

A New Definition of Composition of LTIHA^{*}

Antonia Reißner^[0000-0002-1282-1450] and
Matthias Werner^[0000-0002-3963-2246]

TU Chemnitz Germany
{antonia.reissner|matthias.werner}@informatik.tu-chemnitz.de

Abstract. The aim of this paper is to solve some issues of linear time-invariant hybrid automata as defined in [9]. First these issues are explained on a small example, before the revised definitions of linear time-invariant hybrid automata and their composition operator are presented. Furthermore it is shown that this new composition operator is commutative and associative.

Keywords: Linear Time-Invariant Hybrid Automata · Composition · Hybrid Systems

1 Introduction

Modern systems are becoming more and more complex and hence harder to formally analyze. Because of many catastrophes of well-tested systems in the past the importance of formal verification of systems is widely recognized. Since most modern systems - like embedded systems - have to interact with the physical world but also make discrete computations, a model is needed which can represent both properties. These systems which have analogue (continuous) and digital (discrete) properties are called hybrid systems. A widely considered approach to model a hybrid system is a hybrid automata. These automata are used for example in the field of embedded systems [6].

To model a system as a hybrid automaton all possible states of the system have to be known in advance. Since even small systems could have an immense number of states, modeling errors are quite likely. These considerably small mistakes would invalidate the verification. Hence it is favorable to design only smaller components with fewer states and afterwards construct the whole system using composition.

This property allows to reuse automata which are already known, as well as to incorporate small changes in one of the components of the system with comparatively less effort. A general goal of composition is to transfer properties from the components of a system to the whole system.

A special case of the hybrid automata are the so called linear hybrid automata [2]. For a linear hybrid automaton, the change of a continuous variable

^{*} Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

is described by a linear differential equation. These automata have many subcategories like the timed automata, rectangular automata or hybrid I/O automata. Many aspects of the class of linear hybrid automata are still unknown and in the focus of active research ([3],[4]).

Linear time-invariant hybrid automata (LTIHA) and their composition were first defined by Akhundov et al. ([2],[9]) to model space missions. Properties with regard to the composition were discussed in [1].

In the current paper we are discussing drawbacks of the original approach for LTIHA. We suggest an alternative definition and show some properties of this composition.

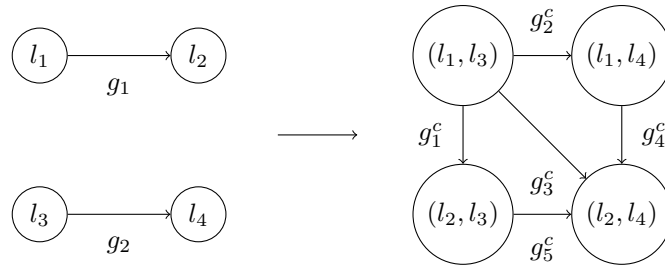
2 Linear Time-Invariant Hybrid Automata

In the first part of this section the original approach is analyzed and two main issues are explained. To overcome these issues a new definition of linear time-invariant hybrid automata and their composition operator is presented in the second part. In the last part the effects of changing the definitions are discussed.

2.1 Issues of the Original Approach

The original definition of LTIHA [9] is based on events. However, in contrast to the typical concept of an event that is connected with an instantaneous change of a system state, events in [9] are associated with a duration. While this may be seen as a clumsy choice of terms, a more crucial issue is an ambiguity in the definition of these events: It is not clear from the definition, what the length of the so-called activation phase is in case of a composition.

A further issue of the former composition operator is discussed using an example. Consider the following automata:¹



A guard is defined by [9] as a triple consisting of three sets. The first is a set of constraints, the second and third sets of events. The transition will only be taken if all constraints are fulfilled and all events in the second set are active. When the transition is taken the events in the third set will be set to active. Consider the guards $g_1 = (C_1, E_1, A_1)$ and $g_2 = (C_2, E_2, A_2)$ for

¹ For the sake of shortness, we assume that the reader is aware of the concepts of [9].

which $E_1 \cap A_2 = E_2 \cap A_1 = \emptyset$ holds. Then the resulting automaton contains the guards $g_1^c = g_4^c = g_1 = (C_1, E_1, A_1)$, $g_2^c = g_5^c = g_2 = (C_2, E_2, A_2)$ and $g_3^c = (C_1 \cup C_2, E_1 \cup E_2, A_1 \cup A_2)$ by definition of the composition operator. In this case when the events in the set $E_1 \cup E_2$ are active and the constraints $C_1 \cup C_2$ are fulfilled, there will always be three guards which are active. Hence this operator constructs a non-determinism that was not present in the former two automata. To overcome this problem guards were changed to two-valued functions mapping to one if a constraint is evaluated to true depending on the variables and their valuation. The new operator was constructed in such a way that if the guard of the diagonal transition is active none of the other guards can be active if there was no non-determinism in the former automata.

