Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 4982974, 14 pages
https://doi.org/10.1155/2018/4982974

Research Article

A New Algebraic Approach to Decision Making in a Railway Interlocking System Based on Preprocess

Antonio Hernando,[1] Roberto Maestre,[1] and Eugenio Roanes-Lozano[2]

[1] *Departamento de Sistemas Inteligentes Aplicados, Escuela Técnica Superior de Ingeniería de Sistemas Informáticos, Universidad Politécnica de Madrid, c/ Alan Turing s/n, 28031 Madrid, Spain*
[2] *Departamento de Álgebra, Geometría y Topología & Instituto de Matemática Interdisciplinar, Facultad de Educación, Universidad Complutense de Madrid, c/ Rector Royo Villanova s/n, 28040 Madrid, Spain*

Correspondence should be addressed to Eugenio Roanes-Lozano; eroanes@mat.ucm.es

Received 6 March 2018; Revised 14 July 2018; Accepted 17 August 2018; Published 1 October 2018

Academic Editor: Luis Payá

The safety of railway networks is a very important issue. Roughly speaking, it can be split into safety along lines and safety of railway facilities such as stations, junctions, yards, etc. In modern networks the safety along lines is controlled by *automatic block systems* that do not give clearance to trains to enter a section (block) until the latter is detected to be unoccupied. Meanwhile, the safety within railway facilities is supervised by *railway interlocking systems*. Decision making in a railway interlocking is a very important issue which is considered to be very labour-intensive. Decision-making in both automatic block systems and railway interlocking systems, unlike road traffic light systems, is not based on time (they are not scheduling problems) but in space. Basically, two different trains should never be allowed to access the same section (whatever time has passed). There are many different approaches to automate decision-making in railway interlocking systems. The classic approaches are offline: only certain routes are allowed and their compatibility is decided in advance. Meanwhile, modern approaches make decisions in real time and are independent from the topology of the railway network, but can be applied only to small or medium size railway networks. Nevertheless, these last approaches have the following drawbacks: the performances are very dependent on the number of trains in the railway network; and are unsuitable to large networks since they take long time to be run. On the other hand, algebraic approaches based on computer algebra concepts have been used in artificial intelligence for implementing expert systems. In this paper we present a completely new algebraic model, based on these concepts of computer algebra that overcomes these drawbacks: the performance of our approach is independent of the number of trains in the railway network and also is suitable for large railway networks.

1. Introduction

The safety of railway networks is a very important issue. Roughly speaking, it can be split in safety along lines and safety of railway facilities such as stations, junctions, yards, etc.

In modern networks the safety along lines is controlled by automatic block systems, that do not give clearance to trains to enter a section (block) until the latter is detected to be unoccupied.

Meanwhile, the safety within railway facilities is supervised by railway interlocking systems. Decision making in a railway interlocking is a very important issue which is considered very labour-intensive.

Railway interlocking systems are conceived so that different trains can be placed on different sections. Semaphores (mechanical devices) or light signals and turnouts must be stated so that two trains cannot collide when moving, if obeying the signalling. A railway interlocking system has the purpose of not allowing unsafe situations in the railway network. In fact the most sophisticated railway interlocking systems also forbid the switches under the train to be changed before the train has left that section, an issue that is not addressed in this article.

Let us underline that decision making in both automatic block systems and railway interlocking systems, unlike road traffic light systems, are not based on time (that is, they are not scheduling problems) but in space: basically, two different trains should never be allowed to access the same section (whatever time has passed since the first one entered the section). The reason is that trains need long distances to break and a train can be unexpectedly found stopped at a certain place due, for instance, to a breakdown. This approach has allowed to introduce semiautomatic train operation and even driverless train operation in some subways and airport shuttle services much earlier than autonomous cars were developed, although that is not the topic of this article.

The first railway interlocking systems were mechanical and were installed in the second half of the nineteenth century (Figure 1). In the mid twentieth century the control was relays-based (Figure 2). From the 1980s, most of the new railway interlocking systems are computer-controlled (electronic interlocking systems) [1–4]. Many of the interlocking systems are specifically designed for a particular railway network and they are not topology-independent. For instance, the first topology-independent railway interlocking system in Spain was installed only in 1993 [5]. Detecting if the situation of trains in a railway network is dangerous is an eye-catching problem. Traditionally, an offline approach is followed: only certain routes are considered and what is allowed by the interlocking is established in advance, usually by hand [6].

Modern approaches use algorithms for checking in real time the safety of a railway network: any proposed change in the position of switches and the signals aspects is analyzed in real time before being authorized. Clearly, an exhaustive analysis of all the possible movements allowed to all trains according to the proposal has to be performed. Obviously, the performance of the algorithm is a crucial issue of these approaches. It is surprising that, in real railway interlocking systems, unacceptable errors can be found (this happened, for instance, in the railway interlocking system of a tiny subway station [7]).

There are many different papers regarding computer applications to railway interlocking systems in general [8] and decision making in particular. These latter works either create a formal specification for an existing railway system (in order to verify it or to create a new decision making tool) or describe a completely new model for railway interlocking systems. Some of these approaches depend strongly on the number of trains in the railway network and are not suitable for large railway networks.

This paper deals with a new algebraic method for detecting the safety in a railway network that overcomes these previous drawbacks. It uses computer algebra concepts (like Boolean polynomials, ideals, Groebner bases, or normal forms) to solve the problem of decision making in a railway interlocking system.

The paper is structured in the following way. In Section 2, we will discuss techniques related to ours. In Section 3, we will define formally concepts related to railway networks. In Section 4, we will describe our method (as a black box) for determining the safety of a railway network. In Section 5, we will explain the basis of our algebraic approach based on the calculations of Groebner bases and normal forms. In Section 6, we will show the advantages of our model. As said above in this section, there are very many different approaches to decision making in a railway interlocking (see, e.g., the survey [8]), but usually the code is not available. That is why different approaches are described in Section 6 but the new system proposed is just compared with four (but very different) approaches, for which the complete implementation was available. Finally, in Section 8, we will set our conclusions.

2. Related Works

2.1. An Overview of Different Approaches. In this section, we will analyze different approaches for the problem of detecting the safety of a certain situation in a railway network.

In the classic approach, admissible train routes are predefined. A *route* denotes a path along the topology of the station or junction (for instance a path from an entrance of the station to a certain track where the train will stop). Establishing a route implies adequately setting the switches of the turnouts and light signals along the train route. Once an engine driver has been given a proceed signal concerning a route, the route is locked (that is, it cannot be changed before the train has completely cleared it). The standard approach to railway interlocking systems design is to predefine the admissible train routes and to manually study in advance their compatibility.

Many works have been proposed for railway interlocking systems (see, for instance, the survey [8]). These works either are specifically designed for an existing railway system or develop a generic new model. The Ph.D. thesis [9] uses a theorem prover implemented in higher-order logic to detect the safety of a certain situation in a railway interlocking. This work is revisited using an annotated logic program with temporal reasoning in [10]. The work [6] uses ordered binary decision diagrams to model railway interlocking systems. The work [11] uses Z notation for an example in the Slovak National Railways. In [12], a VDM model is presented for a case in the Danish State Railways. In [13–15] statecharts are used instead.

An early topology-independent formal model is [16]. It uses different layers of abstraction (called domains). Petri nets are used for the dynamic domains and double point graphs as well as logic invariants for the static domains. It is implemented in *Objective-C* and *PROLOG*. Let us underline that it does not follow the standard approach to railway interlocking systems; the concept of routes has been replaced with a context-free check of the permissibility for each controlling command.

A generic tool for verifying and validating railway interlocking systems is detailed in [17]. The track interlocking tables corresponding to an existing tramway network is represented using a domain-specific language (DSL) as illustration, and they are automatically transformed into an executable control system model expressed in SystemC.

In [18] DSL is also used to represent and analyze the track layout diagrams, interlocking tables, and circuit diagrams of existing relay interlockings of the DSB (Danish state Railways).

