystems (Section 3). The inference engine is based on the use of Groebner bases of polynomial ideals (Section 4). An example of use of the KBS is described in Section 5. In order to compute the Groebner basis, the Computer Algebra System (CAS) CoCoA has been used (Section 6). To ease the communication between the user and the application, a friendly Graphical User Interface (GUI) has been developed (Section 7). This GUI allows to execute the application in remote with any smart device with Internet connection. It also makes transparent to the user both: the use of CoCoA and the management of the different subsystems of the KBS. In Section 8 a brief study of execution times is presented. Finally, acknowledgments and conclusions are stated in Sections 9 and 9, respectively.

2. About the underlying idea, the way it has been developed and others' cars diagnosis research

Let us emphasize that this KBS is not inspired by repair manuals such as the famous *Haynes* series [8], devoted to expert users or auto mechanics, that can dive into the complexities of cars repairs. It is a decision making tool for a driver that suddenly faces a problem when driving, reported by a dashboard light. What it proposes is an *ad hoc* on site acting protocol instead (that can, for instance, suggest a way to keep on driving or alert that the car should be immobilized immediately). As it is neither intended for DIY work at home nor for workshop repairs, it should be accessible from smart mobile devices.

There are several KBS devoted to the problem of finding out the reasons for a malfunction and how to proceed, more in the line of a repair manual for an amateur or professional workshop. Examples can be found in [9–14]. But we do not know of any comparable KBS.

Let us remark that there are many approaches to perform effective computations in logics and KBS. We have chosen a Groebner bases based because of our experience with this approach, its very reasonable performance and, mainly, because of the possibility to choose different logics when designing the KBS (for instance, although this prototype uses Boolean logic, it would be straightforward to change the underlying logic to, for instance, Kleene's or Łukasiewicz's three-valued logic, if desired).

No changes in the rules would be required, only the code of the algebraic inference engine had to be changed, if the logic was changed. This is not the case in other approaches, such as the KBS development environment CLIPS [15] (this is the approach used, for instance, in [9]). When we started developing the rules, we were not sure if it would be better to work with a modal three-valued logic or if a simpler Boolean logic could be used.

The approach followed presents no operating system compatibility disadvantage w.r.t. other KBS development approaches, as it is designed as an Internet service (computations take place at a server).

We finally preferred to choose Boolean logic as the underlying logic because of its much better performance compared to many-valued modal logics (for instance, using the algebraic approach detailed below, the polynomial translation of the logical connectives grow significantly in the many-valued modal, case) as we desired to receive answers almost instantly. This choice could be made since:

- if the driver cannot answer a question, the worst case is considered,
- if we are not sure about the diagnosis, a "take the car to a specialist for further investigation" recommendation is included.

Nevertheless, if in a future enhancement of the prototype presented here, a many-valued logic was preferred, for instance, for one of the subsystems, it would be straightforward to adapt the whole KBS. This is not possible in all approaches to KBS development.

3. KBS structure

The process begins with a dashboard light on (or gauge out of margins). The KBS is divided into three subsystems (what is also good for the performance of the whole KBS):

Please cite this article in press as: E. Roanes-Lozano et al., A portable knowledge-based system for car breakdown evaluation, Appl. Math. Comput. (2014), http://dx.doi.org/10.1016/j.amc.2014.12.001

- First subsystem: tests to be made on site: the system asks the driver to perform some tests *under the hood* (if necessary), perhaps two times.
- Second subsystem: repairs to be made on site, driving style and precautions:
  - determine if there is really a breakdown, and, in the affirmative case, guide the driver to a provisional or definitive repair on site, or conclude that he/she should immobilize the vehicle and call a tow truck,
  - suggest the way of driving, and
  - (perhaps) add some observations.
- Third subsystem: definitive repairs (after the on site repair): determine what has to be repaired at a workshop (or possible DIY) when back (if applies).

The flow diagram of the whole system is shown in Fig. 1.

Remark 1. The DIY repair at home is included in case it is not possible to definitely repair the vehicle on site because at home there could be more tools, fluids, etc. available, or an expert in mechanics could help there.

Remark 2. The DIY repair at home is a possibility, that can obviously be substituted by taking the car to the workshop (but not the other way round).

Remark 3. Whether to repair the car at home (DIY) or to take it to a workshop is many times a personal decision. Someone can prefer the wiper fluid to be refilled at a workshop and someone else can have the appropriate tools and can consider substituting the brake pads a funny DIY activity for the weekend. Other repairs (like those involving ASR or ESP electronics) require of equipments only available at a workshop or even at an official service.

4. The algebraic approach to KBS

This introductory section is included in order the article to be self-contained and can be skipped by an acquainted reader. It is a summary and the interested reader can find the details in [16–18].

The inference engine is based on a mathematical result, that translates the problem of determining whether a propositional formula may be inferred from others or not, into a computer algebra problem.

Let $\vee, \wedge, \neg, \rightarrow$ denote the logic disjunction, conjunction, negation and implication, respectively. Let $(\mathcal{C}, \vee, \wedge, \neg, \rightarrow)$ be the Boolean algebra of the propositions that can be constructed using a finite number of propositional variables $P, Q, \ldots, R$. Let us consider the Boolean algebra $(\mathcal{A}, \dot{+}, \cdot, 1+,$ "is a multiple"), where $\mathcal{A}$ is the residue class ring

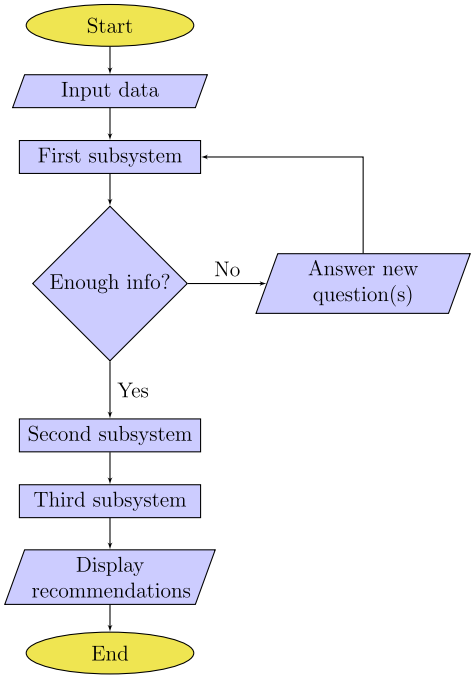

Fig. 1. Flow diagram of the whole system.

