Multi-View Consistency for Infinitary Regular Languages

Maria Pittou
Aalto University

Stavros Tripakis
Aalto University and UC Berkeley

***Abstract*—Multi-view modeling is a system design methodology where different facets of a system are modeled each with a separate model, called *view*. The problem of view consistency then arises, namely, does there exist a system which could generate a given set of views? In previous work this problem has been studied for the case of discrete systems such as finite automata over finite words, on one hand, and finite automata over infinite words, on the other hand. In this work we study the problem for the case of *mixed* automata, which accept both finite and infinite words, and the corresponding infinitary regular languages. This model is particularly useful in the multi-view modeling setting, where views are obtained as projections of the system, and where these projections may turn an infinite behavior into a finite one.**

1. Introduction

Modeling provides an abstract formal description of a system under development, and it has been proved essential in system design. Obtaining a model of a system efficiently allows analysis and detection of failures, that can be resolved prior to experimental work. The construction of complex systems, and hence their modeling, is usually delegated to more than one stakeholders, in order to capture the various aspects of the systems and reduce the involved complexity. In multi-view modeling in particular, the different stakeholders obtain several models, the so called views, of the same system [2], [14].

The views can be either behavioral or structural since they may refer to the behavior or structure of the system respectively, and they are usually expressed in different semantics in order to describe the various tasks of the system. As a result, the views model only partial perspectives of the same system, and therefore the aspects not considered by each view are ignored. However, possible overlaps among the views may give rise to inconsistencies. One of the main challenges in multi-view modeling is to ensure consistency among the different views [14]. Other problems related to multi-view modeling include the construction (or *synthesis* [14]) of a complete system given its views, and also view traceability and reuse [9]. For general discussions on multi-view modeling the reader is referred to [2], [9], and for a formal treatment to [13], [14].

In this work we are interested in solving the consistency problem for the behavioral views of a system under construction. The behavior of a system can be defined in general within any global universe while the views may be obtained by some kind of transformation, like abstraction functions, to the system's behavior. [14] defines a generic multi-view modeling framework and instantiates that framework for the case of symbolic discrete systems. In [13], the behavior of a given system and of its views is described by languages in general, regular languages or ω-regular languages. Projections are used as abstraction functions to obtain the views from the system, and inverse projections for the other direction. Even though the views of a system may be defined only by regular or ω-regular languages, the application of projections and inverse projections may result in an ∞-regular language (*infinitary* regular language, i.e., a language containing both finite and infinite words) for describing the system's behavior. Moreover, it may be the case that the different views of the system can be described only by an ∞-regular language. Hence, in this paper we consider the case of describing the behavior of a system and its views by ∞-regular languages or equivalently by the relevant automata accepting such languages, which we also study in this paper.

In summary, the main contributions of this paper are three: first, we propose a model of (nondeterministic) *mixed* automata accepting infinitary languages; second, we show that mixed automata are closed under the usual set-theoretic operations (union, intersection, complementation), as well as under projection and inverse projection; third, we use these results to solve the multi-view consistency problem in the infinitary language setting.

This work has been partially supported by the Academy of Finland and the National Science Foundation (awards #1329759 and #1139138).

2. Related work

Multi-view modeling is a well known problem, related to the construction of complex systems. Existing literature mainly refers to *structural* views of a system, e.g., see [1], [4]. However, our interest is in describing a system and its views as sets of *behaviors*. Behavioral views are investigated for instance in [11], [12] within the context of cyber physical systems. In particular, the authors address the problem of heterogeneity in such complex systems in order to aid their verification, and consider behavior relations for the different semantic models used to decribe the different aspects of the same system.

Our work follows the framework proposed in [14] where the behavioral view consistency problem is defined formally. [14] provides a generic framework where both the system and views can be defined within any global universe and can be related by any kind of abstraction functions. [14] also includes a notion of conformance to capture faithfulness of a view w.r.t. a system. The problems of view consistency, orthogonality and reduction are also defined. The aforementioned can then be instantiated for versatile formalisms of a system, its views and abstraction functions. In [14] the framework is instatiated for symbolic discrete systems, while recently it has also been extended to languages and automata [13], but not infinitary languages, which is the focus of this paper.

Automata accepting a mix of finite and infinite words have been studied earlier. [10] studies *weighted* such automata for the fuzzy semiring (weighted automata form the quantitative extension of the classical finite state automata). In [3] and [7] one can find futher works for the model of [10]. Moreover, in [15] there is the definition of an (unweighted) finite state automaton that accepts both finite and infinite words, but no further results for this class of automata are presented. In this paper, we consider a definition different, albeit semantically equivalent to the one of [15], and we provide a systematic study for this class which is needed for our multi-view framework.

3. Background

3.1. Languages over finite and infinite words

Alphabet, Finite words, Infinite words: A finite alphabet Σ is a non-empty finite set of symbols. Σ^* is the set of all finite words over Σ and Σ^ω is the set of all infinite words over Σ. The set of all words over Σ is Σ^∞, i.e., $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$.

Languages: A $*$-language (star language) L on Σ is a set of finite words, subset of Σ^*, i.e., $L \subseteq \Sigma^*$. An ω-language (omega language) L on Σ is a set of infinite words, subset of Σ^ω, i.e., $L \subseteq \Sigma^\omega$. An ∞-language (infinitary language) L is a set of finite or infinite words, subset of Σ^∞, i.e., $L \subseteq \Sigma^\infty$.

3.2. Automata over finite and infinite words

Nondeterministic finite automaton: Let $A = (Q_A, \Sigma, I_A, \Delta_A, F_A)$ be a nondeterministic finite state automaton (NFA for short) where Q_A is a finite set of states, Σ is a finite alphabet, $I_A \subseteq Q_A$ is the set of initial states, $\Delta_A \subseteq Q_A \times \Sigma \times Q_A$ is a transition function, and $F_A \subseteq Q_A$ is the set of final states. A path P_w^A of A over a finite word $w = w_0 \ldots w_{n-1} \in \Sigma^*$ is a finite sequence $P_w^A\colon (q_{0A}, w_0, q_{1A}) \ldots (q_{n-1A}, w_{n-1}, q_{nA})$ such that $q_{0A} \in I_A$ is the initial state and $(q_{iA}, w_i, q_{i+1A}) \in \Delta_A$ for every $0 \le i < n$. A path P_w^A of A over a finite word $w \in \Sigma^*$ is called accepting if additionally $q_{nA} \in F_A$. A finite word $w \in \Sigma^*$ is accepted by A if there is an accepting path P_w^A of A over w. The language accepted by A, also called behavior of A, written $L(A)$, is the set of finite words accepted by A: $L(A) = \{w \in \Sigma^* \mid \exists \text{ accepting path } P_w^A \text{ of } A \text{ over } w\}$. A language L is called regular if there exists a nondeterministic finite automaton A over Σ accepting L, i.e., such that $L = L(A)$.