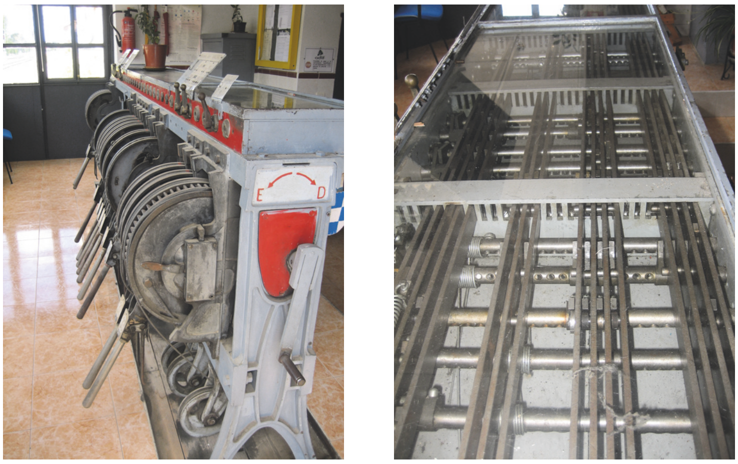

Figure 1: A mechanical railway interlocking system still in use at a junction in 2009. On the left a general view of the system is shown. The levers operated the semaphores (mechanical devices) and the switches of the turnouts. On the right a top view is shown. In the latter the bars that detect if a given lever conflicts with that controlled by another lever are clearly visible (it was common that the mechanical railway interlocking had glass tops so that the internal system could be appreciated).

Figure 2: One of the corridors full of relays of the relay interlocking system of a big railway station in the ‘90s.

In [19] the size of the layouts that can be addressed using formal methods for checking interlocking tables described with control tables is explored. The symbolic model checker NuSMV and the verification system SPIN are used and the conclusion is that these methods cannot be used to address large layouts.

The model checker NuSMV is also used in [20], where a formal model is built from a high-level (logic) description. The interlocking system is interpreted as an Abstract State Machine (ASM). The approach is applied to certain railway interlocking systems of Queensland Rails (QR) network.

Duration calculus, a specific technique for real-time systems [21], has also been applied to decision making in a railway interlocking [22].

2.2. Some Approaches with Available Code. The following topologically independent approaches do not consider the direction of the trains (allowing to directly deal with special situations like reversing loops and reversing triangles).

Model Based on Graphs. In [23] the problem is translated into graph theory language and treated using a matrices-based approach. This approach cannot be used for large railways networks because the matrices involved are square and sparse but $N{\times}N$ (where N is the number of sections of the network).

Algebraic Model. In [24], the problem is directly translated into an algebraic problem. According to this model, the safety of a railway network may be detected by calculating the Groebner basis [25–27] of a polynomial ideal. This approach is not suitable for large stations since the calculation of the Groebner bases of these polynomial ideals usually take a long time. These Groebner bases depend not only on the position of the switches of the turnouts and the color of the light signals, but also on the position of the trains. Curiously, computing times decrease in this model when the number of trains increases, as variables (representing sections) are substituted by numbers (representing trains).

Model Based on Boolean Propositional Logic. In [28], the problem is directly translated into many SAT problems which can be solved using an algorithm based on the DPLL technique [29]. Indeed, the safety of a railway network may be detected by solving N SAT problems where N is the number of sections of the railway network. These SAT problems depend on the position of the switches of the turnouts and the color of the light signals, as well as on the position of the trains. Like the previous approaches, this approach is not suitable for large stations, since many SAT problems need to be solved for a particular configuration of the railway network.

Logic-Algebraic Model. According to the model [30], the safety of the railway network may be detected by calculating

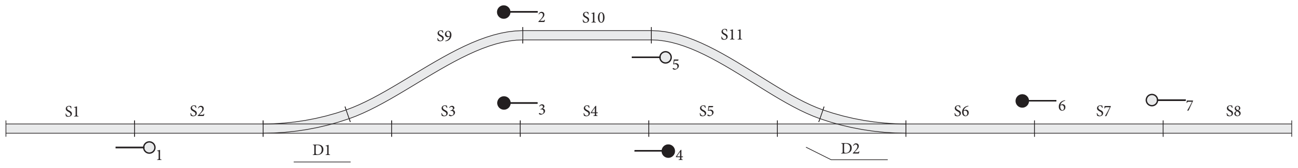

Figure 3: A very simple station.

a Groebner basis of an ideal of Boolean polynomials, G (which usually takes considerable time), and the normal forms of N polynomials modulo G (which is usually very fast computed). Like in [24], only a Groebner bases calculation is required for a particular state of the railway network. However, this model is much faster than [24], since the Groebner basis of this model (unlike the model [24]) is calculated on Boolean polynomials. This model is also faster than the previous one [28]. Nevertheless, this approach depends on the position of the trains and the performing time for calculations strongly depends on the number of trains.

ASP Model. According to [31], the problem to check the safety of the railway network is directly translated in the answer set programming (ASP) paradigm, defining relations and derived relations and solving the problem with logic techniques from ASP. This approach is much faster and efficient than previous approaches and thus can be applied to larger railway stations. From an operational point of view, the response time of this model depends on the number of trains in the station; i.e., more trains means more time to check the safety.

However, all of the previous approaches (except ASP) are not suitable for very large railway networks with many trains, since they involve long time to detect dangerous situation. In this paper, we will propose a completely new approach based on the calculation of a Groebner basis. In this new algebraic approach, unlike the algebraic approach in [24], we deal with Boolean polynomials and we use specialized algorithms for calculating Groebner bases over Boolean polynomials [32]. The approach here presented is completely different from [30]. Indeed, the set of polynomials for which we calculate the Groebner basis does not depend on the trains in the railway network (unlike [30]). As we will see in Section 6, once a precomputation has been performed, our approach is much faster than the previous ones (see Section 6) and is suitable for very large stations.

3. Basic Concepts regarding a Railway Interlocking System

Let us consider the tiny station in Figure 3 (Notice that the turnouts represented in the figures are really punctual elements that do not represent section themselves, they only represent the connectivity between sections.). As may be seen, the railway network represented is divided into sections, denoted in this case $S_1, \dots, S_{11}$ (there is a light signal between sections S_1 and S_2, a turnout D_1 connecting sections S_2, S_3 and S_9, and another light signal between sections S_{10} and S_9, and so on).

Taking D_1 on Figure 3 as example, the position of the switches can hold two states:

(i) Direct track position: the switch of turnout $T1$ connects sections S_2 and S_3.

(ii) Diverted track position: the switch of D_1 connects sections S_2 and S_9.

The position of the switches of the turnouts in Figure 3 is as follows: D_1 is in the direct track position (connecting S_2 and S_3) and D_2 in the diverted track position (connecting S_{11} and S_6). Note the symbols under the turnouts.

In this paper only nontrailable turnouts will be considered. However, it will be easy to extend this approach in order to model trailable turnouts just adding more polynomials. Let us underline that a trailable turnout is a turnout that allows to be passed even when the switch is not in the correct position.

A light signal can hold two states:

(i) Proceed, represented by −○ or ○− in the figures. For instance, in Figure 3, −○1 indicates "proceed", thus it is possible to pass from S_1 to S_2,

(ii) Stop, represented by −● or ●− in the figures. For instance, in Figure 3, −●4 indicates "stop", thus it is not possible to pass from S_4 to S_5.

If, for instance, three trains are placed on S_1, S_{10}, and S_8 (respectively), and the switch of D_1 is in the direct track position, and the switch of D_2 is in the diverted track position; −○1, −○5, ○−7 indicate "proceed" and ●−2, ●−3, −●4, ●−6 indicate "stop" (see Figure 4); the proposed situation is dangerous because section S_7 and S_8 are reachable by the trains placed in S_{10} and S_8.

If, for instance, two trains are placed on S_1 and S_{10} (respectively), and the switch of D_1 is in the direct track position, and the switch of D_2 is in the diverted track position; −○1, −○5, ○−7 indicate "proceed" and ●−2, ●−3, −●4, ●−6 indicate "stop" (see Figure 5); the proposed situation is safe.

More formally, let us consider a railway network with M sections and N trains placed in it. A *section* is a connected (single piece) part of the network, separated from the adjacent (neighbour) sections by a light signal or a turnout.

It is possible to pass from section S_i to the adjacent section S_j if and only if

(i) there is a turnout between sections S_i and section S_j, and its switch directs trains from S_i to section S_j and conversely (like S_2 and S_3 in Figure 3),

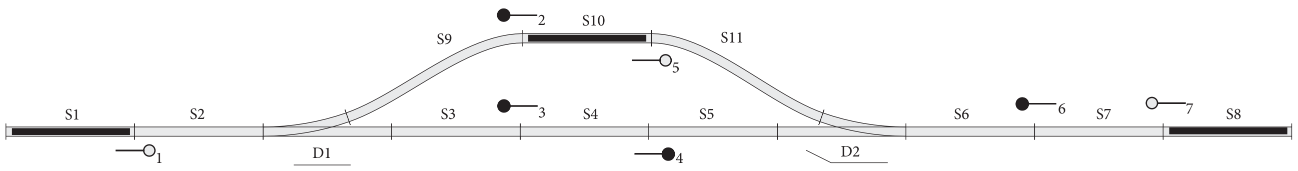

Figure 4: Example of dangerous situation.