To overcome the first issue described above, we suggest a modified approach for LTIHA, that enriches the state by additional variables. Their values at each time point are explicitly described by a function. Alterations of the value at discrete times take the role of events in the original model.

2.2 New Definition

First the formal definition of the linear time-invariant hybrid automata and the corresponding composition operator will be given. To illustrate the changes the aforementioned example is modeled using the new definition.

Definition 1. *A linear time-invariant hybrid automaton is a six tuple*

$$H = (L, X, W, B, T, \mathcal{F})$$

consisting of:

- Set of locations $L = \{l_1, \dots, l_n\}$
- Set X of real valued state variables, decomposable in sets of internal state variables X_I , external state variables X_E with $t \in X_E$ and synchronization variables X_S . The variable t denotes the time of the system.
- A set of valuations

$$W = \{v \mid v : X \times \mathbb{R} \rightarrow \mathbb{R}, \text{ for } t \text{ it holds that } v(t, r_1) = v(t, r_2) \forall r_1, r_2 \in \mathbb{R}\}$$

consisting of functions v which map each variable $x \in X$ to their value at time point $v(t, r_0)$ with r_0 being any real number. This is abbreviated to $t_v = v(t, r_0)$. The valuations for t and all elements in X_E are neither determined nor influenced by the automaton, but given e.g. from the outside world. The valuations of the synchronization variables $s \in X_S$ have the form:

$$v(s, t_v) = \begin{cases} 1 & t_v \in E_s \\ 0 & \text{otherwise} \end{cases}$$

Where $E_s \subseteq \mathbb{R}$ denotes the set of discrete time points at which a synchronization event occurs.

- A set of constraints B depending on a subset M of the set of variables X and their valuation v . All elements in B have to be evaluable to either true (1) or false (0) for every valuation. For each element b in B there is a function

$$g_b(M, v) = \begin{cases} 1 & b \text{ evaluates to true} \\ 0 & \text{otherwise.} \end{cases}$$

The set containing these functions is called guard set C .

- A set of transitions $T \subseteq L \times C \times \mathcal{P}(X_S) \times L$ containing transitions (l_1, g, N, l_2) with $l_1, l_2 \in L$, $g \in C$ and $N \in \mathcal{P}(X_S)$ which can only be taken if $g(M, v) = 1$ holds. In this case M denotes a subset of the variable set X and v a valuation. The set N is a set of events which are generated if the transition is taken. If the transition is taken at the time point \bar{t}_v , then the valuation is modified to:

$$v(s, t_v) = \begin{cases} 1 & t_v \in E_s \cup \{\bar{t}_v\} \\ 0 & \text{otherwise} \end{cases}$$

This holds for all variables $s \in N$. The set E_s is extended to the set $E_s \cup \{\bar{t}_v\}$

- A set of flow functions $\mathcal{F} = \{f_l^x \mid \forall l \in L \forall x \in X_I\}$ containing functions which describe the change of each state variable for each location. These flow functions are solutions to linear ordinary differential equations. These equations have the form :

$$\dot{f}(t) = A \cdot f(t_v) + b u(t_v).$$

Where A and b are constants. The functions $f(t_v)$ and $u(t_v)$ can be vector valued functions.

Definition 2. The state (l_j, v) of a linear time-invariant hybrid automaton consists of the location $l_j \in L$ and the valuation v .

These changes of the linear time-invariant hybrid automata enable a redefinition of the corresponding composition operator. Without the change of the underlying automata the new composition operator would produce an automaton which is not necessarily a LTIHA.

Definition 3. Two automata $H^1 = (L^1, X^1, W^1, B^1, T^1, \mathcal{F}^1)$ and $H^2 = (L^2, X^2, W^2, B^2, T^2, \mathcal{F}^2)$ can be composed using the operator \parallel if

$$\forall x \in X^1 \cap X^2 : v^1(x, t_{v^1}) = v^2(x, t_{v^2})$$

holds. The resulting automaton $H^c = H^1 \parallel H^2$ equals $(L^c, X^c, W^c, B^c, T^c, \mathcal{F}^c)$ with the following properties:

- The set of locations L^c equals $L^1 \times L^2$.
- The set of variables X^c is equal to $X^1 \cup X^2$ and can be decomposed in the set of internal variables $X_I^c = X_I^1 \cup X_I^2$,

A New Definition of Composition of LTIHA

the set of external variables $X_E^c = X_E^1 \cup X_E^2 \setminus X_I^c$ and
the set of synchronization variables $X_S^c = X_S^1 \cup X_S^2$.

