Efficient Synthesis for Monotone Transition Systems and Directed Safety Specifications

Adnane Saoud, Elena A. Ivanova, Antoine Girard

Efficient Synthesis for Monotone Transition Systems and Directed Safety Specifications*.

Adnane Saoud[1,2], Elena Ivanova[1] and Antoine Girard[1]

***Abstract*— In this paper, we introduce an efficient algorithm for control policy synthesis for monotone transition systems and lower (upper) safety specifications. For a monotone transition system the sets of states and inputs are equipped with partial orders, moreover, the transitions preserve the ordering on the states. We propose a lazy algorithm that exploits priorities on the states and inputs. To compute the maximal controlled invariant set, only inputs with the lowest priorities are used. Then, starting from the states with the highest priorities, transitions are computed on-the-fly and only when a particular region of the state space needs to be explored. Once this set is computed, controller synthesis is straightforward by exploring different inputs and using their priorities. We prove the completeness of our algorithm w.r.t the classical safety algorithm. Finally, we illustrate the advantages of the proposed approach on a vehicle platooning problem.**

I. INTRODUCTION

Abstraction-based synthesis techniques have been an ongoing research area in the last decade (see e.g. [1], [2] and the references therein). In symbolic control approaches, a discrete abstraction is constructed from the original dynamical system. When the concrete and abstract systems are related by some behavioral relation such as simulation, alternating simulation or their approximate or directed versions, the discrete controller synthesized for the abstraction can be refined into a controller for the original system. The description of a dynamical system as a discrete abstraction principally enables the use of various techniques developed in the area of supervisory control of discrete event systems.

Symbolic models are often obtained through discretization of the state and input spaces. Due to those discretizations, most symbolic approaches do not scale well.

To tackle this problem different approaches have been proposed. In [3], [4], abstractions were computed using multi-scale state-space discretization. In [5], [6] optimal abstraction parameters were derived to minimize the size of symbolic models. The authors in [7], [8], [9], [10] used lazy synthesis algorithm by exploiting priorities on the inputs. Other approaches have been developed by using compositional methods for abstraction or for controller synthesis [11], [12], [13], [14].

While existing lazy approaches in the literature exploit only priorities on the inputs, in this paper, we also use the priorities[1] on the states to present an efficient synthesis algorithm for monotone transition systems (which are a subclass of transition systems that preserve priorities on the states) and directed safety specifications. The class of monotone transition systems is of practical interest since it arises from monotone dynamical systems, which frequently appears in engineering applications such as traffic networks [15], biological networks [16] and power systems [17]. We show that for the considered problem the maximal controlled invariant is a lower closed set and that it can be computed using only inputs with lower priorities. Then we present an efficient approach to compute the domain of the controller using the concept of basis, which serves as a simpler representation of lower closed sets. Once the maximal controlled invariant is found, we exploit priorities on the inputs to compute the maximal safety controller. Finally, we demonstrate the practicality of our approach on a vehicle platooning problem.

In spirit, the closest works in the literature are [15], [18]. In [15], sparse abstractions were proposed for monotone dynamical systems and directed specifications. We complement their idea by providing an efficient synthesis algorithm for directed safety specifications. In [18], the authors compute controlled invariants for monotone systems using constraint programming. Their notion of s-sequence is relatively close to the characterization of lower closed controlled invariants presented in our work. In this paper, we only focus on lower safety specification, but the results for upper safety specifications can be obtained using the same approach.

The paper is organized as follows. In Section II, some required preliminaries are provided. In Section III, we introduce the class of monotone transition systems. In Section IV, we present an efficient synthesis algorithm for monotone transition systems and lower safety specifications. Finally, in Section V, an illustrative example is proposed in order to show the efficiency of the proposed approach.

*This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 725144). This research was partially supported by Labex DigiCosme (project ANR-11-LABEX-0045-DIGICOSME) operated by ANR as part of the program "Investissement d'Avenir" Idex Paris Saclay (ANR-11-IDEX-0003-02).

[1]Laboratoire des Signaux et Systèmes (L2S), CNRS, CentraleSupélec, Université Paris-Sud, Université Paris-Saclay, 3, rue Joliot-Curie, 91192 Gif-sur-Yvette, cedex, France. adnane.saoud,elena.ivanova,antoine.girard@l2s.centralesupelec.fr

[2]LSV, CNRS, ENS Paris-Saclay, 61, avenue du Président Wilson, 94235 Cachan Cedex, France.

[1]Given a partially ordered set X, we say that a state $x_1 \in X$ has a highest priority than a state $x_2 \in X$ if the state x_1 is bigger than x_2 with respect to the given partial order on the states. The notion of priority is defined similarly for a partially ordred set of inputs.

II. Preliminaries

A. Partial orders

A binary relation $\leq_{\mathcal{L}} \subseteq \mathcal{L} \times \mathcal{L}$ is a partial order if and only if for all $l_1, l_2, l_3 \in \mathcal{L}$ we have: (i) $l_1 \leq_{\mathcal{L}} l_1$, (ii) if $l_1 \leq_{\mathcal{L}} l_2$ and $l_2 \leq_{\mathcal{L}} l_1$ then $l_1 = l_2$ and, (iii) if $l_1 \leq_{\mathcal{L}} l_2$ and $l_2 \leq_{\mathcal{L}} l_3$ then $l_1 \leq_{\mathcal{L}} l_3$. If neither $l_1 \leq_{\mathcal{L}} l_2$ nor $l_2 \leq_{\mathcal{L}} l_1$ holds, we say that l_1 and l_2 are incomparable. The set of all incomparable couples in $\mathcal{L}$ is denoted by $\text{Inc}_{\mathcal{L}}$. We define $\geq_{\mathcal{L}}$ so that $l_1 \geq_{\mathcal{L}} l_2$ if and only if $l_2 \leq_{\mathcal{L}} l_1$.

