Efficient Synthesis for Monotone Transition Systems and Directed Safety Specifications
Adnane Saoud, Elena A. Ivanova, Antoine Girard

To cite this version:
Adnane Saoud, Elena A. Ivanova, Antoine Girard. Efficient Synthesis for Monotone Transition Systems and Directed Safety Specifications. IEEE Conference on Decision and Control, Dec 2019, Nice, France. 10.1109/CDC40024.2019.9029784 . hal-02281945v2

HAL Id: hal-02281945
https://hal.science/hal-02281945v2
Submitted on 27 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 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).

1Laboratoire des Signaux et Systèmes (L2S), CNRS, CentraleSupelec, 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

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

1Given 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 ordered set of inputs.
II. PRELIMINARIES

A. Partial orders

A binary relation \( \leq_L \subseteq L \times L \) is a partial order if and
only if for all \( l_1, l_2, l_3 \in L \) we have: (i) \( l_1 \leq_L l_1 \), (ii) if \( l_1 \leq_L l_2 \) and \( l_2 \leq_L l_3 \) then \( l_1 \leq_L l_3 \), and, (iii) if \( l_1 \leq_L l_2 \) and \( l_2 \leq_L l_3 \) then \( l_1 \leq_L l_3 \). If neither \( l_1 \leq_L l_2 \) nor \( l_2 \leq_L l_1 \) holds, we say that \( l_1 \) and \( l_2 \) are incomparable. The set of all incomparable couples in \( L \) is denoted by \( Inc_L \). We define \( \geq_L \) so that \( l_1 \geq_L l_2 \) if and only if \( l_2 \leq_L l_1 \).

For a partially ordered set \( L \), half closed-open intervals
are \( (x,y] = \{ z \mid x <_L z \leq_L y \} \). Given a partially ordered
set \( L \), for \( a \in L \) let \( \uparrow a = \{ x \in L \mid x \leq_L a \} \) and \( \downarrow a = \{ x \in L \mid a \leq_L x \} \). When \( A \subseteq L \) then its lower closure is
\( \downarrow A = \bigcup_{a \in A} \downarrow a \). A subset \( A \subseteq L \) 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: \( \Delta(\downarrow x, u) \subseteq \downarrow \Delta(x, u) \).

B. Abstraction of a monotone control system

Let a monotone discrete-time control system \( \Sigma \) of the form:
\[
x(k + 1) = f(x(k), u(k), d(k)), \quad x(0) \in X_0.
\]
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 \( \Sigma \) 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 \( C : X \to U \). 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.
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 $C : X \rightarrow U$ defined for $x \in X$ by: $C(x) = \bigcup_{x' \in \{x\}} C^*(x')$. 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) $\downarrow \text{dom}(C^*)$ is a lower closed set;
(ii) $\forall x_1, x_2 \in X$, if $x_1 \leq x_2$ 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 $x_2 \leq x_1$ ($u_2 \leq_U u_1$).

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 $U = \bigcup_{i=1}^N U_i$ defined as follows: $U_{min} = U_1 = \min(U)$ and $U_{i+1} = \min(U \setminus U_i)$, $i \in \{1, \ldots, N - 1\}$, where $U_0 = \bigcup_{i=1}^N U_i$. An illustration of the input partitioning 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 $\text{dom}(C^*_r) = \text{dom}(C^*)$.

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 $Z = \bigcup_{i=1}^N \downarrow s_i$ 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:

$$\forall x \in \text{Bas}(Z), \exists u \in U_1 \text{ s.t } \Delta(x, u) \subseteq Z$$ (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 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 set of the basis $X^S$. Once an element $s \in B$ satisfies condition (1) for a given control input $u \in U_i$ (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 $i \in \{1, \ldots, N\}$, we define the set $Z_i = \text{Pre}(Z^*, U_i) \cap Z^* = \{x \in Z^* \mid \exists u \in U_i, \Delta(x, u) \subseteq Z^*\}$. Let us remark that $Z_1 = Z^*$. Similarly, we define the set $Z_T = \text{Pre}(Z^*, U_T) \cap Z^*$, where $U_T = \bigcup_{j=i:N} U_j$.

Lemma 3: For any $i \in \{1, \ldots, N\}$ 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:

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

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 $Z_i$, $i \in \{2, \ldots, N\}$, 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: $Z = \text{InvariantSet}(S, Z_{ex}, Z_c)$**