- The set W^c consists of valuations v^c having the form

$$v^c(x, t_{v^c}) = \begin{cases} v^1(x, t_{v^c}) & x \in X^1 \\ v^2(x, t_{v^c}) & \text{otherwise.} \end{cases}$$

- The set of constraints B^c equals the set of all to true or false evaluable constraints over a subset of the set of variables X^c and the valuation v^c . The set C^c is the set of two-valued functions which map to 1 if and only if the respective constraint evaluates to true.
- Let $(l_i^1, g_{b^1}, N^1, l_k^1) \in T^1$ and $(l_j^2, g_{b^2}, N^2, l_l^2) \in T^2$ be transitions in the corresponding automata.
 - The transitions $((l_i^1, l^2), g_{b^1 \wedge b_{l^2}}, N^1, (l_k^1, l^2))$ and $((l^1, l_j^2), g_{b^2 \wedge b_{l^1}}, N^2, (l^1, l_l^2))$ are elements of the set of transitions T^c for all locations $l^2 \in L^2$ and $l^1 \in L^1$. Let $B_o^1(l^1)$ be the set of the constraints b_n of all guards g_{b_n} such that there exists $l_n \in L^1$ with $(l_1, g_{b_n}, N, l_n) \in T^1$ for any $N \in \mathcal{P}(X_S^1)$. The constraint $b_{l^1}^1$ denotes $\bigwedge_{b_i \in B_o^1(l^1)} \neg b_i$. The new guard constraints $b^1 \wedge b_{l^2}^2$ and $b^2 \wedge b_{l^1}^1$ are defined by:

$$b^1 \wedge b_{l^2}^2(M, v^c) = b^1(M \cap X^1, v^c|_1) \wedge b_{l^2}^2(M \cap X^2, v^c|_2)$$

$$b^2 \wedge b_{l^1}^1(M, v^c) = b^2(M \cap X^2, v^c|_2) \wedge b_{l^1}^1(M \cap X^1, v^c|_1).$$

With the valuation $v^c|_2 : X^2 \times \mathbb{R} \rightarrow \mathbb{R}$ which is defined as

$$\forall y \in X^2, n \in \mathbb{R} : v^c|_2(y, n) = v^c(y, n).$$

The valuation $v^c|_1$ is defined analogously.

- The transition $((l_i^1, l_j^2), g_{b^1 \wedge b^2}, N^1 \cup N^2, (l_k^1, l_l^2))$ is an element of T^c with the guard constraint $b^1 \wedge b^2$ defined as:

$$b^1 \wedge b^2(M, v^c) = b^1(M \cap X^1, v^c|_1) \wedge b^2(M \cap X^2, v^c|_2).$$

- The set \mathcal{F}^c consists of flow functions $f_{(l_i^1, l_j^2)}^x$. At the location (l_i^1, l_j^2) the equation $f_{(l_i^1, l_j^2)}^x = f_{l_i^1}^x + f_{l_j^2}^x$ holds.

In the following table guards for one specific example based on the above example are given for both the new and the old definition.

	Old Notation	New Notation
g_1	$(\{x \leq 10\}, \{event1\}, \{event2\})$	$(g_{b_1}, \{event2\})$
g_2	$(\{x \leq 10, y > 3\}, \{event3\}, \{event4\})$	$(g_{b_2}, \{event4\})$
g_1^c	g_1	$(g_{b_1 \wedge b_{l_3}^2}, \{event2\})$
g_2^c	g_2	$(g_{b_{l_2}^2 \wedge b_2}, \{event4\})$
g_3^c	$(\{x \leq 10, y > 3\}, \{event1, event3\}, \{event2, event4\})$	$(g_{b_1 \wedge b_2}, \{event2, event4\})$
g_4^c	g_1	$(g_{b_1}, \{event2\})$
g_5^c	g_2	$(g_{b_2}, \{event4\})$

Here the constraints are defined as follows:

$$b_1(\{x, event1\}, v) = (v(x, v(t)) \leq 10) \wedge v(event1, v(t)),$$

$$b_2(\{x, y, event3\}, v) = (v(x, v(t)) \leq 10) \wedge (v(y, v(t)) > 3) \wedge v(event3, v(t)).$$