For a partially ordered set $\mathcal{L}$, half closed-open intervals are $(x, y]_{\mathcal{L}} = \{z \mid x <_{\mathcal{L}} z \leq_{\mathcal{L}} y\}$. Given a partially ordered set $\mathcal{L}$, for $a \in \mathcal{L}$ let $\downarrow a = \{x \in \mathcal{L} \mid x \leq_{\mathcal{L}} a\}$ and $\uparrow a = \{x \in \mathcal{L} \mid a \leq_{\mathcal{L}} x\}$. When $A \subseteq \mathcal{L}$ then its lower closure is $\downarrow A = \bigcup_{a \in A} \downarrow a$. A subset $A \subseteq \mathcal{L}$ is said to be lower closed if $\downarrow A = A$.

B. Transition systems

Definition 1: A transition system is a tuple $S = (X, X_0, U, \Delta, Y, H)$ where X is a set of states, $X_0 \subseteq X$ is a set of initial states, U is a set of inputs, $\Delta \subseteq X \times U \times X$ is a transition relation, Y is a set of outputs, and $H : X \to Y$ is an injective output map.

A transition system is said to be finite if X and U are finite. We introduce notation $x' \in \Delta(x, u)$ as an alternative representation for a transition $(x, u, x') \in \Delta$, where state x' is called a u-successor of state x, for input $u \in U$. This notion could be generalized toward sets in the natural way: for $A \subseteq X$ and $W \subseteq U$, $\Delta(A, W) = \bigcup_{a \in A} \bigcup_{u \in W} \Delta(a, u)$. Similarly we define $\text{Pre}(A, W) = \{x \in X \mid \exists u \in W,\ \Delta(x, u) \subseteq A\}$. For the transition system S, we assume that for all $x \in X$ and for all $u \in U$, $\Delta(x, u) \neq \emptyset$. This means that for any state all the inputs are admissible.

III. Monotone transition systems

A. Monotone transition systems

Let a transition system $S = (X, X_0, U, \Delta, Y, H)$ where the set of outputs is equipped with a partial order $\leq_Y$. Using the injectivity of the output map H, a partial order $\leq_X$ can be defined on the state space X as follows: for $x_1, x_2 \in X$, $x_1 \leq_X x_2$ if and only if $H(x_1) \leq_Y H(x_2)$.

In the paper, we consider a class of transition systems for which transitions (and then trajectories) preserve some partial order on the states.

Definition 2: A transition system $S = (X, X_0, U, \Delta, Y, H)$ is said to be monotone if for all $x_1, x_2 \in X$ and for all $u_1, u_2 \in U$, if $x_1 \leq_X x_2$ and $u_1 \leq_U u_2$, then for any $x_1' \in \Delta(x_1, u_1)$, there exists $x_2' \in \Delta(x_2, u_2)$ satisfying $x_1' \leq_X x_2'$.

Now, let us give some characterizations of monotone transition systems. We first introduce an auxiliary lemma.

Lemma 1: Let a partially ordered set X, and let subsets $A, B \subseteq X$. The set A is included in lower closure of the set B (i.e. $A \subseteq \downarrow B$) if and only if for any $a \in A$, there exists $b \in B$ such that $a \leq_X b$.

Proposition 1: For a transition system $S = (X, X_0, U, \Delta, Y, H)$ the following statements are equivalent:

(i) S is a monotone transition system;

(ii) for all $x_1, x_2 \in X$ and $u_1, u_2 \in U$, if $x_1 \leq_X x_2$ and $u_1 \leq_U u_2$ then $\Delta(x_1, u_1) \subseteq \downarrow \Delta(x_2, u_2)$;

(iii) for all $x \in X$, $u \in U$ we have: $\Delta(\downarrow x, \downarrow u) \subseteq \downarrow \Delta(x, u)$.

B. Abstraction of a monotone control system

Let a monotone discrete-time control system Σ of the form:

$$x(k+1) = f(x(k), u(k), d(k)),\ x(0) \in X_0.$$

where $x(k) \in X$ is a state, $u(k) \in U$ is a control input and $d(k) \in D$ is a disturbance input. It was shown in [15] how to construct a sparse abstraction S related to the original system Σ by an upper alternating simulation relation. This relation is useful for controller refinement when dealing with lower closed specifications. Moreover, the abstraction S satisfies the monotonicity property given in Definition 2. In the following, we provide an efficient synthesis algorithm allowing to deal with monotone transition systems and lower closed safety specifications.

IV. Controller synthesis for safety specifications

A. Maximal safety controller

Consider a transition system S and a safety specification $X^S \subseteq X$ (which can be easily obtained from a subset $Y^S \subseteq Y$ of safe outputs, $X^S = H^{-1}(Y^S)$). We consider a synthesis problem that consists in determining a controller that keeps the trajectories of the system inside a safe set X^S. Let us remark that if a controller keeps the trajectories of the system S into X^S, the output specification $Y^S = H(X^S)$ is satisfied by construction. The safety specification is said to be lower closed if X^S is a lower closed set (which can be obtained from a lower closed set of outputs Y^S).

Given a transition system $S = (X, X_0, U, \Delta, Y, H)$, a controller for S is a set-valued map $C : X \rightrightarrows U$. We define the domain of the controller as $\text{dom}(C) = \{x \in X \mid C(x) \neq \emptyset\}$.

Definition 3: A safety controller C for the transition system S and the safe set X^S satisfies:

- $\text{dom}(C) \subseteq X^S$;
- $\forall x \in \text{dom}(C)$ and $\forall u \in C(x)$, $\Delta(x, u) \subseteq \text{dom}(C)$.

There are in general several controllers that solve the safety problem. A suitable solution is a controller that enables as many actions as possible. This controller C^* is said to be *a maximal safety controller*, in the sense that for any other controller C and for all $x \in X$, we have $C(x) \subseteq C^*(x)$.

Definition 4: Given a transition system S and a safety specification $X^S \subseteq X$. A subset $A \subseteq X^S$ is said to be a controlled invariant if for all $x \in A$ there exists $u \in U$ such that $\Delta(x, u) \subseteq A$.