Example 1. *Consider the NFA A shown in Figure 1, $A = (\{q_{0A}, q_{1A}\}, \{a, b\}, \{q_{0A}\}, \Delta_A, \{q_{1A}\})$, with $\Delta_A = \{(q_{0A}, a, q_{0A}), (q_{0A}, a, q_{1A}), (q_{1A}, b, q_{1A})\}$. The language accepted by A is $L(A) = a^+b^*$, where a^+ denotes a non-empty finite sequence of the letter a and b^* denotes a (possibly empty) finite sequence of the letter b.*

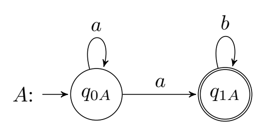

Figure 1: NFA example.

Union of NFA: Let $A^i = (Q_A^i, \Sigma, I_A^i, \Delta_A^i, F_A^i)$, for $i = 1, 2$, be two NFA. The union of A^1 and A^2 is the NFA $A^1 \cup A^2 = (Q_A, \Sigma, I_A, \Delta_A, F_A)$, with $Q_A = Q_A^1 \cup Q_A^2$, $I_A = I_A^1 \cup I_A^2$, $F_A = F_A^1 \cup F_A^2$ and the transition function $\Delta_A \subseteq Q_A \times \Sigma \times Q_A$ defined by $\Delta_A = \{(q_A, \sigma, q'_A) \mid (q_A, \sigma, q'_A) \in \Delta_A^1 \text{ or } (q_A, \sigma, q'_A) \in \Delta_A^2\}$.

Intersection of NFA: The intersection of A^1 and A^2 is the NFA $A^1 \times A^2 = (Q_A, \Sigma, I_A, \Delta_A, F_A)$, with $Q_A = Q_A^1 \times Q_A^2$, $I_A = I_A^1 \times I_A^2$, $F_A = F_A^1 \times F_A^2$ and the transition function $\Delta_A \subseteq Q_A \times \Sigma \times$

Q_A defined by $\Delta_A = \{((q_A^1, q_A^2), \sigma, (q_A'^1, q_A'^2)) \mid (q_A^1, \sigma, q_A'^1) \in \Delta_A^1$ and $(q_A^2, \sigma, q_A'^2) \in \Delta_A^2\}$.

It is well known that $L(A^1 \cup A^2) = L(A^1) \cup L(A^2)$ and $L(A^1 \times A^2) = L(A^1) \cap L(A^2)$ [6].

Closure properties of regular languages: The class of regular languages is closed under union, intersection, and complementation [6], as well as under projections [13], which will be presented below.

Nondeterministic Büchi automaton: Let $B = (Q_B, \Sigma, I_B, \Delta_B, C_B)$ be a nondeterministic Büchi automaton (NBA for short) where Q_B is a finite set of states, Σ is a finite alphabet, $I_B \subseteq Q_B$ is the set of initial states, $\Delta_B \subseteq Q_B \times \Sigma \times Q_B$ is a transition function, and $C_B \subseteq Q_B$ is the set of final states. A path P_w^B of B over an infinite word $w = w_0 w_1 \ldots \in \Sigma^\omega$ is an infinite sequence $P_w^B \colon (q_{0B}, w_0, q_{1B})(q_{1B}, w_1, q_{2B}) \ldots$ such that $q_{0B} \in I_B$ is the initial state and $(q_{iB}, w_i, q_{i+1B}) \in \Delta_B$ for every $i \geq 0$. For every path P_w^B of B over an infinite word $w \in \Sigma^\omega$ we denote with $Inf\left(P_w^B\right)$ the set of states occurring an infinite number of times along P_w^B. Then, a path P_w^B of B over $w \in \Sigma^\omega$ is called accepting if additionally $Inf(P_w^B) \cap C_B \neq \emptyset$. An infinite word $w \in \Sigma^\omega$ is accepted by B if there is an accepting path P_w^B of B over w. The language accepted by B, also called behavior of B, written $L(B)$, is the set of infinite words accepted by B: $L(B) = \{w \in \Sigma^\omega \mid \exists$ infinite accepting path P_w^B of B over $w\}$. A language L is called ω-regular if there exists a nondeterministic Büchi automaton B over Σ accepting L, i.e., $L = L(B)$.

Example 2. *Consider the NBA of Figure 2, $B = (\{q_{0B}, q_{1B}\}, \{a, b\}, \{q_{0B}\}, \Delta_B, \{q_{1B}\})$ with $\Delta_B = \{(q_{0B}, b, q_{0B}), (q_{0B}, b, q_{1B}), (q_{1B}, a, q_{0B})\}$. Then the language accepted by B is $L(B) = (b^+ab^+)^\omega$.*

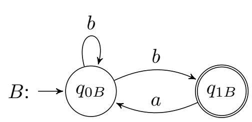

Figure 2: NBA example.

Union of NBA: Consider two NBA $B^i = (Q_B^i, \Sigma, I_B^i, \Delta_B^i, C_B^i)$ for $i = 1, 2$. The union of B^1 and B^2 is a NBA $B^1 \cup B^2 = (Q_B, \Sigma, I_B, \Delta_B, C_B)$, with $Q_B = Q_B^1 \cup Q_B^2$, $I_B = I_B^1 \cup I_B^2$, $C_B = C_B^1 \cup C_B^2$ and $\Delta_B \subseteq Q_B \times \Sigma \times Q_B$ defined by $\Delta_B = \{(q_B, \sigma, q'_B) \mid (q_B, \sigma, q'_B) \in \Delta_B^1$ or $(q_B, \sigma, q'_B) \in \Delta_B^2\}$.

Intersection of NBA: The intersection of B^1 and B^2 is the NBA $B^1 \times B^2 = (Q_B, \Sigma, I_B, \Delta_B, C_B)$, with $Q_B = Q_B^1 \times Q_B^2 \times \{1, 2\}$, $I_B = I_B^1 \times I_B^2 \times \{1\}$, $C_B = Q_B^1 \times C_B^2 \times \{2\}$ and $\Delta_B \subseteq Q_B \times \Sigma \times Q_B$ is defined by $\Delta_B = \{((q_B^1, q_B^2, 1), \sigma, (q_B'^1, q_B'^2, j)) \mid (q_B^1, \sigma, q_B'^1) \in \Delta_B^1$ and $(q_B^2, \sigma, q_B'^2) \in \Delta_B^2$ and if $q_B^1 \in C_B^1$, $j = 2$ else $j = 1\} \cup \{((q_B^1, q_B^2, 2), \sigma, (q_B'^1, q_B'^2, j)) \mid (q_B^1, \sigma, q_B'^1) \in \Delta_B^1$ and $(q_B^2, \sigma, q_B'^2) \in \Delta_B^2$ and if $q_B^2 \in C_B^2$ $j = 1$ else $j = 2\}$.

