A Portable Knowledge Based System for Car Breakdown Evaluation

Eugenio Roanes-Lozano$^1$, José Luis Galán-García$^2$, Gabriel Aguilera-Venegas$^2$

$^1$Instituto de Matemática Interdisciplinar (IMI) & Depto. de Algebra, Universidad Complutense de Madrid, Spain

$^2$Department of Applied Mathematics. Universidad de Málaga, Spain

4$^{th}$ European Seminar on Computing
ESCO 2014
June 15-20. Pilsen, Czech Republic
Contents

1 Introduction
   - Motivation of the problem
   - Color symbols
   - Interpretation problems

2 Work developed
   - Backgrounds
   - Justification of the application
   - Work developed

3 RBES structure
   - Three subsystems
   - Mathematical background
   - Computer algebra approach to RBES

4 The RBES and its GUI
   - First subsystem
   - Second subsystem
   - Third subsystem
   - The GUI
Motivation of the problem

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 wasn’t correctly evaluated!
Color symbols

- 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.
- **Red symbols** usually indicate a safety issue or a serious problem.
- **Yellow symbols** use to indicate a not so urgent problem.
- **Green** and **blue symbols** usually provide information about the systems connected.
Interpretation problems

- But not all the **red icons** require of the same action.
- To **stop the car immediately** is not always required, although can be the best option sometimes.
- 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.
We had previously developed knowledge based systems (KBS). The underlying logic was Boolean or many-valued modal. Different tasks such as managing medical appropriateness criteria and the early detection of illnesses. In all cases we used computational techniques borrowed from computer algebra (Gröbner bases).
Justification of the application

- **Smart devices** have become popular and have an outstanding computing power plus Internet connection.
- 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 by the driver regarding:
  - the possible **repair** on site or at a workshop.
  - its **urgency**.
  - the need to **immobilize the vehicle** immediately.
Work developed

- We have designed and developed a **KBS** for the described problem.
- The code have been developed using a CAS (specifically, Co-Coa) and the GUI to get the data from the user is done by means of a web page which allows to be accessed at any time from any smart device.
- It returns explanations about the repairs to be made such as **substitute brake pads**
- Recommendations such as **drive carefully** $\land$ **increase safety distance**
- Suggestions about where to have the car to be repaired such as **have the car repaired at a workshop** $\lor$ **Do it yourself (DIY) repair at home**
We have developed the application using a Rule-Based Expert System (RBES).

The RBES is divided into three subsystems:

1. **Tests to be made on site**: the system asks the driver to perform some tests under the hood (if necessary), perhaps two times.

2. **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.

3. **Definitive repairs** (after the on site repair): determine what has to be repaired at a workshop (or possible DIY) when back.
Mathematical background

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.
Mathematical background

Let $\lor, \land, \neg, \rightarrow$ denote the logic disjunction, conjunction, negation and implication, respectively.

Let $(\mathcal{C}, \lor, \land, \neg, \rightarrow)$ be the Boolean algebra of the propositions that can be constructed using a finite number of propositional variables $P, Q, ..., R$.

Let us consider the Boolean algebra $(\mathcal{A}, \tilde{+}, \cdot, 1+, \text{“is a multiple”})$, where $\mathcal{A}$ is the residue class ring

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

$(\langle p^2 - p, q^2 - q, ..., r^2 - r \rangle$ denotes the polynomial ideal generated by $p^2 - p, q^2 - q, ..., r^2 - r)$.\]
Mathematical background

Let us define:

\[ \varphi : (\mathcal{C}, \lor, \land, \neg, \rightarrow) \rightarrow (\mathcal{A}, \hat{+}, \cdot, 1+, \text{“is a multiple”}) \]

the following way:

for propositional variables

\[ P \rightarrow p \]
\[ Q \rightarrow q \]
\[ \ldots \ldots \]
\[ R \rightarrow r \]

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

\[ A \lor B \rightarrow a \hat{+} b \]
\[ \neg A \rightarrow 1 + a \]

Then, as an immediate consequence of the De Morgan laws:

\[ A \land B \rightarrow a \cdot b \]
Mathematical background

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

The main result is based in the following theorem:

$$\{\beta_1, \ldots, \beta_m\} \models \alpha \iff \varphi(\neg \alpha) \in \langle \varphi(\neg \beta_1), \ldots, \varphi(\neg \beta_m) \rangle$$

which translates the problem of checking whether a propositional formula is a $\alpha$ tautological consequence of (i.e., can be inferred from) others or not, by a polynomial ideal membership.
Computer algebra approach to RBES

The algebraic computations are performed in a computer algebra system 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**, that are unique for each polynomial ideal.