It was shown in [1] that there exists a maximal controlled invariant given by $Z^* = \text{dom}(C^*)$, which is the union of all controlled invariants.

In this part, we give some characterizations of the maximal safety controller for monotone transition systems and lower closed safety specifications $X^S \subseteq X$. We first introduce the following instrumental lemma.

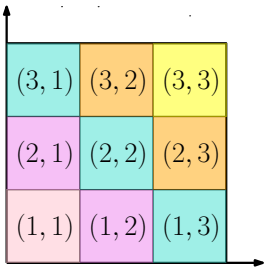

Fig. 1: Illustration of the partitioning of the input set $U = \{1, 2, 3\}^2$. The input set U is equipped with the partial order $\leq$ defined on $\mathbb{R}^2$. The input set U is partitioned as follows: $U_1 = U_{\min} = \{(1, 1)\}$, $U_2 = \{(1, 2), (2, 1)\}, U_3 = \{(1, 3), (2, 2), (3, 1)\}, U_4 = \{(2, 3), (3, 2)\}, U_5 = \{(3, 3)\}$.

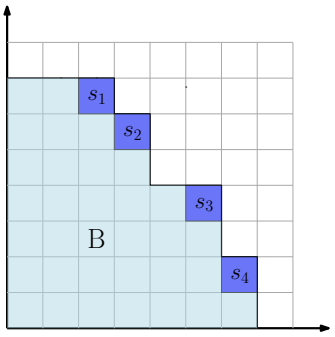

Fig. 2: Illustration of Definition 5. A lower closed set B and its basis $\text{Bas}(B) = \{s_1, s_2, s_3, s_4\}$.

Lemma 2: Let the monotone transition system $S = (X, X_0, U, \Delta, Y, H)$. Let C^* the maximal safety controller for the system S and the lower closed safety specification $X^S \subseteq X$. Let the controller $C : X \rightrightarrows U$ defined for $x \in X$ by: $C(x) = \bigcup_{x' \in (\uparrow x)} C^*(x')$. We have:

(i) $\downarrow \text{dom}(C^*) = \text{dom}(C)$;
(ii) $\text{dom}(C) = \text{dom}(C^*)$.

Proposition 2: Consider a monotone transition system $S = (X, X_0, U, \Delta, Y, H)$. Let C^* be the maximal safety controller enforcing the lower closed safety specification $X^S \subseteq X$. The following properties hold:

(i) $\text{dom}(C^*)$ is a lower closed set;
(ii) $\forall x_1, x_2 \in X$, if $x_1 \leq_X x_2$ then $C^*(x_2) \subseteq C^*(x_1)$;
(iii) $\forall x \in X$, $C^*(x)$ is a lower closed set.

B. Control synthesis for monotone transition systems and directed safety specifications

In this section, we propose an efficient safety algorithm which exploits priorities on states and inputs. The synthesis of the maximal safety controller is done in two steps: first we use only inputs with lower priorities to compute the maximal controlled invariant set Z^*. We then synthesize the maximal controller by exploring different inputs and using their priorities. In the rest of the paper, we only consider finite monotone transition systems. Let us remind that a state $x_1 \in X$ (a control $u_1 \in U$) has a higher priority than a state $x_2 \in X$ (a control $u_2 \in U$) if and only if $x_2 \leq_X x_1$ ($u_2 \leq_U u_1$).

1) Domain of the controller: Given the partial order on the inputs $\leq_U$, we can introduce for $U' \subseteq U$, the operator $\min(U') = \{u \in U' \mid \forall u_1 \in U',\ u \leq_U u_1 \text{ or } (u, u_1) \in \text{Inc}_U\}$. Using this operator the input set U can be partitioned into finite number of sets $U = \cup_{i=1}^N U_i$ defined as follows: $U_{\min} = U_1 = \min(U)$ and $U_{i+1} = \min(U \setminus U_{\underline{i}})$, $i \in \{1, \ldots, N-1\}$, where $U_{\underline{i}} = \cup_{j=1}^i U_j$. An illustration of the input partioning technique is given in Figure 1.

For a monotone transition system $S = (X, X^0, U, \Delta, Y, H)$ we define its reduced transition system by $S_r = (X, X^0, U_1, \Delta, Y, H)$, where $U_1 \subseteq U$ is the set of minimal inputs.

Proposition 3: Let the transition system $S = (X, X^0, U, \Delta, Y, H)$. Let C^* be the maximal safety controller for the system S and the lower closed safety specification $X^S \subseteq X$. Let the reduced transition system $S_r = (X, X^0, U_1, \Delta, Y, H)$ and C_r^* the maximal safety controller for the transition system S_r and safety specification X^S. We have $\text{dom}(C^*) = \text{dom}(C_r^*)$.

The previous result states that to compute the maximal controlled invariant set $Z^* = \text{dom}(C^*)$, it is sufficient to use inputs with lower priorities.

In the sequel, we define the notion of a basis which is adapted from [19]. Indeed the concept of basis serves as a simpler representation of lower closed sets.

Definition 5: Let A be a finite partially ordered set. Let $Z \subseteq A$ be a lower closed set. A set $B = \{s_1, \ldots, s_N\} \subseteq A$ is said to be the basis of Z, denoted $B = \text{Bas}(Z)$, if $Z = \bigcup_{i=1,\ldots,N} \downarrow s_i$ and for all $s_i, s_j \in B$, if $s_i \neq s_j$ then $(s_i, s_j) \in \text{Inc}_A$.

The existence and uniqueness of a finite basis of a lower closed set follows from the fact that the relation $\leq_A$ is a well-quasi-order [19], [20]. An illustration of the concept of basis is given in Figure 2. In the following result, we give a characterization of lower closed controlled invariant sets based on the notion of basis.

Proposition 4: Let the reduced transition system $S_r = (X, X^0, U_1, \Delta, Y, H)$ and the lower closed safety specification $X^S \subseteq X$. Let $Z \subseteq X^S$ be a lower closed set. Z is a controlled invariant for the system S_r and the safe set X^S if and only if the following property holds:

$$\forall x \in \text{Bas}(Z),\ \exists u \in U_1 \text{ s.t } \Delta(x, u) \subseteq Z \qquad (1)$$

We now give the main result of the paper, which states that the maximal controlled invariant set is the maximal by inclusion lower closed set satisfying condition (1).