Then, it can be proved that $L(B^1 \cup B^2) = L(B^1) \cup L(B^2)$ and $L(B^1 \times B^2) = L(B^1) \cap L(B^2)$.

Closure properties of ω-regular languages: The class of ω-regular languages is closed under union, intersection, complementation [8], infinite projection, and inverse projection [13].

3.3. Projections and inverse projections

Projection of words and languages: Consider two alphabets Σ and Σ' such that $\Sigma' \subseteq \Sigma$. The projection of a word $w \in \Sigma^\infty$ onto the subalphabet Σ', is performed by the function $\Pi_{\Sigma \to \Sigma'} : \Sigma^\infty \to \Sigma'^\infty$ defined as follows (where $\cdot$ denotes word concatenation):

$$\Pi_{\Sigma \to \Sigma'}(w) = \begin{cases} \epsilon & \text{if } w = \epsilon \\ \sigma \cdot \Pi_{\Sigma \to \Sigma'}(u) & \text{if } w = \sigma \cdot u \text{ and } \sigma \in \Sigma' \\ \Pi_{\Sigma \to \Sigma'}(u) & \text{if } w = \sigma \cdot u \text{ and } \sigma \notin \Sigma' \end{cases}$$

The projection of a finite word is always a finite word while the projection of an infinite word may be either a finite or infinite word. For example, if $\Sigma = \{a, b\}$ and $\Sigma' = \{b\}$, then for $w^1 = abab \in \Sigma^*$ we have $\Pi_{\Sigma \to \Sigma'}(w^1) = bb \in \Sigma'^*$ while for $w^2 = ba^\omega \in \Sigma^\omega$ we have $\Pi_{\Sigma \to \Sigma'}(w^2) = b \in \Sigma'^*$.

The projection of a language $L \subseteq \Sigma^\infty$ is defined as $\Pi_{\Sigma \to \Sigma'}(L) = \{\Pi_{\Sigma \to \Sigma'}(w) \mid w \in L\}$. The projection of a $*$-language is always a $*$-language, while the projection of an ω-language is generally an ∞-language. For example, if $\Sigma = \{b, c\}$ and $\Sigma' = \{b\}$, then for $L^1 = cb^*c \subseteq \Sigma^*$, we obtain that $\Pi_{\Sigma \to \Sigma'}(L^1) = b^* \subseteq \Sigma^*$, while for $L^2 = b^*c^\omega \cup c^*b^\omega \subseteq \Sigma^\omega$, we obtain that $\Pi_{\Sigma \to \Sigma'}(L^2) = b^* \cup b^\omega \subseteq \Sigma^\infty$.

Inverse projection of words and languages: Consider two alphabets Σ and Σ' such that $\Sigma' \supseteq \Sigma$. We define the inverse projection of $w \in \Sigma^\infty$ onto Σ' as the set $\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(w) = \{u \text{ over } \Sigma' \mid \Pi_{\Sigma' \to \Sigma}(u) = w\}$. The inverse projection of a finite word can be either a finite or infinite word, while the inverse projection of an infinite word is always an infinite word. For example, if $\Sigma = \{b\}$ and $\Sigma' = \{a, b\}$, then for $w^1 = b \in \Sigma^*$ we obtain that $\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(w^1) = a^*b(a^* \cup a^\omega) \subseteq \Sigma^\infty$, and for $w^2 = b^\omega \in \Sigma^\omega$ we obtain that $\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(w^2) = a^*(a^*ba^*)^\omega \subseteq \Sigma^\omega$.

Moreover, the inverse projection of a language $L \subseteq \Sigma^\infty$ is defined as $\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(L) = \{w$ over $\Sigma' \mid \Pi_{\Sigma' \to \Sigma}(w) \in L\}$. The inverse projection of an ω-language is always an ω-language, while the inverse projection of a $*$-language is generally an ∞-language. For instance, if $\Sigma = \{b\}$ and $\Sigma' = \{b, c\}$, then for $L^1 = b^* \subseteq \Sigma^*$, we obtain that $\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(L^1) = c^* b^* (c^* \cup c^\omega) \subseteq \Sigma^\infty$, while for $L^2 = (bb)^\omega \subseteq \Sigma^\omega$, we obtain that $\Pi_{\Sigma' \leftarrow \Sigma}(L^2) = c^*(c^* bc^* bc^*)^\omega \subseteq \Sigma^\omega$.

More examples of projections and inverse projections of languages are given below where we discuss how these operations are implemented on automata.

Lemma 1. *1. The projection of the inverse projection of a language yields the original language, i.e.,: for every language $L \subseteq \Sigma^\infty$ and $\Sigma \subseteq \Sigma'$ it holds that $\Pi_{\Sigma' \to \Sigma}(\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(L)) = L$.*

2. The inverse projection of the projection of a language generally yields a superset of the original language: for every language $L \subseteq \Sigma^\infty$ and $\Sigma' \subseteq \Sigma$ it holds that $\Pi^{-1}_{\Sigma \leftarrow \Sigma'}(\Pi_{\Sigma \to \Sigma'}(L)) \supseteq L$.

Next, we show how the projection and inverse projection operations can be implemented on NFA and NBA. We omit the details and correctness proofs of these constructions, which can be found on [13], and we rather illustrate the results with some examples.

Projection of NFA: Consider two alphabets Σ and Σ' such that $\Sigma' \subseteq \Sigma$, and a NFA $A = (Q_A, \Sigma, I_A, \Delta_A, F_A)$ over Σ. The projection of A on Σ' denoted by $\Pi_{\Sigma \to \Sigma'}(A)$, is a NFA $(Q_A, \Sigma', I_A, \Delta'_A, F_A)$ with $\Delta'_A = \{(q_A, \sigma, q'_A) \mid \sigma \in \Sigma'$ and $(q_A, \sigma, q'_A) \in \Delta_A\} \cup \{(q_A, \epsilon, q'_A) \mid \sigma \in \Sigma \setminus \Sigma'$ and $(q_A, \sigma, q'_A) \in \Delta_A\}$. An example of NFA projection is shown in Figure 3.

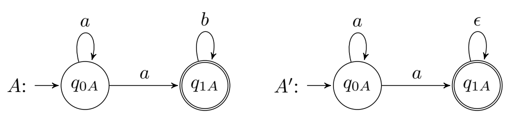

Figure 3: Example of NFA projection. NFA A over $\Sigma = \{a, b\}$ and NFA $A' = \Pi_{\Sigma \to \{a\}}(A)$.