Please cite this article in press as: E. Roanes-Lozano et al., A portable knowledge-based system for car breakdown evaluation, Appl. Math. Comput. (2014), http://dx.doi.org/10.1016/j.amc.2014.12.001

$$\mathcal{A} = \mathbb{Z}_2[p, q, \ldots, r]/\langle p^2 - p, q^2 - q, \ldots, r^2 - r\rangle$$

($\langle p^2 - p, q^2 - q, \ldots, r^2 - r\rangle$ denotes the polynomial ideal generated by $p^2 - p, q^2 - q, \ldots, r^2 - r$). Let us define:

φ: $(\mathcal{C}, \vee, \wedge, \neg, \rightarrow) \longrightarrow (\mathcal{A}, \tilde{+}, \cdot, 1+$,"is a multiple").

The following way; for propositional variables:

$$P \longrightarrow p,$$
$$Q \longrightarrow q,$$
$$\ldots\ldots\ldots\ldots$$
$$R \longrightarrow r$$

and for any $A, B \in \mathcal{C}$

$$A \vee B \longrightarrow a \tilde{+} b,$$
$$\neg A \longrightarrow 1 + a,$$

where $a \tilde{+} b = a + b - a \cdot b = a + b + a \cdot b$. Then, as an immediate consequence of the De Morgan laws:

$$A \wedge B \longrightarrow a \cdot b.$$

This correspondence turns out to be a Boolean algebra isomorphism. Moreover, if $\vee$ is substituted by *xor* and $\tilde{+}$ by $+$, a (Boolean) ring isomorphism is obtained.

The main result is Theorem 1, translating the problem of checking whether a propositional formula is a *tautological consequence of* (i.e., *can be inferred from*) others or not, by a polynomial ideal membership:

Theorem 1. *A propositional formula α is a tautological consequence of a set of formulae $\{\beta_1, \ldots, \beta_m\}$, if and only if*

$$\varphi(\neg\alpha) \in \langle \varphi(\neg\beta_1), \ldots, \varphi(\neg\beta_m)\rangle.$$

Moreover, the logical inconsistency of the KBS (i.e., what happens when a statement turns out to be true and false at the same time) is translated in the polynomial model in the degeneracy of the ring into a ring with only one element (that is, a ring where $0 = 1$). This can be checked with the computer by calculating whether a certain Groebner basis is $\{1\}$ or not.

The first works in this approach to logic are [19,20] (Boolean case) and [21,22] (many-valued propositional logics with a prime number of truth values). An algebraic model for these logics can be found in [18,23], and its extension to KBS can be found in [23,24].

The algebraic computations are performed in a CAS using commands *Groebner basis* (of a polynomial ideal) and *Normal Form* (of a polynomial modulo an ideal: the residue of the polynomial modulo the ideal). Bruno Buchberger developed a theory and an algorithm for finding specific basis of ideals, which he called *Groebner basis* (named after his PhD advisor), that are unique for each polynomial ideal (once the order for the variables is fixed and which monomial ordering is to be used is chosen) [6].

A very important application is the solution of the *ideal membership problem*: if g is a polynomial and L is an ideal:

$$g \in L \text{ if and only if } \mathrm{NF}(g, L) = 0$$

(where NF denotes *Normal Form*).

5. Example of KBS

This section illustrates the possibilities of the approach with a simple example of KBS. As the existent dashboard lights and their symbols vary slightly with the mark and model, the KBS would ideally be tailor-made for the precise mark, model and year of the vehicle.

We would like to remark that this is a first prototype showing the possibilities of such an KBS, not a final version after years of development and refinement.

5.1. First set of potential facts: dashboard lights

The dashboard lights considered (potential facts) are:

d_1 low brake fluid level
d_2 brake pads worn out
d_3 ESP failure
d_4 ASR/TCS failure
d_5 ABS failure
d_6 park brake engaged
d_7 Airbag/SRS failure
d_8 low coolant level

Please cite this article in press as: E. Roanes-Lozano et al., A portable knowledge-based system for car breakdown evaluation, Appl. Math. Comput. (2014), http://dx.doi.org/10.1016/j.amc.2014.12.001

d_9 high coolant temperature
d_{10} low oil pressure
d_{11} low oil level
d_{12} engine failure
d_{13} low timing belt tension
d_{14} servicing required
d_{15} battery charge failure
d_{16} bulb failure
d_{17} low fuel level
d_{18} door open
d_{19} seat belt not fasten
d_{20} low wiper fluid
d_{21} high beam
d_{22} glow plug (Diesel engine only, dashboard light on for a few seconds after starting the engine)
d_{23} Diesel particulates filter (Diesel engine only)
d_{24} low tire pressure
d_{25} extremely low tire pressure.