Hence the resulting guard functions are:

$$\begin{aligned} & (b_1 \wedge b_{i_3}^2)(\{x, y, event1, event3\}, v) \\ &= b_1(\{x, event1\}, v) \wedge b_{i_3}^2(\{x, y, event3\}, v) \\ &= (v(x, v(t)) \leq 10) \wedge v(event1, v(t)) \\ &\quad \wedge \neg((v(x, v(t)) \leq 10) \wedge (v(y, v(t)) > 3) \wedge v(event3, v(t))) \\ &= (v(x, v(t)) \leq 10) \wedge v(event1, v(t)) \\ &\quad \wedge ((v(x, v(t)) > 10) \vee (v(y, v(t)) \leq 3) \vee \neg v(event3, v(t))) \\ &= ((v(x, v(t)) \leq 10) \wedge v(event1, v(t))) \wedge ((v(x, v(t)) > 10) \\ &\quad \vee ((v(x, v(t)) \leq 10) \wedge v(event1, v(t))) \wedge ((v(y, v(t)) \leq 3) \vee \neg v(event3, v(t)))) \\ &= 0 \vee ((v(x, v(t)) \leq 10) \wedge v(event1, v(t))) \wedge ((v(y, v(t)) \leq 3) \vee \neg v(event3, v(t))) \\ &= ((v(x, v(t)) \leq 10) \wedge v(event1, v(t))) \wedge ((v(y, v(t)) \leq 3) \vee \neg v(event3, v(t))) \\ & (b_{i_1}^1 \wedge b_2)(\{x, y, event1, event3\}, v) \\ &= b_{i_1}^1(\{x, event1\}, v) \wedge b_2(\{x, y, event3\}, v) \\ &= \neg((v(x, v(t)) \leq 10) \wedge v(event1, v(t))) \\ &\quad \wedge ((v(x, v(t)) \leq 10) \wedge (v(y, v(t)) > 3) \wedge v(event3, v(t))) \\ &= ((v(x, v(t)) < 10) \vee \neg v(event1, v(t))) \\ &\quad \wedge ((v(x, v(t)) > 10) \wedge (v(y, v(t)) \leq 3) \wedge v(event3, v(t))) \\ &= (v(x, v(t)) < 10) \wedge ((v(x, v(t)) > 10) \wedge (v(y, v(t)) \leq 3) \wedge v(event3, v(t))) \\ &\quad \vee \neg v(event1, v(t)) \wedge ((v(x, v(t)) > 10) \wedge (v(y, v(t)) \leq 3) \wedge v(event3, v(t))) \\ &= 0 \vee \neg v(event1, v(t)) \wedge ((v(x, v(t)) > 10) \wedge (v(y, v(t)) \leq 3) \wedge v(event3, v(t))) \\ &= \neg v(event1, v(t)) \wedge ((v(x, v(t)) > 10) \wedge (v(y, v(t)) \leq 3) \wedge v(event3, v(t))) \\ & (b_1 \wedge b_2)(\{x, y, event1, event3\}, v) \\ &= b_1(\{x, event1\}, v) \wedge b_2(\{x, y, event3\}, v) \\ &= (v(x, v(t)) \leq 10) \wedge v(event1, v(t)) \wedge ((v(x, v(t)) \leq 10) \wedge (v(y, v(t)) > 3) \\ &\quad \wedge v(event3, v(t))) \\ &= (v(x, v(t)) \leq 10) \wedge v(event1, v(t)) \wedge (v(y, v(t)) > 3) \wedge v(event3, v(t)) \end{aligned}$$

Depending on the active events only one of these three guards can be active by construction. Therefore no non-determinism is introduced.

2.3 Discussion

Besides solving the issues mentioned in the first part of this section, the redefinition has some further advantages.

In the new definition the underlying graph structure of the composed automaton - expressed as locations and transitions - is independent of the respective guards. The composition of the new automata with regard to the locations and transitions is equivalent to the strong graph product. This graph product is already known to be commutative up to an isomorphism and associative [5].

Furthermore by definition of functions as guards, more conditions to change the location of an automaton can be expressed. Consider the inactivity of an event as a condition. This can not be implemented in the original automata. Some conditions can be modeled in a more compact way using the new introduced guard functions. For example to express that one of two guards is active, multiple transitions or guards were necessary in the former version. Whereas now only one function is necessary.

Besides that the original LTIHA only allowed transitions with disjoint input and output event sets. For the new definition this constraint is not necessary.

The introduction of a variable subset X_E allows to design dependencies on values which are not manipulated by the automaton itself. With these variables the automaton can interact besides synchronization with other automata or other systems which are not designed as LTIHA.

3 Properties of the composition

The general idea of the composition operator is to enable the construction of the whole system by composing small automata. The composition of two automata should behave equivalent to the two automata running parallel while interacting using shared variables. From a formal perspective running parallel can only be considered as commutative, hence the operator should have the same property. Since there should be no difference in which order the components are build and composed together, it is needed that the composition is associative as well as commutative.