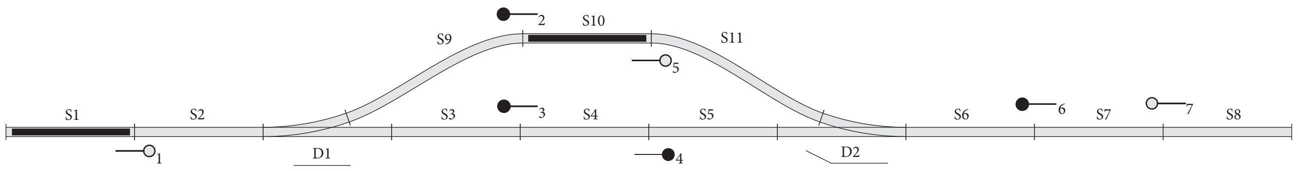

Figure 5: Example of safe situation.

(ii) there is a light signal controlling the pass from S_i to S_j, and it indicates "proceed" (for instance, the light signal —○1 controlling the movement from S_1 to S_2 in Figure 3, indicates "proceed"),

(iii) there is no light signal controlling the pass from S_i to S_j, and they are not connected by a turnout, but there is a light signal controlling the pass from S_j to S_i (like S_2 and S_1 in Figure 3).

We provide afterwards a recursive definition illustrating the idea that a train may reach a given section.

Definition 1. The notion "a train t_i can reach section S_j" is recursively defined as

(i) if train t_i is in section S_j then it can reach section S_j,

(ii) if train t_i is in section S_l and it can reach section S_k and it is possible to pass from section S_k to section S_j according to the position of the switches and the color of the light signals, then train t_i can reach section S_j.

Definition 2. Let M be a positive integer, $M \leq N$, and let $\{w_1, w_2, \ldots, w_M\} \subseteq \{1, 2, \ldots, N\}$. The position of M trains in sections $S_{w_1}, \ldots, S_{w_M}$ (one in each section) is dangerous if and only if there is a section S_i reachable from two sections S_j and S_k $(j \neq k)$ where two different trains are placed.

4. Overview of Our Approach

4.1. Description. In this section we will describe the approach proposed for determining the safety of a railway network composed of N sections, denoted as $S_1, \ldots, S_N$.

Our approach is based on defining a set of Boolean polynomials (that is to say, polynomials whose coefficients are in $\{0, 1\}$) in variables $s_1, \ldots, s_N, t_1, \ldots, t_N$. A variable s_i and a variable t_i are considered for each section S_i in the railway network. More precisely,

(i) a polynomial variable, s_i, is assigned to each section S_i;

(ii) another polynomial variable, t_i, is considered for each section S_i. Let us underline that variables t_i represent neither a section nor a train. The polynomial variables t_i will be used to introduce both the allowed connectivity between sections (see the paragraphs immediately below and Section 5) and the sections occupancy (through the membership of some of these polynomial variables to a certain polynomial monomial that introduces the positions of the trains).

For a configuration of the railway network (defining the position of the switch of each turnout and the color of each light signal), we will consider the following steps.

Step 1. Obtain a set K of polynomials according to the connectivity of the railway network: if it is possible to pass from section S_i to section S_j, we will consider the polynomial $t_i + s_i t_j$. That is to say, we have that

$$K = \{t_i + s_i t_j \mid \text{it is possible to pass from section } S_i \text{ to section } S_j\} \tag{1}$$

Step 2. Calculate the Groebner basis, G, of the following ideal:

$$I = K + \{s_1^2 + s_1, \ldots, s_N^2 + s_N, t_1^2 + t_1, \ldots, t_N^2 + t_N\} \tag{2}$$

(with respect to a certain order for the monomials and a certain order for the polynomial variables). Although Groebner bases [25–27] is a complicated concept of algebra to be completely detailed here, in Section 5.1, we will describe those properties of this concept which are relevant for our purpose.

Once these two steps are calculated, we can determine whether the position of a set of trains is dangerous or not. In this approach, a train only can be placed in one section at the same time, however, as we will see later in Section 7, we can extend the proposed model to take into account when a train occupies several sections at the same time.

Step 3. Let M be a positive integer, $M \leq N$, and let $\{w_1, w_2, \ldots, w_M\} \subseteq \{1, 2, \ldots, N\}$. For determining if the position of M trains in sections $S_{w_1}, S_{w_2}, \ldots, S_{w_M}$ (one in each section) is dangerous, we calculate the normal form of the monomial $t_{w_1}t_{w_2}\ldots t_{w_M}$ modulo I (I can be given through the Groebner basis G previously calculated):

$$\mathrm{NF}\left(t_{w_1}t_{w_2}\ldots t_{w_M}, I\right) \tag{3}$$

Like Groebner bases, the concept of normal form is also enough complicated to be completely described here. Nevertheless, in Section 5.1, we will describe those properties relevant for our purpose.

In this step we can detect if the position of the trains is dangerous. Indeed, as we will see in Theorem 9, we have that the position of the trains is dangerous if and only if $\mathrm{NF}(t_{w_1}t_{w_2}\ldots t_{w_M}, I)$ contains a number of t_α variables lower than M.

As we will see, Step 1 and Step 2 are only needed to be performed again if the configuration of the railway network changes. Only Step 3 must be performed (which is very fast done, around 0.0001 seconds) for different placement of the trains. As we have previously stated, this is an important advantage of the present approach over the previous approaches, which require to completely run the algorithm for different positions of the trains.

4.2. Example of the Approach. Let us consider the example of the railway network depicted in Figure 3. Since this railway network contains 11 sections, we will make use of polynomials in variables $s_1, \ldots, s_{11}, t_1, \ldots, t_{11}$.

Step 1. We calculate the set K of $t_i + s_i t_j$ polynomials related to the connectivity of the railway network (defined in Step 1 of Section 4.1):

$$\begin{aligned} K = \{&t_1 + s_1t_2, t_2 + s_2t_1, t_2 + s_2t_3, t_3 + s_3t_2, t_3 + s_3t_4, t_5 \\ &+ s_5t_4, t_9 + s_9t_{10}, t_{10} + s_{10}t_{11}, t_{11} + s_{11}t_{10}, t_6 \\ &+ s_6t_{11}, t_6 + s_6t_7, t_7 + s_7t_8, t_8 + s_8t_7\} \end{aligned} \tag{4}$$

(for instance, $t_1+s_1t_2$ is included in K because it is possible to pass from section S_1 to section S_2). Observe that trailing through a switch set against is not allowed. Then it is as if the turnout was guarded by (possibly) nonexistent light signals. For instance, it is considered that moving from s_9 to s_2 and from s_5 to s_6 is forbidden.

Step 2. We calculate the Groebner basis of the ideal:

$$I = K + \left\{s_1^2 + s_1, \ldots, s_{11}^2 + s_{11}, t_1^2 + t_1, \ldots, t_{11}^2 + t_{11}\right\} \ : \tag{5}$$

and

$$\begin{aligned} G = \{&t_1 + s_1t_2, t_2 + s_2t_1, t_2 + s_2t_3, t_3 + s_3t_2, t_3 + s_3t_4, t_5 \\ &+ s_5t_4, t_9 + s_9t_{10}, t_{10} + s_{10}t_{11}, t_{11} + s_{11}t_{10}, t_6 \\ &+ s_6t_{11}, t_6 + s_6t_7, t_7 + s_7t_8, t_8 + s_8t_7\} \end{aligned} \tag{6}$$

is obtained.

Step 3. Once the Groebner basis is calculated, we can immediately determine if the position of a set of trains is dangerous for this configuration of the railway network.

(i) As example, if there are three trains respectively placed in sections $S_1S_{10}S_8$, in order to determine if the position of the trains is dangerous, we just simply need to calculate:

$$\mathrm{NF}\left(t_1t_{10}t_8, I\right) = t_4t_{11} \tag{7}$$

As may be seen, there are only two variables t_α in $\mathrm{NF}(t_1t_{10}t_8, I) = t_4t_{11}$, which is lower than the number of trains. Therefore, we determine that the position of the trains in $\{S_1, S_{10}, S_8\}$ is dangerous (as may be seen, section S_7 is reachable from sections S_8 and section S_{11}, where trains are placed).

