

Article

# *n*-Ary Cartesian Composition of Multiautomata with Internal Link for Autonomous Control of Lane Shifting

Štěpán Křehlík 

CDV—Transport Research Centre, Líšeňská 33a, 636 00 Brno, Czech Republic; stepan.krehlik@cdv.cz;  
Tel.: +420-724-863-410

Received: 27 April 2020; Accepted: 19 May 2020; Published: 21 May 2020



**Abstract:** In this paper, which is based on a real-life motivation, we present an algebraic theory of automata and multi-automata. We combine these (multi-)automata using the products introduced by W. Dörfler, where we work with the cartesian composition and we define the internal links among multiautomata by means of the internal links' matrix. We used the obtained product of *n*-ary multi-automata as a system that models and controls certain traffic situations (lane shifting) for autonomous vehicles.

**Keywords:** hypergroup; automata theory; cartesian composition; autonomous vehicles; cooperative intelligent transport systems

---

## 1. Introduction

### 1.1. Motivation

Every physical object in real time and space is defined by its specific properties such as its position in spacetime, temperature, shape, or dimension. This set of properties may be regarded as a state in which the physical object evinces. While focusing on transport infrastructure, there are a lot of elements of characteristic properties. These characteristics are considered as states of elements in the transport infrastructure, for example, a state of traffic lights, a state of velocity, a state of mileage, or a state of traffic density. A clear description of such states is not the only requirement. The possibility of a state change is an equally important requirement. In this respect, the algebraic theory of automata suggests a specific tool (automata or multi-automata) that enables to change a state using the transition function and input words. In other words, the input symbol (or input word) is applied to a state by a transition function, and consequently, a new state is obtained. In the past, these structures were considered as systems that transmitted a specific type of information. Nowadays, this concept cannot sufficiently describe or even control real-life applications as these are too complex. In such complex and difficult systems, a change of a state causes a change of another state, for example, a train that passes a railroad crossing stops vehicles on a road. In current traffic, a driver of a vehicle usually responds to the change in the state of the surrounding elements in transport—generally, this is what we consider a human factor. Traffic situations will get more complicated when autonomous vehicles or even whole autonomous systems that control the traffic are included. In the case of autonomous systems, we suggest that every vehicle communicates with other vehicles or the infrastructure, and as a result, it influences the next state of a system. The fact that autonomous vehicles can communicate and send information about their state or information about a planned change of their state is considered as an advantage that makes the traffic infrastructure more effective. In this paper, we construct a cartesian composition of multi-automata, i.e., we combine some multi-automata (or automata) and we add internal links between their particular states. This new

approach to a system of multi-automata with an internal link (SMAil) is defined by matrices and enables to describe and control the above-mentioned problems.

### 1.2. Basic Terminology of the Hyperstructure Theory

Before we introduce a theory of automata, recall some basic notions of the algebraic hyperstructure theory (or theory of algebraic hypercompositional structures). For further reference see, for example, books [1,2]. A hypergroupoid is a pair  $(H, *)$ , where  $H$  is a nonempty set and the mapping  $* : H \times H \rightarrow \mathcal{P}^*(H)$  is a binary hyperoperation (or hypercomposition) on  $H$  (here  $\mathcal{P}^*(H)$  denotes the system of all nonempty subsets of  $H$ ). If  $a * (b * c) = (a * b) * c$  holds for all  $a, b, c \in H$ , then  $(H, *)$  is called a semihypergroup. If moreover the reproduction axiom, i.e., relation  $a * H = H = H * a$  for all  $a \in H$ , is satisfied, then the semihypergroup  $(H, *)$  is called hypergroup. By extensive hypergroup in the sense of hypercompositional structures we mean a hypergroup that  $\{a, b\} \subseteq a * b$  for all  $a, b \in H$ , i.e., that the elements are included in its “result of hyperoperation”. For the same application on this theory in other science or real-life problems see, for example, [3–7].

### 1.3. Some Notions of the (Multi-)Automata Theory

In the field of automata theory, several types of automata are considered. In this work, we focus on automata without outputs, i.e., a structure composed from a triad of an input alphabet, a set of states, and a transition function. For completeness, let us note that the automaton is a quintuple consisting of the above-mentioned triad plus an output alphabet, and an output function. For more details, see [8–11]. For details regarding terminology and some minor deviations from standard usage, see Novák, Křehlík, and Staněk [12]. Further on, we recall the following definition (notice that condition 1 is sometimes omitted if we regard a semigroup instead of a monoid).

**Definition 1** ([13]). *By a quasi-automaton we mean a structure  $\mathbb{A} = (I, S, \delta)$  such that  $I \neq \emptyset$  is a monoid,  $S \neq \emptyset$  and  $\delta : I \times S \rightarrow S$  satisfies the following condition:*

1. *There exists an element  $e \in I$  such that  $\delta(e, s) = s$  for any state  $s \in S$ .*
2.  *$\delta(y, \delta(x, s)) = \delta(xy, s)$  for any pair  $x, y \in I$  and any state  $s \in S$ .*

*The set  $I$  is called the input set or input alphabet, the set  $S$  is called the state set and the mapping  $\delta$  is called next-state or transition function. Condition 1 is called the unit condition (UC) while condition 2 is called the Mixed Associativity Condition (MAC).*

Next, we are going to work with the idea of a quasi-multiautomaton as a hyperstructure generalization of a quasi-automaton, see [14–18]. When adjusting the conditions imposed on the transition function  $\delta$ , it must be defined cautiously because we get a state on the left-hand side of Definition 1, condition 2, whereas we get a set of states on the right-hand side.

**Definition 2** ([13]). *A quasi-multiautomaton is a triad  $\mathbb{MA} = (I, S, \delta)$ , where  $(I, *)$  is a semihypergroup,  $S$  is a non-empty set and  $\delta : I \times S \rightarrow S$  is a transition function satisfying the condition:*

$$\delta(b, \delta(a, s)) \in \delta(a * b, s) \text{ for all } a, b \in I, s \in S. \tag{1}$$

*The semihypergroup  $(I, *)$  is called the input semihypergroup of the quasi-multiautomaton  $\mathbb{A}$  ( $I$  alone is called the input set or input alphabet), the set  $S$  is called the state set of the quasi-multiautomaton  $\mathbb{A}$ , and  $\delta$  is called next-state or transition function. Elements of the set  $S$  are called states, elements of the set  $I$  are called input symbols or letters. Condition (1) is called Generalized Mixed Associativity Condition (GMAC).*

In the theory of automata, ways to combine automata into one structure have been described. We will recall homogeneous, heterogeneous products, and cartesian composition, which was introduced by [9]. These products were constructed and investigated primarily on classical automata without output

in [19]. In the following definition, we recall all three types of products/compositions introduced by W. Dörfler

**Definition 3** ([20]). Let  $\mathbb{A}_1 = (I, S, \delta)$ ,  $\mathbb{A}_2 = (I, R, \tau)$  and  $\mathbb{B} = (J, T, \sigma)$  be quasi-automata. By the homogeneous product  $\mathbb{A}_1 \times \mathbb{A}_2$  we mean the quasi-automaton  $(I, S \times R, \delta \times \tau)$ , where  $\delta \times \tau : I \times (S \times R) \rightarrow S \times R$  is a mapping satisfying, for all  $s \in S, r \in R, a \in I$ ,  $(\delta \times \tau)(a, (s, r)) = (\delta_1(a, s), \tau(a, r))$ , while the heterogeneous product  $\mathbb{A}_1 \otimes \mathbb{B}$  is the quasi-automaton  $(I \times J, S \times T, \delta \otimes \sigma)$ , where  $\delta \otimes \sigma : (I \times J) \times (S \times T) \rightarrow S \times T$  is a mapping satisfying, for all  $a \in I, b \in J, s \in S, t \in T$ ,  $\delta \otimes \sigma((a, b), (s, t)) = (\delta(a, s), \sigma(b, t))$ . For  $I, J$  disjoint we, by  $\mathbb{A} \cdot \mathbb{B}$ , denote the cartesian composition of  $\mathbb{A}$  and  $\mathbb{B}$ , i.e., the quasi-automaton  $(I \cup J, S \times T, \delta \cdot \sigma)$ , where  $\delta \cdot \sigma : (I \cup J) \times (S \times T) \rightarrow S \times T$  is defined, for all  $x \in I \cup J, s \in S$  and  $t \in T$ , by

$$(\delta \cdot \sigma)(x, (s, t)) = \begin{cases} (\delta(x, s), t) & \text{if } x \in I, \\ (s, \sigma(x, t)) & \text{if } x \in J. \end{cases}$$

One can see that in the homogeneous product, we have the same input set, which operates on every component of the state set. In the heterogeneous product, the input is a pair of symbols such that each input symbol from the pair is operated on the respective component of the state pair. In the cartesian composition, the input is directed to the corresponding state and operates on one component of the state pair only.

Notice that generalizing the homogeneous or heterogeneous product of automata to the case of quasi-multiautomata is rather straightforward; see Chvalina, Novák, and Křehlík [20]. However, applying the GMAC condition, which distinguishes the transition function of a quasi-multiautomaton from a transition function of a quasi-automaton, on cartesian composition of quasi-multiautomata is not straightforward. In Chvalina, Novák, and Křehlík [13] two extensions of this condition:

- Extension Generalized Mixed Associativity Condition (E-GMAC),
- Small Extension Generalized Mixed Associativity Condition (SE-GMAC)

and their applications were suggested.