5.2. Other propositional variables: repairs to be made

The explanations of the repairs to be made are:

$x_{1,1}$ refill brake fluid
$x_{1,2}$ repair brake circuit leak or servo brake booster leak
$x_{2,1}$ substitute brake pads
$x_{3,1}$ repair ESP (Electronic Stability Program)
$x_{4,1}$ repair ASR/TCS (Automatic Skid Reduction system/Traction Control System)
$x_{5,1}$ repair ABS (Anti-locking Brake System)
$x_{6,1}$ release handbrake
$x_{7,1}$ repair airbag/SRS (Supplemental Restraint System)
$x_{8,1}$ refill coolant (wait for engine cold)
$x_{8,2}$ repair coolant leak
$x_{9,1}$ substitute servo fan fuse
$x_{9,2}$ repair servo fan
$x_{10,1}$ repair engine
$x_{11,1}$ refill oil
$x_{13,1}$ tighten timing belt
$x_{14,1}$ service the vehicle
$x_{15,1}$ repair alternator or wires to battery
$x_{16,1}$ substitute lamp's fuse
$x_{16,2}$ substitute lamp
$x_{17,1}$ refill the tank
$x_{18,1}$ close door(s)
$x_{19,1}$ fasten seat belt
$x_{20,1}$ refill wiper fluid
$x_{21,1}$ switch to low beam if necessary
$x_{22,1}$ nothing to be done, perfectly normal
$x_{23,1}$ substitute Diesel particulates filter (Diesel engine)
$x_{24,1}$ inflate tire when possible
$x_{25,1}$ substitute wheel
$x_{26,1}$ take the car to a specialist for further investigation.

5.3. First subsystem: tests to be made on site

This first subsystem is run twice in order to obtain feedback from the driver. The tests that the driver could be asked to perform on site are (observe that more propositional variables are introduced):

$y_{1,1}$ test if any important brake fluid leak in the brake circuit
$y_{1,2}$ test if any important brake fluid leak in the servo brake booster
$y_{8,1}$ test if any important coolant leak in the cooling system

Please cite this article in press as: E. Roanes-Lozano et al., A portable knowledge-based system for car breakdown evaluation, Appl. Math. Comput. (2014), http://dx.doi.org/10.1016/j.amc.2014.12.001

$y_{9,1}$ test if servo fan off (ignition key in position *on*)
$y_{9,2}$ test if servo fan's fuse is blown
$y_{16,1}$ test if lamp's fuse is blown
$y_{16,2}$ test lamp

Note that the driver will asked to perform only some (or none) of these tests, according to the data introduced to the system. The possible answers to the previous tests are:

$z_{1,1}$ there is an important brake fluid leak in the brake circuit

$z_{1,2}$ there is an important brake fluid leak in the servo brake booster
$z_{8,1}$ there is an important coolant leak in the cooling system
$z_{9,1}$ servo fan not working
$z_{9,2}$ servo fan's fuse is blown
$z_{16,1}$ lamp's fuse blown
$z_{16,2}$ lamp burned out.
and their negations.

If the driver cannot make the test for some reason, the answer regarding the failure must be affirmative (worst-case considered), for example $z_{1,1}$, that is, all $z_{i,j}$ must be affirmed or negated. Observe that $y_{9,2}$ and $y_{16,2}$ correspond to the second knowledge extraction using the first subsystem (see below).

Rules of the first subsystem (first knowledge extraction):

R101: $d_1 \rightarrow y_{1,1} \wedge y_{1,2}$
R102: $d_8 \rightarrow y_{8,1}$
R103: $d_9 \rightarrow y_{9,1}$
R104: $d_{16} \rightarrow y_{16,1}$

For instance, rule R103 says:
IF "high coolant temperature" (d_9) THEN "test if servo fan off" ($y_{9,1}$).

Rules of the first subsystem (second knowledge extraction):

R105: $z_{9,1} \rightarrow y_{9,2}$
R106: $\neg z_{16,1} \rightarrow y_{16,2}$
R107: $\neg z_{9,1} \rightarrow y_{8,1}$

For instance, rule R105 says:

IF "servo fan not working" ($z_{9,1}$)

THEN "test if servo fan's fuse is blown" ($y_{9,2}$)

(note that $z_{9,1}$ and $\neg z_{9,1}$ are the possible answers to the test $y_{9,1}$).

5.4. Second subsystem: repairs to be made on site, driving style and precautions

Only some repairs can be made on site. These are the ones that appear in this subsystem.
The recommended driving styles are:

c_1 keep on driving normally
c_2 drive carefully and smoothly
c_3 absolutely do not keep on driving (call a tow truck).

Cautions to observe: while driving is still possible, in some cases care should be extreme regarding particular issues:

o_1 keep observing the *low brake fluid level* dashboard light
o_2 increase safety distance (breaking distance possibly increased)
o_3 keep observing the *low coolant level* dashboard light
o_4 keep observing the *high coolant temperature* dashboard light/gauge
o_5 the autonomy will be very restricted (the battery will be completely discharged soon)
o_6 keep observing the *tire pressure* dashboard light/gauge.

Please cite this article in press as: E. Roanes-Lozano et al., A portable knowledge-based system for car breakdown evaluation, Appl. Math. Comput. (2014), http://dx.doi.org/10.1016/j.amc.2014.12.001

Rules of the second subsystem:

R201: $d_1 \wedge \neg z_{1,1} \wedge \neg z_{1,2} \rightarrow x_{1,1} \wedge c_2 \wedge o_1 \wedge o_2$
R202: $d_1 \wedge (z_{1,1} \vee z_{1,2}) \rightarrow c_3$
R203: $d_2 \rightarrow c_2 \wedge o_2$
R204: $d_3 \rightarrow c_2$
R205: $d_4 \rightarrow c_2$
R206: $d_5 \rightarrow c_2$
R207: $d_6 \rightarrow x_{6,1} \wedge c_1$
R208: $d_7 \rightarrow c_2$
R209: $d_8 \wedge \neg z_{8,1} \rightarrow x_{8,1} \wedge c_2 \wedge o_3$
R210: $d_8 \wedge z_{8,1} \rightarrow c_3$
R211: $d_9 \wedge z_{9,2} \rightarrow x_{9,1} \wedge c_2 \wedge o_4$
R212: $d_9 \wedge \neg d_8 \wedge \neg z_{9,2} \rightarrow c_3$
R213: $d_{10} \rightarrow c_3$
R214: $d_{11} \rightarrow x_{11,1} \wedge c_1$
R215: $d_{12} \rightarrow c_3$
R216: $d_{13} \rightarrow c_2$
R217: $d_{14} \rightarrow c_1$
R218: $d_{15} \rightarrow c_2 \wedge o_5$
R219: $d_{16} \wedge z_{16,1} \rightarrow x_{16,1} \wedge c_1$
R220: $d_{16} \wedge z_{16,2} \rightarrow x_{16,2} \wedge c_1$
R221: $d_{16} \wedge \neg z_{16,1} \wedge \neg z_{16,2} \rightarrow c_2$
R222: $d_{17} \rightarrow x_{17,1} \wedge c_1$
R223: $d_{18} \rightarrow x_{18,1} \wedge c_1$
R224: $d_{19} \rightarrow x_{19,1} \wedge c_1$
R225: $d_{20} \rightarrow x_{20,1} \wedge c_1$
R226: $d_{21} \rightarrow x_{21,1} \wedge c_1$
R227: $d_{22} \rightarrow c_1$
R228: $d_{23} \rightarrow c_1$
R229: $d_{24} \rightarrow x_{24,1} \wedge c_2 \wedge o_6$
R230: $d_{25} \rightarrow x_{25,1} \wedge c_1$
R231: $d_9 \wedge \neg z_{9,1} \rightarrow c_3$