(ii) As example, if there are now two trains respectively placed in sections S_1, S_{10}, in order to determine if the the position of the trains is dangerous we just simply need to calculate:

$$\mathrm{NF}\left(t_1t_{10}, G\right) = t_1t_{11} \tag{8}$$

As may be seen, there are two variables t_α in $\mathrm{NF}(t_1t_{10}, I) = t_1t_{10}$, which is equal to the number of trains in the railway network. Therefore, we determine that the position of the trains in $\{S_1, S_{10}\}$ is not dangerous.

5. Theoretical Foundations of Our Approach

In this section we will detail the theoretical foundations of our algebraic approach for the problem of detecting dangerous situation in a railway network. In Section 5.1, we will give some outlines about concepts of computer algebra (like Boolean polynomials, ideals, Groebner bases, normal forms) which are relevant for this paper. In Section 5.2 we will describe the problem of detecting dangerous situation in an interlocking system by means of states and transformation of the states by means of operations. As we will see in Section 5.3, these states may be represented in algebraic terms (by means of Boolean monomials) which will allow us to determine whether the position of the trains is dangerous or not.

5.1. Some Introductory Notes about Boolean Polynomials and Ideals. In this section we will describe some outlines about Boolean polynomial. A polynomial $p \in \mathbb{Z}_2[x_1, \ldots, x_N]$ is a polynomial in the variables $x_1, \ldots, x_N$ whose coefficients lie in the field $\mathbb{Z}_2 = \{0, 1\}$. An example of this kind of polynomials is

$$p = x_1x_2^7 + x_3 + x_2x_3x_4 \tag{9}$$

We must take into account the fact that the coefficients lie in $\mathbb{Z}_2$, and therefore, we have that (since 1+1=0 in $\mathbb{Z}_2$), for example,

$$x_1x_2 + x_3x_4 + x_3x_4 = x_1x_2 + (1+1)\,x_3x_4 = x_1x_2 \tag{10}$$

As we have previously seen, the concept of ideal will play an important role in our approach. An ideal is a special kind of subset of a ring: it is also a ring and has the curious property that the product of an element of the subring by an element of the ring (and vice versa, if the ring is not a commutative one) always belongs to the subring. For instance, the set of even numbers is an ideal of the integers. In our case, the ideal J, generated by $p_1, \ldots, p_m$, denoted $J = \langle p_1, \ldots, p_m \rangle$, turns out to be the set of polynomials of the following form:

$$\beta_1 p_1 + \ldots + \beta_m p_m \tag{11}$$

where $\beta_1, \ldots, \beta_N$ are polynomials in $\mathbb{Z}_2[x_1, \ldots, x_N]$, that is, J is the set of algebraic combinations of $p_1, \ldots, p_m$.

Given a polynomial p, we define $p + J$ as the set of polynomials of the form $p + q$ where $q \in J$. It is possible that $p + J = q + J$ although $p \neq q$.

Example 3. Let us consider the ideal $J = \langle x_1 + x_2 x_3, x_2 + x_3 x_4 \rangle$ and the polynomial $p = x_1 x_4 + x_2$. We have that

$$\begin{aligned} p + J &= x_1 x_4 + x_2 + J \\ &= x_2 x_3 x_4 + x_2 + J \\ &\qquad\qquad (x_1 = x_2 x_3 \textit{ because } x_1 + x_2 x_3 \in J) \\ &= x_3^2 x_4^2 + x_2 + J \\ &(1\textit{st ocurrence of} x_2 = x_3 x_4 \textit{ because } x_2 + x_3 x_4 \in J) \\ &= x_3^2 x_4^2 + x_3 x_4 + J \\ &(2\textit{nd ocurrence of } x_2 = x_3 x_4 \textit{ because } x_2 + x_3 x_4 \in J) \end{aligned} \tag{12}$$

In this paper, we will consider polynomials in $\mathbb{Z}_2[s_1, \ldots, s_N, t_1, \ldots, t_N]$. The ideal I defined in Section 4 is generated by the following polynomials:

(i) The polynomials $s_i^2 + s_i, t_i^2 + t_i \in I, i = 1, \ldots n,$. This involves that, for every polynomial p, we can get a polynomial q such that $p + I = q + I$ and the exponents of all the variables in q is 1.

Example 4. Let us consider, for example, the ideal I and the polynomial $p = s_1^2 t_2^3 + s_1 t_2 + s_1^3$. We have that

$$\begin{aligned} p + I &= s_1^2 t_2^3 + s_1 t_2 + s_1^3 + I \\ &= s_1 t_2^2 t_2 + s_1 t_2 + s_1^2 s_1 + I \\ &\qquad\qquad \left(s_1^2 = s_1,\ t_2^3 = t_2^2 t_2 \textit{ and } s_1^3 = s_1^2 s_1\right) \\ &= s_1 t_2 t_2 + s_1 t_2 + s_1 s_1 + I \\ &\qquad\qquad\qquad \left(t_2^2 = t_2 \textit{ and } s_1^2 = s_1\right) \\ &= s_1 t_2^2 + s_1 t_2 + s_1^2 + I \\ &\qquad\qquad\qquad \left(t_2 t_2 = t^2 \textit{ and } s_1 s_1 = s_1^2\right) \\ &= s_1 t_2 + s_1 t_2 + s_1 + I \quad \left(t_2^2 = t_2 \textit{ and } s_1^2 = s_1\right) \\ &= s_1 + I \quad (2 s_1 t_2 = 0) \end{aligned} \tag{13}$$

Remember that (as we have seen before)

$$K = \{t_i + s_i t_j \mid \text{it is possible to pass from section } S_i \text{ to section } S_j\} \tag{14}$$

and

$$I = K + \left\{s_1^2 + s_1, \ldots, s_N^2 + s_N, t_1^2 + t_1, \ldots, t_N^2 + t_N\right\} \tag{15}$$

(ii) The polynomial $t_i + s_i t_j \in I$ if it is possible to pass from section S_i to section S_j. This involves that for every monomial $t_1 \cdot \ldots \cdot t_n$ we can obtain other monomials in variables t and s.

Example 5. In the example of the railway network described in Figure 3, we have that $p = t_1 t_{10} t_8$:

$$\begin{aligned} p + I &= t_1 t_{10} t_8 + I \\ &= t_1 t_{10} s_8 t_7 + I \\ &\qquad\qquad (t_8 = s_8 t_7 \textit{ because } t_8 + s_8 t_7 \in I) \\ &= t_1 s_{10} t_{11} s_8 t_7 + I \\ &\qquad\qquad (t_{10} = s_{10} t_{11} \textit{ because } t_{10} + s_{10} t_{11} \in I) \\ &= t_1 s_{10} s_{11} t_6 s_8 t_7 + I \\ &\qquad\qquad (t_{11} = s_{11} t_6 \textit{ because } t_{11} + s_{11} t_6 \in I) \\ &= t_1 s_{10} s_{11} s_6 t_7 s_8 t_7 + I \\ &\qquad\qquad (t_6 = s_6 t_7 \textit{ because } t_6 + s_6 t_7 \in I) \\ &= t_1 s_{10} s_{11} s_6 t_7^2 s_8 + I \quad \left(t_7 t_7 = t_7^2\right) \\ &= t_1 s_{10} s_{11} s_6 t_7 s_8 + I \quad \left(t_7^2 = t_7\right) \\ &= t_1 s_{10} s_{11} s_6 t_8 + I \\ &\qquad\qquad (s_8 t_7 = t_8 \textit{ because } t_8 + s_8 t_7 \in I) \end{aligned} \tag{16}$$

By means of these substitutions induced by the ideal I, we can obtain, for every monomial p, another monomial q whose variables are to power 1 and such that $p + I = q + I$.

An important solved issue in computer algebra is to find the "simplest" monomial q such that $q + I = p + I$. In a formal way, we pose the problem of finding the minimal polynomial q (where a total order between monomials, $\preceq$, must be defined) such that $p + I = q + I$. This polynomial q is what is called normal form of p modulo the ideal I (under the order $\preceq$), $\mathrm{NF}(p, I)$, which can be calculated by performing pseudodivisions of the polynomial p by a Groebner basis of the ideal I (which are required to be previously calculated). In this way, the concept of normal form requires defining a monomial order.

For the purpose of this paper, we will define the following order between monomials:

(i) If p contains a lower number of t_α variables than q, we have that $p \preceq q$.