**Definition 4** ([13]). Let  $\mathbb{A} = (I, S, \delta), \mathbb{B} = (J, T, \sigma)$  be e-quasi-multiautomata with input semihypergroups  $(I \circ), (J, *)$  and transition maps  $\delta : I \times S \rightarrow S, \sigma : J \times T \rightarrow T$  satisfying conditions

$$\delta(y, \delta(x, s)) \in \delta(x \cdot y, s) \cup \delta(I, s) \text{ for all } x, y \in I, s \in S, \tag{2}$$

$$\sigma(y, \sigma(x, t)) \in \sigma(x \cdot y, t) \cup \sigma(J, t) \text{ for all } x, y \in J, t \in T. \tag{3}$$

By the cartesian composition of e-quasi-multiautomata  $\mathbb{A}$  and  $\mathbb{B}$ , denoted as  $\mathbb{A} \cdot_E \mathbb{B}$ , we mean the e-quasi-multiautomaton  $\mathbb{A} \cdot_E \mathbb{B} = ((I \cup J), \diamond, S \times T, \delta \cdot \sigma)$ , where  $\delta \cdot \sigma$  is for all  $x \in I \cup J, s \in S$  and  $t \in T$  defined by

$$(\delta \cdot \sigma)(x, (s, t)) = \begin{cases} (\delta(x, s), t) & \text{if } x \in I, \\ (s, \sigma(x, t)) & \text{if } x \in J, \end{cases} \tag{4}$$

and  $\diamond : (I \cup J) \times (I \cup J) \rightarrow \mathcal{P}^*(I \cup J)$  is for all  $x, y \in I \cup J$  defined by

$$x \diamond y = \begin{cases} x \circ y \subseteq I & \text{if } x, y \in I, \\ x * y \subseteq J & \text{if } x, y \in J, \\ \{x, y\} & \text{if } x \in I, y \in J \text{ or } x \in J, y \in I \end{cases}$$

and  $\delta \cdot \sigma : (I \cup J) \times (S \times T) \rightarrow (S \times T)$

When modelling the process of data aggregation from underwater wireless sensor networks, Křehlík and Novák [21] suggested that using an internal link between respective automata in their cartesian composition describes the real-life context better because the internal link influences not only the given individual automaton but to a certain extent the whole system. In this paper we make use of [21]. Therefore, we include the following definition. For an example explaining the concept (together with calculations) see [21], Example 1.

**Definition 5 ([21]).** Let  $\mathbb{A} = (I, S, \delta_{\mathbb{A}})$  and  $\mathbb{B} = (J, T, \delta_{\mathbb{B}})$  be two quasi-automata  $\delta_{\mathbb{A}} : I \times S \rightarrow S, \delta_{\mathbb{B}} : J \times T \rightarrow T$  with disjoint input sets  $I, J$ . By  $\mathbb{A} \cdot \mathbb{B}$  we denote the automaton  $\mathbb{A} \cdot \mathbb{B} = (I \cup J, S \times T, \delta_{\mathbb{A}} \cdot \delta_{\mathbb{B}})$ , where  $\delta_{\mathbb{A}} \cdot \delta_{\mathbb{B}} : (I \cup J) \times (S \times T) \rightarrow S \times T$  and  $\varphi : S \rightarrow J, \varrho : T \rightarrow I$  is defined by

$$(\delta_{\mathbb{A}} \cdot \delta_{\mathbb{B}})(x, (s, t)) = \begin{cases} (\delta_{\mathbb{A}}(x, s); \delta_{\mathbb{B}}(\varphi(\delta_{\mathbb{A}}(x, s)), t)) & \text{if } x \in I, \\ (\delta_{\mathbb{A}}(\varrho(\delta_{\mathbb{B}}(x, t)), s); \delta_{\mathbb{B}}(x, t)) & \text{if } x \in J, \end{cases} \tag{5}$$

for all  $x \in I \cup J, s \in S$  and  $t \in T$ . The quasi-automaton  $\mathbb{A} \cdot \mathbb{B}$  is called the cartesian composition of quasi-automata  $\mathbb{A}$  and  $\mathbb{B}$  with an internal link. If in Definition 4 we replace Condition (4) with (5), we call the resulting quasi-multiautomaton the cartesian composition of quasi-multiautomata  $\mathbb{A}$  and  $\mathbb{B}$  with an internal link.

### 2. New Theoretical Model

In the classical theory, automata were considered as systems for transferring information of specific types. However, given complicated systems used nowadays, their benefits may not seem sufficient. In [9,13,19,20] we presented some real-life applications. In this section, we introduce an extension of ideas first included in [21] regarding internal links in the cartesian composition of quasi-multiautomata. We discuss systems in which there is a whole set of internal links. For our theoretical purposes we will organize them in a matrix considered by, for example, Golestan et al. [22]. In Section 3, we show the application of our theoretical results in the context of autonomous cars and their navigation.

**Notation 1.** We are going to use the following notation:

$$\begin{aligned} \text{Cartesian product of state sets :} & \quad \bigotimes_{i=1}^n S_i = S_1 \times S_2 \times S_3 \times \dots \times S_n \\ \text{Cartesian product of transition functions :} & \quad \prod_{i=1}^n \delta_i = \delta_1 \cdot \delta_2 \cdot \delta_3 \cdot \dots \cdot \delta_n \end{aligned}$$

**Definition 6.** Let  $\mathbb{M}\mathbb{A}_i = (I_i, S_i, \delta_i)$  be  $e$ -quasi-multiautomata with input semihypergroups  $(I_i, *_i)$  and  $\delta_i : I_i \times S_i \rightarrow S_i$  satisfying condition

$$\delta_i(y, (\delta_i(x, s))) \in \delta_i(y *_i x, s) \cup \delta_i(I_i, s) \text{ for all } x, y \in I_i, s \in S_i, i \in \{1, 2, \dots, n\}. \tag{6}$$

By  $n$ -ary cartesian composition of the  $e$ -quasi-multiautomata we mean the following system of  $e$ -quasi-multiautomata with internal link ( $S\mathbb{M}\mathbb{A}il$ ):

$$\mathbb{A}^{ncc} = \left( \left( \bigcup_{i=1}^n I_i, \diamond \right), \bigotimes_{i=1}^n S_i, \prod_{i=1}^n \delta_i, \mathbf{M}_{nn}(\varphi) \right),$$

where  $\prod_{i=1}^n \delta_i : \bigcup_{i=1}^n I_i \times \bigotimes_{i=1}^n S_i \rightarrow \bigotimes_{i=1}^n S_i$  and matrix  $\mathbf{M}_{nn}(\varphi)$ , called matrix of internal links, where  $\varphi_{ij} : S_i \rightarrow I_j$  for  $i, j \in \{1, 2, \dots, n\}$ , i.e.,

$$\mathbf{M}_{nm}(\varphi) = \begin{pmatrix} \varphi_{11} & \varphi_{12} & \dots & \varphi_{1n} \\ \varphi_{21} & \varphi_{22} & \dots & \varphi_{2n} \\ \vdots & \dots & \ddots & \vdots \\ \varphi_{n1} & \varphi_{n2} & \dots & \varphi_{nn} \end{pmatrix} \text{ is defined by}$$

$$\left(\prod_{i=1}^n \delta_i\right) (x, (s_1; s_2; \dots; s_n)) = (\delta_1 \cdot \delta_2 \cdot \dots \cdot \delta_n) (x, (s_1; s_2; \dots; s_n)) =$$

$$= \begin{cases} (\delta_1(\varphi_{11}(\delta_1(x, s_1)), s_1); \delta_2(\varphi_{12}(\delta_1(x, s_1)), s_2); \dots; \delta_n(\varphi_{1n}(\delta_1(x, s_1)), s_n)) & \text{if } x \in I_1, \\ (\delta_1(\varphi_{21}(\delta_2(x, s_2)), s_1); \delta_2(\varphi_{22}(\delta_2(x, s_2)), s_2); \dots; \delta_n(\varphi_{2n}(\delta_2(x, s_2)), s_n)) & \text{if } x \in I_2, \\ \vdots & \vdots \\ (\delta_1(\varphi_{n1}(\delta_n(x, s_n)), s_1); \delta_2(\varphi_{n2}(\delta_n(x, s_n)), s_2); \dots; \delta_n(\varphi_{nn}(\delta_n(x, s_n)), s_n)) & \text{if } x \in I_n, \end{cases} \quad (7)$$

and  $\diamond : \bigcup_{i=1}^n I_i \times \bigcup_{i=1}^n I_i \rightarrow \mathcal{P}^*\left(\bigcup_{i=1}^n I_i\right)$  is for all  $x, y \in \bigcup_{i=1}^n I_i$  defined by

$$x \diamond y = \begin{cases} x *_i y \subseteq I_i & \text{if } x, y \in I_i, \\ \{x, y\} & \text{if } x \in I_i, y \in I_k \text{ where } i \neq k. \end{cases}$$

satisfies the condition:

$$\left(\prod_{i=1}^n \delta_i\right) \left(y, \left(\prod_{i=1}^n \delta_i\right) (x, (s_1; s_2; \dots; s_n))\right) \in \left(\prod_{i=1}^n \delta_i\right) (x \diamond y, (s_1; s_2; \dots; s_n)) \cup \bigotimes_{i=1}^n \delta_i(I_i, s_i). \quad (8)$$

One can easily explain what is meant by Definition 6 using Figure 1. When an arbitrary input  $x \in \bigcup_{i=1}^n I_i$  is applied, the system, i.e., the cartesian composition, has to find out in which input set  $x$  belongs. Therefore, we determine the correct line in Equation (7), which through using we obtain a new state of the system. If, for example,  $x \in I_1$ , the new state  $s'_1 \in S_1$  will be computed. Then, it will be adjusted to  $s''_1 \in S_1$  using  $\varphi_{11}$ , i.e., by the self-mapping internal link. This new state  $s''_1 \in S_1$  will be mapped to remaining input sets  $I_2, I_3, \dots$  as defined by the matrix of internal links. In other words, the matrix determines which other states can be influenced by the change of the primary state. We apply inputs on states using respective transition functions. If there is 0 instead of  $\varphi_{ij}$  in the matrix of internal links, then the link between the state set  $S_{ij}$  and input set  $I_j$  does not exist. Notice that in Equation (7) e.g.,  $\delta_1(\varphi_{n1}(\delta_n(x, s_n)), s_1) = \delta_1(\varphi_{n1}(s'), s_1) = \delta_1(i_1, s_1) = s''$ . Since we obviously have to regard directions (see the oriented arrows), the matrix of internal links is not symmetric.