For instance, rule R211 says:
IF "high coolant temperature" (d_9) AND "servo fan's fuse is blown" ($z_{9,2}$).

THEN "substitute servo fan fuse" ($x_{9,1}$)

AND "drive carefully and smoothly" (c_2)

AND "keep observing the high coolant level dashboard light/gauge" (o_4). meanwhile (its complementary) rule R212 says:

IF "high coolant temperature" (d_9)

AND NOT "low coolant level" ($\neg d_8$)

AND NOT "servo fan's fuse is blown" ($\neg z_{9,2}$)

THEN "absolutely do not keep on driving (call a tow truck)" (c_3).
(note that $z_{9,2}$ and $\neg z_{9,2}$ are the two possible answers to the test $y_{9,2}$).

5.5. Third subsystem: definitive repairs (after the on site repair)

The possible conclusions of the third subsystem are:

g_0 no other repair required,
g_1 have the car repaired at a workshop,
g_2 DIY repair at home.

Rules of the third subsystem:

R301: $d_1 \rightarrow x_{1,2} \wedge g_1$

R302: $d_2 \rightarrow x_{2,1} \wedge (g_1 \vee g_2)$
R303: $d_3 \rightarrow x_{3,1} \wedge g_1$
R304: $d_4 \rightarrow x_{4,1} \wedge g_1$
R305: $d_5 \rightarrow x_{5,1} \wedge g_1$
R306: $d_6 \rightarrow g_0$
R307: $d_7 \rightarrow x_{7,1} \wedge g_1$
R308: $d_8 \rightarrow x_{8,2} \wedge (g_1 \vee g_2)$
R309: $d_9 \wedge z_{9,2} \rightarrow g_0$
R310: $d_9 \wedge \neg d_8 \wedge \neg z_{9,2} \rightarrow x_{9,2} \wedge (g_1 \vee g_2)$
R311: $d_{10} \vee d11 \vee d12 \rightarrow x_{10,1} \wedge (g_1 \vee g_2)$
R312: $d_{13} \rightarrow x_{13,1} \wedge g_1$
R313: $d_{14} \rightarrow x_{14,1} \wedge (g_1 \vee g_2)$
R314: $d_{15} \rightarrow x_{15,1} \wedge (g_1 \vee g_2)$
R315: $d_{16} \rightarrow g_0$
R316: $d_{17} \rightarrow g_0$
R317: $d_{18} \rightarrow g_0$
R318: $d_{19} \rightarrow g_0$
R319: $d_{20} \rightarrow g_0$
R320: $d_{21} \rightarrow g_0$
R321: $d_{22} \rightarrow g_0$
R322: $d_{23} \rightarrow x_{23,1} \wedge (g_1 \vee g_2)$
R323: $d_{24} \rightarrow g_0$
R324: $d_{25} \rightarrow g_0$
R325: $d_9 \wedge \neg z_{9,1} \wedge z_{8,1} \rightarrow x_{8,2} \wedge (g_1 \vee g_2)$
R326: $d_9 \wedge \neg z_{9,1} \wedge \neg z_{8,1} \rightarrow x_{26,1} \wedge g_1$

For instance, rule R309 says:

IF "high coolant temperature" (d_9) AND "servo fan's fuse is blown" ($z_{9,2}$).

THEN "no other repair required" (g_0)

(the substitution of the fuse is supposed to have taken place) meanwhile (its complementary) rule R310 says:

IF "high coolant temperature" (d_9)

AND NOT "low coolant level" ($\neg d_8$)

AND NOT "servo fan's fuse is blown" ($\neg z_{9,2}$)

THEN $x_{9,2}$repair servo fan ($x_{9,2}$)

AND "have the car repaired at a workshop" (g_2)

OR "DIY repair at home" (g_2)

6. *CoCoA 4.6/4.7* implementation

6.1. CoCoA code of the underlying logic

The required code is really simple and straightforward. Firstly, the polynomial ring A and the ideal I (for introducing idempotency) are defined:

```
A ::= Z/(2)[d[1..25], x[1..26, 1..2], y[1..16, 1..2], z[1..16, 1..2],
          c[1..3], o[1..6], g[0..2]];
USE A;
MEMORY.I := Ideal(d[1]^2 - d[1], d[2]^2 - d[2], d[3]^2 - d[3], d[4]^2 - d[4],
                  ..., d[23]^2 - d[23], d[24]^2 - d[24], d[25]^2 - d[25],
                  x[1, 1]^2 - x[1, 1], x[1, 2]^2 - x[1, 2], x[2, 1]^2 - x[2, 1],
                  ..., x[25, 1]^2 - x[25, 1], x[26, 1]^2 - x[26, 1], ...,
                  g[0]^2 - g[0], g[1]^2 - g[1], g[2]^2 - g[2]);
```

Please cite this article in press as: E. Roanes-Lozano et al., A portable knowledge-based system for car breakdown evaluation, Appl. Math. Comput. (2014), http://dx.doi.org/10.1016/j.amc.2014.12.001

(dots are included for the sake of space, here all variables should appear).