Theorem 1: Let the reduced transition system $S_r = (X, X^0, U_1, \Delta, Y, H)$ and the lower closed safety specification $X^S \subseteq X$. The maximal controlled invariant set for the system S_r and the specification X^S is the maximal lower closed $Z \subseteq X^S$ satisfying (1).

Intuitively, the previous result means that the computation of the maximal controlled invariant for monotone transition systems and lower safety specifications can be efficiently done using Proposition 4. Indeed, the invariance condition for a set $Z \subseteq X^S$ needs to be checked only on the elements

of the basis (see equation (1)), instead of all the elements of the set Z (see Definition 4), in the case of classical safety synthesis.

2) Computation of the maximal controlled invariant set: In this section, we propose a lazy fixed-point algorithm to compute the maximal controlled invariant. The algorithm is based on condition (1) and deals only with the elements of the basis in each iteration. To compute the maximal controlled invariant, the inputs to Algorithm 1 are $S = S_r$ which represents the reduced transition system, $Z_{ex} = X^S$ is the safety specification and $Z_c = \emptyset$ (this input to the algorithm will not be used for the computation of the maximal controlled invariant set Z^* but for the computation of the maximal controller, as it will be shown in the next section). Algorithm 1 works as follows: the for loop in line 5 iterates over all elements of the basis B. Initially, this is the basis of the set X^S. Once an element $s \in B$ satisfies condition (1) for a given control input $u \in U_1$ (which is equivalent to the condition given in line 9 since $Z_c = \emptyset$), we move to the next element of B, without exploring other inputs. If all control inputs have been explored but none leads to the acceptance condition, the element s is removed and the basis B is updated (lines 14 and 16). Once all elements in B satisfy condition (1) in line 7, the algorithm terminates and the maximal controlled invariant set Z^* is returned. One can check that this maximal controlled invariant is lower closed by construction $Z^* = \downarrow B$. The maximality comes from the fact that we start from the elements with the highest priority (elements of the basis of X^S) and keep removing elements that did not satisfy condition (1) until the fixed-point is reached.

Let us remark that the abstraction is computed on the fly during the synthesis algorithm. Therefore, the elements with lower priorities are only explored when necessary.

3) Maximal safety controller: In this section, we propose an approach that lazily computes the maximal safety controller by exploiting priorities on the inputs. First we introduce some notations: for $i \in \{1, \dots, N\}$, we define the set $Z_i = \mathrm{Pre}(Z^*, U_i) \cap Z^* = \{x \in Z^* \mid \exists\, u \in U_i,\ \Delta(x,u) \subseteq Z^*\}$. Let us remark that $Z_1 = Z^*$. Similarly, we define the set $Z_{\bar{i}} = \mathrm{Pre}(Z^*, U_{\bar{i}}) \cap Z^*$, where $U_{\bar{i}} = \bigcup_{j=i:N} U_j$.

Lemma 3: For any $i \in \{1, \dots, N\}$ the set Z_i defined above is a lower closed set.

Now, similarly to the result of Proposition 4, we will characterize the set Z_i using its basis.

Proposition 5: Let the set Z_i defined above. For a lower closed set $Z \subseteq Z^*$, we have $Z \subseteq Z_i$ if and only if the following property holds:

$$\forall x \in \mathrm{Bas}(Z),\ \exists\, u \in U_i \text{ s.t } \Delta(x,u) \subseteq Z^* \tag{2}$$

We have from Lemma 3 that Z_i is lower closed set. Then, from Proposition 5, Z_i is the maximal set in Z^* satisfying condition (2). Hence, to compute the set Z_i, $i \in \{2, \dots, N\}$, we rely on Algorithm 1, where the used inputs to the algorithm are $S_i = (X, X^0, U_i, \Delta, Y, H)$, $Z_{ex} = Z^*$ and $Z_c = Z^*$.

Algorithm 1: Z = InvariantSet(S, Z_{ex}, Z_c)

Input: Transition system $S = (X, X^0, U, \Delta, Y, H)$, explored set Z_{ex}, controllable set Z_c.
Output: Invariant set Z

```
1  begin
2      B := Bas(Z_ex);
3      B^pr = ∅;
4      while B^pr ≠ B do
5          for all s ∈ B do
6              B^us := ∅;
7              for all u ∈ U do
8                  U_int = ∅;
9                  if Δ(s,u) ⊆ (↓B) ∪ Z_c then
10                     break;
11                 else
12                     U_int = U_int ∪ {u};
13             if U_int = U then
14                 B^us := B^us ∪ {s};
15         B^pr := B;
16         B := Bas(↓(B) \ B^us);
17     return ↓B;
```

Remark 1: Since we start the computation from the set Z^*, all the basis B generated by the algorithm satisfies $\downarrow B \subseteq Z^*$. Then, the condition in line 7 of Algorithm 1 can be written as: there exits $u \in U_i$ such that $\Delta(x,u) \subseteq Z^*$, which is equivalent to condition (2) of Proposition 5.

We now present the key result for the efficient computation of the maximal safety controller C^*.

Proposition 6: Let the sets Z_i, $i \in \{1, \dots, N\}$, defined above, the following properties holds:

(i) for all $i \in \{2, \dots, N\}$, $Z_i \subseteq Z_{i-1}$;
(ii) for all $i \in \{1, \dots, N\}$, $Z_i = Z_{\bar{i}}$.