In Section 3 we apply the matrix of internal links in the context of modelling the navigation of autonomous vehicles. In this context, each vehicle obviously adjusts its behaviour based on the behaviour of other vehicles. Also obviously, not all vehicles need to be autonomous, which explains the use of zeros in the matrix (only autonomous vehicles communicate).

The above reasoning can be seen applied in the following example. Notice that we will use it also for the construction of the state set in Theorems 2 and 4 of Section 3.

**Example 1.** Consider a section of a road in front of an autonomously controlled intersection in Figure 2. All vehicles are autonomous with compatible self-driving control parameters. Every car is in a certain state  $s$ , for example, car  $A_3$  is in state  $s = (3; 45; 250; 1; 0; 0)$ , which means, for example, that car number 3 going at 45 km/h is in the position 250 m from the intersection in the 1st lane from the right. The first 0 means no change in speed while the second 0 means that the car is about to turn left.

In our system it is obvious that vehicle  $A_3$  does not need to communicate with vehicle  $A_4$  because it is not an obstacle in its intended driving operation. Thus, introducing an internal link between  $A_3$  and  $A_4$  is

not necessary. However, vehicles  $A_1$  and  $A_2$  are problematic for  $A_3$  because it intends to turn left. Therefore,  $A_1$  should increase its speed while  $A_2$  should slow down.

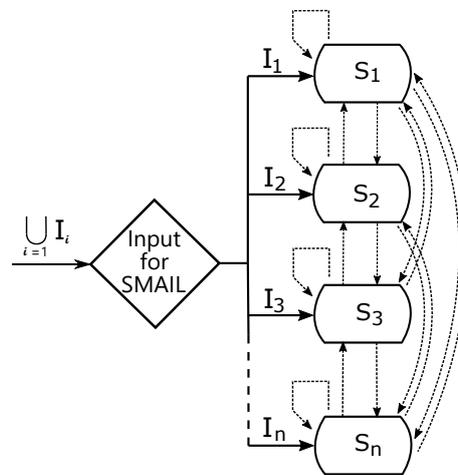


Figure 1. SMAil—A system of multiautomata with internal links.

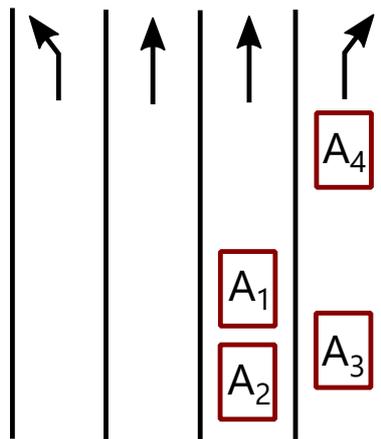


Figure 2. Lane shifting.

**Remark 1.** Notice that some theoretical requests in the above example are not possible or even absurd, i.e., we have to keep in mind their feasibility. We already faced this problem in [12,23], where it was eliminated by using a suitable state set, input set, or operations on them. In this paper, we will choose special input and state sets as well.

### 3. SMAil—Application of the Theoretical Model for Autonomous Cooperation

We are going to construct our system, called SMAil, for the above context of Example 1. Naturally, Example 1 is an example of a possible usage of SMAil only.

First, we need to define suitable sets  $\mathbb{S}_{\mathcal{R},\mathcal{M}}^p$  and  $I_{\mathcal{V},\mathcal{M}}^p$ , where  $p \in \mathbb{N}$  is the index. The elements of the state set are ordered pairs—the first component being an ordered sextuple of numbers while the second component is a matrix. The state set is

$$\mathbb{S}_{\mathcal{R},\mathcal{M}}^p = \left\{ [s; \mathbf{A}], | s \in \mathcal{R}_6^p, \mathbf{A} \in \mathbf{M}_{m,n} \right\}, \tag{9}$$

where

$$\mathcal{R}_6^p = \left\{ s = (s_1, s_2, \dots, s_6) | s_1 = p; s_2, s_3, s_5 \in \mathbb{R}_0^+; s_4 \in \mathbb{N}; s_6 \in \left\langle -\frac{\pi}{4}, \frac{\pi}{4} \right\rangle \right\} \tag{10}$$

and

$$\mathbf{M}_{m,n} = \left\{ \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \mid a_{ij} \in \begin{cases} \{0\} & \text{for } i = \frac{m+1}{2}, j = \frac{n+1}{2}; \\ \{-1, 0, 1\} & \text{for otherwise} \end{cases} \right\} \quad (11)$$

where  $m, n$  are a odd numbers. The state set describes the complete position of every vehicle in all lanes. Notice that this is the same approach as in [24,25]. Now, the elements of  $\mathcal{R}_6^p$  defined by Equation (10) correspond to states mentioned in Example 1, i.e.,  $s_1$  stands for the vehicle number,  $s_2$  for its speed,  $s_3$  for distance from the intersection,  $s_4$  for the respective lane (calculated from the right using odd numbers only, even number are reserved for positions between two lanes),  $s_5$  stands for changing velocity (i.e., an interval  $(0, 1)$  is deceleration, an interval  $(1, \infty)$  is acceleration, 1 stands for constant speed),  $s_6$  stands for changing direction (i.e., an interval  $[-\frac{\pi}{4}, 0)$  means manoeuvring left, interval  $(0, \frac{\pi}{4}]$  means manoeuvring right, 0 means straight direction). Of course, more parameters can be used; for an example see [26]. Matrices  $\mathbf{A}$  used in Equation (9) are taken from the set  $\mathbf{M}_{m,n}$  defined by Equation (11). Notice that we use these matrices in the form suggested in [25,27,28] where, for the purpose of its control policy, the intersection is divided into a grid of reservation tiles. To be more precise,

$$\mathbf{A} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 0 & 0 & 0 & -1 & -1 & -1 \\ -1 & -1 & -1 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \in \mathbf{M}_{m,n} \quad (12)$$

where 0 are parts of the grid (or tiles) occupied by the given vehicle, 1 stands for tiles occupied by some other vehicles and  $-1$  are free tiles. The above matrix  $\mathbf{A}$  describes the situation depicted in Figure 3 from the point of view of  $A_3$ .

Next, for construction of the  $i$ -th quasi-multiautomaton and consequently  $SMAil$ , we will need an input alphabet, i.e., input sets (since we will be working with quasi-multiautomata, these will be algebraic hyperstructures). As mentioned above, we will define  $I_{\mathcal{V},\mathcal{M}}^p$ , where  $p \in \mathbb{N}$  is the index of the input set.

$$I_{\mathcal{V},\mathcal{M}}^p = \left\{ \left[ \vec{i}, \mathbf{C}_{m,n} \right] = \left[ (i_1, i_2, \dots, i_6), \begin{bmatrix} c_{11} & \dots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{m1} & \dots & c_{mn} \end{bmatrix} \right] \mid i_1 = p, i_2 \in \mathbb{R}_0^+, i_3 \in \langle 0, 1 \rangle, i_4 \in \mathbb{N}_0, i_5 \in \mathbb{R}^+, i_6 \in \left\langle -\frac{\pi}{4}, \frac{\pi}{4} \right\rangle, \right. \\ \left. c_{ij} \in \{-1, 1\}, i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\} \right\} \quad (13)$$

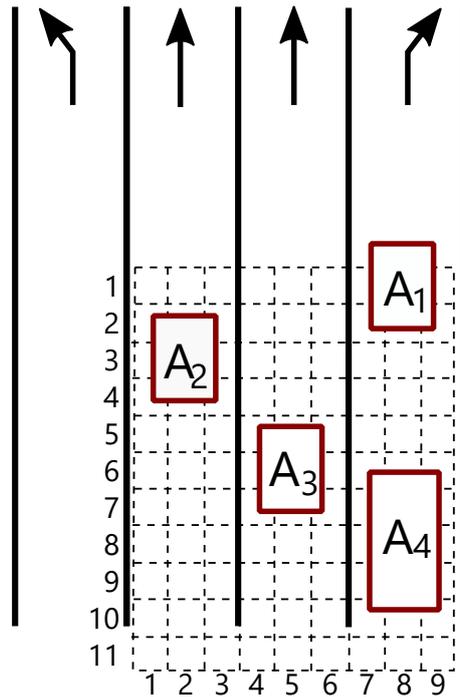


Figure 3. Lane shifting; grid view of vehicle  $A_3$ .

It is obvious that input sets  $I_{V,M}^p$  and  $I_{V,M}^q$  are disjoint for  $p \neq q$ , because the first component of every vector used in  $[\vec{v}, \mathbf{A}_{n,m}] \in I_{V,M}^p$  is  $p$  while the first component of every vector used in  $[\vec{u}, \mathbf{B}_{n,m}] \in I_{V,M}^q$  is  $q$ , i.e.,

$$I_{V,M}^p \cap I_{V,M}^q = \emptyset, \text{ for every } p, q \in \mathbb{N}, \text{ where } p \neq q.$$

Since vector components are real (or even natural) numbers, we can suppose that their sets are ordered; we write  $\leq$  for this ordering. For all  $p \in \mathbb{N}$  we define hyperoperation  $*_p : I_{V,M}^p \times I_{V,M}^p \rightarrow \mathcal{P}^*(I_{V,M}^p)$  by:

$$[\vec{u}, \mathbf{A}_{m,n}] *_p [\vec{v}, \mathbf{B}_{m,n}] = \{[\vec{w}, \mathbf{C}_{m,n}] \mid \min\{u_i, v_i\} \leq w_i \leq \max\{u_i, v_i\}, \text{ for every } i \in \mathbb{N}, \mathbf{C}_{m,n} \in \{\mathbf{A}_{m,n}, \mathbf{B}_{m,n}\}\}. \tag{14}$$

In the following example we show what we mean by the hyperoperation. Notice that the hyperoperation will be later on used to construct a hypergroup in Theorem 1, which will fulfil the assumption regarding input alphabet stated in Definition 2.