Projection of NBA: Let $\Sigma' \subseteq \Sigma$ and consider a NBA $B = (Q_B, \Sigma, I_B, \Delta_B, C_B)$ over Σ. Since the projection of a ω-language can contain both finite and infinite words, two kinds of projections are obtained, the NBA finite and infinite projection. The NBA finite projection of B on Σ', denoted by $\Pi^*_{\Sigma \to \Sigma'}(B)$, is a NFA $(Q_B, \Sigma', I_B, \bar{\Delta}_B, \bar{F}_B)$ where $\bar{\Delta}_B = \{(q_B, \sigma, q'_B) \mid \sigma \in \Sigma'$ and $(q_B, \sigma, q'_B) \in \Delta_B\} \cup \{(q_B, \epsilon, q'_B) \mid \sigma \in \Sigma \setminus \Sigma'$ and $(q_B, \sigma, q'_B) \in \Delta_B\}$, and $\bar{F}_B = \{q_B \mid q_B \in C_B$ and $\exists w \in (\Sigma \setminus \Sigma')^+ : (q_B, \sigma, q_B) \in \Delta^*_B\}$, where Δ^*_B denotes the reflexive and transitive closure of Δ_B. The NBA infinite projection of B on Σ', denoted by $\Pi^\omega_{\Sigma \to \Sigma'}(B)$, is a NBA $(Q_B, \Sigma', I_B, \Delta'_B, C_B)$ with $\Delta'_B = \{(q_B, \sigma, q'_B) \mid \sigma \in \Sigma'$ and $\exists w \in \Sigma^* : \Pi_{\Sigma \to \Sigma'}(w) = \sigma$ and $(q_B, w, q'_B) \in \Delta^*_B\}$. An example of NBA projection is shown in Figure 4.

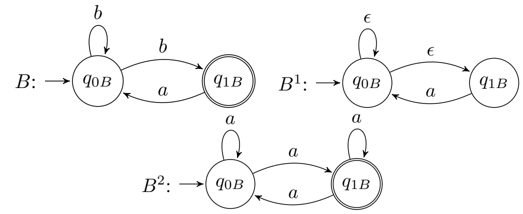

Figure 4: Example of NBA projection. NBA B over $\Sigma = \{a, b\}$, NFA $B^1 = \Pi^*_{\Sigma \to \{a\}}(B)$, and NBA $B^2 = \Pi^\omega_{\Sigma \to \{a\}}(B)$.

NFA inverse projection: Let $\Sigma' \supseteq \Sigma$ and consider a NFA $A = (Q_A, \Sigma, I_A, \Delta_A, F_A)$ over Σ. Since the inverse projection of a $*$-language can contain both finite and infinite words, we consider the relevant $*$-part and ω-part of the inverse projection of A. In particular, the former is denoted by $\Pi^{-1,*}_{\Sigma' \leftarrow \Sigma}(A)$ and is a NFA $(Q_A, \Sigma', \bar{\Delta}_A, I_A, F_A)$, where $\bar{\Delta}_A = \{(q_A, \sigma, q'_A) \mid \sigma \in \Sigma$ and $(q_A, \sigma, q'_A) \in \Delta_A\} \cup \{(q_A, \sigma, q_A) \mid q_A \in Q_A$ and $\sigma \in \Sigma' \setminus \Sigma\}$. The infinite, ω-part of the inverse projection is denoted by $\Pi^{-1,\omega}_{\Sigma' \leftarrow \Sigma}(A)$ and is a NBA $(Q_A \cup \{q_{\omega A}\}, \Sigma', I_A, \Delta'_A, \{q_{\omega A}\})$ where $\Delta'_A = \{(q_A, \sigma, q'_A) \mid \sigma \in \Sigma$ and $(q_A, \sigma, q'_A) \in \Delta_A\} \cup \{(q_A, \sigma, q_A) \mid q_A \in Q_A$ and $\sigma \in \Sigma' \setminus \Sigma\} \cup \{(q_A, \sigma, q_{\omega A}) \mid q_A \in F_A$ and $\sigma \in \Sigma' \setminus \Sigma\} \cup \{(q_{\omega A}, \sigma, q_{\omega A}) \mid \sigma \in \Sigma' \setminus \Sigma\}$. An example of NFA inverse projection is shown in Figure 5.

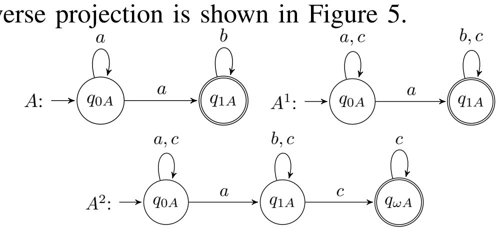

Figure 5: Example of NFA inverse projection. NFA A over $\Sigma = \{a, b\}$. NFA $A^1 = \Pi^{-1,*}_{\{a,b,c\} \leftarrow \Sigma}(A)$, and NBA $A^2 = \Pi^{-1,\omega}_{\{a,b,c\} \leftarrow \Sigma}(A)$.

NBA inverse projection: Let $\Sigma' \supseteq \Sigma$ and consider a NBA $A = (Q_B, \Sigma, I_B, \Delta_B, C_B)$ over Σ. The inverse projection of B on Σ' denoted by $\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(B)$, is a NBA $(Q'_B, \Sigma', I_B, \Delta'_B, C_B)$ where $Q'_B = Q_B \cup \{\hat{q_B} \mid q_B \in C_B\}$ and $\Delta'_B = \Delta_B \cup \{(\hat{q_B}, \sigma, q'_B) \mid q_B \in C_B$ and $(q_B, \sigma, q'_B) \in \Delta_B\} \cup \{(q_B, \sigma, \hat{q_B}) \mid q_B \in C_B$ and $\sigma \in \Sigma' \setminus \Sigma\} \cup \{(\hat{q_B}, \sigma, \hat{q_B}) \mid q_B \in C_B$ and $\sigma \in \Sigma' \setminus \Sigma\} \cup \{(q_B, \sigma, q_B) \mid q_B \in Q_B \setminus C_B$ and $\sigma \in \Sigma' \setminus \Sigma\}$. An example of NBA inverse projection is shown in Figure 6.

3.4. Multi-view modeling

For our multi-view modeling framework we consider a system and its views as sets of behaviors [14].

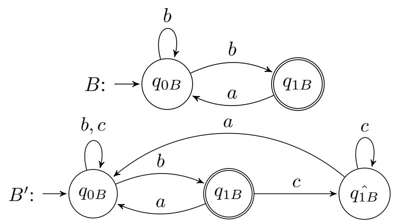

Figure 6: Example of NBA inverse projection. NBA B over $\Sigma = \{a, b\}$ and NBA $B' = \Pi^{-1}_{\{a,b,c\}\leftarrow\Sigma}(B)$.