Now the logical connectives (uppercase, prefix) can be defined in ring A:

```
DefineNEG(M)
  ReturnNF(1 + M, MEMORY.I);
 EndDefine;
DefineO(M,N)
  ReturnNF(M + N + M * N, MEMORY.I);
 EndDefine;
DefineY(M,N)
  ReturnNF(M * N, MEMORY.I);
 EndDefine;
  DefineIMP(M,N)
ReturnNF(1 + M + M * N, MEMORY.I);
 EndDefine;
```

6.2. CoCoA code of the KBS

Now the rules of the different subsystems can be introduced. For instance the rules of the first subsystem are:

```
R101 := IMP(d[1] , Y(y[1,1],y[1,2]));
R102 := IMP(d[8] , y[8,1]);
R103 := IMP(d[9], y[9,1]);
R104 := IMP(d[16] , y[16,1]);
R105 := IMP(z[9,1] , y[9,2]);
R106 := IMP(NEG(z[16,1]) , y[16,2]);
R107 := IMP(NEG(z[9,1]) , y[8,1]);
```

and the ideal of the negation of the rules is:

```
J1 := Ideal(NEG(R101),NEG(R102),NEG(R103),NEG(R104),NEG(R105),
          NEG(R106),NEG(R107));
```

(the other subsystems are similar and are not detailed for the sake of space).

6.3. Extracting knowledge using CoCoA

Now, for a set of facts, for example {d16}, we can extract knowledge. Those of $y_{1,1}, y_{1,2}, y_{8,1}, y_{9,1}, y_{16,1}$ which NF is 0 can be obtained from that set of facts:

```
Kla := Ideal(NEG(d[16]));
GBli := Ideal(MEMORY.I + J1 + Kla);
NF(NEG(y[1,1]), GBli);
NF(NEG(y[1,2]), GBli);
NF(NEG(y[8,1]), GBli);
NF(NEG(y[9,1]), GBli);
NF(NEG(y[16,1]), GBli);
```

(the result is 0 only in the last case, so only $y_{16,1}$ can be obtained).

The knowledge extraction takes place a second time in the first subsystem and then in the second and third subsystems (these two last processes determine the actions recommended to solve the problem). The process is exactly the same and the details are omitted for the sake of space. In any case, the interested reader can freely download all the CoCoA code from: http://www.matap.uma.es/jlgalan/Breakdown/CoCoA-Code.html.

7. Graphic User Interface

One of the main goals of the work described in this paper is that the final application can be run in smart devices (smartphones, tablets, notebooks, ...). To achieve this aim, a Graphical User Interface (*GUI*) has been developed in *HTML* and *PHP*. This combination of *HTML* and *PHP* for the development of the *GUI* has been firstly used by authors in this work, being a

Please cite this article in press as: E. Roanes-Lozano et al., A portable knowledge-based system for car breakdown evaluation, Appl. Math. Comput. (2014), http://dx.doi.org/10.1016/j.amc.2014.12.001

novelty and improvement with respect to their previous works. The previous KBS were intended to be used in a computer at a hospital, a laboratory, at home, etc., and each GUI was written in BASIC specifically for the corresponding application.

The GUI obtained is a friendly environment that can be run in any smart devices with Internet access. Nowadays many studies suppose the existence of smartphones on board road vehicles (for instance for automatic data collection regarding positioning [25]).

It is important to remark that, nowadays, the use of smartphones with internet access is widely spread all over the world. Therefore, the GUI allows to run the application just in the moment that the problem arises.

The only need for an user to run the application is just an Internet connection. Neither the CAS COCOA nor other applications need to be installed in the smart device since all the needed software is installed and executed remotely in the server where the GUI is allocated.

The user enters the initial data throughout a graphical menu which allows the user to identify the problem. This is done in a friendly way since common visual icons are displayed.

Let us remember that the KBS has been structured in three different subsystems. These subsystems need to be executed in different steps with successive calls to COCOA in which possible additional information is required. In order to ease these tasks, the GUI asks the possible needed questions for each subsystem and runs COCOA in a transparent way. That is, the user does not need to know the subsystem structure. The user only has to mark in the main menu the dashboard lights which are on and answer the required questions (if any). According to the marked icons, the GUI builds the input file to be run from COCOA, reads and gets information from the output file (and possibly from the user), and uses this information to build the new input file for next step. After the last step, the suggestions to the user are displayed.

The GUI shows four different windows which require information from the user and provides the final suggestions. These windows are built dynamically depending on the input data and the results obtained after each execution of COCOA. An example of each type of windows can be found in Figs. 2–5.

Specifically, in Fig. 2, the main menu is shown. After marking the dashboard lights that are on, the user must press the send bottom (note that not all the menu is shown since the figure has been cut for space and legibility reasons). With this