**Example 2.** Consider two elements  $[\vec{a}, \mathbf{A}_{2,3}], [\vec{c}, \mathbf{C}_{2,3}] \in I_{V,M}^2$ , where  $[\vec{a}, \mathbf{A}_{2,3}] = \left[ (2; 6.2; 0; 1; 4; -\frac{\pi}{8}), \begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \right]$  and  $[\vec{c}, \mathbf{C}_{2,3}] = \left[ (2; 8.1; 1; 2; 5; \frac{\pi}{6}), \begin{bmatrix} -1 & -1 & -1 \\ 1 & -1 & -1 \end{bmatrix} \right]$ . Then the hyperoperation

$$[\vec{a}, \mathbf{A}_{2,3}] *_p [\vec{c}, \mathbf{C}_{2,3}] = \left[ (2; 6.2; 0; 1; 4; -\frac{\pi}{8}), \begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \right] *_p \left[ (2; 8.1; 1; 2; 5; \frac{\pi}{6}), \begin{bmatrix} -1 & -1 & -1 \\ 1 & -1 & -1 \end{bmatrix} \right] = \left\{ [(2; r_2; r_3; r_4; r_5; r_6), \mathbf{D}_{2,3}] \mid 6.2 \leq r_2 \leq 8.1; 0 \leq r_3 \leq 1; 1 \leq r_4 \leq 2; 4 \leq r_5 \leq 5; -\frac{\pi}{8} \leq r_6 \leq \frac{\pi}{6} \right\},$$

where  $[\vec{r}, \mathbf{D}_{2,3}]$  is defined by Equation (14), i.e.,  $\mathbf{D}_{2,3} \in \left\{ \begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix}, \begin{bmatrix} -1 & -1 & -1 \\ 1 & -1 & -1 \end{bmatrix} \right\}$ .

**Theorem 1.** For every index  $p \in \mathbb{N}$ , the pairs  $(I_{\mathcal{V}, \mathcal{M}}^p, *_p)$  are hypergroups.

**Proof.** First, we have to show that the associativity axiom holds, i.e., for all  $[\vec{u}, \mathbf{A}_{m,n}]; [\vec{v}, \mathbf{B}_{m,n}]; [\vec{w}, \mathbf{C}_{m,n}] \in I_{\mathcal{V}, \mathcal{M}}^p$  there is  $[\vec{u}, \mathbf{A}_{m,n}] *_p ([\vec{v}, \mathbf{B}_{m,n}] *_p [\vec{w}, \mathbf{C}_{m,n}]) = ([\vec{u}, \mathbf{A}_{m,n}] *_p [\vec{v}, \mathbf{B}_{m,n}]) *_p [\vec{w}, \mathbf{C}_{m,n}]$ . Without losing generality, we can show that associativity axiom holds for both matrices and vectors. First, we consider vectors:

$$\begin{aligned} \vec{u} *_p (\vec{v} *_p \vec{w}) &= \vec{u} *_p \{ \vec{r}, \min\{v_i, w_i\} \leq r_i \leq \max\{v_i, w_i\} \} = \\ &= \bigcup_{\vec{s} \in \{ \vec{r}, \min\{v_i, w_i\} \leq r_i \leq \max\{v_i, w_i\} \}} \vec{u} *_p \vec{s} = \bigcup_{\vec{s} \in \{ \vec{r}, \min\{v_i, w_i\} \leq r_i \leq \max\{v_i, w_i\} \}} \{ \vec{t}, \min\{u_i, s_i\} \leq t_i \leq \max\{u_i, s_i\} \} = \\ &= \{ \vec{t}, \min\{u_i, \min\{v_i, w_i\}\} \leq t_i \leq \max\{u_i, \max\{v_i, w_i\}\} \} = \{ \vec{t}, \min\{u_i, v_i, w_i\} \leq t_i \leq \max\{u_i, v_i, w_i\} \} = \\ &= \{ \vec{t}, \min\{\min\{u_i, v_i\}, w_i\} \leq t_i \leq \max\{\min\{u_i, v_i\}, w_i\} \} = \\ &= \bigcup_{\vec{a} \in \{ \vec{b}, \min\{u_i, v_i\} \leq b_i \leq \max\{u_i, v_i\} \}} \{ \vec{c}, \min\{w_i, a_i\} \leq c_i \leq \max\{w_i, a_i\} \} = \bigcup_{\vec{a} \in \{ \vec{b}, \min\{u_i, v_i\} \leq b_i \leq \max\{u_i, v_i\} \}} \vec{a} *_p \vec{w} = \\ &= \{ \vec{b}, \min\{u_i, v_i\} \leq b_i \leq \max\{u_i, v_i\} \} *_p \vec{w} = (\vec{u} *_p \vec{v}) *_p \vec{w}. \end{aligned}$$

In the case of matrices, the proof is straightforward:

$$\begin{aligned} \mathbf{A} *_p (\mathbf{B} *_p \mathbf{C}) &= \mathbf{A} *_p \{ \mathbf{B}, \mathbf{C} \} = \mathbf{A} *_p \mathbf{B} \cup \mathbf{A} *_p \mathbf{C} = \{ \mathbf{A}, \mathbf{B} \} \cup \{ \mathbf{A}, \mathbf{C} \} = \{ \mathbf{A}, \mathbf{B}, \mathbf{C} \} \\ &= \{ \mathbf{A}, \mathbf{C} \} \cup \{ \mathbf{B}, \mathbf{C} \} = \mathbf{A} *_p \mathbf{C} \cup \mathbf{B} *_p \mathbf{C} = \{ \mathbf{A}, \mathbf{B} \} *_p \mathbf{C} = (\mathbf{A} *_p \mathbf{B}) *_p \mathbf{C}. \end{aligned}$$

The reproductivity axiom holds automatically because the hyperoperation defined above is obviously extensive, i.e., for all  $\mathbf{A}, \mathbf{B} \in I_{\mathcal{V}, \mathcal{M}}^p$  there is  $\{ \mathbf{A}, \mathbf{B} \} \subseteq \mathbf{A} *_p \mathbf{B}$ . So for arbitrary  $[\vec{v}, \mathbf{B}_{m,n}]$  we have  $[\vec{v}, \mathbf{B}_{m,n}] *_p I_{\mathcal{V}, \mathcal{M}}^p = \bigcup_{[\vec{u}, \mathbf{C}_{m,n}] \in I_{\mathcal{V}, \mathcal{M}}^p} [\vec{v}, \mathbf{B}_{m,n}] *_p [\vec{u}, \mathbf{C}_{m,n}] = I_{\mathcal{V}, \mathcal{M}}^p$ . Thus, for all indices  $p \in \mathbb{N}$  the structures  $(I_{\mathcal{V}, \mathcal{M}}^p, *_p)$  are hypergroups.  $\square$

At this point, everything is ready for the construction of an e-quasi-multiautomaton with the state set  $\mathbb{S}_{\mathcal{R}, \mathcal{M}}^p$  and input set  $I_{\mathcal{V}, \mathcal{M}}^p$  for  $p$ -th quasi-automaton, where  $p \in \mathbb{N}$ .

**Remark 2.** We consider ordered pairs of vectors and matrices as elements of the input set and also of the state set. As a result, we will denote elements with index  $i$  for input and with index  $s$  for the state, i.e.,  $[\vec{a}^i, {}^i m, n]$   $A$  is an input word and  $[u^s, \mathbf{M}_{m,n}^s]$  is a state.

**Theorem 2.** For every index  $p \in \mathbb{N}$  the triple  $\mathbb{M}\mathbb{A}_p = (I_{\mathcal{V}, \mathcal{M}}^p, \mathbb{S}_{\mathcal{R}, \mathcal{M}}^p, \delta_p)$  is an e-quasi-multiautomaton with input hypergroup  $(I_{\mathcal{V}, \mathcal{M}}^p, *_p)$ , where transition function  $\delta_p : I_{\mathcal{V}, \mathcal{M}}^p \times \mathbb{S}_{\mathcal{R}, \mathcal{M}}^p \rightarrow \mathbb{S}_{\mathcal{R}, \mathcal{M}}^p$  is defined by

$$\delta_p \left( [\vec{a}^i, {}^i m, n] C, [v^s, \mathbf{K}_{m,n}^s] \right) = \tag{15}$$

$$\left[ \left( \sqrt{a_1 \cdot v_1}; a_2 \cdot v_2 + a_2; a_3 \cdot v_3; a_4 + v_4 \equiv \text{mod } 2l - 1; a_5 \cdot v_5; a_6 + 0 \cdot v_6 \right)_s, \begin{bmatrix} c_{11}k_{11} & \dots & c_{1n}k_{1n} \\ \dots & \dots & \dots \\ c_{m1}k_{m1} & \dots & c_{mn}k_{mn} \end{bmatrix}^s \right].$$

**Proof.** This proof is constructed as follows: first we calculate left hand side condition E-GMAC, second we show that left hand side is included in right hand side.