Formally, *a system $\mathcal{S}$ over a domain of behaviors $\mathcal{U}$, is a subset of $\mathcal{U}$: $\mathcal{S} \subseteq \mathcal{U}$*. A view is intuitively an incomplete picture of a system, and may be obtained by some kind of transformation of the system behaviors into (incomplete) behaviors in another domain. Following [14], such a transformation is defined by means of an *abstraction function* $a : \mathcal{U} \to \mathcal{D}$, where $\mathcal{D}$ is the *view domain*. A *view $\mathcal{V}$ over view domain $\mathcal{D}$, is a subset of $\mathcal{D}$: $\mathcal{V} \subseteq \mathcal{D}$.*

Let us now state the multi-view consistency problem as defined in [14]. A set of views $\mathcal{V}_1, \ldots, \mathcal{V}_n$ over view domains $\mathcal{D}_1, \ldots, \mathcal{D}_n$ respectively, are *consistent with respect to a set of abstraction functions* $a_1, ..., a_n$, if there exists a system $\mathcal{S}$ over $\mathcal{U}$ so that $\mathcal{V}_i = a_i(\mathcal{S})$, for all $i = 1, ..., n$. We call such a system $\mathcal{S}$ a *witness system* to the consistency of $\mathcal{V}_1, \ldots, \mathcal{V}_n$. Obviously, if there is no such system, then we conclude that the views are inconsistent.

We consider the setting where a system and its views are described by languages over finite and infinite words. Specifically, two languages L^1, L^2 (describing two views) on alphabets Σ^1, Σ^2, respectively, are consistent if there exists language L on some alphabet Σ such that $\Pi_{\Sigma\to\Sigma^i}(L) = L^i$, for $i = 1, 2$. It is not necessary that the alphabet Σ is known, however, it should be a superset of $\Sigma^1 \cup \Sigma^2$. It is in fact proved in [13] that it suffices to consider only $\Sigma = \Sigma^1 \cup \Sigma^2$, and to check whether $L^\sharp = \Pi^{-1}_{\Sigma^1\cup\Sigma^2\leftarrow\Sigma^1}(L^1) \cap \Pi^{-1}_{\Sigma^1\cup\Sigma^2\leftarrow\Sigma^2}(L^2)$ is a valid witness:

Theorem 1. *[13] Let L^1 and L^2 be two languages on alphabets Σ^1 and Σ^2 respectively, and let $L^\sharp = \Pi^{-1}_{\Sigma^1\cup\Sigma^2\leftarrow\Sigma^1}(L^1) \cap \Pi^{-1}_{\Sigma^1\cup\Sigma^2\leftarrow\Sigma^2}(L^2)$. Then:*

1) *L^1 and L^2 are consistent if and only if there exists language L on alphabet $\Sigma^1 \cup \Sigma^2$, such that $\Pi_{\Sigma^1\cup\Sigma^2\to\Sigma^i}(L) = L^i$, for $i = 1, 2$.*
2) *Let L be a language on $\Sigma^1 \cup \Sigma^2$ that is a witness to the consistency of L^1 and L^2. Then: (a) If both L^1 and L^2 are $*$-languages, then L is also a $*$-language, i.e., $L \subseteq \Sigma^*$. (b) If both L^1 and L^2 are ω-languages, then L is also a ω-language, i.e., $L \subseteq \Sigma^\omega$.*
3) *L^1 and L^2 are consistent if and only if $\Pi_{\Sigma^1\cup\Sigma^2\to\Sigma^i}(L^\sharp) \supseteq L^i$, for $i = 1, 2$.*
4) *If L^1 and L^2 are consistent then $L^\sharp$ is a witness to their consistency, i.e., $\Pi_{\Sigma^1\cup\Sigma^2\to\Sigma^i}(L^\sharp) = L^i$, for $i = 1, 2$. Moreover, $L^\sharp$ is the greatest such witness, that is, any other witness L is such that $L \subseteq L^\sharp$.*

4. Multi-view consistency for ∞-regular languages

In this paper, we define the behavior of a system and its views by ∞-regular languages, that is, by ∞-languages accepted by a type of finite automata that we call *mixed* automata, and which will be defined below. We consider projections as abstraction functions, and we propose algorithmic solutions to the view consistency problem in this setting.

4.1. Mixed automata and ∞-regular languages

In order to solve the consistency problem for ∞-languages, we need the notion of automata that can accept an infinitary language, i.e., a language that may contain both finite and infinite words. As we have already mentioned in Section 2, such automata have been considered in the literature, but a systematic study has been lacking. [15] defines an automaton with two accepting conditions, so that it can accept both finite and infinite words. However, [15] provides no further results for this type of automata, apart from their definition. In this paper, we propose an even more intuitive definition of an automaton accepting an infinitary language, the so called *mixed* automaton, which is a *pair* of a finite automaton and an ω-automaton. We fix the ω-automaton to be a Büchi automaton, and we consider in particular nondeterministic mixed automata.

One should observe that the notion of a pair automaton is equivalent with the definition of a single automaton with two accepting sets [15]. Indeed, from the latter model one can obtain a pair automaton by making two distinct copies of the single automaton, and associating the appropriate accepting set of each element of the pair. Vice-versa, from a mixed automaton, i.e., from a pair automaton, one can construct a single automaton by taking the product of the two automata in the pair.

Nondeterministic Mixed Automaton: Let Σ denote a finite alphabet. A nondeterministic mixed automaton (NXA for short) over the alphabet Σ is defined as a pair $M = (A, B)$ where $A = (Q_A, \Sigma, I_A, \Delta_A, F_A)$ is a nondeterministic finite state automaton and $B = (Q_B, \Sigma, I_B, \Delta_B, C_B)$ is a nondeterministic Büchi automaton, with $Q_A \cap Q_B = \emptyset$. The language $L(M)$ of the mixed automaton M is defined by $L(M) = L(A) \cup L(B) = \{w \in \Sigma^* \mid w$ is accepted by $A\} \cup \{w \in \Sigma^\omega \mid w$ is accepted by $B\} = \{w \in \Sigma^\infty \mid w$ is accepted by A or $B\}$. It should be clear that this is disjoint union since A accepts only finite words over Σ and B accepts only infinite words over Σ. A language $L \subseteq \Sigma^\infty$ is called ∞-regular, if there exists a mixed automaton M such that $L(M) = L$. Moreover, every NXA M over Σ can be considered either as nondeterministic finite automaton whenever $C_B = \emptyset$, or as a nondeterministic Büchi automaton, whenever $F_A = \emptyset$.