To compute the maximal safety controller, Algorithm 2 works as follows: the sets Z_i, $i \in \{2, \dots, N\}$ are computed iteratively, starting from Z^*. At each step $i \in \{2, \dots, N\}$, the algorithm starts from the set Z_{i-1} and firstly computes the set Z_i (line 5), (Initially, the algorithm starts from the set $Z^* = Z_1$ and computes the set Z_2). Once this set is computed, for all $x \in Z_{i-1} \setminus Z_i$ the algorithm selects all the inputs $u \in U_{\underline{i-1}}$ satisfying $\Delta(x,u) \subseteq Z^*$ (line 7). Hence, the controller given by Algorithm 2 can be defined for all $x \in Z_{i-1} \setminus Z_i$ by:

$$C(x) = \{u \in U_{\underline{i-1}} \mid \Delta(x,u) \subseteq Z^*\}. \tag{3}$$

and for all $x \in \mathrm{dom}(Z_N)$ by

$$C(x) = \{u \in U_{\underline{N}} \mid \Delta(x,u) \subseteq Z^*\}. \tag{4}$$

Remark 2: We can remark from (i) in Proposition 6 that to compute the set Z_i, $i \in \{2, \dots, N\}$, the explored set Z_{ex} in Algorithm 1 can be given by $Z_{ex} = Z_{i-1}$, instead of $Z_{ex} = Z^*$ (see line 5 in Algorithm 2), which allows the synthesis to be more efficient.

Algorithm 2: Maximal Safety Controller

Input: Transition system $S = (X, X^0, U, \Delta, Y, H)$, Safety specification X^S
Output: Controller C

begin
  $C(X) := \emptyset$;
  $Z^* :=$ InvariantSet$(S_1, X^S, \emptyset)$;
  for $i = 2 \colon N$ **do**
    $Z_i :=$ InvariantSet(S_i, Z_{i-1}, Z^*);
    for $s \in Z_{i-1} \setminus Z_i$ **do**
      $C(s) := \{u \in \underline{U_{i-1}} \mid \Delta(s,u) \subseteq Z^*\}$;
  for $s \in Z_N$ **do**
    $C(s) := \{u \in \underline{U_N} \mid \Delta(s,u) \subseteq Z^*\}$;
  return C;

We are now ready to show the completeness of the controller given by Algorithm 2 w.r.t the maximal safety controller C^*.

Proposition 7: Let the transition system $S = (X, X^0, U, \Delta, Y, H)$ and the lower closed safety specification $X^S \subseteq X$. Let C^* be the maximal safety controller for the system S and specification X^S, and let C defined as in (3) and (4). We have that $C^*(x) = C(x)$ for all $x \in Z^*$.

Remark 3: Let us emphasis that when the partial order on the inputs $\leq_U$ satisfies the following property: for all $i \in \{2, \ldots, N\}$ and for any $(u_{i-1}, u_i) \in U_{i-1} \times U_i$, we have $u_i \leq u_{i-1}$. The synthesis is more efficient. Indeed, for all $x \in Z_{i-1} \setminus Z_i$, we have the existence of $u \in U_{i-1}$ such that $\Delta(x,u) \subseteq Z^*$. Hence, from (iii) in Proposition 2 and since $U_{i-2} \subseteq\downarrow u$, we have that $U_{i-2} \subseteq C^*(x)$. Then, at each step $i \in \{1, \ldots, N\}$, only the set of inputs U_i needs to be explored, instead of $U_{\underline{i}}$ in the general case, which allows to speedup the synthesis of the maximal safety controller.

V. Numerical example

A. Model description and control objective

We consider a vehicle modeled as a point mass m moving along a straight road. The dynamics of the vehicle is adapted from [21] and given by:

$$m\dot{v} = \alpha(u,v) = \begin{cases} u - f_0 - f_1 v - f_2 v^2 & \text{if} \quad v > 0 \\ \max(u - f_0, 0) & \text{if} \quad v = 0 \end{cases} \tag{5}$$

where $v \geq 0$ represents the velocity of the vehicle, $m > 0$ its mass, u is the net engine torque applied to the wheels and the term $f_0 + f_1 v + f_2 v^2$ include the rolling resistance and aerodynamics ($f_0, f_1, f_2 \in \mathbb{R}^+$). In this equation, u is the control input and satisfies $u \in [U_{\min}, U_{\max}]$, where $U_{\min} < 0 < U_{\max}$. Moreover, we include a lead vehicle $w \in W$ (considered as a bounded disturbance) in the system description, the dynamics of the global system is given by:

$$\begin{cases} \dot{d} &= w - v \\ m\dot{v} &= \alpha(u,v). \end{cases} \tag{6}$$

TABLE I: Vehicle and safety parameters

Parameter	Value	Unit
M	1370	Kg
f_0	51.0709	N
f_1	0.3494	Ns/m
f_2	0.4161	Ns^2/m^2
$U_{\min}$	-4031.9	mKg/s^2
$U_{\max}$	2687.9	mKg/s^2
$d_{\min}$	10	m
d'	70	m
$v_{\max}$	15	m/s

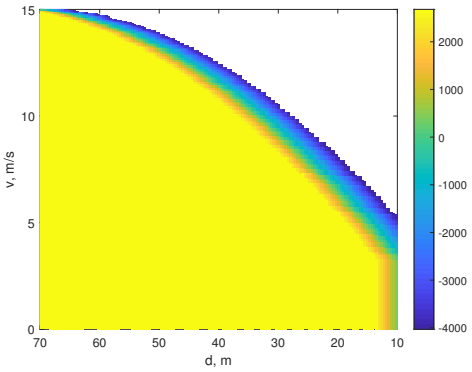

Fig. 3: Maximal safety controller C^*

Remark 4: Let us remark that the system can be easily transformed to a monotone one by using the following change of coordinates: $h = -d$ and $z = -w$.

The objective is to synthesize a controller for the follower vehicle, giving values of input u such that the velocity remains between 0 and $v_{\max}$, and the relative distance between the leader and the follower remains larger than $d_{\min} \geq 0$, while assuming that the velocity of the leader w belongs to the set $W = [0, v_{\max}]$. One can check that since the constraint $v \geq 0$ is directly satisfied from (5), the safety specification is a lower closed set.

From this continuous-time system, we generate a discrete-time model using the sampling period $\tau = 0.5s$, while conserving the monotonicity property of the system.