The left hand side:

$$\begin{aligned} & \delta_p \left( \left[ \vec{a}^i, \mathbf{A}_{m,n}^i \right] A, \delta_p \left( \left[ \vec{b}^i, \mathbf{B}_{m,n}^i \right], \left[ u^s, \mathbf{U}_{m,n}^s \right] \right) \right) = \\ & \delta_p \left( \left[ \vec{a}^i, \mathbf{A}_{m,n}^i \right] A, \left[ \left( \sqrt{b_1 u_1}; b_2 u_2 + b_2; b_3 u_3; b_4 + u_4 \equiv \text{mod } 2l - 1; b_5 u_5; b_6 + 0 \cdot u_6 \right)_s, (b_{ij} u_{ij})^s \right] \right) = \\ & \delta_p \left( \left[ \vec{a}^i, \mathbf{A}_{m,n}^i \right] A, \left[ \left( \sqrt{p^2}; b_2 u_2 + b_2; b_3 u_3; b_4 + u_4 \equiv \text{mod } 2l - 1; b_5 u_5; b_6 \right)_s, (b_{ij} u_{ij})^s \right] \right) = \\ & \left[ \left( \sqrt{a_1 p}; a_2 (b_2 u_2 + b_2) + a_2; a_3 b_3 u_3; a_4 + b_4 + u_4 \equiv \text{mod } 2l - 1; a_5 b_5 u_5; a_6 + (0 \cdot b_6) \right)_s, (a_{ij} b_{ij} u_{ij})^s \right] = \\ & \left[ \left( \sqrt{p^2}; a_2 (b_2 u_2 + b_2) + a_2; a_3 b_3 u_3; a_4 + b_4 + u_4 \equiv \text{mod } 2l - 1; a_5 b_5 u_5; a_6 \right)_s, (a_{ij} b_{ij} u_{ij})^s \right] = \\ & \left[ (p; a_2 (b_2 u_2 + b_2) + a_2; a_3 b_3 u_3; a_4 + b_4 + u_4 \equiv \text{mod } 2l - 1; a_5 b_5 u_5; a_6)_s, (a_{ij} b_{ij} u_{ij})^s \right] = \\ & \left[ v^s, \mathbf{V}_{m,n}^s \right]. \end{aligned}$$

The right hand side:

$$\delta_p \left( \left[ \vec{a}^i, \mathbf{A}_{m,n}^i \right] A, *_p \left[ \vec{b}^i, \mathbf{B}_{m,n}^i \right], \left[ u^s, \mathbf{U}_{m,n}^s \right] \right) \cup \delta_p \left( I_{\mathcal{V}, \mathcal{M}}^p, \left[ u^s, \mathbf{U}_{m,n}^s \right] \right).$$

From the definition of hyperproduct  $*_p$  in Equation (14), we simply observe that the left hand side is not included in the right part of the right hand side union. Therefore, we have to proof, that the left hand side belongs to the left part of the right hand side union. Therefore we calculate:

$$\delta_p \left( I_{\mathcal{V}, \mathcal{M}}^p, \left[ u^s, \mathbf{U}_{m,n}^s \right] \right) = \bigcup_{\left[ \vec{p}^i, \mathbf{P}_{m,n}^i \right] \in I_{\mathcal{V}, \mathcal{M}}^p} \delta_p \left( \left[ \vec{p}^i, \mathbf{P}_{m,n}^i \right], \left[ u^s, \mathbf{U}_{m,n}^s \right] \right)$$

There exist an input word  $\left[ \vec{c}^i, \mathbf{C}_{m,n}^i \right] \in I_{\mathcal{V}, \mathcal{M}}^p$  such that the vector is  $\vec{c} = \left( p; \frac{a_2 i_2 s + a_2 i_2 + a}{s+1}; a_3 b_3; a_4 + b_4; a_5 b_5; a_6 \right)$  and the matrix is  $\mathbf{C} = \begin{bmatrix} a_{11} b_{11} & \dots & a_{1n} b_{1n} \\ \dots & \dots & \dots \\ a_{m1} b_{m1} & \dots & a_{mn} b_{mn} \end{bmatrix}^i$ . Thus,

$$\left[ v^s, \mathbf{V}_{m,n}^s \right] = \delta_p \left( \left[ \vec{c}^i, \mathbf{C}_{m,n}^i \right], \left[ u^s, \mathbf{U}_{m,n}^s \right] \right) \in \bigcup_{\left[ \vec{p}^i, \mathbf{P}_{m,n}^i \right] \in I_{\mathcal{V}, \mathcal{M}}^p} \delta_p \left( \left[ \vec{p}^i, \mathbf{P}_{m,n}^i \right], \left[ u^s, \mathbf{U}_{m,n}^s \right] \right),$$

As a result, E-GMAC holds and the structure  $\mathbb{M}\mathbb{A}_p = (I_{\mathcal{V}, \mathcal{M}}^p, S_{\mathcal{V}, \mathcal{M}}^p, \delta_p)$  is an e-quasi-multiautomaton.  $\square$

Once Theorem 2 is proved we will show a practical application of e-quasi-multiautomata in intelligent transport systems. In the following example, the e-quasi-multiautomaton represents an autonomous vehicle. Its state is described by parameters organized into a sextuple while the matrix is used to detect its environment. This example is also linked to Example 1.

**Example 3.** We consider matrix  $\mathbf{B}$  equivalent to matrix  $\mathbf{A}$  of Equation (12), which describes the situation in Figure 3. The vehicle  $A_3$ —in Figure 4 depicted in green (while other, non-autonomous, vehicles are red)—detects its surroundings and saves the data to matrix  $\mathbf{B}$ . This model is suitable for a situation where only one vehicle, such as  $A_3$ , is autonomous and it is not possible to establish communication with other vehicles (because they are not autonomous). We consider a state  $\left[ a^s, \mathbf{B}_{m,n}^s \right] \in S_{\mathcal{V}, \mathcal{M}}^3$ , where  $\vec{a}_s = (3; 30; 200; 2; 0; 0)$  and values in matrix  $\mathbf{B}$  corresponding to values in the first picture of Figure 4. If we apply the input  $\left[ \vec{s}^i, \mathbf{S}_{m,n}^i \right] \in I_{\mathcal{V}, \mathcal{M}}^3$  by the transition

function  $\delta_3$ , where  $\vec{s}^i = (3; 1; 0, 99; 1; 0; -\frac{\pi}{4})$  and  $\mathbf{S}_{m,n}^i =$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \end{bmatrix},$$

then the

first component of the input word is used to control the direction, i.e., it changes the state of the sextuple. The second component of the input word changes the matrix that detects the surroundings of a vehicle. This way we obtain a new state  $[r^s, \mathbf{R}_{m,n}^s]$ , where  $\vec{s}_i = (3; 30; 198; 3; 0; -\frac{\pi}{4})$  and the entries of the matrix  $\mathbf{R}_{m,n}^s$  are  $-1$  for  $a_{21}, a_{24}, a_{31}, a_{34}, a_{41}, a_{44}, a_{17}, a_{27}, a_{67}, a_{77}, a_{87}, a_{97}, a_{10,7}$  and  $1$  for other cases.

We know from the new state  $[r^s, \mathbf{R}_{m,n}^s]$  that the vehicle  $A_3$  does not change its velocity or its acceleration. We also know that the vehicle  $A_3$  is positioned about 2m far from the intersection border and is between lane 2 and 3, and it turns to the left. We can see the position surroundings in Figure 4. Consequently, this is what we use in the next step as input for full inclusion in lane 3.

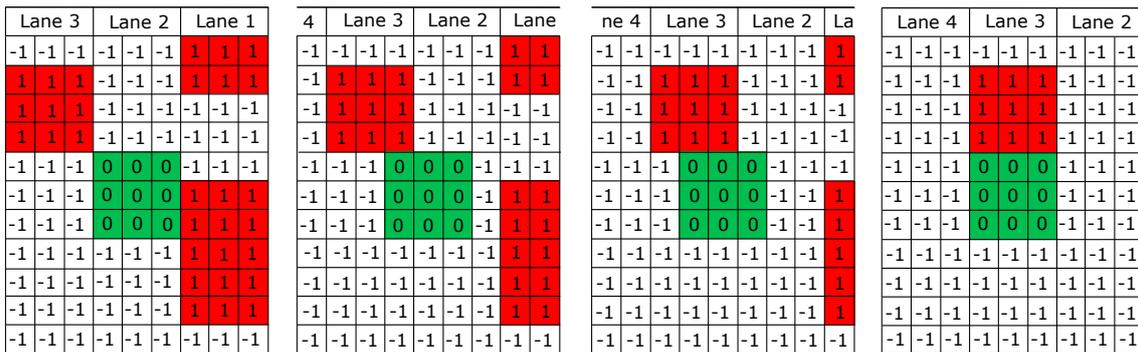


Figure 4. Lane change.

We will use the e-quasi-multiautomata from Theorem 2 for the construction of a system called SMAil. Necessarily, a change on any component of the state of the system must trigger a change on another component of the state. However, there may be situations where the change of other components are not necessarily required. Therefore, we include the following theorem in which we assert that there are such inputs for which the state of the multiautomata do not change.

**Theorem 3.** For every e-quasi-multiautomaton  $\mathbb{M}\mathbb{A}_p = (I_{V,M}^p, S_{R,M}^p, \delta_p)$  there is an input  $[\vec{e}^i, E_{m,n}^i]$  for which holds

$$\delta_p \left( [\vec{e}^i, E_{m,n}^i], [s^s, S_{m,n}^s] \right) = [s^s, S_{m,n}^s]. \tag{16}$$

**Proof.** Consider an arbitrary state  $[s^s, \mathbf{S}_{m,n}^s] \in \mathbb{S}_{\mathcal{R}\mathcal{M}}^p$  of the e-quasi-multiautomaton  $\mathbb{M}\mathbb{A}_p$ . For input words  $[\vec{e}^i, \mathbf{E}_{m,n}^i] = \left[ (e_1; e_2; e_3; e_4; e_5; e_6), \begin{bmatrix} 1_{11} & \dots & 1_{1n} \\ \dots & \dots & \dots \\ 1_{m1} & \dots & 1_{mn} \end{bmatrix}^i \right]$ , where  $e_1 = p; e_2 = \frac{s_2}{s_2+1}; e_3 = 1; e_4 = 0; e_5 = 1; e_6 = s_6$ , there holds Equation (16). Indeed,

$$\begin{aligned} \delta_p \left( [\vec{e}^i, \mathbf{E}_{m,n}^i], [s^s, \mathbf{S}_{m,n}^s] \right) &= \\ \delta_p \left( \left[ \left( p; \frac{s_2}{s_2+1}; 1; 0; 1; s_6 \right)_i, \begin{bmatrix} 1_{11} & \dots & 1_{1n} \\ \dots & \dots & \dots \\ 1_{m1} & \dots & 1_{mn} \end{bmatrix}^i \right], \left[ (s_1; s_2; s_3; s_4; s_5; s_6)_s, \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}^s \right] \right) &= \\ \left[ \left( \sqrt{p^2}, \frac{s_2}{s_2+1}s_2 + \frac{s_2}{s_2+1}, 1 \cdot s_3, 0 + s_4 \equiv \text{mod } k - 1, 1 \cdot s_5, s_6 + 0 \cdot s_6 \right) \begin{bmatrix} 1_{11}a_{11} & \dots & 1_{1n}a_{1n} \\ \dots & \dots & \dots \\ 1_{m1}a_{m1} & \dots & 1_{mn}a_{mn} \end{bmatrix}^s \right] &= \\ \left[ \left( p, \frac{s_2(s_2+1)}{s_2+1} s_3, s_4, s_5, s_6 \right), \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}^s \right] &= [s^s, \mathbf{S}_{m,n}^s] \end{aligned}$$

□

In Example 4 we consider different e-quasi-multiautomata as parts of a cooperative intelligent transport system. Some e-quasi-multiautomata represent autonomous vehicles while some are non-autonomous vehicles. Obviously, non-autonomous vehicles cannot detect their surroundings. In such cases we will use e-quasi-multiautomata, in which all entries of state matrices are 0. This explains the inclusion of the following lemma.