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

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

(where NF denotes **Normal Form**).
First set of potential facts: dashboard lights

- $d_1$ low brake fluid level
- $d_3$ ESP failure
- $d_5$ ABS failure
- $d_7$ Airbag / SRS failure
- $d_9$ high coolant temperature
- $d_{11}$ low oil level
- $d_{13}$ low timing belt tension
- $d_{15}$ battery charge failure
- $d_{17}$ low fuel level
- $d_{19}$ seat belt not fasten
- $d_{21}$ high beam
- $d_{23}$ Diesel particulates filter
- $d_{25}$ extremely low tire pressure
- $d_2$ brake pads worn out
- $d_4$ ASR / TCS failure
- $d_6$ park brake engaged
- $d_8$ low coolant level
- $d_{10}$ low oil pressure
- $d_{12}$ engine failure
- $d_{14}$ servicing required
- $d_{16}$ bulb failure
- $d_{18}$ door open
- $d_{20}$ low wiper fluid
- $d_{22}$ glow plug
- $d_{24}$ low tire pressure
Repairs to be made

- $x_{1,1}$ refill brake fluid
- $x_{2,1}$ substitute brake pads
- $x_{4,1}$ repair ASR / TCS
- $x_{6,1}$ release handbrake
- $x_{8,1}$ refill coolant (wait for engine cold)
- $x_{9,1}$ substitute servo fan fuse
- $x_{10,1}$ repair engine
- $x_{13,1}$ tighten timing belt
- $x_{15,1}$ repair alternator or wires to battery
- $x_{16,2}$ substitute lamp
- $x_{18,1}$ close door(s)
- $x_{20,1}$ refill wiper fluid
- $x_{22,1}$ nothing to be done, perfectly normal
- $x_{24,1}$ inflate tire when possible
- $x_{1,2}$ repair brake circuit leak
- $x_{3,1}$ repair ESP
- $x_{5,1}$ repair ABS
- $x_{7,1}$ repair airbag / SRS
- $x_{8,2}$ repair coolant leak
- $x_{9,2}$ repair servo fan
- $x_{11,1}$ refill oil
- $x_{14,1}$ service the vehicle
- $x_{16,1}$ substitute lamp's fuse
- $x_{17,1}$ refill the tank
- $x_{19,1}$ fasten seat belt
- $x_{21,1}$ switch to low beam if necessary
- $x_{23,1}$ substitute Diesel particulates filter
- $x_{25,1}$ substitute wheel
Tests to be made on site

\[ y_{1,1} \text{ test if any important brake fluid leak in the brake circuit} \]
\[ y_{1,2} \text{ test if any important brake fluid leak in the servo brake booster} \]
\[ y_{8,1} \text{ test if any important coolant leak in the cooling system} \]
\[ y_{9,1} \text{ test if servo fan off (ignition key in position } \text{on}) \]
\[ y_{9,2} \text{ test if servo fan’s fuse is blown} \]
\[ y_{16,1} \text{ test if lamp’s fuse is blown} \]
\[ y_{16,2} \text{ test lamp} \]
Tests to be made on site

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.
Rules of the first subsystem

- First knowledge extraction:
  - R101: \( d_1 \rightarrow (y_{1,1} \land y_{1,2}) \)
  - R102: \( d_8 \rightarrow y_{8,1} \)
  - R103: \( d_9 \rightarrow y_{9,1} \)
  - R104: \( d_{16} \rightarrow y_{16,1} \)

- Second knowledge extraction:
  - R105: \( z_{9,1} \rightarrow y_{9,2} \)
  - R106: \( \neg z_{16,1} \rightarrow y_{16,2} \)
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 these 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 level* dashboard light
- $o_5$ the autonomy will be very restricted (the battery will be completely discharged soon)
- $o_6$ keep observing the *tire pressure* dashboard light / gauge.
Rules of the second subsystem

- R201: \((d_1 \land \neg z_{1,1} \land \neg z_{1,2}) \rightarrow (x_{1,1} \land c_2 \land o_1 \land o_2)\)
- R202: \((d_1 \land (z_{1,1} \lor z_{1,2})) \rightarrow c_3\)
- R203: \(d_2 \rightarrow (c_2 \land o_2)\)
- R204: \(d_3 \rightarrow c_2\)
- R205: \(d_4 \rightarrow c_2\)
- R206: \(d_5 \rightarrow c_2\)
- R207: \(d_6 \rightarrow (x_{6,1} \land c_1)\)
- R208: \(d_7 \rightarrow c_2\)
- R209: \((d_8 \land \neg z_{8,1}) \rightarrow (x_{8,1} \land c_2 \land o_3)\)
- R210: \((d_8 \land z_{8,1}) \rightarrow c_3\)
Rules of the second subsystem