The importance of the composition operator to be commutative and associative was already stated in [7]. If these properties would not hold the composition operator could not be used in the above described way.

In the following the automata H^1 , H^2 and H^3 are defined as:

$$\begin{aligned} H^1 &= (L^1, X^1, W^1, B^1, T^1, \mathcal{F}^1) \\ H^2 &= (L^2, X^2, W^2, B^2, T^2, \mathcal{F}^2) \\ H^3 &= (L^3, X^3, W^3, B^3, T^3, \mathcal{F}^3). \end{aligned}$$

3.1 Commutative

Theorem 1. *The composition operator is commutative up to an isomorphism. This means if*

$$F = H^1 \parallel H^2 = (L^F, X^F, W^F, B^F, T^F, \mathcal{F}^F)$$

and

$$G = H^2 \parallel H^1 = (L^G, X^G, W^G, B^G, T^G, \mathcal{F}^G)$$

then there exists a bijective map $\phi : L^F \rightarrow L^G$ such that

$\forall l_1, l_2 \in L^F : (l_1, g, N, l_2) \in T^F$ if and only if $(\phi(l_1), g, N, \phi(l_2)) \in T^G$. Besides that $X^F = X^G$, $W^F = W^G$, $B^F = B^G$ and $\mathcal{F}^F = \mathcal{F}^G$ holds.

Proof.

$L^F \cong L^G$ Let ϕ be a map $\phi : (l_1, l_2) \mapsto (l_2, l_1)$ with $l_1 \in L^1, l_2 \in L^2$ then the location (l_1, l_2) is an element of $L^1 \times L^2 = L^F$ and (l_2, l_1) an element of $L^2 \times L^1 = L^G$. Therefore ϕ is a map $\phi : L^F \rightarrow L^G$. This map is injective and L^F and L^G have by construction the same number of elements so ϕ is a bijection.

$X^F = X^G$ The sets X^F and X^G are equal due to the commutative nature of the set union. By the same argument the internal and synchronization variables are equal in both automata. Consider the set of external variables of the automata H^F . This set is by construction equal to

$$(X_E^1 \cup X_E^2) \setminus X_I^F = (X_E^1 \cup X_E^2) \setminus X_I^G.$$

This equals the set of external variables of H^G .

$W^F = W^G$ Consider an element of the set W^F :

$$v^F(x, t_{v^F}) = \begin{cases} v^1(x, t_{v^F}) & x \in X^1 \\ v^2(x, t_{v^F}) & \text{otherwise} \end{cases}$$

and an element of the set W^G :

$$v^G(x, t_{v^G}) = \begin{cases} v^2(x, t_{v^G}) & x \in X^2 \\ v^1(x, t_{v^G}) & \text{otherwise.} \end{cases}$$

It can be observed that the only difference between these valuations is for variables in $X^1 \cap X^2$. By definition of the composition operator v^1 and v^2 have to be equal for all variables in the set $X^1 \cap X^2$. So there is no difference between these functions. Hence every function in W^F is also an element of W^G and every function in W^G is also an element of W^F . Therefore the sets have to be equal.

$B^F = B^G$ It was already shown that $X^F = X^G$ and $W^F = W^G$ so it follows that $B^F = B^G$ holds by definition. Since the constraint sets are equal, the guard sets are equal as well.

$T^F \cong T^G$ Let $l_a = (l_1, l_2)$ and $l_b = (l_3, l_4)$ be locations in the automaton F . The transition $(l_a, g_b, N, l_b) \in T^F$ exists in F if and only if :

- 1) $(l_1, g_{b^1}, N^1, l_3) \in T^1$, $(l_2, g_{b^2}, N^2, l_4) \in T^2$,
 $b = b^1 \wedge b^2$ and $N = N^1 \cup N^2$ or
- 2) $(l_1, g_{b^1}, N^1, l_3) \in T^1$, $l_2 = l_4$,
 $b = b^1 \wedge b_{l_2}^2$ and $N = N^1$ or
- 3) $(l_2, g_{b^2}, N^2, l_4) \in T^2$, $l_1 = l_3$,
 $b = b^2 \wedge b_{l_1}^1$ and $N = N^2$

The transition $(\phi(l_a), g_b, N, \phi(l_b)) = ((l_2, l_1), g_b, N, (l_4, l_3)) \in T^G$ exists in the automaton G if and only if:

- a) $(l_2, g_{b^2}, N^2, l_4) \in T^2$, $(l_1, g_{b^1}, N^1, l_3) \in T^1$,
 $b = b^2 \wedge b^1$ and $N = N^2 \cup N^1$ or
- b) $(l_2, g_{b^2}, N^2, l_4) \in T^2$, $l_1 = l_3$,
 $b = b^2 \wedge b_{l_1}^1$ and $N = N^2$ or
- c) $(l_1, g_{b^1}, N^1, l_3) \in T^1$, $l_2 = l_4$,
 $b = b^1 \wedge b_{l_2}^2$ and $N = N^1$

It can be observed that the condition 1) and a) as well as 2) and c) as well as 3) and b) are equivalent. In these three cases the guards and the sets of activated synchronization variables are equal as well. So the automata F and G have the same underlying graph structure except for an isomorphism.

$\mathcal{F}^F = \mathcal{F}^G$ Follows trivially by the commutativity of addition.

3.2 Associative

Theorem 2. *The composition operator is associative, this means that*

$$H^1 \parallel (H^2 \parallel H^3) = (H^1 \parallel H^2) \parallel H^3$$

holds for all linear time-invariant hybrid automata H^1, H^2 and H^3 .

Proof. Two automata

$$F = H^1 \parallel (H^2 \parallel H^3) = (L^F, X^F, W^F, B^F, T^F, \mathcal{F}^F)$$

and

$$G = (H^1 \parallel H^2) \parallel H^3 = (L^G, X^G, W^G, B^G, T^G, \mathcal{F}^G)$$

are equal if and only if

$$(L^F, X^F, W^F, B^F, T^F, \mathcal{F}^F) = (L^G, X^G, W^G, B^G, T^G, \mathcal{F}^G)$$

holds. For shortness of notation the variable set of an automaton $(H^2 \parallel H^3)$ is addressed by $X^{2,3}$ a valuation by $v^{2,3}$ and a guard constraint by $b^{2,3}$. For an automaton $H^1 \parallel H^2$ this is defined analogously.

$L^F = L^G$ It holds that $L^F = L^1 \times (L^2 \times L^3) = L^1 \times L^2 \times L^3 = L^1 \times (L^2 \times L^3) = L^G$.
 $X^F = X^G$ By definition it holds that $X^F = X^1 \cup (X^2 \cup X^3)$. Because the set union is associative this is equal to $(X^1 \cup X^2) \cup X^3 = X^G$. The same holds for the internal and the synchronization variables. The set of external variables in F is equal to

$$X_E^F = (X_E^1 \cup X_E^{2,3}) \setminus X_I^F = (X_E^1 \cup (X_E^2 \cup X_E^3 \setminus X_I^{2,3})) \setminus X_I^F.$$

By construction $X_I^{2,3} \subseteq X_I^F$ so it holds that

$$X_E^F = (X_E^1 \cup X_E^2 \cup X_E^3) \setminus X_I^F.$$

Antonia Reißner, Matthias Werner

The set of external variables in G is equal to

$$X_E^G = (X_E^{1,2} \cup X_E^3) \setminus X_I^G = ((X_E^1 \cup X_E^2 \setminus X_I^{1,2}) \cup X_E^3) \setminus X_I^G.$$

By construction $X_I^{1,2} \subseteq X_I^G$ so it holds that

$$X_E^G = (X_E^1 \cup X_E^2 \cup X_E^3) \setminus X_I^G = X_E^F.$$

$W^F = W^G$ A valuation in W^F has the form

$$v^F(x, t_{v^F}) = \begin{cases} v^1(x, t_{v^F}) & x \in X^1 \\ v^{2,3}(x, t_{v^F}) & \text{otherwise} \end{cases} = \begin{cases} v^1(x, t_{v^F}) & x \in X^1 \\ v^2(x, t_{v^F}) & x \in X^2 \\ v^3(x, t_{v^F}) & \text{otherwise} \end{cases}$$

and in W^G the form

$$v^G(x, t_{v^G}) = \begin{cases} v^{1,2}(x, t_{v^G}) & x \in X^{1,2} \\ v^3(x, t_{v^G}) & \text{otherwise} \end{cases} = \begin{cases} v^1(x, t_{v^G}) & x \in X^1 \\ v^2(x, t_{v^G}) & x \in X^{1,2} \setminus X^1 \\ v^3(x, t_{v^G}) & \text{otherwise.} \end{cases}$$

Hence they denote the same function. So each function that is in W^F is also in W^G and each function that is in W^G is also in W^F . Therefore the equality $W^F = W^G$ holds.

$B^F = B^G$ It was already shown that $X^F = X^G$ and $W^F = W^G$ so this follows by definition. The set of constraints are equal so the set of guards are equal in both automata as well.