Example 3. *Consider the NXA $M = (A, B)$ where $A = (\{q_{0A}, q_{1A}\}, \{a, b\}, \{q_{0A}\}, \Delta_A, \{q_{1A}\})$ with $\Delta_A = \{(q_{0A}, a, q_{0A}), (q_{0A}, a, q_{1A}), (q_{1A}, b, q_{1A})\}$, and $B = (\{q_{0B}, q_{1B}\}, \{a, b\}, \{q_{0B}\}, \Delta_B, \{q_{1B}\})$ with $\Delta_B = \{(q_{0B}, b, q_{0B}), (q_{0B}, b, q_{1B}), (q_{1B}, a, q_{0B})\}$, as shown in Figure 7. Then the language accepted by M is $L(M) = L(A) \cup L(B) = a^+b^* \cup (b^+ab^+)^\omega$.*

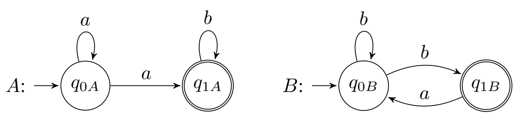

Figure 7: NXA example with NFA A and NBA B.

Throughout the paper, when we refer to mixed automata we mean nondeterministic mixed automata.

4.2. Closure properties of ∞-regular languages

We now show that the class of ∞-regular languages is closed under union, intersection, complementation, projection, and inverse projection.

Union of mixed automata: Consider two NXA $M^i = (A^i, B^i)$ where $A^i = (Q_A^i, \Sigma, I_A^i, \Delta_A^i, F_A^i)$ and $B^i = (Q_B^i, \Sigma, I_B^i, \Delta_B^i, C_B^i)$, for $i = 1, 2$ and $(Q_A^1 \cup Q_B^1) \cap (Q_A^2 \cup Q_B^2) = \emptyset$. The union of M^1 and M^2 is a NXA $M = (A, B)$ where: $A = A^1 \cup A^2 = (Q_A, \Sigma, I_A, \Delta_A, F_A)$ and $B = B^1 \cup B^2 = (Q_B, \Sigma, I_B, \Delta_B, C_B)$.

Proposition 1. *The class of ∞-regular languages is closed under union.*

Proof. One can observe that $L(M) = L(A) \cup L(B) = (L(A^1) \cup L(A^{2)})) \cup (L(B^1) \cup L(B^2)) = (L(A^1) \cup L(B^1)) \cup (L(A^2) \cup L(B^2)) = L(M^1) \cup L(M^2)$, which completes our proof. □

Intersection of mixed automata: Consider two NXA $M^i = (A^i, B^i)$ where $A^i = (Q_A^i, \Sigma, I_A^i, \Delta_A^i, F_A^i)$ and $B^i = (Q_B^i, \Sigma, I_B^i, \Delta_B^i, C_B^i)$, for $i = 1, 2$ and $(Q_A^1 \cup Q_B^1) \cap (Q_A^2 \cup Q_B^2) = \emptyset$. The intersection of M^1 and M^2 is a NXA $M = (A, B)$ where: $A = A^1 \times A^2 = (Q_A, \Sigma, I_A, \Delta_A, F_A)$ and $B = B^1 \times B^2 = (Q_B, \Sigma, I_B, \Delta_B, C_B)$.

Proposition 2. *The class of ∞-regular languages is closed under intersection.*

Proof. We observe that $L(M^1) \cap L(M^2) = (L(A^1) \cap L(A^2)) \cup (L(A^1) \cap L(B^2)) \cup (L(B^1) \cap L(A^2)) \cup (L(B^1) \cap L(B^2)) = (L(A^1) \cap L(A^2)) \cup (L(B^1) \cap L(B^2))$. Moreover, by construction of B, we have that $L(M) = (L(A^1) \cap L(A^2)) \cup (L(B^1) \cap L(B^2)) = L(M^1) \cap L(M^2)$, which completes our proof. □

Complementation of mixed automata: Consider a NXA $M = (A, B)$ where $A = (Q_A, \Sigma, I_A, \Delta_A, F_A)$ and $B = (Q_B, \Sigma, I_B, \Delta_B, C_B)$, and $Q_A \cap Q_B = \emptyset$. The complement of M is a NXA $M^c = (A^c, B^c)$, where A^c is the complement automaton of the NFA A and B^c is the complement automaton of the NBA B. (One can find the relevant constructions for A^c and B^c in [6] and [17] respectively).

Proposition 3. *The class of ∞-regular languages is closed under complementation.*

Proof. We have that $L(M^c) = L(A^c) \cup L(B^c) = (\Sigma^* \setminus L(A)) \cup (\Sigma^\omega \setminus L(B)) = \Sigma^\infty \setminus (L(A) \cup L(B)) = \Sigma^\infty \setminus L(M)$, and our proof is completed. □

Proposition 4. *The emptiness problem of an ∞-regular language is decidable.*

Proof. The proof can be obtained by decidability of the emptiness problem for both regular and ω-regular languages. □

Proposition 5. *The equality problem of two ∞-regular languages is decidable.*

Proof. The proof can be obtained by decidability of the equality problem for both regular and ω-regular languages. □

4.3. Projections and inverse projections on mixed automata

In the sequel, we consider the closure of ∞-regular languages over projections and inverse projections. For this, we need to obtain the constructions of these operations on NXA, and we do so by using the relevant constructions for NFA and NBA as defined in 3.3. It should be clear that since mixed automata accept ∞-regular languages, both the projection and inverse projection of an ∞-regular language is always an ∞-regular language.

Projection of NXA: Let Σ, Σ' denote two finite alphabets such that $\Sigma' \subseteq \Sigma$ and consider the NXA $M = (A, B)$ where $A = (Q_A, \Sigma, I_A, \Delta_A, F_A)$ and $B = (Q_B, \Sigma, I_B, \Delta_B, C_B)$, with $Q_A \cap Q_B = \emptyset$. The projection of M on Σ', denoted by $\Pi_{\Sigma \to \Sigma'}(M)$ is a NXA $M' = (A', B')$ over Σ', where $A' = A^1 \cup A^2$, $A^1 = \Pi_{\Sigma \to \Sigma'}(A)$ and $A^2 = \Pi^*_{\Sigma \to \Sigma'}(B)$, and $B' = \Pi^\omega_{\Sigma \to \Sigma'}(B)$. An example of NXA projection is shown in Figure 8.

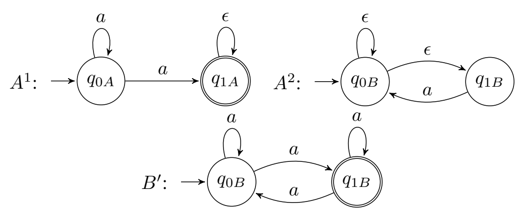

Figure 8: Example of NXA projection $\Pi_{\Sigma \to \{a\}}(M)$, where M is the NXA of the Example 3.