For the construction of the symbolic abstraction, we use the same partitioning technique presented in [22] with n_x and n_u as state and input discretization parameters.

B. Numerical results

In this section, we numerically illustrates the benefit of the proposed approach. The values shown in Table I are taken from [22]. The implementations has been done in MATLAB, Processor Intel Core i7-8700, 3.20 Hg, RAM 16 GB.

We compute the maximal controlled invariant Z^* using Algorithm 1 and synthesize the maximal safety controller C^* using Algorithm 2. Figure 3 represents the resulting maximal safety controller C^* using the following values of abstraction parameters: $n_u = 500$ for the input discretization and $n_x = (300, 150)$ for the state-space discretization. The color bar represent the given input set $U = [U_{\min}, U_{\max}]$ where the blue color corresponds to the minimal input $U_{\min} =$

TABLE II: Runtime comparison when varying the state-space discretization parameter

Number of states	$T_{la}\,(s)$	$T_{cl}\,(s)$	T_{cl}/T_{la}
(61,31)	0.41s	6.84s	16.79s
(122,62)	1.04s	26.85s	25.85s
(244,124)	3.71s	107.55s	28.98s
(488,248)	14.11s	432.22s	30.64s
(976,496)	54.58s	1695.00s	31.05s

TABLE III: Runtime comparison when varying the input discretization parameter

Number of inputs	$T_{la}\,(s)$	$T_{cl}\,(s)$	T_{cl}/T_{la}
10	0.48s	7.17s	14.8s
20	0.49s	13.87s	29.19s
40	0.64s	27.47s	43.2s
80	0.98s	54.81s	56.06s
160	1.66s	109.47s	65.85s

$-4031.9\ mKg/s^2$ and the yellow color corresponds to the maximal input $U_{\max} = 2687.9\ mKg/s^2$. For a given state (d, v) of the vehicle, Figure 3 shows the maximal allowed control input. Let us mention that following Remark 3, if an input $u \in U$ is allowed by the maximal safety controller C^*, then all inputs satisfying $u' \leq_U u$ are allowed by C^*. Let us also mention that for each state, the controller has only to save the basis of the enabled inputs, which allows to reduce the size of the controller when implementing it in an embedded platform.

We evaluate the performance of the proposed approach w.r.t the classical safety synthesis using two different scenarios. In the first case we vary the state-space discretization parameter n_x while keeping the input discretization parameter as a constant $n_u = 10$. The results of run time comparison are represented in Table II. In the second case we fix $n_x = (61, 31)$ and vary the input discretization parameter n_u. The computational results are given in Table III. In Tables II and III, T_{cl} and T_{la} represent the time needed to compute the maximal safety controller C^*, in seconds. The last column T_{cl}/T_{la} represents the ratio between the classical and lazy synthesis approaches. Tables II and III highlight the practical speedups that can be attained using the lazy approach, while ensuring completeness w.r.t the classical safety algorithm.

VI. Conclusion

In this paper, we have presented an efficient approach to controller synthesis for monotone transition systems and directed safety specifications. The synthesis of the maximal safety controller is done in two steps: first we use only inputs with lower priorities to compute the maximal controlled invariant set. Once this set is computed, we use a lazy approach to efficiently explore different inputs while using their priorities. Numerical results highlight the practical speedups that can be attained using the proposed approach, while ensuring completeness w.r.t the classical safety algorithm.

In future work we will develop more general algorithms allowing to extend the approach to other types of directed specifications, such as reachability, stability or more general properties described by temporal logic formula.

References

[1] P. Tabuada, *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media, 2009.

[2] C. Belta, B. Yordanov, and E. Gol, *Formal methods for discrete-time dynamical systems*. Springer, 2017.

[3] A. Girard, G. Gössler, and S. Mouelhi, "Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models," *IEEE Transactions on Automatic Control*, vol. 61, no. 6, pp. 1537–1549, 2016.

[4] K. Hsu, R. Majumdar, K. Mallik, and A.-K. Schmuck, "Multi-layered abstraction-based controller synthesis for continuous-time systems," in *International Conference on Hybrid Systems: Computation and Control*, pp. 120–129, 2018.

[5] A. Weber, M. Rungger, and G. Reissig, "Optimized state space grids for abstractions," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5816–5821, 2017.

[6] A. Saoud and A. Girard, "Optimal multirate sampling in symbolic models for incrementally stable switched systems," *Automatica*, vol. 98, pp. 58–65, 2018.

[7] J. Camara, A. Girard, and G. Gössler, "Safety controller synthesis for switched systems using multi-scale symbolic models," in *IEEE Conference on Decision and Control and European Control Conference*, pp. 520–525, 2011.

[8] K. Hsu, R. Majumdar, K. Mallik, and A.-K. Schmuck, "Lazy abstraction-based control for safety specifications," in *IEEE Conference on Decision and Control*, pp. 4902–4907, 2018.

[9] O. Hussien and P. Tabuada, "Lazy controller synthesis using three-valued abstractions for safety and reachability specifications," in *IEEE Conference on Decision and Control*, pp. 3567–3572, 2018.

[10] A. Kader, A. Saoud, and A. Girard, "Safety controller design for incrementally stable switched systems using event-based symbolic models," in *European Control Conference*, pp. 1269–1274, 2019.

[11] A. Swikir, A. Girard, and M. Zamani, "From dissipativity theory to compositional synthesis of symbolic models," in *Indian Control Conference*, pp. 30–35, 2018.

[12] E. S. Kim, M. Arcak, and M. Zamani, "Constructing control system abstractions from modular components," in *International Conference on Hybrid Systems: Computation and Control*, pp. 137–146, 2018.

[13] P.-J. Meyer, A. Girard, and E. Witrant, "Safety control with performance guarantees of cooperative systems using compositional abstractions," in *IFAC Conference on Analysis and Design of Hyrbid Systems*, pp. 317–322, 2015.