$T^F = T^G$ The automata $F = H^1 \parallel (H^2 \parallel H^3)$ and $G = (H^1 \parallel H^2) \parallel H^3$ have the same underlying edge structure since the strong graph product is known to be associative. It has to be shown that those edges have the same guard and produce the same events in both automata.

Since the logical and as well as the set union are known to be associative it is clear that most of the edges have the same guard constraint and set the same variables to one if taken. It remains to show that the edge between (l_1, l_2, l_3) and (l_4, l_5, l_6) with $l_2 = l_5$ and $l_3 = l_6$ have the same guard in both automata.

A New Definition of Composition of LTIHA

Consider the constraint of the guard mapping to 1 of this edge in $F = H^1 \parallel (H^2 \parallel H^3)$:

$$b \tag{1}$$

$$= b_1 \wedge b_{(l_2, l_3)}^{2,3} \tag{2}$$

$$= b_1 \wedge_{b_i \in B_o^{2,3}(l_2, l_3)} \neg b_i \tag{3}$$

$$= b_1 \wedge \wedge_{b_2 \in B_o^2(l_2)} \wedge_{b_3 \in B_o^3(l_3)} (\neg(b_2 \wedge b_3) \wedge \neg(b_{l_2}^2 \wedge b_3) \wedge \neg(b_2 \wedge b_{l_3}^3)) \tag{4}$$

$$= b_1 \wedge \wedge_{b_2 \in B_o^2(l_2)} \wedge_{b_3 \in B_o^3(l_3)} ((\neg b_2 \vee \neg b_3) \wedge (\neg b_{l_2}^2 \vee \neg b_3) \wedge (\neg b_2 \vee \neg b_{l_3}^3)) \tag{5}$$

$$= b_1 \wedge \wedge_{b_2 \in B_o^2(l_2)} \wedge_{b_3 \in B_o^3(l_3)} ((\neg b_2 \vee \neg b_3) \wedge (\neg \wedge_{b_2 \in B_o^2(l_2)} \neg b_2 \vee \neg b_3) \wedge (\neg b_2 \vee \neg \wedge_{b_3 \in B_o^3(l_3)} \neg b_3)) \tag{6}$$

$$= b_1 \wedge \wedge_{b_2 \in B_o^2(l_2)} \wedge_{b_3 \in B_o^3(l_3)} ((\neg b_2 \vee \neg b_3) \wedge (\vee_{b_2 \in B_o^2(l_2)} b_2 \vee \neg b_3)) \tag{7}$$

$$\wedge (\neg b_2 \vee \vee_{b_3 \in B_o^3(l_3)} b_3)) \tag{8}$$

$$= b_1 \wedge (\wedge_{b_2 \in B_o^2(l_2)} \wedge_{b_3 \in B_o^3(l_3)} (\neg b_2 \vee \neg b_3)) \tag{9}$$

$$\wedge (\wedge_{b_3 \in B_o^3(l_3)} (\vee_{b_2 \in B_o^2(l_2)} b_2 \vee \neg b_3)) \tag{10}$$

From line 9 follows that if $\forall b_a \in B_o^2(l_2) : b_a = 0$ then $\forall b_3 \in B_o^3(l_3)$ it holds that $b_3 = 0$. Line 10 indicates that if $\forall b_3 \in B_o(l_3) : b_3 = 0$ then it follows that $\forall b_2 \in B_o(l_2) : b_2 = 0$ holds. Consider now the case that there exists at least one $b_i \in B_o^2(l_2)$ with $b_i = 1$ then from line 8 it is concluded that $\forall b_3 \in B_o(l_3) : b_3$ equals to 0. But by the observation from line 9 it follows $\forall b_a \in B_o^2(l_2) : b_a$ equals to 0. This is a contradiction to the existence of a $b_i \in B_o^2(l_2)$ with $b_i = 1$. For a $b_j \in B_o^3(l_3)$ with $b_j = 1$ it follows analogously. In conclusion b can only be true if $\forall b_a \in B_o^2(l_2) : b_a = 0$ and $\forall b_3 \in B_o^3(l_3) : b_3 = 0$ and $b_1 = 1$ holds. So b can be written as:

$$b = b_1 \wedge (\wedge_{b_2 \in B_o(l_2)} \neg b_2) \wedge (\wedge_{b_3 \in B_o(l_3)} \neg b_3) = b_1 \wedge b_{l_2}^2 \wedge b_{l_3}^3.$$