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

**Output:** Invariant set $Z$.

```plaintext
begin
  $B := \text{Bas}(Z_{ex});$
  $B^\text{pr} = \emptyset;$
  while $B^\text{pr} \neq B$ do
    for all $s \in B$ do
      $B^{us} := \emptyset;$
      for all $u \in U$ do
        $U_{int} = \emptyset;$
        if $\Delta(s, u) \subseteq (\downarrow B) \cup Z_c$ then
          break;
        else
          $U_{int} = U_{int} \cup \{u\};$
          if $U_{int} = U$ then
            $B^{us} := B^{us} \cup \{s\};$
          $B^\text{pr} := B;$
          $B := \text{Bas}(\downarrow (B \setminus B^{us}));$
        return $\downarrow B;$

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 exists $u \in U_i$ 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 $Z_i$, $i \in \{1, \ldots, N\}$, defined above, the following properties hold:

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

To compute the maximal safety controller, Algorithm 2 works as follows: the sets $Z_i$, $i \in \{2, \ldots, N\}$ are computed iteratively, starting from $Z^*$. At each step $i \in \{2, \ldots, N\}$, 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 $u \in U_{i-1}$ 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:

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

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

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

Remark 2: We can remark from (i) in Proposition 6 that to compute the set $Z_i$, $i \in \{2, \ldots, N\}$, 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.


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 \( U_{i-2} \subseteq U_{i-1} \), 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 \( U_i \) 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:

\[
\begin{align*}
\dot{m}v = \alpha(u, v) = \begin{cases} 
 u - f_0 - f_1 v - f_2 v^2 & \text{if } v > 0 \\
 \max(u - f_0, 0) & \text{if } v = 0
\end{cases}
\end{align*}
\]

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:

\[
\begin{align*}
\dot{d} &= w - v \\
\dot{m}v &= \alpha(u, v).
\end{align*}
\]

**TABLE I: Vehicle and safety parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \gamma )</td>
<td>1370</td>
<td>( Kg )</td>
</tr>
<tr>
<td>( f_0 )</td>
<td>51.0709</td>
<td>( N )</td>
</tr>
<tr>
<td>( f_1 )</td>
<td>0.3494</td>
<td>( N/m )</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>0.4161</td>
<td>( N\times m/s )</td>
</tr>
<tr>
<td>( U_{\min} )</td>
<td>-4931.9</td>
<td>( m\times Kg/m/s )</td>
</tr>
<tr>
<td>( U_{\max} )</td>
<td>2687.9</td>
<td>( m\times Kg/m/s )</td>
</tr>
<tr>
<td>( d_{\min} )</td>
<td>10</td>
<td>( m )</td>
</tr>
<tr>
<td>( v_{\max} )</td>
<td>15</td>
<td>( m/s )</td>
</tr>
</tbody>
</table>

**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 ensuring 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 \( n_u \) and \( n_v \) 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

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

TABLE III: Runtime comparison when varying the input discretization parameter

<table>
<thead>
<tr>
<th>Number of inputs</th>
<th>$T_{la}$ (s)</th>
<th>$T_{cl}$ (s)</th>
<th>$T_{cl}/T_{la}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.48s</td>
<td>7.17s</td>
<td>14.8s</td>
</tr>
<tr>
<td>20</td>
<td>0.49s</td>
<td>13.87s</td>
<td>29.19s</td>
</tr>
<tr>
<td>40</td>
<td>0.64s</td>
<td>27.47s</td>
<td>43.2s</td>
</tr>
<tr>
<td>80</td>
<td>0.98s</td>
<td>54.81s</td>
<td>56.06s</td>
</tr>
<tr>
<td>160</td>
<td>1.66s</td>
<td>109.47s</td>
<td>65.85s</td>
</tr>
</tbody>
</table>


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 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

APPENDIX

Proof of Proposition 1

Proof: (i) $\iff$ (ii): Let $x_1, x_2 \in X$ and $u_1, u_2 \in U$ with $x_1 \leq x_2$ and $u_1 \subseteq u_2$. From Lemma 1, we have that $\Delta(x_1, u_1) \subseteq \Delta(x_2, u_2)$ iff for any $x'_1 \in \Delta(x_1, u_1)$, there exists $x'_2 \in \Delta(x_2, u_2)$ with $x'_1 \leq x'_2$. Hence, (i) $\iff$ (ii).

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

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

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 $x \leq x'$, $x' \in \text{dom}(C^*)$ and $u \in C^*(x')$. Then, we have that $\Delta(x, u) \subseteq \Delta(x', u) \subseteq \text{dom}(C^*) = \text{dom}(C)$, 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 $\downarrow \text{Dom}(C^*) = \text{dom}(C)$. 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 $x_1 \leq x_2$. Let $u \in C^*(x_2)$. Then, $\Delta(x_2, u) \subseteq \text{dom}(C^*)$. Hence, we have that $\Delta(x_1, u) \subseteq \Delta(x_2, u) \subseteq \text{dom}(C^*) = \text{dom}(C^*)$, where the first inclusion comes from the fact that $C^*$ 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 $u' \in u$. We have that $\Delta(x, u) \subseteq \Delta(x, u') \subseteq \text{dom}(C^*) = \text{dom}(C^*)$, 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 closedness 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 $X^{S_r}$ 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 $u' \subseteq C^*(x)$. 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 $x \leq x'$. 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 $x \leq x'$, we have that $\Delta(x, u) \subseteq \Delta(x', u) \subseteq Z = Z$. 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 $x' \leq x$. From definition of $Z_i$, we have the existence of $u \in U_i$ such that $\Delta(x, u) \subseteq Z^*$. Then, we have $\Delta(x', u) \subseteq \downarrow Z = Z^*$, where the first inclusion comes from the monotonicity of the transition system $S$, the second inclusion comes the construction of the set $Z$, 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, \( \Delta(x, u') \subseteq \Delta(x, u) \subseteq Z^* = Z^* \), 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 \( Z_i = \bigcup_{j=i:N} Z_j \). ■

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 \( U_N = U \), it follows from (4) that \( u \in C(x) \). Now if there exists \( i \in \{2, \ldots, N\} \) such that \( x \in Z_{i-1} \setminus Z_i = Z_{i-1} \setminus Z_i \), where the last equality comes from (ii) in Proposition 6. Hence, we have that \( u \notin U_i \). Then, \( u \in U_{i-1} \). 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 \( x \in Z^* \). ■