(ii) If p contains exactly the same number of t_α variables as q, we have that $p \preceq q \Longleftrightarrow p \preceq_{Lex} q$, where $\preceq_{Lex}$ is the typical lexicographical order used in computer algebra.

By means of this order $\preceq$, the normal form of p modulo I, $\mathrm{NF}(p, I)$, returns the monomial q with minimal number of variables in t such that $p + I = q + I$. Next equations are examples of $\preceq$ order used in this approach:

$$\begin{aligned} s_3 \preceq t_1 &= True \\ t_1 \preceq s_3 &= False \\ t_2 \preceq t_1 &= True \\ t_1 \preceq t_2 &= False \\ s_1 \preceq s_2 &= False \end{aligned} \tag{17}$$

5.2. An Approach to the Railway Interlocking Problem Based on States. Here we will see how the interlocking problem can be described by means of states which are transformed by means of operations. On the one hand a state describes the sections where the trains are presently placed in the railway network at a given moment; on the other one, past states describe the sections where the trains were placed in the railway network in the past. Operations of the states (i.e., transitions between states) represent possible movements of the trains in the railway network.

For our purpose, each state is described by a subset of variables $s_1, \ldots, s_N, t_1, \ldots, t_N$.

The following are examples of possible states of the railway network in Figure 3: $\mathscr{E}_0 = \{t_1 t_{10} t_8\}$, $\mathscr{E}_1 = \{t_1 t_{10} s_8 t_7\}$, $\mathscr{E}_2 = \{t_1 s_{10} t_{11} s_8 t_7\}$, $\mathscr{E}_3 = \{t_1 s_{10} s_{11} t_6 s_8 t_7\}$, $\mathscr{E}_4 = \{t_1 s_{10} s_{11} s_6 t_7 s_8 t_7\}$, $\mathscr{E}_5 = \{t_1 s_{10} s_{11} s_6 t_8\}$.

We can transform a state $\mathscr{E}_i$ into another one $\mathscr{E}_j$ by means of one of these operations (we will denote $\mathscr{E}_i \longrightarrow \mathscr{E}_j$):

(i) When it is possible, substitute all t_i in $\mathscr{E}_i$ by the monomial $s_i t_j \in I$ if it is possible to pass from S_i to S_j. For example, we have that $\mathscr{E}_0 \longrightarrow \mathscr{E}_1$ ($\mathscr{E}_0 = \{t_1 t_{10} t_8\}$ and $\mathscr{E}_1 = \{t_1 t_{10} s_8 t_7\}$) by substituting t_8 by $s_8 t_7$, because $t_8 + s_8 t_7 \in I$.

(ii) When it is possible, substitute all $s_i t_j$ in $\mathscr{E}_i$ by the variable $t_i \in I$ if it is possible to pass from S_i to S_j. For example, we have that $\mathscr{E}_4 \longrightarrow \mathscr{E}_5$ ($\mathscr{E}_4 = \{t_1 s_{10} s_{11} s_6 t_7 s_8 t_7\}$ and $\mathscr{E}_5 = \{t_1 s_{10} s_{11} s_6 t_8\}$) by substituting $s_8 t_7$ by t_8, because $t_8 + s_8 t_7 \in I$.

Variables t and s in these states inform about possible movements of the trains. Variables t indicate the present position of the trains, and variables s the sections through which these trains have gone. Indeed,

(i) the state $\mathscr{E}_0 = \{t_1 t_{10} t_8\}$ represents that there are trains in the sections S_1, S_{10}, S_8 (variables t_1, t_{10}, t_8 appear in $\mathscr{E}_0$);

(ii) the state $\mathscr{E}_1 = \{t_1 t_{10} s_8 t_7\}$ represents that there are trains in the sections S_1, S_7, S_{10} (variables t_1, t_7, t_{10} appear in $\mathscr{E}_1$) and that (at least) one of these trains has passed through the section S_8 (variable s_8 appears in $\mathscr{E}_1$);

(iii) the state $\mathscr{E}_2 = \{t_1 s_{10} t_{11} s_8 t_7\}$ represents that there are trains in the sections S_1, S_7, S_{11} (variables t_1, t_7, t_{11} appear in $\mathscr{E}_2$) and that (at least) one of these trains has passed through the sections S_8, S_{10} (variables s_{10}, s_8 appear in $\mathscr{E}_2$);

(iv) the state $\mathscr{E}_3 = \{t_1 s_{10} s_{11} t_6 s_8 t_7\}$ represents that there are trains in the sections S_1, S_6, S_7 (variables t_1, t_6, t_7 appear in $\mathscr{E}_3$) and that (at least) one of these trains has passed through the sections S_8, S_{10}, S_{11} (variables s_8, s_{10}, s_{11} appear in $\mathscr{E}_3$).

As may be seen, states $\mathscr{E}_0, \mathscr{E}_1, \mathscr{E}_2, \mathscr{E}_3$ contain the same number of variables t representing the position of the trains in different times. However, we can transform the state $\mathscr{E}_3 = \{t_1 s_{10} s_{11} t_6 s_8 t_7\}$ into the state $\mathscr{E}_4 = \{t_1 s_{10} s_{11} s_6 t_7 s_8 t_7\} = \{t_1 s_{10} s_{11} s_6 t_7 s_8\}$ by substituting t_6 with $s_6 t_7$ (the train which was in section S_6 moves to section S_7). In this case, the number of "t" variables is lower than the cardinal of $\mathscr{E}_3$, since the variable t_7 (that we add with the operation) indeed belongs to the previous state $\mathscr{E}_3$. The operation means that the train which was in section S_6 moves to section S_7, and, since section S_7 was occupied by another train, a possible collision may happen.

Following this reasoning, we have the following.

Proposition 6. *Let M be a positive integer, $M \leq N$, and let $\{w_1, w_2, \ldots, w_M\} \subseteq \{1, 2, \ldots, N\}$. The position of M trains in sections $S_{w_1}, S_{w_2}, \ldots, S_{w_M}$ (one in each section) is dangerous $\Longleftrightarrow$ the state $\mathscr{E}_0 = \{t_1 \ldots t_M\}$ can be transformed into a state $\mathscr{E}_z$ with a number of t_α variables lower than M.*

In the previous example we had $\{t_1 t_{10} t_8\} = \mathscr{E}_0 \longrightarrow \mathscr{E}_1 \longrightarrow \mathscr{E}_3 \longrightarrow \mathscr{E}_4 = \{t_1 s_{10} s_{11} s_6 t_7 s_8\}$. According to the previous proposition, the position of the trains in sections S_1, S_{10}, S_8 (related to the state $\mathscr{E}_0 = \{t_1 t_{10} t_8\}$) is dangerous in the railway network (as we saw in Section 3).

5.3. Interpretation of States in Algebraic Terms. In this section, we will represent each of the states described in the previous section by means of a Boolean monomial. As we will see, this representation will be useful for determining whether the position of the trains is dangerous or not.

We represent each state $\mathscr{E}$, a set consisting in one monomial, by means of precisely such monomial, so $\varphi(\mathscr{E}) \in \mathbb{Z}_2[s_1 \ldots, s_N, t_1 \ldots, t_N]$, and can be expressed as follows:

$$\varphi(\mathscr{E}) = \prod_{i \in \widehat{\mathscr{S}}} s_i \prod_{j \in \widehat{\mathscr{T}}} t_j \tag{18}$$

where $\widehat{\mathscr{S}}, \widehat{\mathscr{T}} \subseteq \{1, \ldots, N\} \wedge \widehat{\mathscr{T}} \neq \emptyset$. In this way, we have the following:

(i) $\varphi(\mathscr{E}_0) = \varphi(\{t_1 t_{10} t_8\}) = t_1 t_{10} \mathrm{t}_8$

(ii) $\varphi(\mathscr{E}_1) = \varphi(\{t_1 t_{10} s_8 t_7\}) = t_1 t_{10} s_8 t_7$

(iii) $\varphi(\mathscr{E}_2) = \varphi(\{t_1 s_{10} t_{11} s_8 t_7\}) = t_1 s_{10} t_{11} s_8 t_7$

(iv) $\varphi(\mathscr{E}_3) = \varphi(\{t_1 s_{10} s_{11} t_6 s_8 t_7\}) = t_1 s_{10} s_{11} t_6 s_8 t_7$

(v) $\varphi(\mathscr{E}_4) = \varphi(\{t_1 s_{10} s_{11} s_6 t_7 s_8\}) = t_1 s_{10} s_{11} s_6 t_7 s_8$

As we saw in Section 5.1, the operations previously defined for transforming a state $\mathscr{E}_1$ into $\mathscr{E}_2$ are completely related to the fact that $\varphi(\mathscr{E}_1) + I = \varphi(\mathscr{E}_2) + I$. Therefore, we have the following.