- R211: \((d_9 \land z_{9,2}) \rightarrow (x_{9,1} \land c_2 \land o_4)\)
- R212: \((d_9 \land \neg z_{9,2}) \rightarrow c_3\)
- R213: \(d_{10} \rightarrow c_3\)
- R214: \(d_{11} \rightarrow (x_{11,1} \land c_1)\)
- R215: \(d_{12} \rightarrow c_3\)
- R216: \(d_{13} \rightarrow c_2\)
- R217: \(d_{14} \rightarrow c_1\)
- R218: \(d_{15} \rightarrow (c_2 \land o_5)\)
- R219: \((d_{16} \land z_{16,1}) \rightarrow (x_{16,1} \land c_1)\)
- R220: \((d_{16} \land z_{16,2}) \rightarrow (x_{16,2} \land c_1)\)
Rules of the second subsystem

- R221: \((d_{16} \land \neg z_{16,1} \land \neg z_{16,2}) \rightarrow c_2\)
- R222: \(d_{17} \rightarrow (x_{17,1} \land c_1)\)
- R223: \(d_{18} \rightarrow (x_{18,1} \land c_1)\)
- R224: \(d_{19} \rightarrow (x_{19,1} \land c_1)\)
- R225: \(d_{20} \rightarrow (x_{20,1} \land c_1)\)
- R226: \(d_{21} \rightarrow (x_{21,1} \land c_1)\)
- R227: \(d_{22} \rightarrow c_1\)
- R228: \(d_{23} \rightarrow c_1\)
- R229: \(d_{24} \rightarrow (x_{24,1} \land c_2 \land o_6)\)
- R230: \(d_{25} \rightarrow (x_{25,1} \land c_1)\)
Conclusions of the third subsystem

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: \( d_1 \rightarrow (x_{1,2} \land g_1) \)
- R302: \( d_2 \rightarrow (x_{2,1} \land (g_1 \lor g_2)) \)
- R303: \( d_3 \rightarrow (x_{3,1} \land g_1) \)
- R304: \( d_4 \rightarrow (x_{4,1} \land g_1) \)
- R305: \( d_5 \rightarrow (x_{5,1} \land g_1) \)
- R306: \( d_6 \rightarrow (x_{6,1} \land g_0) \)
- R307: \( d_7 \rightarrow (x_{2,1} \land (g_1 \lor g_2)) \)
- R308: \( d_8 \rightarrow (x_{8,2} \land (g_1 \lor g_2)) \)
- R309: \( (d_9 \land z_{9,2}) \rightarrow g_0 \)
- R310: \( (d_9 \land \neg z_{9,2}) \rightarrow (x_{9,2} \land (g_1 \lor g_2)) \)
- R311: \( (d_{10} \lor d_{11} \lor d_{12}) \rightarrow (x_{10,1} \land (g_1 \lor g_2)) \)
- R312: \( d_{13} \rightarrow (x_{13,1} \land g_1) \)
Rules of the third subsystem

- R313: $d_{14} \rightarrow (x_{14,1} \land (g_1 \lor g_2))$
- R314: $d_{15} \rightarrow (x_{15,1} \land (g_1 \lor g_2))$
- 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: $d_{23} \rightarrow (x_{23,1} \land (g_1 \lor g_2))$
- R323: $d_{24} \rightarrow g_0$
- R324: $d_{25} \rightarrow g_0$
The GUI

- The **GUI** is still under development.
- We have decided to implement a web accessible GUI that asks the driver different questions with drop-down menus.
- Implementing it as a web page that calls the CAS has the advantage that it will be accessible from electronic devices:
  - just with an Internet connection, or
  - running on their own if the CAS and RBES are installed in the electronic device.
- After each subsystem is run, and according to the knowledge extracted by the subsystem, the GUI will ask the driver more questions in order to complete the input to run the next subsystem.

**Link to the GUI**
A Portable Knowledge Based System for Car Breakdown Evaluation

Eugenio Roanes-Lozano\textsuperscript{1}, José Luis Galán-García\textsuperscript{2}, Gabriel Aguilera-Venegas\textsuperscript{2}

\textsuperscript{1}Instituto de Matemática Interdisciplinar (IMI) & Depto. de Algebra, Universidad Complutense de Madrid, Spain

\textsuperscript{2}Department of Applied Mathematics. Universidad de Málaga, Spain

4\textsuperscript{th} European Seminar on Computing
ESCO 2014
June 15-20. Pilsen, Czech Republic