[14] A. Saoud, P. Jagtap, M. Zamani, and A. Girard, "Compositional abstraction-based synthesis for cascade discrete-time control systems," in *IFAC Conference on Analysis and Design of Hybrid Systems*, pp. 13–18, 2018.

[15] E. S. Kim, M. Arcak, and S. A. Seshia, "Symbolic control design for monotone systems with directed specifications," *Automatica*, vol. 83, pp. 10–19, 2017.

[16] D. Angeli and E. D. Sontag, "Monotone control systems," *IEEE Transactions on automatic control*, vol. 48, no. 10, pp. 1684–1698, 2003.

[17] D. Zonetti, A. Saoud, A. Girard, and L. Fribourg, "A symbolic approach to voltage stability and power sharing in time-varying DC microgrids," in *European Control Conference*, pp. 903–909, 2019.

[18] S. Sadraddini and C. Belta, "Safety control of monotone systems with bounded uncertainties," in *IEEE Conference on Decision and Control*, pp. 4874–4879, 2016.

[19] A. Finkel and P. Schnoebelen, "Well-structured transition systems everywhere!," *Theoretical Computer Science*, vol. 256, no. 1-2, pp. 63–92, 2001.

[20] G. Higman, "Ordering by divisibility in abstract algebras," *Proceedings of the London Mathematical Society*, vol. 3, no. 1, pp. 326–336, 1952.

[21] P. Ioannou and C.-C. Chien, "Autonomous intelligent cruise control," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, pp. 657–672, 1993.

[22] A. Saoud, A. Girard, and L. Fribourg, "Contract based design of symbolic controllers for interconnected multiperiodic sampled-data systems," in *IEEE Conference on Decision and Control*, pp. 773–779, 2018.

APPENDIX

Proof of Proposition 1

Proof: $(i) \iff (ii)$: Let $x_1, x_2 \in X$ and $u_1, u_2 \in U$ with $x_1 \leq_X x_2$ and $u_1 \leq_U u_2$. From Lemma 1, we have that $\Delta(x_1, u_1) \subseteq\downarrow \Delta(x_2, u_2)$ iff for any $x_1' \in \Delta(x_1, u_1)$, there exists $x_2' \in \Delta(x_2, u_2)$ with $x_1' \leq_X x_2'$. Hence, $(i) \iff (ii)$.

$(ii) \implies (iii)$: Let $x \in X$, $u \in U$, $x_1 \in (\downarrow x)$ and $u_1 \in (\downarrow u)$. We have $x_1 \leq_X x$ and $u_1 \leq_U u$. Hence, from (ii) we have that $\Delta(x_1, u_1) \subseteq\downarrow \Delta(x, u)$, for any $x_1 \in (\downarrow x)$ and any $u_1 \in (\downarrow u)$. Then, $\Delta(\downarrow x, \downarrow u)) \subseteq\downarrow \Delta(x, u)$.

$(iii) \implies (ii)$: Let $x_1, x_2 \in X$ and $u_1, u_2 \in U$ with $x_1 \leq_X x_2$ and $u_1 \leq_U u_2$. We have that $x_1 \in (\downarrow x_2)$ and $u_1 \in (\downarrow u_2)$. Hence, from (iii), we have that $\Delta(x_1, u_1) \subseteq \Delta(\downarrow x_2, \downarrow u_2) \subseteq\downarrow \Delta(x_2, u_2)$. ■

Proof of Lemma 2

Proof: From construction of the controller C, it follows immediately that $\downarrow \text{dom}(C^*) = \text{dom}(C)$. Let us prove that $\text{dom}(C^*) = \text{dom}(C)$. Let $x \in X$, since $x \in (\uparrow x)$ we have that $\text{dom}(C^*) \subseteq \text{dom}(C)$. To prove the second inclusion, it is sufficient to show that C is a safety controller for the transition system S and the safety specification X^S. We have that $\text{dom}(C) =\downarrow \text{dom}(C^*) \subseteq\downarrow X^S = X^S$, where the first equality comes from (i), the second inclusion comes from the fact that C^* is a safety controller and the last equality comes from the lower closedness of X^S. Hence, the first condition of Definition 3 is satisfied. Now let $x \in \text{dom}(C)$ and $u \in C(x)$. From construction of the controller C, we have the existence of $x' \in X$ such that $x \leq_X x'$, $x' \in \text{dom}(C^*)$ and $u \in C^*(x')$. Then, we have that $\Delta(x, u) \subseteq\downarrow \Delta(x', u) \subseteq\downarrow \text{dom}(C^*) = \text{dom}(C)$, where the first inclusion comes from (ii) in Proposition 1 and the second inclusion comes from the fact that C^* is a safety controller. Then, condition (ii) in Definition 3 is satisfied and C is safety controller. Since C^* is the maximal controller we have that $\text{dom}(C) \subseteq \text{dom}(C^*)$. ■

Proof of Proposition 2

Proof: (i) We have from (ii) in Lemma 2 that $\text{dom}(C^*) = \text{dom}(C)$. Then, $\downarrow \text{dom}(C^*) =\downarrow \text{dom}(C) = \text{dom}(C^*)$, where the last equality comes from (i) in Lemma 2. Hence, $\text{dom}(C^*)$ is a lower closed set.

(ii) Let $x_1, x_2 \in X$ with $x_1 \leq_X x_2$. Let $u \in C^*(x_2)$. Then, $\Delta(x_2, u) \subseteq \text{dom}(C^*)$. Hence, we have that $\Delta(x_1, u) \subseteq\downarrow \Delta(x_2, u) \subseteq\downarrow \text{dom}(C^*) = \text{dom}(C^*)$, where the first inclusion comes from the fact that S is a monotone transition system and the last equality comes from (i). Hence, by maximality of C^*, we have that $u \in C^*(x_1)$. Then, $C^*(x_2) \subseteq C^*(x_1)$.