Consider the constraint of the guard mapping to 1 of the same edge in $G = (H^1 \parallel H^2) \parallel H^3$:

$$b = (b_1 \wedge b_{l_2}^2) \wedge b_{l_3}^3$$

So the same transition has the same guard constraint in both automata. The following table gives an overview over the guard constraints and set of synchronization variables of each transition in both automata.

	automaton F		automaton G	
	b	N	b	N
(l_1, l_2, l_3) to (l_4, l_5, l_6)	$b_1 \wedge (b_2 \wedge b_3)$	$N^1 \cup (N^2 \cup N^3)$	$(b_1 \wedge b_2) \wedge b_3$	$(N^1 \cup N^2) \cup N^3$
$l_2 = l_5$	$b_1(\wedge b_{i_2}^2 \wedge b_3)$	$N^1 \cup N^3$	$(b_1 \wedge b_{i_2}^2) \wedge b_3$	$N^1 \cup N^3$
$l_3 = l_6$	$b_1(\wedge b^2 \wedge b_{i_3}^3)$	$N^1 \cup N^2$	$(b_1 \wedge b^2) \wedge b_{i_3}^3$	$N^1 \cup N^2$
$l_1 = l_4$	$b_{i_1}^1 \wedge (b_2 \wedge b_3)$	$N^2 \cup N^3$	$(b_{i_1}^1 \wedge b_2) \wedge b_3$	$N^2 \cup N^3$
$l_1 = l_4$ and $l_2 = l_5$	$b_{i_1}^1 \wedge (b_{i_2}^2 \wedge b_3)$	N^3	$\star : (b_{i_1}^1 \wedge b_{i_2}^2) \wedge b_3$	N^3
$l_1 = l_4$ and $l_3 = l_6$	$b_{i_1}^1 \wedge (b_2 \wedge b_{i_3}^3)$	N^2	$(b_{i_1}^1 \wedge b_2) \wedge b_{i_3}^3$	N^2
$l_2 = l_5$ and $l_3 = l_6$	$\star : b_1 \wedge (b_{i_2}^2 \wedge b_{i_3}^3)$	N^1	$(b_1 \wedge b_{i_2}^2) \wedge b_{i_3}^3$	N^1

A \star signifies that the entry follows by the above discussion. Any other entry follows by construction.

$\mathcal{F}^F = \mathcal{F}^G$ Follows trivially because addition is associative.

4 Conclusion and Future Work

Initially we showed some issues of the original LTIHA and the composition operator definition. These issues were then solved by changing the respective definitions.

By these changes the expressiveness of LTIHA has been expanded. Beyond that the new composition operator was proven to be commutative and associative, which are both necessary conditions for the operator to behave as wanted. Neither of these properties were shown for the original version.

After these issues are solved future work will focus on the properties of the redefined composition operator.

It will be researched which properties the composition of two automata has if these automata were known to have a zeno execution. For this analysis first the zeno definition of [8] will be adjusted to fit the redefinition of LTIHA.

One step further is the analysis of reachability. It will be analyzed if reachability is decidable for LTIHA and if reachability results of small automata can be used to analyze the composition of these automata.

References

1. Akhundov, Jafar and Tröger, Peter and Werner, Matthias: Superposition Principle in Composable Hybrid Automata. *Fundamenta Informaticae* **157**(4), 321–339(2018)
2. Akhundov, Jafar and Reißner, Michael and Werner, Matthias: Using Hybrid Automata for Early Spacecraft Design Evaluation. *Proceedings of the 26th International Workshop on Concurrency, Specification and Programming*. Warsaw, Poland(2017)

A New Definition of Composition of LTIHA

3. Doyen, Laurent and Frehse, Goran and Pappas, George J. and Platzer, André: Verification of hybrid systems. Handbook of Model Checking, 1047–1110(2018)
4. André, Étienne: What's decidable about parametric timed automata? International Journal on Software Tools for Technology Transfer **21**(2) 203–219(2019)
5. Harary, Frank and Wilcox, Gordon W.: Boolean operations on graphs. *Mathematica Scandinavica* 41–51(1967)
6. Roehm, Hendrik and Oehlerking, Jens and Woehrle, Matthias and Althoff, Matthias: Reachset conformance testing of hybrid automata. Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control 277–286(2016)
7. Van der Schaft, A.J. and Schumacher, Johannes Maria: Compositionality issues in discrete, continuous, and hybrid systems. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* **11**(5) 417–434(2001)
8. Lygeros, John and Tomlin, Claire and Sastry, Shankar: Hybrid systems: modeling, analysis and control. (1999)
9. Akhundov, Jafar and Reißner, Michael and Werner, Matthias: Compositional Expressiveness of Hybrid Models. CS&P on Proceedings, (2018)