Proposition 6. *The class of ∞-regular languages is closed under projections.*

Proof. By [13], it holds that $L\left(\Pi_{\Sigma \to \Sigma'}(A)\right) = \Pi_{\Sigma \to \Sigma'}(L(A))$, $L\left(\Pi^*_{\Sigma \to \Sigma'}(B)\right) = \Pi_{\Sigma \to \Sigma'}(L(B)) \cap \Sigma'^*$, and $L\left(\Pi^\omega_{\Sigma \to \Sigma'}(B)\right) = \Pi_{\Sigma \to \Sigma'}(L(B)) \cap \Sigma'^\omega$. Then, $L(M') = L\left(\Pi_{\Sigma \to \Sigma'}(M)\right) = L(A') \cup L(B') = \left(L(A^1 \cup A^2)\right) \cup L(B') = L(A^1) \cup L(A^2) \cup L(B') = L\left(\Pi_{\Sigma \to \Sigma'}(A)\right) \cup L\left(\Pi^*_{\Sigma \to \Sigma'}(B)\right) \cup L\left(\Pi^\omega_{\Sigma \to \Sigma'}(B)\right) = \Pi_{\Sigma \to \Sigma'}(L(A)) \cup (\Pi_{\Sigma \to \Sigma'}(L(B)) \cap \Sigma'^*) \cup (\Pi_{\Sigma \to \Sigma'}(L(B)) \cap \Sigma'^\omega) = \Pi_{\Sigma \to \Sigma'}(L(M))$. □

NXA inverse projection: Let Σ, Σ' denote two finite alphabets such that $\Sigma' \supseteq \Sigma$ and consider the NXA $M = (A, B)$ where $A = (Q_A, \Sigma, I_A, \Delta_A, F_A)$ and $B = (Q_B, \Sigma, I_B, \Delta_B, C_B)$, with $Q_A \cap Q_B = \emptyset$. The inverse projection of M on Σ', denoted by $\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(M)$ is a NXA $M' = (A', B')$ over Σ' such that $A' = \Pi^{-1,*}_{\Sigma' \leftarrow \Sigma}(A)$, and $B' = B^1 \cup B^2$ where $B^1 = \Pi^{-1,\omega}_{\Sigma' \leftarrow \Sigma}(A)$ and $B^2 = \Pi^{-1}_{\Sigma' \leftarrow \Sigma}(B)$. An example of NXA inverse projection is shown in Figure 9.

Proposition 7. *The class of ∞-regular languages is closed under inverse projections.*

Proof. By [13], it holds that $L\left(\Pi^{-1,*}_{\Sigma' \leftarrow \Sigma}(A)\right) = \Pi^{-1}_{\Sigma' \leftarrow \Sigma}(L(A)) \cap \Sigma'^*$, $L\left(\Pi^{-1,\omega}_{\Sigma' \leftarrow \Sigma}(A)\right) = \Pi^{-1}_{\Sigma' \leftarrow \Sigma}(L(A)) \cap \Sigma'^\omega$, and $L\left(\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(B)\right) = \Pi^{-1}_{\Sigma' \leftarrow \Sigma}(L(B))$. Hence, $L(M') = L\left(\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(M)\right) = L(A') \cup L(B') = L(A') \cup \left(L(B^1 \cup B^2)\right) = L(A') \cup L(B^1) \cup L(B^2) = L\left(\Pi^{-1,*}_{\Sigma' \leftarrow \Sigma}(A)\right) \cup L\left(\Pi^{-1,\omega}_{\Sigma' \leftarrow \Sigma}(A)\right) \cup L\left(\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(B)\right) = \left(\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(L(A)) \cap \Sigma'^*\right) \cup \left(\Pi^{-1}_{\Sigma' \leftarrow \Sigma}(L(A)) \cap \Sigma'^\omega\right) \cup \Pi^{-1}_{\Sigma' \leftarrow \Sigma}(L(B)) = \Pi^{-1}_{\Sigma' \leftarrow \Sigma}(L(M))$. □

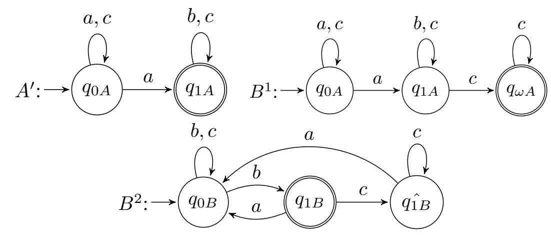

Figure 9: Example of NXA inverse projection $\Pi^{-1}_{\{a,b,c\} \leftarrow \Sigma}(M)$, where M is the NXA of the Example 3.

4.4. Solution to the multi-view consistency problem

Checking consistency for infinitary languages: We consider the setting discussed in 3.4 and we solve the view consistency problem, given that the views of a certain system are described by ∞-regular languages. We obtain two variants of the problem, and then we prove that they are equivalent:

Problem 1: Consider two NXA $M^i = \left(A^i, B^i\right)$ over the alphabet Σ^i for $i = 1, 2$ respectively. Check whether $L(M^1)$ and $L(M^2)$ are consistent, i.e., whether there exists an ∞-language L over the alphabet $\Sigma = \Sigma^1 \cup \Sigma^2$ such that $\Pi_{\Sigma^1 \cup \Sigma^2 \to \Sigma^1}(L) = L\left(M^1\right)$ and $\Pi_{\Sigma^1 \cup \Sigma^2 \to \Sigma^2}(L) = L\left(M^2\right)$.

Problem 2: Consider two NXA $M^i = \left(A^i, B^i\right)$ over the alphabet Σ^i for $i = 1, 2$ respectively. Check whether there exists a NXA M over the alphabet $\Sigma = \Sigma^1 \cup \Sigma^2$ such that $\Pi_{\Sigma^1 \cup \Sigma^2 \to \Sigma^1}(L(M)) = L\left(M^1\right)$ and $\Pi_{\Sigma^1 \cup \Sigma^2 \to \Sigma^2}(L(M)) = L\left(M^2\right)$.

Theorem 2. *There is a solution to Problem 1 if and only if there is a solution to Problem 2.*

Proof. The *if* part is trivial. For the *only if* part, we assume that $L\left(M^1\right)$ and $L\left(M^2\right)$ are consistent. Then by Theorem 1, the language $L^\sharp = \Pi^{-1}_{\Sigma^1 \cup \Sigma^2 \leftarrow \Sigma^1}(L(M^1)) \cap \Pi^{-1}_{\Sigma^1 \cup \Sigma^2 \leftarrow \Sigma^2}(L(M^2))$ is a witness to the consistency of $L\left(M^1\right)$ and $L\left(M^2\right)$. It suffices to obtain a NXA that accepts $L^\sharp$. It holds, by Proposition 7, that $\Pi^{-1}_{\Sigma^1 \cup \Sigma^2 \leftarrow \Sigma^i}(L(M^i)) = L(\Pi^{-1}_{\Sigma^1 \cup \Sigma^2 \leftarrow \Sigma^i}(M^i))$, for $i = 1, 2$. Moreover, by Proposition 2, we have that $L^\sharp = L(\Pi^{-1}_{\Sigma^1 \cup \Sigma^2 \leftarrow \Sigma^1}(M^1)) \cap L(\Pi^{-1}_{\Sigma^1 \cup \Sigma^2 \leftarrow \Sigma^2}(M^2)) = L(M^\sharp)$, where $M^\sharp = \Pi^{-1}_{\Sigma^1 \cup \Sigma^2 \leftarrow \Sigma^1}(M^1) \times \Pi^{-1}_{\Sigma^1 \cup \Sigma^2 \leftarrow \Sigma^2}(M^2)$, and the NXA $M^\sharp$ is a solution to Problem 2. □