(iii) Let $x \in X$, $u \in C^*(x)$ and $u' \in\downarrow u$. We have that $\Delta(x, u') \subseteq\downarrow \Delta(x, u) \subseteq\downarrow \text{dom}(C^*) = \text{dom}(C^*)$, where the first inclusion comes from the monotonicity of the transition system S, the second inclusion comes from the fact that C^* is a safety controller and the last equality comes from the lower closedeness of $\text{dom}(C^*)$. Hence, we have $\Delta(x, u') \subseteq \text{dom}(C^*)$. Then, by maximality of C^*, $u' \in C^*(x)$. ■

Proof of Proposition 3

Proof: Let us define the controller C_r of the reduced transition system S_r and the safe set X^S as follows: for $x \in X$, $C_r(x) = C^*(x) \cap U_1$. First let us prove that $\text{dom}(C_r) = \text{dom}(C^*)$. The inclusion $\text{dom}(C_r) \subseteq \text{dom}(C^*)$ follows immediately from the construction of the controller C_r. Now let $x \in \text{dom}(C^*)$ and let $u \in C(x)$. From (iii) in Proposition 2 we have that $\downarrow u \subseteq C^*(x)$, then there exists $u' \in U_1$ such that $u' \in C^*(x)$. Then, $u' \in C_r(x)$. Hence $x \in \text{dom}(C_r)$ and $\text{dom}(C_r) = \text{dom}(C^*)$. Now let us prove that for all $x \in X$, $C_r(x) = C_r^*(x)$. The first inclusion $C_r(x) \subseteq C_r^*(x)$ follows from maximality of the controller C_r^*. For the second inclusion, we have from maximality of C^* and since $U_1 \subseteq U$ that $C_r^*(x) \subseteq C^*(x)$ for all $x \in X$. Moreover, by construction of C_r^*, we have that $C_r^*(x) \subseteq U_1$ for all $x \in X$. Then, $C_r(x) = C_r^*(x)$ for all $x \in X$. Since $\text{dom}(C_r) = \text{dom}(C^*)$, we have that $\text{dom}(C_r^*) = \text{dom}(C^*)$. ■

Proof of Proposition 4

Proof: Let $Z =\downarrow Z$, we first assume that Z is a controlled invariant. Using the fact that $\text{Bas}(Z) \subseteq Z$, condition (1) is directly satisfied. Now let us prove the second implication. Assume that condition (1) is satisfied, and let us prove that Z is a controlled invariant. For $x \in Z$, there exists $x' \in \text{Bas}(Z)$ such that $x \leq_X x'$. Since $x' \in \text{Bas}(Z)$, we have from (1) the existence of $u \in U_1$ such that $\Delta(x', u) \subseteq Z$. From monotonicity of the transition system S and since $x \leq_X x'$, we have that $\Delta(x, u) \subseteq\downarrow \Delta(x', u) \subseteq\downarrow Z = Z$. Hence, Z is a controlled invariant. ■

Proof of Theorem 1

Proof: Given the transition system S_r and the lower closed safety specification X^S. We have from (i) in Proposition 2 that the maximal controlled invariant for S_r and the safe set X^S, $Z^* = \text{dom}(C^*)$ is a lower closed set. Hence, from Proposition 4, it follows immediately that the maximal controlled invariant set for the system S_r and the specification X^S is the maximal lower closed set $Z \subseteq X^S$ satisfying (1). ■

Proof of Lemma 3

Proof: Let $i \in \{1, \ldots, N\}$, $x \in Z_i$ and $x' \leq_X x$. From definition of Z_i we have the existence of $u \in U_i$ such that $\Delta(x, u) \subseteq Z^*$. Then, we have $\Delta(x', u) \subseteq\downarrow \Delta(x, u) \subseteq\downarrow Z^* = Z^*$, where the first inclusion comes from the monotonicity of the transition system S, the second inclusion comes the construction of the set Z_i and the last inclusion comes from (i) in Proposition 2 ($Z^* = \text{dom}(C^*)$ is a lower closed set). Then, Z_i is a lower closed set. ■

Proof of Proposition 5

The proof is similar to the one of Proposition 4 and then omitted.

Proof of Proposition 6

Proof: (i) Let $i \in \{2, \ldots, N\}$ and $x \in Z_i$. Hence, $x \in Z^*$ and the exists $u \in U_i$ such that $\Delta(x, u) \subseteq Z^*$. Since $U_{i-1} \leq_U U_i$, we have the existence of $u' \in U_{i-1}$ such that $u' \leq_U u$. Then, $\Delta(x, u') \subseteq\downarrow \Delta(x, u) \subseteq\downarrow Z^* = Z^*$, where the first inclusion comes from the monotonicity of the transition system S, the second inclusion comes the construction of the set Z_i and the last inclusion comes from (i) in Proposition 2. Hence, $x \in Z_{i-1}$.

(ii) For $i \in \{1, \ldots, N\}$, the proof follows immediately from (i) and the fact that $Z_{\bar{i}} = \cup_{j=i:N} Z_j$. ■

Proof of Proposition 7

Proof: From the construction of C in (3) and (4) we have that $C(x) \subseteq C^*(x)$ for all $x \in Z^*$. Let Z_i be defined as above, We have from (i) in Proposition 6 that:

$$Z^* = Z_1 \cup Z_2 \cup \ldots \cup Z_N = (Z_1 \setminus Z_2) \cup \ldots \cup (Z_{N-1} \setminus Z_N) \cup Z_N,$$

Let $x \in Z^*$ and $u \in C^*(x)$. If $x \in Z_N$, then using the fact that $U_{\overline{N}} = U$, it follows from (4) that $u \in C(x)$. Now if there exists $i \in \{2, \ldots, N\}$ such that $x \in Z_{i-1} \setminus Z_i = Z_{i-1} \setminus Z_{\bar{i}}$, where the last equality comes from (ii) in Proposition 6. Hence, we have that $u \notin U_{\bar{i}}$. Then, $u \in U_{\underline{i-1}}$. Moreover, C^* is the maximal safety controller, then using the fact that $\Delta(x, u) \subseteq Z^*$, we have from construction of the controller C in (3) that $u \in C(x)$. Then, $C^*(x) = C(x)$ for all $x \in Z^*$. ■