Proposition 7. *Let $\mathscr{E}_a, \mathscr{E}_b$ be two states. We have that*

$$\begin{gathered} \mathscr{E}_a \longrightarrow \mathscr{E}_b \Longleftrightarrow \\ \varphi(\mathscr{E}_a) + I = \varphi(\mathscr{E}_b) + I \end{gathered} \tag{19}$$

Let us clarify this proposition with the example of Figure 6. In this case we have that $K = \{t_1 + s_1 t_2\}$ because the light signal represented by $-\!\circ_1$ indicates "proceed", thus $I = K + \{t_1 + t_1^2, t_2 + t_2^2, s_1 + s_1^2\}$.

We can define as possible states $\mathscr{E}_a, \mathscr{E}_b, \mathscr{E}_c$ where

$$\begin{aligned} \varphi(\mathscr{E}_a) &= t_1 + I \quad (\text{train } t_1 \text{ is placed in } s_1) \\ \varphi(\mathscr{E}_b) &= t_2 + I \quad (\text{train } t_2 \text{ is placed in } s_2) \\ \varphi(\mathscr{E}_c) &= s_1 t_2 + I \quad (t_1 = s_1 t_2 \text{ because } t_1 + s_1 t_2 \in I) \end{aligned} \tag{20}$$

State $\mathscr{E}_c$ is equivalent to state $\mathscr{E}_a$ and different from state $\mathscr{E}_b$ ($\mathscr{E}_a \longrightarrow \mathscr{E}_c$ and $\mathscr{E}_c \longrightarrow \mathscr{E}_a$). Calculating $\mathrm{NF}(\varphi(\mathscr{E}_a) \cdot \varphi(\mathscr{E}_b))$ that represents two trains placed in sections s_1 and s_2 respectively, we have that

$$\begin{aligned} \mathrm{NF}(\varphi(\mathscr{E}_a) \cdot \varphi(\mathscr{E}_b)) &= t_1 t_2 + I \\ &= s_1 t_2 t_2 + I \\ &\quad (t_1 = s_1 + t_2 \text{ because } t_1 + s_1 t_2 \in I) \\ &= s_1 t_2 + I \quad (t_2^2 = t_2) \end{aligned} \tag{21}$$

Consequently, Proposition 6 can be stated in the following way.

Proposition 8. *Let us consider a railway network composed of N sections, let M be a positive integer, $M \leq N$ and let $\{w_1, w_2, \ldots, w_M\} \subseteq \{1, 2, \ldots, N\}$. The position of M trains in sections $S_{w_1}, S_{w_2}, \ldots, S_{w_M}$ (one in each section) is dangerous $\Longleftrightarrow$ there is a monomial p with a number of t_α variables lower than M, such that $p + I = t_{w_1} t_{w_2} \ldots t_{w_M} + I$.*

In order to find this polynomial p, we make use of $\mathrm{NF}(t_{w_1} t_{w_2} \ldots t_{w_M}, I)$, the normal form of the monomial modulo the ideal I. According to Section 5.1, $\mathrm{NF}(t_{w_1} t_{w_2} \ldots t_{w_M}, I)$, returns the monomial p with minimal number of t_α variables such that $p + I = t_{w_1} t_{w_2} \ldots t_{w_M} + I$. Consequently, we have the following result.

Theorem 9. *Let us consider a railway network composed of N sections and let M be a positive integer, $M \leq N$. The position of M trains located in sections $S_{w_1}, S_{w_2}, \ldots, S_{w_M}$ is dangerous $\Longleftrightarrow$ $\mathrm{NF}(t_{w_1} t_{w_2} \ldots t_{w_M}, I)$ contains a number of t_α variables lower than M.*

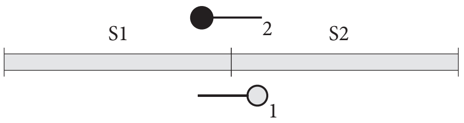

Figure 6: Two sections connected.

6. Evaluation and Discussion

In this section, we will analyze the performance of the approach presented here, comparing it with other techniques with available implementations. Indeed, we have made a comparison between the times required to calculate the safety of the railway network with different N sections and M trains. All experiments have been carried out on the same computer.

6.1. Comparing the Performances of the Logic, Algebraic, and Logic-Algebraic Models. In Table 1, a comparison between the times required to calculate the safety of certain railway networks in different situations using different models is shown. The Logic [28], Algebraic [24], and Logic-Algebraic [30] models were implemented in the computer algebra system *Maple*.

We can state that these three models are strongly dependent on the number of sections and trains. From the experiments carried out (and as can be seen in the first two columns of Table 1), we can affirm that timings grow very quickly with the number of sections. The timings obtained make it clear that these models are not appropriate for middle size or large size railway networks.

Regarding the influence of the number of trains on timings, they grow quickly both in the Logic and Logic-Algebraic models. In the Algebraic Model, the polynomial variable corresponding to the section is substituted by an integer number when a train is declared to be in a certain section. Therefore more trains mean less variables in the Groebner basis. So, curiously, if the number of sections is fixed, timings decrease if the number of trains increases. In the degenerated case of one different train in each section, we would have time to be equal to 0 in the Groebner basis computation, as there is no variable left.

6.2. Comparing the Performances of the ASP and the New Model. Meanwhile, the ASP model [31] and the new proposed model can be applied to large railway networks. We have used as example for the benchmarks Madrid Chamartin station in its state prior to the transformation of some of its tracks into international gauge [33], with 300 sections and more than 200 turnouts and light signals.

The ASP model was implemented in *Smodels*. Finally, the model introduced in this article has been implemented in *PolyBoRi* [32], an specialized software to calculate Groebner bases and normal forms on Boolean rings. Table 2 shows the times obtained considering different numbers of trains (Times in *Smodels* are calculated taking into account the *lparse* and *smodels* executions. We apply the same rule in *PolyBoRi*.).

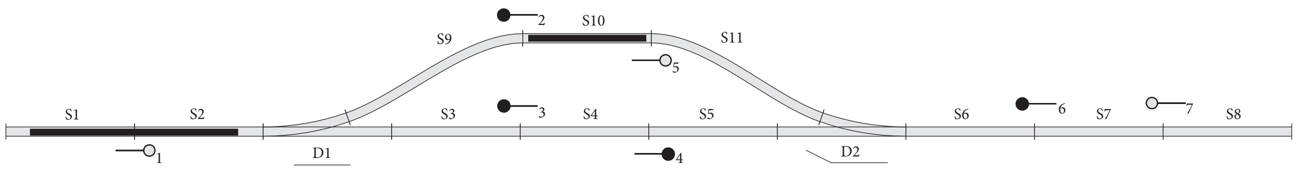

Figure 7: Possible situation of trains occupy several sections at the same time. The situation is safe.

Table 1: Time comparative of methods implemented in for detecting the safety in a railway interlocking system with N sections and M trains.

	$M = 15$ $N = 32$	$M = 15$ $N = 64$	$M = 45$ $N = 93$	$M = 60$ $N = 127$	$M = 65$ $N = 160$	$M = 105$ $N = 224$
Logic	10.23 s	250.250 s	> 1 h	> 1h	> 1h	> 1h
Algebraic	15.356 s	380.368 s	> 1 h	> 1h	> 1h	> 1h
Logic-Algebraic	0.005 s	0.060 s	1.1 s	3.348 s	10.323 s	38.392 s

Table 2: Time comparative between the ASP model and the new model, with 300 sections and $M = 5, 15, 30$ trains.

	$M = 5$ $N = 300$	$M = 15$ $N = 300$	$M = 30$ $N = 300$
ASP	0.646s	0.716s	0.799s
New model (GB preprocess)	2.3s	2.3s	2.3s
New model (NF)	0.0001s	0.00039s	0.0054s

The main results from Table 2 are, one the one hand, that the efficiency of our method is always high superior to the other approaches tested when the GB is previously calculated; on the other hand, the time to calculate the GB is reasonably fast (e.g., ~ 2.3s for Chamartin Station), providing very competitive times for large networks. Moreover, the Groebner bases computed in the preprocess step can be stored and used in case the same situation is proposed again. Let us observe that our two steps method connects somehow with the traditional methods that check the compatible routes in advance.