**Lemma 1.** Let  $\overline{\mathbb{S}}_{\mathcal{R}\mathcal{M}}^p = \{[s; \mathbf{A}] | s \in \mathcal{R}_{\mathcal{M}}^p, a_{ij} = 0\}$ , then a structure  $\text{sub}\mathbb{M}\mathbb{A}_p = (I_{\mathcal{V}\mathcal{M}}^p, \overline{\mathbb{S}}_{\mathcal{R}\mathcal{M}}^p, \delta_p)$  is a sub-e-quasi-multiautomaton of e-quasi-multiautomaton  $\mathbb{M}\mathbb{A}_p = (I_{\mathcal{V}\mathcal{M}}^p, \mathbb{S}_{\mathcal{R}\mathcal{M}}^p, \delta_p)$ .

**Proof.** It is obvious that  $\overline{\mathbb{S}}_{\mathcal{R}\mathcal{M}}^p \subset \mathbb{S}_{\mathcal{R}\mathcal{M}}^p$  and  $\delta_p : I_{\mathcal{V}\mathcal{M}}^p \times \overline{\mathbb{S}}_{\mathcal{R}\mathcal{M}}^p \rightarrow \overline{\mathbb{S}}_{\mathcal{R}\mathcal{M}}^p$ . Next, we can see that E-GMAC holds because all entries in the state matrix are zero. Thus, the second component of the state can not change for any input. Therefore, the proof for the first component is the same as the proof of the Theorem 2. □

In what follows, we will note that every state  $[p s^s, p \mathbf{S}_{m,n}^s]$  is equal to  $[(p s_1^s, \dots, p s_6^s), p b_{ij}^s] \in \mathbb{S}_{\mathcal{R}\mathcal{M}}^p$  and every input  $[p \vec{a}^i, p \mathbf{A}_{m,n}^i]$  is equal to  $[(p a_1^i, \dots, p a_6^i), p a_{ij}^i] \in I_{\mathcal{V}\mathcal{M}}^p$  from e-quasi-multiautomaton  $\mathbb{M}\mathbb{A}_p = (I_{\mathcal{V}\mathcal{M}}^p, \mathbb{S}_{\mathcal{R}\mathcal{M}}^p, \delta_p)$ , where  $p \in \mathbb{N}$  is an index. In the other words, such as  $p s_1^s$ , the upper index  $s$  denotes state element, lower index 1 denotes position in the sextuple and the lower index  $p$  denotes the  $p$ -th e-quasi-multiautomaton. The indices in matrices have the same meaning. As far as input words are regarded, we use upper index  $i$ ; other indices have the same meaning.

Now we need to define the matrix of internal links. First, we define  $\varphi_{pp} : \mathbb{S}_{\mathcal{R}\mathcal{M}}^p \rightarrow I_{\mathcal{V}\mathcal{M}}^p$  for all  $p \in \{1, 2, \dots, n\}$  by

$$\varphi_{pp} \left( [(p s_1^s, p s_2^s, p s_3^s, p s_4^s, p s_5^s, p s_6^s), p \mathbf{A}_{m,n}^s] \right) = [(p v_1^i, p v_2^i, p v_3^i, p v_4^i, p v_5^i, p v_6^i), p \mathbf{B}_{m,n}^i].$$

where  $p v_1^i = p s_1^s = p; p v_2^i = \frac{p s_2^s}{p s_2^s + \frac{1}{p s_5^s}}; p v_3^i = 0, 95; p v_4^i = p s_4^s + l + \text{sgn}(s_6^s); p v_5^i = \frac{1}{p s_5^s}; p v_6^i = 0$  and entries of the input matrix are dependent on the occupancy of the tiles.

**Corollary 1.** For mapping  $\varphi_{pq}$  added to every e-quasi-multiautomaton  $\mathbb{MA}_p = (I_{\mathcal{VM}}^p, \mathbb{S}_{\mathcal{RM}}^p, \delta_p)$  from Theorem 2 there holds the condition E-GMAC.

**Proof.** The proof is obvious. On the left hand side we have up to four changes of the original state in the E-GMAC condition (two inputs and two applications of the internal link). However, we have a suitable input for the right hand side, similar to the proof of Theorem 2.  $\square$

Now we will define a mapping for two different e-quasi-multiautomata, i.e., from the state set of the  $p$ th e-quasi-multiautomaton to the input set of the  $q$ th e-quasi-multiautomaton. We define mapping  $\varphi_{pq} : \mathbb{S}_{\mathcal{RM}}^p \rightarrow I_{\mathcal{VM}}^q$  for all  $p, q \in \{1, 2, \dots, n\}$ , where  $p \neq q$  between two e-quasi-multiautomata  $\mathbb{MA}_p = ((I_{\mathcal{VM}}^p, \mathbb{S}_{\mathcal{RM}}^p, \delta_p))$  and  $\mathbb{MA}_q = ((I_{\mathcal{VM}}^q, \mathbb{S}_{\mathcal{RM}}^q, \delta_q))$  by

$$\varphi_{pq} \left( [(p s_1^s, p s_2^s, p s_3^s, p s_4^s, p s_5^s, p s_6^s), p \mathbf{A}_{n,m}^s] \right) = [(q v_1^l, q v_2^l, q v_3^l, q v_4^l, q v_5^l, q v_6^l), q \mathbf{B}_{n,m}^l], \tag{17}$$

where

$$q v_1^l = q \neq p s_1^s; q v_2^l = \frac{p s_2^s}{p s_2^s + 1} + \alpha p s_5^s + \lambda \cdot \frac{|p s_6^s|}{10}; q v_3^l = 0, 99; q v_4^l = 0; q v_5^l = 1; q v_6^l = 0.$$

Next we add a meaning of the parameters  $\lambda$  and  $\alpha$

$$\lambda = \begin{cases} 1 & \text{if } p s_4^s > q s_4^s \text{ and } p s_6^s > 0 \\ 1 & \text{if } p s_4^s < q s_4^s \text{ and } p s_6^s < 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \alpha = \begin{cases} 1 & \text{if } p s_4^s = q s_4^s \text{ and } q s_3^s < p s_4^s \text{ and } p s_5^s > 1 \\ 0 & \text{otherwise} \end{cases}$$

At this point we can give the main theorem of this section in which we are going to use all results obtained above, i.e., e-quasi-multiautomata, hypergroups, and internal links. By Definition 6 we obtain the  $n$ -ary cartesian composition of e-quasi-multiautomata with internal links.

**Theorem 4.** Let  $\mathbb{MA}_1 = (I_{\mathcal{VM}}^1, \mathbb{S}_{\mathcal{RM}}^1, \delta_1), \mathbb{MA}_2 = ((I_{\mathcal{VM}}^2, \mathbb{S}_{\mathcal{RM}}^2, \delta_2)), \dots; \mathbb{MA}_p = ((I_{\mathcal{VM}}^p, \mathbb{S}_{\mathcal{RM}}^p, \delta_p))$  be an e-quasi-multiautomata with disjoint input-sets  $I_{\mathcal{VM}}^p$ , and  $p \geq 2$ . Then a quadruple

$$\text{SMail} = \left( \left( \bigcup_{i=1}^p I_{\mathcal{VM}}^i, \diamond \right), \bigotimes_{i=1}^p \mathbb{S}_{\mathcal{RM}}^i, \prod_{i=1}^p \delta_i, \mathbf{M}_{nn}(\varphi) \right)$$

is a system of the cartesian composition of e-quasi-multiautomata with internal links.

**Proof.** We have to demonstrate that the condition E-GMAC holds, i.e., that there is

$$\left( \prod_{i=1}^p \delta_i \right) \left( [\vec{a}^l, \mathbf{A}_{m,n}^l], \left( \prod_{i=1}^p \delta_i \right) \left( [\vec{b}^l, \mathbf{B}_{m,n}^l], \bigotimes_{i=1}^p [i^s, \mathbf{S}_{m,n}^s] \right) \right) \in \left( \prod_{i=1}^p \delta_i \right) \left( [\vec{a}^l, \mathbf{A}_{m,n}^l] * p [\vec{b}^l, \mathbf{B}_{m,n}^l], \bigotimes_{i=1}^p [i^s, \mathbf{S}_{m,n}^s] \right) \cup \bigotimes_{i=1}^p \delta_i \left( I_{\mathcal{VM}}^i, [i^s, \mathbf{S}_{m,n}^s] \right). \tag{18}$$

We prove, while maintaining generality, that

$$\left( \prod_{i=1}^t \delta_i \right) \left( [\vec{a}^l, \mathbf{A}_{m,n}^l], \left( \prod_{i=1}^t \delta_i \right) \left( [\vec{b}^l, \mathbf{B}_{m,n}^l], \bigotimes_{i=1}^t [i^s, \mathbf{S}_{m,n}^s] \right) \right) \in \bigotimes_{i=1}^t \delta_i \left( I_{\mathcal{VM}}^i, [i^s, \mathbf{S}_{m,n}^s] \right), \tag{19}$$

i.e., Formula (18) without the left part of the right hand side of E-GMAC. There are two cases. The first, both input words are in the same input set, and the second, input words are from different input sets.