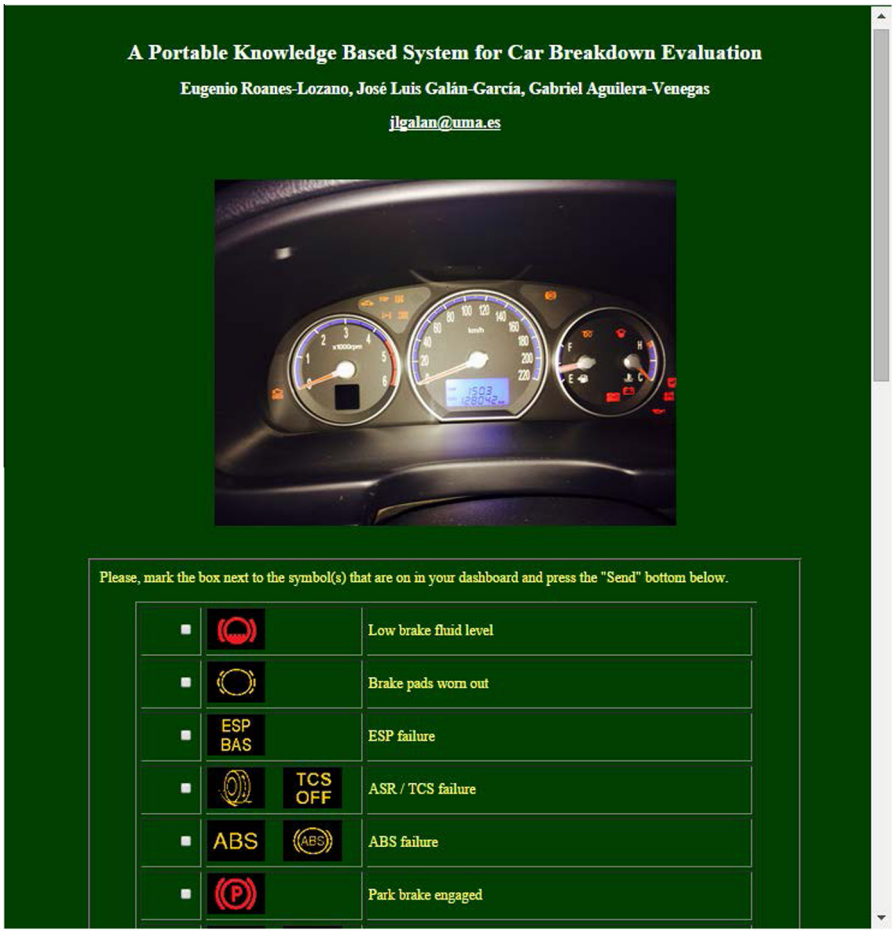

Fig. 2. Upper part of the menu.

Please cite this article in press as: E. Roanes-Lozano et al., A portable knowledge-based system for car breakdown evaluation, Appl. Math. Comput. (2014), http://dx.doi.org/10.1016/j.amc.2014.12.001

information, the GUI builds a first ideal (containing the variables associated to the lights that the user has marked) and generates the CoCoA input file for the first subsystem. After running CoCoA and processing the result, the GUI checks the need to ask for more information or not. The possible questions are then shown in the window in Fig. 3. When the user marks the possible answers and presses the confirmation bottom, the GUI builds a second ideal from the obtained information and generates a new CoCoA file for the second part of the first subsystem. After processing the new result, the GUI checks once more the need of new questions. Fig. 4 shows the window that handles this step. The GUI builds a third ideal from the variables associated to the user's answers and generates the final CoCoA file to run subsystems 2 and 3. Finally, after processing the result file, the suggestions to the user are displayed in the window shown in Fig. 5.

If there are more than one fault the GUI takes care of showing only the most restrictive recommended driving style and way to perform the repair.

The application described in this paper can be tested throughout this GUI which is allocated at:
http://www.matap.uma.es/jlgalan/Breakdown/.

8. Performance tests

It is well known that the problem of deciding if a formula (conclusion) can be inferred from a set of formulae (hypothesis) in the Propositional Classical Logic is a NP-complete problem. This means that for the problem dealt in this paper, in the general case, there not exists any known polynomial algorithm with respect to the number of propositional symbols (in our case, with respect to the number of variables) which solves the problem.

In any case, since the number of variables used in this prototype is not high (80 variables), and not all of them appears in all subsystems, the execution time of each subsystem is reasonably short. Table 1 describes the execution times of each subsystem with respect to the number of dashboard lights which are on. The computer used for these timings is a standard notebook which processor is an Intel® Core™ i5 (2.5 GHz) and the server is a standard Macintosh desktop computer (Mac mini) which processor is an Intel® Core™ i7 (2 GHz).

The values showed in Table 1 have been obtained computing exclusively the time (in micro seconds and, in the table, printed in seconds) used for each call to the server. This way, only the times spent in the CoCoA executions are computed. Therefore, the time spent in the access and management of communication files and the time consumed by the user answering the different required information are not considered.

As it can be seen in Table 1, the time consumed is very similar with independence of the number of dashboard lights on. This is due to the fact that, in all cases, all the possible questions to extract knowledge are done. In any case, using the computer previously described, the total time is reasonably short (less than 2.5 s).

The prototype developed in this work uses 80 variables. For a specific car model, an *ad hoc* adaptation of this prototype should be done. In any case, the number of variables would be very similar to the number used in our prototype since the number of dashboard lights and the number of actions to be taken are similar for the different car models. Therefore, time consumed in this kind of *ad hoc* KBS would be similar to the values shown in Table 1. In any case, if a particular situation required of a significatively greater number of variables, time consumed could increase too much. In this case, the KBS could

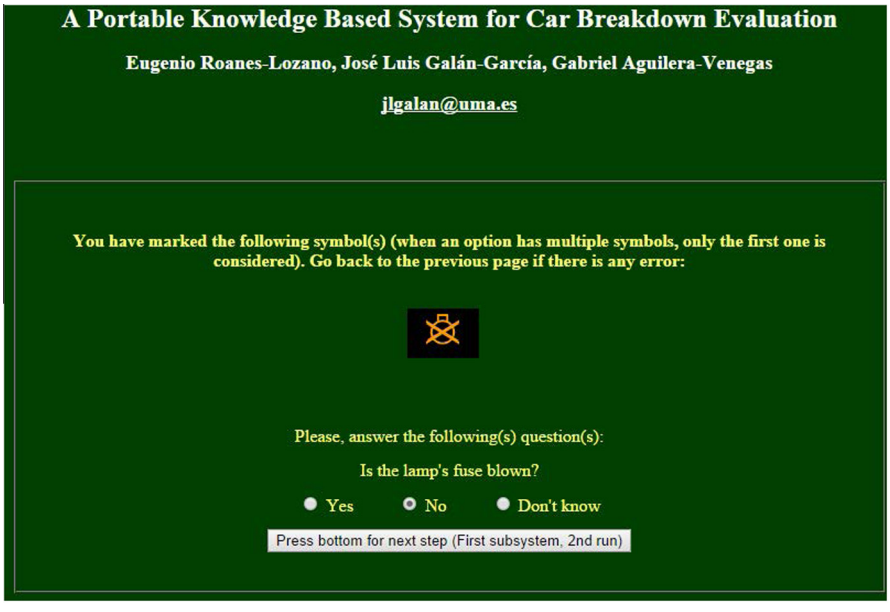

Fig. 3. Example of questions in the first step of the first subsystem.