In our approach, computation timings for the Groebner bases preprocess grow when the number of sections increases (as happens in all the other models).

But, unlike the other approaches, ours does not need to perform all the calculations for the different positions of the trains for a given specific particular state of the turnouts and light signals. Indeed, we only need to calculate the normal form of one monomial (once the Groebner basis is calculated for this particular state of turnouts and light signals), as detailed in Theorem 9. The time for calculating this normal form grows with the number of trains, as can be seen in Table 2. Nevertheless, these timings of this second step have always been negligible (lower than 0.01s in all the experiments carried out, regardless of the size of the railway network or the number of trains).

On the contrary, the other approaches require to recalculate all the computations.

6.3. Comparison: Final Remarks. The approach presented in this paper only focuses on the safeness of the proposed situation; meanwhile other approaches like [23, 34] also consider other properties such as reachability of sections or the situation of the switches of the turnouts under a train (that shouldn't be changed under any circumstance to avoid derailments due to a premature change of the switch during the transit of the train). Another issue not treated in our model is the confirmation that the switches of the turnouts and the light signals have really been set to the ordered situation (something considered, for instance, in [34]; this latter approach uses a real-time system, a flexible approach with a wide variety of applications [21]).

On the other hand, our new approach can handle really big layouts (like the one in the example of Table 1, with, as said above, 300 sections and more than 200 turnouts and light signals, what is a far bigger layout than any one that we have found treated in real time). For instance, the authors of [34] mention that "our study is unique in the fact that modeling and verification of mid to large size railway yard is being undertaken in UPPAAL model checker (...)" and the example presented (Rawalpindi Cantt station) has 25 semaphores and 27 mechanically operated points.

7. Model Extension for Trains Occupying Several Sections at the Same Time

Two trains are represented in Figure 7, one placed on section S_{10} and another one placed on sections S_1, S_2 at the same time. It is common to find these situations in a real railway when several trains can occupied several sections at the same time.

In order to check the safety of a railway network with M trains that can occupy several sections at the same time, we will redefine Step 3 of Section 4 as follows.

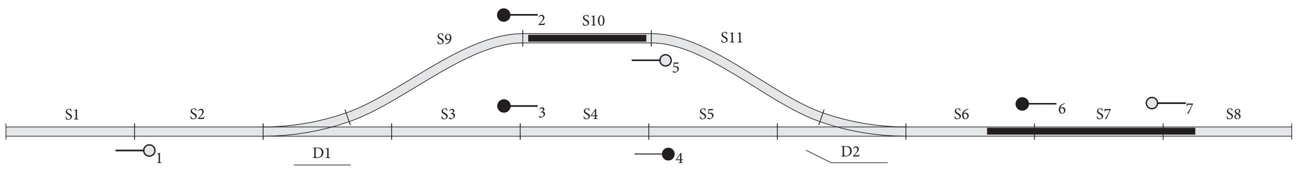

Figure 8: Possible situation of trains occupying several sections at the same time. The situation is dangerous.

Step 3.1. As long as $\mathcal{E}$ represents the sections occupied by a given train T, we calculate $\mathrm{NF}(\varphi(\mathcal{E}), I) = \mathcal{E}'$. $\mathcal{E}'$ is a reduced initial state representing the sections occupied by a train.

Step 3.2. Once M initial states $\mathcal{E}'_1, \ldots, \mathcal{E}_{M'}$ have been calculated, we calculate the joint state $\mathcal{E}''$ which $\varphi(\mathcal{E}'') = \mathrm{NF}(\varphi(\mathcal{E}'_1) \cdot \ldots \cdot \varphi(\mathcal{E}'_M), I)$ of all initial states.

Step 3.3. For determining if the position of M trains with initial states $\mathcal{E}'_1, \ldots, \mathcal{E}_{M'}$ is dangerous, we need to check if

$$\left(\sum_{i=1}^{M} \eta\left(\mathcal{E}'_i\right)\right) > \eta\left(\mathcal{E}''\right) \tag{22}$$

where $\eta(\mathcal{E}) \longrightarrow \mathbb{N}$ is a function that returns the number of t_α variables in the monomial $\varphi(\mathcal{E})$.

Example 10. As example, the safety of the railway network on Figure 7 will be checked.

Step 3.1. The first train is represented by $\mathcal{E}_1 = \{t_1 t_2\}$ and the second one is represented by $\mathcal{E}_2 = \{t_{11}\}$. Therefore, the initial train states will be

$$\begin{aligned} \mathcal{E}'_1 &= \mathrm{NF}\left(t_1 t_2, I\right) = t_4 s_3 \\ \mathcal{E}'_2 &= \mathrm{NF}\left(t_{10}, I\right) = t_{11} \end{aligned} \tag{23}$$

Step 3.2. Joint calculations of all initial states are

$$\mathcal{E}'' = \mathrm{NF}\left(\varphi\left(\mathcal{E}'_1\right) \cdot \varphi\left(\mathcal{E}'_2\right), I\right) = t_4 t_{11} \tag{24}$$

Step 3.3. Check the number of variables t in both monomials:

$$\eta\left(\mathcal{E}'_1\right) + \eta\left(\mathcal{E}'_2\right) = \eta\left(\mathcal{E}''\right) = 2 \tag{25}$$

so we have that the proposed situation of the trains on Figure 7 is safe.

Example 11. Let us propose another example where the situation in the railway network is dangerous. Figure 8 represents two trains, one located in S_{10} and another one located on S_6, S_7, S_8. In order to check if exists a dangerous situation, we will calculate the following.

Step 3.1. The first train is represented by $\mathcal{E}_1 = \{t_{10}\}$ and the second one is represented by $\mathcal{E}_2 = \{t_6 t_7 t_8\}$. Therefore, the initial train states will be as follows:

$$\begin{aligned} \mathcal{E}'_1 &= \mathrm{NF}\left(t_{10}, I\right) = t_{11} \\ \mathcal{E}'_2 &= \mathrm{NF}\left(t_6 t_7 t_8, I\right) = t_{11} \end{aligned} \tag{26}$$

Step 3.2. Joint calculations of all initial states are

$$\mathcal{E}'' = \mathrm{NF}\left(\varphi\left(\mathcal{E}'_1\right) \cdot \varphi\left(\mathcal{E}'_2\right), I\right) = t_{11}, \tag{27}$$

Step 3.3. Check the number of variables t in both monomials:

$$\eta\left(\mathcal{E}'_1\right) + \eta\left(\mathcal{E}'_2\right) > \eta\left(\mathcal{E}''\right) \tag{28}$$

So we have that the proposed situation of the trains on Figure 8 is dangerous.

8. Conclusions

In this paper we have presented a new algebraic model for railway interlocking systems. According to this new model, the position of the trains in a railway network is dangerous if and only if the normal form of a certain monomial (representing the present position of the trains) contains a lower number of t_α variables than the number of trains in the railway network. We have implemented this model on the computer algebra system *PolyBoRi* resulting in a very short program code. Moreover, we have compared the execution times with other very fast models implemented previously.

Appendix

Smodels Code

As said above, the code has been developed in Smodels. The maximum number of trains and sections has been set to 300 in the ring definition (function `__init__(self)`) in order to book memory space, but these values can be increased. The whole code is included afterwards, except

(i) the body of the definition of Buchberger function (that computes Groebner bases), taken from `http://polybori.sourceforge.net/doc/tutorial/tutorialse3.html` has been substituted by "...",

(ii) the input corresponding to the connectivity between sections, included in function `define_ideals(self)` has been shortened (substituting all the polynomials but three by "...").