- (a) For the first case, we consider  $[p\vec{a}^i, p\mathbf{A}_{m,n}^i], [p\vec{b}^i, p\mathbf{B}_{m,n}^i] \in I_{\mathcal{VM}}^p$ . On the left hand of Equation (19), we know that the input  $[p\vec{b}^i, p\mathbf{B}_{m,n}^i]$  works upon one component from  $([1^s, \mathbf{1S}_{m,n}^s], [2^s, \mathbf{2S}_{m,n}^s], \dots, [t^s, \mathbf{tS}_{m,n}^s])$  and the state can alter by the internal link  $\varphi_{pp}$ . This is performed again for input  $[p\vec{a}^i, p\mathbf{A}_{m,n}^i]$ , then we have an input  $[p\vec{c}^i, p\mathbf{C}_{m,n}^i] \in I_{\mathcal{VM}}^p$  for which holds following (considering the proof of Theorem 2):

$$\delta_p([p\vec{a}^i, p\mathbf{A}_{m,n}^i], \delta_p([p\vec{b}^i, p\mathbf{B}_{m,n}^i], [p^s, p\mathbf{S}_{m,n}^s])) = \delta_p([p\vec{c}^i, p\mathbf{C}_{m,n}^i], [p^s, p\mathbf{S}_{m,n}^s]).$$

When the inputs  $[p\vec{a}^i, p\mathbf{A}_{m,n}^i]$  or  $[p\vec{b}^i, p\mathbf{B}_{m,n}^i]$  are applied on the state  $[t^s, \mathbf{S}_{m,n}^s]$  other states react to this change by mapping  $\varphi_{pq}$ . It is obvious that mapping  $\varphi_{pr}$  influences other states by the input from the respective input set. Then for every component of the state on the right side there exists an element from the corresponding input set, that state on the left hand side is included on the right hand side.

- (b) For the second case, we consider different inputs  $[p\vec{a}^i, p\mathbf{A}_{m,n}^i] \in I_{\mathcal{VM}}^p, [q\vec{b}^i, q\mathbf{B}_{m,n}^i] \in I_{\mathcal{VM}}^q$  now. On the left-hand side, we have an influence on two different components in the tuple  $([1^s, \mathbf{1S}_{m,n}^s], [2^s, \mathbf{2S}_{m,n}^s], \dots, [t^s, \mathbf{tS}_{m,n}^s])$  it is evident that the same inputs are the same on the right-hand side if internal links  $\varphi_{pp}, \varphi_{qq}$  do not change. At the moment,  $\varphi_{pp}, \varphi_{pp}$  influence corresponding components of the state, we have suitable inputs on the right-hand side, as in proof of Theorem 2. For mapping the influence of  $\varphi_{pr}$  and  $\varphi_{qs}$  on other components, we have the same situation as case a). Thus, it is obvious that *SMAil* holds condition E-GMAC.

□

In the conclusion of this section, we will demonstrate the theory explained by using the example of *SMAil* to describe and model a situation with several autonomous vehicles in traffic lanes each intending to perform some action.

**Example 4.** We will consider five e-quasi-multiautomata  $\mathbb{MA}_1 = (I_{\mathcal{VM}}^1, \mathbb{S}_{RM}^1, \delta_1), \mathbb{MA}_2 = (I_{\mathcal{VM}}^2, \mathbb{S}_{RM}^2, \delta_2), \mathbb{MA}_3 = (I_{\mathcal{VM}}^3, \mathbb{S}_{RM}^3, \delta_3), \mathbb{MA}_4 = (I_{\mathcal{VM}}^4, \mathbb{S}_{RM}^4, \delta_4), \mathbb{MA}_5 = (I_{\mathcal{VM}}^5, \mathbb{S}_{RM}^5, \delta_5)$ , where every e-quasi-multiautomaton represents a vehicle. The e-quasi-multiautomata  $\mathbb{MA}_2, \mathbb{MA}_3, \mathbb{MA}_4$  are autonomous vehicles and  $\mathbb{MA}_1, \mathbb{MA}_5$  are ordinary vehicles. Figure 5 depicts the state of every vehicle, where the vehicles  $\mathbb{MA}_1, \mathbb{MA}_5$  are denoted in red colour in the same as verge of the road and other colour are used for autonomous vehicles. The complete situation with detection field of each vehicle is presented in Figure 6. In fact, Figure 5 is state of *SMAil*, i.e.,

$$([1^r, \mathbf{1R}_{m,n}^s], [2^s, \mathbf{2S}_{m,n}^s], [3^t, \mathbf{3T}_{m,n}^s], [4^u, \mathbf{4U}_{m,n}^s], [5^v, \mathbf{5V}_{m,n}^s]).$$

Next, we need a matrix of the internal link

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \varphi & \varphi & \varphi & 0 \\ 0 & \varphi & \varphi & \varphi & 0 \\ 0 & \varphi & \varphi & \varphi & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where 0 means no internal link and  $\varphi$  mean internal link between vehicles given by the respective indices. We are going to describe how to proceed, i.e., what input symbols to use, if the grey car, represented by e-quasi-multiautomaton  $\mathbb{MA}_3$ , wants to change lane and turn left to lane 5.

We have two approaches to start changing the lane: we can correct of the state of the vehicle directly by the component on the 2nd, 3rd and 4th position in the input word, or to use the internal link and 6th component of the input. We will use an input  $[(3^3, 3^1, 3^0.98, 3^0, 3^1, 3^{\frac{\pi}{5}}), 3(b_{i,j})^i = 1] \in I_{\mathcal{VM}}^3$ , which will operate on

the 6th component. Then, the resulting state—full correction—will be made by the internal link. The calculation will be done according to Equations (7) and (15), i.e.,

$$\delta_3 \left( \left[ \left( {}_33^t, {}_31^t, {}_30.99^t, {}_30^t, {}_31^t, {}_3\frac{\pi^t}{5} \right), {}_3(b_{i,j})^t = 1 \right], [{}_3t^s, {}_3\mathbf{T}_{n,m}^s] \right) = \left[ ({}_33, {}_330, {}_3192.06, {}_33, {}_31, {}_3\frac{\pi}{5}), {}_3\mathbf{T}_{n,m}^s \right] \quad (20)$$

We obtain a state with new elements on the 5<sup>th</sup>, 6th positions where these components have an effect on other components by means of the internal link  $\varphi_{33}$  between the state and input of the e-quasi-multiautomaton  $\mathbb{M}\mathbb{A}_3$ . With this internal link, we obtain a new input:

$$\varphi_{33} \left( \left[ \left( {}_33^s, {}_330^s, {}_3192.06^s, {}_33^s, {}_31^s, {}_3\frac{\pi^s}{5} \right), {}_3\mathbf{T}_{n,m}^s \right] \right) = [({}_33^t, {}_31^t, {}_31^t, {}_31^t, {}_31^t, {}_30^s), {}_3\mathbf{C}_{n,m}^t],$$

where  ${}_3\mathbf{C}_{n,m}^t = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$ , i.e., matrix to correct the state matrix

detecting other vehicles or obstacles near vehicle  $A_3$ .

When we apply a new input on the state  $[({}_33, {}_330, {}_3192.06, {}_33, {}_31, {}_3\frac{\pi}{5}), {}_3\mathbf{T}_{n,m}^s]$ , we obtain

$$\delta_3 \left( [({}_33^t, {}_31^t, {}_31^t, {}_31^t, {}_31^t, {}_30^t), {}_3\mathbf{C}_{n,m}^s], [{}_3t^s, {}_3\mathbf{T}_{n,m}^s] \right) = [({}_33^s, {}_330^s, {}_3192.06^s, {}_34^s, {}_31^s, {}_30^s), {}_3\mathbf{D}_{n,m}^s],$$

where matrix  $\mathbf{D}$  has the same size and entries as depicted for  $\mathbb{M}\mathbb{A}_3$  in Figure 7.

Now consider internal link  $\varphi_{32}$ , i.e., state of the e-quasi-multiautomaton  $\mathbb{M}\mathbb{A}_2$  will be changed by the state  $[({}_33^s, {}_330^s, {}_3192.06^s, {}_33^s, {}_31^s, {}_3\frac{\pi^s}{5}), {}_3\mathbf{T}_{n,m}^s]$  of the e-quasi-multiautomaton  $\mathbb{M}\mathbb{A}_3$ , which we obtained in Equation (20).

We will proceed using the definition of the link between two different e-quasi-multiautomata given as Equation (17).

$$\varphi_{32} \left( \left[ \left( {}_33^s, {}_330^s, {}_3192.06^s, {}_33^s, {}_31^s, {}_3\frac{\pi^s}{5} \right), {}_3\mathbf{T}_{n,m}^s \right] \right) = \left[ \left( {}_22^t, {}_2\left(\frac{30}{31} + 0.0628\right)^t, {}_20.99^t, {}_20^t, {}_21^t, {}_20^t \right), {}_2\mathbf{M}_{n,m}^t \right], \quad (21)$$

where  ${}_2\mathbf{M}_{n,m}^t$  is a suitable matrix which enables us to obtain a state matrix with entries given in Figure 7 for the state of  $\mathbb{M}\mathbb{A}_2$ . After we apply the input obtained by the internal link  $\varphi_{32}$  in Equation (21) with the help of transition function  $\delta_2$ , we get a new state depicted in Figure 7 for e-quasi-multiautomaton  $\mathbb{M}\mathbb{A}_2$ .

Next state of the (vehicle)  $\mathbb{M}\mathbb{A}_4$  will not affect velocity by internal link  $\varphi_{24}$ , because the input obtained by  $\varphi_{24}$  from state  $[({}_33^s, {}_330^s, {}_3192.06^s, {}_33^s, {}_31^s, {}_3\frac{\pi^s}{5}), {}_3\mathbf{T}_{n,m}^s]$  has parameters  $\lambda = 0$  and  $\alpha = 0$ .