Theorem 3. *Problems 1 and 2 are PSPACE-complete.*

Proof. Since Problems 1 and 2 are equivalent, it suffices to consider only Problem 1. By part 3 of Theorem 1, by Theorem 2, and by Proposition 6, $L(M^1)$ and $L(M^2)$ are consistent iff $L(\Pi_{\Sigma^1 \cup \Sigma^2 \to \Sigma^i}(M^\sharp)) \supseteq L(M^i)$ for $i = 1, 2$. $\Pi_{\Sigma^1 \cup \Sigma^2 \to \Sigma^i}(M^\sharp)$ can be computed in polynomial time, and for checking language containment of NXA, one has to check language containment of the NFA and NBA parts, which are both PSPACE-complete. Therefore, checking consistency of $L(M^1)$ and $L(M^2)$ is in PSPACE. Moreover, since NFA and NBA language equivalence are both PSPACE-hard [5], [16], we obtain that NXA language equivalence is PSPACE-hard.

For PSPACE hardness of Problem 1, we prove that NFA language equivalence, which is PSPACE-complete [5], is reducible to Problem 1. Let A^1, A^2 be two NFA over the same alphabet Σ. The NFA language equivalence problem is to check whether $L(A^1) = L(A^2)$. We let $\Sigma^1 = \Sigma^2 = \Sigma$ and define two NXA $M^1 = (A^1, B_\emptyset)$ and $M^2 = (A^2, B_\emptyset)$ by setting their NBA parts to be a NBA $B_\emptyset$ accepting the empty language. Then $L(M^i) = L(A^i)$, for $i = 1, 2$. We claim that $L(A^1) = L(A^2)$ iff $L(M^1)$ and $L(M^2)$ are consistent. Assume that $L(A^1) = L(A^2)$, i.e., $L(M^1) = L(M^2) = L$. Then, $\Pi_{\Sigma \cup \Sigma \to \Sigma}(L) = L(M^1)$ and $\Pi_{\Sigma \cup \Sigma \to \Sigma}(L) = L(M^2)$. Hence, $L(M^1)$ and $L(M^2)$ are consistent. Conversely, now assume that $L(M^1)$ and $L(M^2)$ are consistent. Then, by part 1 of Theorem 1 there is a language L over Σ, such that $L = \Pi_{\Sigma \cup \Sigma \to \Sigma}(L) = L(M^1)$ and $L = \Pi_{\Sigma \cup \Sigma \to \Sigma}(L) = L(M^2)$, which implies that $L(M^1) = L(M^2)$, thus $L(A^1) = L(A^2)$.

□

5. Conclusions and future work

One of the main challenges in multi-view modeling, where different models (views) are used to represent different facets of the same system, is to ensure that the views are consistent. In this work we solved the consistency problem for behavioral views defined by ∞-regular languages. As a special case, our solution also implies the solution of the consistency problem in the case where some of the views are regular languages while some others are ω-regular languages, an open problem from [13]. Indeed, both a regular and an ω-regular language is an ∞-regular language, and in the positive cases of the consistency problem the behavior of the witness system is also defined as an ∞-regular language, i.e., can be described by the behavior of a nondeterministic mixed automaton.

Possible directions for future work would be to develop the multi-view modeling for different frameworks and solve the consistency problem. Moreover, one can consider other abstraction functions apart from projections for obtaining the views of a system or in general other types of transformations.

References

[1] A. Bhave, B. H. Krogh, D. Garlan, and B. R. Schmerl. View consistency in architectures for cyber-physical systems. In *ICCPS*, pages 151–160, 2011.

[2] D. Broman, E. Lee, S. Tripakis, and M. Törngren. Viewpoints, Formalisms, Languages, and Tools for Cyber-Physical Systems. In *6th International Workshop on Multi-Paradigm Modeling (MPM'12)*, 2012.

[3] M. Droste and D. Kuske. Skew and infinitary formal power series. *Theor. Comput. Sci.*, 366(3):199–227, 2006.

[4] A. Finkelstein, D. M. Gabbay, A. Hunter, J. Kramer, and B. Nuseibeh. Inconsistency handling in multperspective specifications. *IEEE Trans. Soft. Eng.*, 20(8):569–578, 1994.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman, 1979.

[6] J. E. Hopcroft and J. D. Ullman. *Introduction To Automata Theory, Languages, And Computation*. 1990.

[7] W. Kuich and G. Rahonis. Fuzzy regular languages over finite and infinite words. *Fuzzy Sets and Systems*, 157(11):1532–1549, 2006.

[8] D. Perrin and J.-E. Pin. *Infinite words : automata, semigroups, logic and games*. Elsevier, 2004.

[9] M. Persson, M. Törngren, A. Qamar, J. Westman, M. Biehl, S. Tripakis, H. Vangheluwe, and J. Denil. A characterization of integrated multi-view modeling in the context of embedded and cyber-physical systems. In *Embedded Software, EMSOFT*, 2013.

[10] G. Rahonis. Infinite fuzzy computations. *Fuzzy Sets and Systems*, 153(2):275–288, 2005.

[11] A. Rajhans and B. H. Krogh. Heterogeneous verification of cyber-physical systems using behavior relations. In *HSCC*, pages 35–44, 2012.

[12] A. Rajhans and B. H. Krogh. Compositional heterogeneous abstraction. In *Hybrid Systems: Computation and Control, HSCC'13*, pages 253–262, 2013.

[13] J. Reineke, C. Stergiou, and S. Tripakis. Basic problems in multi-view modeling. Submitted journal version of [14].

[14] J. Reineke and S. Tripakis. Basic problems in multi-view modeling. In *Tools and Algorithms for the Construction and Analysis of Systems - TACAS*, volume 8413 of *LNCS*, pages 217–232. Springer, 2014.

[15] J. A. Robinson and A. Voronkov, editors. *Handbook of Automated Reasoning*. Elsevier and MIT Press, 2001.

[16] A. P. Sistla, M. Y. Vardi, and P. Wolper. The Complementation Problem for Büchi Automata with Applications to Temporal Logic. In *ICALP*, volume 194 of *LNCS*, pages 465–474. Springer, 1985.

[17] M. Tsai, S. Fogarty, M. Y. Vardi, and Y. Tsay. State of Büchi Complementation. *Logical Methods in Computer Science*, 10(4), 2014.