The code used in the tests is as follows:

```
import time
from polybori.PyPolyBoRi import *
from polybori.gbcore import *
from polybori import *
from polybori.interred import *
from polybori.nf import *
from polybori.gbrefs import *
class Safety:

    def __init__(self):
        """ Variables ring declaration in Z2 by default in
        PolyBoRi
        """
        r = declare_ring([Block("t", 300), Block ("s", 300)],
                         globals());

    def _buchberger(self, l):
        ...
        ...
        return g

    def define_ideals(self):
        """ Define the main Ideal
        """
        # Init timer to calculate elapsed time
        t_start = time()

        # Ideal definition
        self.K = [t(2) + s(2) * t(1),
                  t(3) + s(3) * t(2),
                  ...
                  ...
                  (t(275) + s(275) * t(265))]

    def calculate_gb(self):
        """ Computes the Groebner Basis of a given Ideal K
        """
        # Calculate Groebner basis
        self.gb = self._buchberger(self.K)

    def get_gb_polys(self, gb):
        """ Returns the polynomials of a computed Groebner Basis
        """
        return [poly for poly in gb]

    def calculate_nf(self, poly):
        """ Computes and returns the normal form of a polynomial
        """
        return self.gb.nf(poly)

if __name__ == ‘ __main__’:
    t_start = time()
    # Create object
    safety = Safety()
    # Define main ideal
    safety.define_ideals()
```

```
# Calculate Groebner Basis
safety.calculate_gb()
print("Elapsed time: %f" % (time() - t_start))

 # Get polynomials from the Groebner basis
 print(safety.get_gb_polys(safety.gb))
 # Calculate Normal Form of a given polynomial
 t_start = time()
 print(safety.calculate_nf(t(0) * t(1) * t(4)))
 print("Elapsed time: %f" % (time() - t_start))
```

(In the last procedure that there are trains in sections 0, 1, and 4 introduced as data.)

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the research projects TIN2015-66471-P (Government of Spain) and CASI-CAM S2013/ICE-2845 (Comunidad Autónoma de Madrid).

References

[1] Anonymous, "Proyecto y obra del enclavamiento electrónico de la estación de Madrid-Atocha. Proyecto Técnico," Siemens, Madrid, 1988.

[2] Anonymous, "Microcomputer Interlocking Hilversum," Siemens, Munich, 1986.

[3] Anonymous, "Microcomputer Interlocking Rotterdam," Siemens, Munich, 1989.

[4] Anonymous, "Puesto de enclavamiento con microcomputadoras de la estación de Chiasso de los SBB," Siemens, Munich, 1989.

[5] L. Villamandos, *Sistema informático concebido por Renfe para diseñar los enclavamientos*, vol. 65, Va Libre, 1993.

[6] K. Winter, W. Johnston, P. Robinson, P. Strooper, and L. van den Berg, "Tool Support for Checking Railway Interlocking Designs," in *Proceedings of the 10th Australian Workshop on Safety Related Programmable Systems*, T. Cant, Ed., pp. 101–107, Australian Computer Society, Inc., Sydney, Australia, 2006.

[7] A. Borälv, "Case Study: Formal Verification of a Computerized Railway Interlocking," *Formal Aspects of Computing*, vol. 10, no. 4, pp. 338–360, 1998.

[8] D. Bjørner, "The FMERail/TRain Annotated Rail Bibliography," http://www2.imm.dtu.dk/db/fmerail/fmerail/, 2005.

[9] M. J. Morley, *Modelling British Rail's Interlocking Logic: Geographic Data Correctness*, Technical Report ECS-LFCS-91-186, Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, 1991.

[10] K. Nakamatsu, Y. Kiuchi, and A. Suzuki, "EVALPSN Based Railway Interlocking Simulator," in *Knowledge-Based Intelligent Information and Engineering Systems*, M. G. Negoita et al., Ed., pp. 961–967, Springer LNAI 3214, Berlin, Heidelberg, Germany, 2004.

[11] A. Janota, "Using Z specification for railway interlocking safety," *Periodica Polytechnica Transportation Engineering*, vol. 28, no. 1-2, pp. 39–53, 2000.

[12] K. M. Hansen, "Formalising Railway Interlocking Systems," in *Nordic Seminar on Dependable Computing Systems*, pp. 83–94, Department of Computer Science, Technical University of Denmark, Lyngby, Denmark, 1994.

[13] X. Chen, H. Huang, and Y. He, "Automatic Generation of Relay Logic for Interlocking System Based on Statecharts," in *Proceedings of the 2010 Second WRI World Congress on Software Engineering WCSE*, vol. 2, pp. 183–188, IEEE, Los Alamitos, CA, USA, 2010.

[14] X. Chen, Y. He, and H. Huang, "An approach to automatic development of interlocking logic based on statechart," *Enterprise Information Systems*, vol. 5, no. 3, pp. 273–286, 2011.

[15] X. Chen, Y. He, and H. Huang, "A component–based topology model for railway interlocking systems," *Mathematics and Computers in Simulation*, vol. 81, no. 9, pp. 1892–1900, 2011.

[16] M. Montigel, *Modellierung und Gewährleistung von Abhängigkeiten in Eisenbahnsicherungsanlagen (Ph.D. Thesis)*, ETH Zurich, Zurich, Switzerland, 1994, http://www.inf.ethz.ch/research/disstechreps/theses.

[17] A. E. Haxthausen, J. Peleska, and S. Kinder, "A formal approach for the construction and verification of railway control systems," *Formal Aspects of Computing*, vol. 23, no. 2, pp. 191–219, 2011.

[18] A. E. Haxthausen, "Automated generation of safety requirements from railway interlocking tables," in *ISoLA 2012, Part II*, T. Margaria and B. Steffen, Eds., pp. 261–275, Springer, LNCS 7610, Berlin, Germany, 2012.

[19] A. Ferrari, G. Magnani, D. Grasso, and A. Fantechi, "Model checking interlocking control tables," in *Proceedings of the Formal Methods for Automation and Safety in Railway and Automotive Systems FORMS/FORMAT 2010*, E. Schnieder and G. Tarnai, Eds., pp. 107–115, Springer, Berlin, Germany, 2011.

[20] K. Winter and N. J. Robinson, "Modelling large interlocking systems and model checking small ones," in *Proceedings of the 26th Australasian Computer Science Conference (ACSC'2003)*, M. Oudshoorn, Ed., vol. 16, pp. 309–316, Australian Computer Science Communications, 2003.

[21] C. Zhou and M. R. Hansen, *Duration Calculus: A Formal Approach to Real-Time Systems*, Monographs in Theoretical Computer Science. An EATCS Series, Springer-Verlag, NY, USA, 2004.

[22] S. Veloudis and N. Nissanke, "Duration Calculus in the specification of safety requirements," in *Formal Techniques in Real-Time and Fault-Tolerant Systems. FTRTFT 1998*, A. P. Ravn and H. Rischel, Eds., pp. 103–112, Springer, LNCS 1486, Berlin, Heidelberg, Germany, 1998.

[23] E. Roanes-Lozano and L. M. Laita, "An applicable topology-independent model for railway interlocking systems," *Mathematics and Computers in Simulation*, vol. 45, no. 1-2, pp. 175–183, 1998.

[24] E. Roanes-Lozano, E. Roanes-Macías, and L. M. Laita, "Railway interlocking systems and Gröbner bases," *Mathematics*, vol. 51, no. 5, pp. 473–481, 2000.

[25] A. G. Akritas, *Elements of Computer Algebra with Applications*, John Wiley Interscience, NY, USA, 1989.

[26] B. Buchberger, "An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal," *Journal of Symbolic Computation*, vol. 41, no. 3-4, pp. 475–511, 2006.

[27] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties and Algorithms*, Springer, Verlag, Berlin, Heidelberg, Germany, NY, USA, 1992.

[28] E. Roanes-Lozano, A. Hernando, J. Alonso, and L. M. Laita, "A logic approach to decision taking in a railway interlocking system using Maple," *Mathematics and Computers in Simulation*, vol. 82, no. 1, pp. 15–28, 2011.

[29] M. Davis, G. Logemann, and D. Loveland, "A machine program for theorem-proving," *Communications of the ACM*, vol. 5, pp. 394–397, 1962.

[30] A. Hernando, E. Roanes-Lozano, R. Maestre-Martínez, and J. Tejedor, "A logic-algebraic approach to decision taking in a railway interlocking system," *Annals of Mathematics and Artificial Intelligence*, vol. 65, pp. 317–328, 2012.

[31] E. Roanes-Lozano, J. A. Alonso, and A. Hernando, "An approach from answer set programming to decision making in a railway interlocking system," *Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas*, vol. 108, no. 2, pp. 973–987, 2014.

[32] M. Brickenstein and A. Dreyer, "A framework for Groebner-basis computations with Boolean polynomials," *Journal of Symbolic Computation*, vol. 44, no. 9, pp. 1326–1345, 2009.

[33] Anonymous, "Atlas of High Speed Rail in Spain," Fundación de los Ferrocarriles Españoles, Madrid, http://www.ave-altavelocidad.es/AtlasAV.pdf, 2017.

[34] U. Khan, J. Ahmad, T. Saeed, and S. H. Mirza, "On the real time modeling of interlocking system of passenger lines of Rawalpindi Cantt train station," *Complex Adaptive Systems Modeling*, vol. 4, no. 1, 2016.