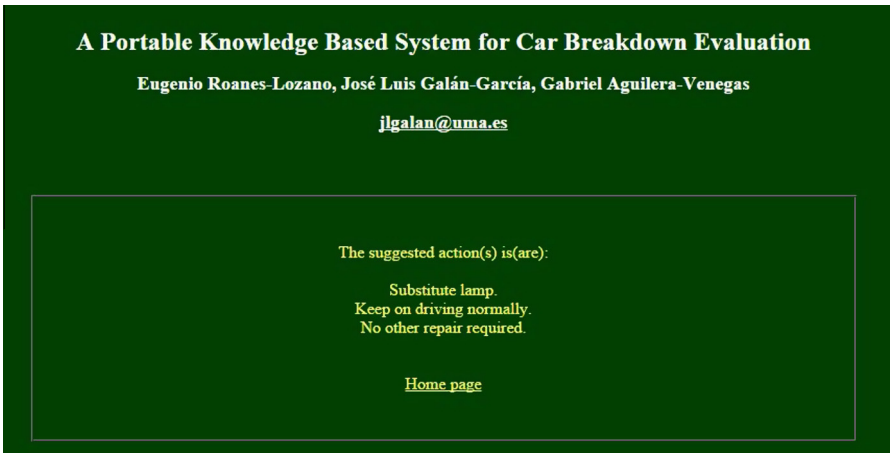

Fig. 4. Example of questions in the second step of the first subsystem.

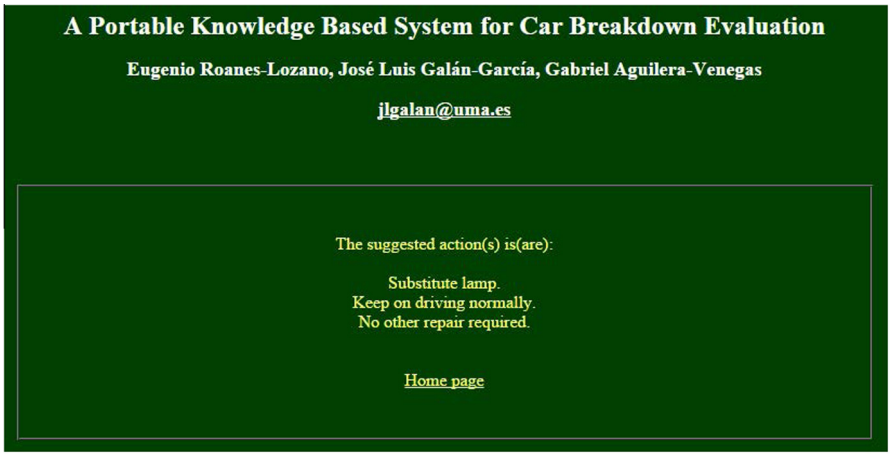

Fig. 5. Example of suggested actions.

Table 1
Times in seconds.

Lights on	Subs. 1–1	Subs. 1–2	Subs. 2 + Subs. 3	Total
1	0.273	0.259	1.852	2.384
1 and 2	0.276	0.262	1.770	2.308
1–3	0.270	0.266	1.668	2.204
1–5	0.277	0.265	1.597	2.139

be split in more subsystems (with a similar number of variables than the one used in our prototype), devoted to different aspects, for instance: engine, transmission, electrical systems, etc. Each subsystem would need to work with its associated variables (not the total) and therefore the total time would be the sum of these partial times, remaining reasonably short.

9. Conclusions

We do no pretend to have implemented a complete and detailed KBS on the topic, but a possible design and a simple example of development as illustration.

We have shown in the paper that the rules of the KBS are intuitive and easy to write since COCOA works with a direct transcription from the logic rules. Therefore, the example developed in this paper could be adapted to other dashboard lights and different actions. This fact, together with the easiness of adapting the PHP code of the GUI, make it possible to develop *ad hoc* applications for a specific car model.

Please cite this article in press as: E. Roanes-Lozano et al., A portable knowledge-based system for car breakdown evaluation, Appl. Math. Comput. (2014), http://dx.doi.org/10.1016/j.amc.2014.12.001

The facility of executing the application remotely is useful when any emergency arises while driving, just using a smartphone.

Vehicle fault diagnosis has been a topic of knowledge systems since the early times of artificial intelligence, and has been proposed or applied to different kinds of vehicles: automobiles [26], locomotives [27], etc. Different techniques have been applied along time, and this is still an active topic [12–14]. Nevertheless we know of no similar KBS: oriented to solve the problem reported by the dashboard lights, based on the use of an algebraic inference engine and oriented to help a driver situated anywhere through the use of smart devices and an Internet connection.

Acknowledgments

This work was partially supported by the research project TIN2012-32482 (Government of Spain) and by Grant No. TIN2011-28084 of the Science and Innovation Ministry of Spain, co-funded by the European Regional Development Fund (ERDF).

Finally, we thank the anonymous reviewers for their suggestions and comments which have improved the quality of the paper.