Thus, the input  $\left[ \left( {}_44^t, {}_3\left(\frac{30}{30+1}\right)^t, {}_30.99^t, {}_30^t, {}_31^t, {}_30^s \right), {}_3\mathbf{N}_{n,m}^t \right]$  operates as neutral input on states of  $\mathbb{M}\mathbb{A}_4$  except for the distance given by  ${}_4s_3^s$ . Thus, we will present a new situation on the lanes, i.e., a new state after the application of one input. See Figure 7.

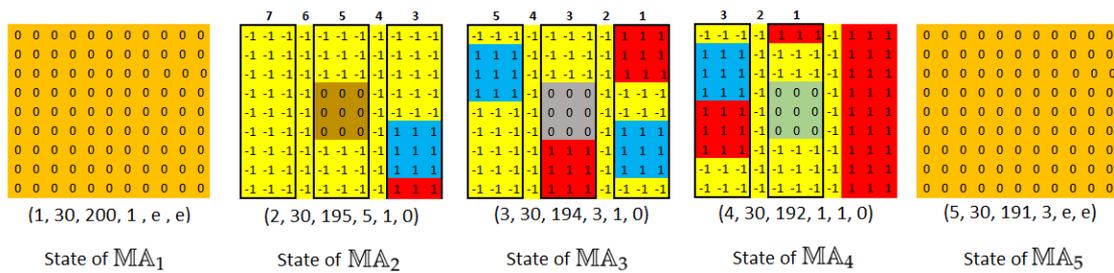


Figure 5. *SMAil* for lane change.

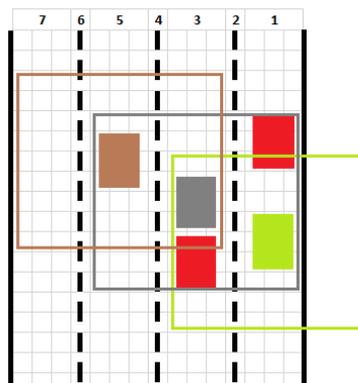


Figure 6. Complete situation in lanes with detection field of each vehicle.

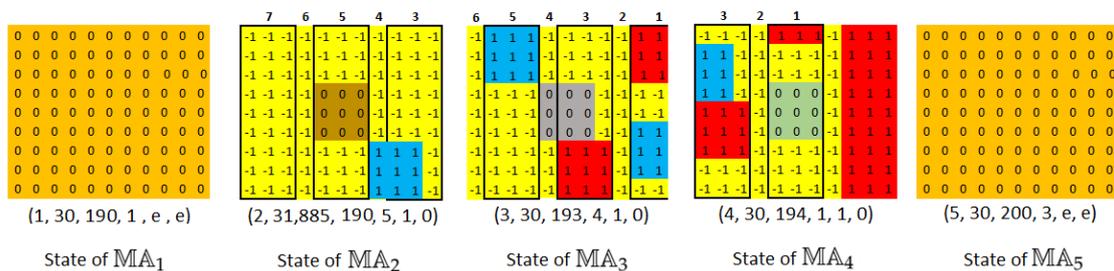


Figure 7. A new state of *SMAil* with the release position.

#### 4. Conclusions

In this paper, we presented the concept of cartesian compositions, which was first introduced by W. Dörfler. In the past, cartesian compositions were generalized in the sense of hyperstructure theory (using complete hypergroups). Now we considered the internal link between the cartesian product of state sets. We described these internal links by matrices and decision functions. These functions determine which state will influence other components. Our modified concept of the cartesian composition is suitable to describe and control systems used for real-life applications, as was shown in examples throughout the paper. While constructing our system called *SMAil*, we assumed that it was made up of e-quasi-multiautomata of a similar nature. This fact affected the proof of the E-GMAC condition (see Theorem 4), which was a substantial simplification of the procedure discussed in [13]. In our future research, we shall concentrate on answering the question of whether the internal link may not remove the necessity to use the extensions of GMAC, i.e., whether the pure GMAC could be used instead of E-GMAC.

**Funding:** This article was produced with the financial support of the Ministry of Education, Youth and Sports within the National Sustainability Programme I, project of Transport R&D Centre (LO1610), on the research infrastructure acquired from the Operation Programme Research and Development for Innovations (CZ.1.05/2.1.00/03.0064).

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. Corsini, P.; Leoreanu, V. *Applications of Hyperstructure Theory*; Kluwer Academic Publishers: Dodrecht, The Netherlands; Boston, MA, USA; London, UK, 2003.
2. Davvaz, B.; Leoreanu-Fotea, V. *Applications of Hyperring Theory*; International Academic Press: Palm Harbor, FL, USA, 2007.
3. Chvalina, J.; Smetana, B. Series of Semihypergroups of Time-Varying Artificial Neurons and Related Hyperstructures. *Symmetry* **2019**, *11*, 927. [[CrossRef](#)]
4. Cristea, I.; Kocijan, J.; Novák, M. Introduction to Dependence Relations and Their Links to Algebraic Hyperstructures. *Mathematics* **2019**, *7*, 885. [[CrossRef](#)]
5. Křehlík, Š.; Vyroubalová, J. The Symmetry of Lower and Upper Approximations, Determined by a Cyclic Hypergroup, Applicable in Control Theory. *Symmetry* **2020**, *12*, 54. [[CrossRef](#)]
6. Novák, M.; Křehlík, Š. EL-hyperstructures revisited. *Soft. Comput.* **2018**, *22*, 7269–7280. [[CrossRef](#)]
7. Al Tahan, M.; Hoskova-Mayerova, Š.; Davvaz, B. Some Results on (Generalized) Fuzzy Multi-Hv-Ideals of Hv-Rings. *Symmetry* **2019**, *11*, 1376. [[CrossRef](#)]
8. Chvalina, J. *Functional Graphs, Quasi-Ordered Sets and Commutative Hypergroups*; Masaryk University: Brno, Czech Republic, 1995. (In Czech)
9. Dörfler, W. The cartesian composition of automata. *Math. Syst. Theory* **1978**, *11*, 239–257. [[CrossRef](#)]
10. Gécseg, F.; Peák, I. *Algebraic Theory of Automata*; Akadémia Kiadó: Budapest, Hungary, 1972.
11. Ginzburg, A. *Algebraic Theory of Automata*; Academic Press: New York, NY, USA, 1968.
12. Novák, M.; Křehlík, Š.; Staněk, D.  $n$ -ary Cartesian composition of automata. *Soft. Comput.* **2019**, *24*, 1837–1849. [[CrossRef](#)]
13. Chvalina, J.; Křehlík, Š.; Novák, M. Cartesian composition and the problem of generalising the MAC condition to quasi-multiautomata. *An. Univ. Ovidius Constanta-Ser. Mat.* **2016**, *24*, 79–100.
14. Ashrafi, A.R.; Madanshekar, A. Generalized action of a hypergroup on a set. *Ital. J. Pure Appl. Math.* **1998**, *15*, 127–135.
15. Chvalina, J. Infinite multiautomata with phase hypergroups of various operators. In Proceedings of the 10th International Congress on Algebraic Hyperstructures and Applications, Brno, Czech Republic, 3 September 2008; Hořková, Š., Ed.; University of Defense: Washington, DC, USA; pp. 57–69.
16. Chvalina, J.; Chvalinová, L. State hypergroups of automata. *Acta Math. Inform. Univ. Ostrav.* **1996**, *4*, 105–120.
17. Chvalina, J.; Hořková-Mayerová, Š.; Dehghan Nezhad, A. General actions of hyperstructures and some applications. *An. Univ. Ovidius Constanta-Ser. Mat.* **2013**, *21*, 59–82 [[CrossRef](#)]
18. Massouros, G.G. Hypercompositional structures in the theory of languages and automata. *An. Șt. Univ. A.I. Țuza Iași, Sect. Inform.* **1994**, *3*, 65–73.
19. Dörfler, W. The direct product of automata and quasi-automata. In *Mathematical Foundations of Computer Science: 5th Symposium*; Mazurkiewicz, A., Ed.; Springer: Gdansk, Poland, 1976; pp. 6–10.
20. Chvalina, J.; Novák, M.; Křehlík, Š. Hyperstructure generalizations of quasi-automata induced by modelling functions and signal processing. In Proceedings of the 16th International Conference of Numerical Analysis and Applied Mathematics, Rhodes, Greece, 13–18 September 2018.
21. Křehlík, Š.; Novák, M. Modified Product of Automata as a Better Tool for Description of Real-Life Systems. In Proceedings of the 17th International Conference of Numerical Analysis and Applied Mathematics, Rhodes, Greece, 23–29 September, 2019.
22. Golestan, K.; Seifzadeh, S.; Kamel, M.; Karray, F.; Sattar, F. Vehicle Localization in VANETs Using Data Fusion and V2V Communication. In Proceedings of the Second ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, Paphos, Cyprus, 21 October 2012; pp. 123–130.
23. Novák, N.; Křehlík, Š.; Ovaliadis, K. Elements of hyperstructure theory in UWSN design and data aggregation. *Symmetry* **2019**, *11*, 734. [[CrossRef](#)]

24. Dresner, K.; Stone, P. A multiagent approach to autonomous intersection management, *J. Artif. Intell. Res.* **2008**, *31*, 591–656. [[CrossRef](#)]
25. Fajardo, D.; Au, T.-C.; Waller, S.T.; Stone, P.; Yang, D. Automated Intersection Control: Performance of Future Innovation Versus Current Traffic Signal Control. *Transp. Res. Rec.* **2011**, *2259*, 223–232. [[CrossRef](#)]
26. Huifu, J.; Jia, H.; Shi, A.; Meng, W.; Byungkyu, B.P. Eco approaching at an isolated signalized intersection under partially connected and automated vehicles environment. *Transp. Res. Part C Emerg. Technol.* **2017**, *79*, 290–307.
27. Latombe, J. *Robot Motion Planning*; Kluwer Academic Publishers: Boston, MA, USA, 1991.
28. Liu, F.; Naraynan, A.; Bai, Q. Effective methods for generating collision free paths for multiple robots based on collision type (demonstration). In Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, Valencia, Spain, 4–8 June 2012; Volume 3, pp. 1459–1460.



© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).