References

[1] <http://dashboardsymbols.com/the-symbols/>.
[2] L.M. Laita, E. Roanes-Lozano, V. Maojo, E. Roanes-Macías, L. de Ledesma, L. Laita, An expert system for managing medical appropriateness criteria based on computer algebra techniques, Comput. Math. Appl. 42 (12) (2001) 1505–1522.
[3] C. Pérez-Carretero, L.M. Laita, E. Roanes-Lozano, L. Lázaro, J. González-Cajal, L. Laita, A logic and computer algebra-based expert system for diagnosis of anorexia, Math. Comput. Simul. 58 (3) (2002) 183–202.
[4] C. Rodríguez-Solano, L.M. Laita, E. Roanes-Lozano, L. López-Corral, L. Laita, A computational system for diagnosis of depressive situations, Expert Syst. Appl. 31 (2006) 47–55.
[5] J. Piury, L.M. Laita, E. Roanes-Lozano, A. Hernando, F.J. Piury-Alonso, J.M. Gómez- Argüelles, L. Laita, A Gröbner bases-based rule based expert system for fibromyalgia diagnosis, Rev. R. Acad. Cien. Ser. A. Mat. (RACSAM) 106 (2) (2012) 443–456.
[6] B. Buchberger, Bruno Buchberger's PhD thesis 1965: an algorithm for finding the basis elementals of the residue class ring of a zero dimensional polynomial ideal, J. Symb. Comput. 41 (3–4) (2006) 475–511.
[7] E. Arnold, I. Kotsireas, M. Rosenkranz (Eds.), Groebner Bases and Applications (Special Issue), J. Symb. Comput. 46(5), 2011.
[8] <http://www.haynes.com/>.
[9] A.T. Al-Taani, An expert system for car failure diagnosis, in: C. Ardil (Ed.), IEC (Prague)'05, Enformatika, Çanakkale, Turkey, 2005, pp. 457–560.
[10] S.A. Mostafa, M.S. Ahmad, M.A. Mohammed, O.I. Obaid, Implementing an expert diagnostic assistance system for car failure and malfunction, Int. J. Comput. Sci. Issues 9 (2) (2012) 1–7.
[11] W.T. Scherer, C.C. White III, A survey of expert systems for equipment maintenance and diagnostics, in: S.G. Tzafestas (Ed.), Knowledge-Based System Diagnosis, Supervision, and Control, Springer Science+Business Media, New York, 1989, pp. 285–300, http://dx.doi.org/10.1007/978-1-4899-2471-116.
[12] P. Chen, Study on neural network automobile fault diagnosis expert system, J. Appl. Sci. 14 (2014) 348–354, http://dx.doi.org/10.3923/jas.2014.348.354.
[13] J.-D. Wu, J.-D. Road, C.-H. Liu, An expert system for fault diagnosis in internal combustion engines using wavelet packet transform and neural network, J. Exp. Syst. Appl. 36 (3) (2009) 4278–4286, http://dx.doi.org/10.1016/j.eswa.2008.03.008.
[14] Y. Youjun, L. Xiang, Z. Qun, Development of automobile fault diagnosis expert system based on fault tree – Neural network ensemble, in: International Conference Electronics, Communications and Control (ICECC) 2011, Institute of Electrical and Electronics Engineers Inc, Piscataway, NJ, 2011, http://dx.doi.org/10.1109/ICECC.2011.6067801.
[15] <http://clipsrules.sourceforge.net/>.
[16] L.M. Laita, E. Roanes-Lozano, L. de Ledesma, J.A. Alonso, A computer algebra approach to verification and deduction in many-valued knowledge systems, Soft Comput. 3 (1999) 7–19.
[17] E. Roanes-Lozano, L.M. Laita, E. Roanes-Macías, A polynomial model for multivalued logics with a touch of algebraic geometry and computer algebra, Math. Comput. Simul. 45 (1) (1998) 83–99.
[18] E. Roanes-Lozano, L.M. Laita, A. Hernando, E. Roanes-Macías, An algebraic approach to rule based expert systems, Rev. R. Acad. Cien. Ser. A. Mat. (RACSAM) 104 (1) (2011) 19–40, http://dx.doi.org/10.5052/RACSAM.2010.04.
[19] J. Hsiang, Reputational theorem proving using term-rewriting systems, Art. Intell. 25 (1985) 255–300.
[20] D. Kapur, P. Narendran, An equational approach to theorem proving in first-order predicate calculus, in: A.K. Joshi (Ed.), Proceedings of IJCAI-85, Morgan Kaufmann, San Francisco, CA, 1985, pp. 1146–1153.
[21] J.A. Alonso, E. Briales, Lógicas polivalentes y bases de Gröbner, in: M. Vide (Ed.), Proceedings of the V Congress on Natural Languages and Formal Languages, Barcelona Univ. Press, Barcelona, 1989, pp. 307–315.
[22] J. Chazarain, A. Riscos, J.A. Alonso, E. Briales, Many-valued logic and Gröbner bases with applications to modal logic, J. Symb. Comput. 11 (1991) 181–194.
[23] E. Roanes-Lozano, L.M. Laita, E. Roanes-Macías, A polynomial model for many-valued logics with a touch of algebraic geometry and computer algebra, Math. Comput. Simul. 45/1 (1998) 83–99.
[24] E. Roanes-Lozano, L.M. Laita, E. Roanes-Macías, Maple V in A.I.: the Boolean algebra associated to a KBS, CAN Nieuwsbrief 14 (1995) 65–70.
[25] N. Cáceres Sánchez, J. Wideberg, F. García Benítez, Review of traffic data estimations extracted from cellular networks, IET Intell. Transp. Syst. 2 (3) (2008) 179–192.
[26] K.L. Clark, F.G. McCabe, PROLOG: a language for implementing expert systems, in: J. Hayes, D. Michie, Y.H. Pao (Eds.), Machine Intelligence 10, Ellis Horwood Publications, Wiley, New York, 1982, pp. 455–470.
[27] H.E. Johnson Jr., P.P. Bonissone, Expert system for diesel electric locomotive repair, J. Forth Appl. Res. 1/ (1/) (1983) 7–16.