Universal Solutions in Temporal Data Exchange

Zehui Cheng
University of California Santa Cruz, CA, USA
zecheng@ucsc.edu

Phokion G. Kolaitis
University of California Santa Cruz, CA, USA
IBM Research, Almaden, CA, USA
kolaitis@ucsc.edu

Abstract

During the past fifteen years, data exchange has been explored in depth and in a variety of different settings. Even though temporal databases constitute a mature area of research studied over several decades, the investigation of temporal data exchange was initiated only very recently. We analyze the properties of universal solutions in temporal data exchange with emphasis on the relationship between universal solutions in the context of concrete time and universal solutions in the context of abstract time. We show that challenges arise even in the setting in which the data exchange specifications involve a single temporal variable. After this, we identify settings, including data exchange settings that involve multiple temporal variables, in which these challenges can be overcome.

2012 ACM Subject Classification Information systems → Data management systems; Theory of computation → Data exchange; Information systems → Temporal data

Keywords and phrases temporal databases, database dependencies, data exchange, universal solutions, abstract time, concrete time, Allen's relations

Digital Object Identifier 10.4230/LIPIcs.TIME.2020.8

Funding The research of Phokion Kolaitis is partially supported by NSF Grant IIS-1814152.

Acknowledgements We thank Jing Ao and Rada Chirkova for numerous fruitful conversations concerning temporal data exchange.

1 Introduction and Summary of Results

Data exchange is concerned with the transformation of data structured under one schema, called the *source* schema, into data structured under a different schema, called the *target* schema. Since the original formalization of the data exchange problem between relational schemas in [9] about fifteen years ago, an extensive study of data exchange has been carried out in several different settings, including XML data exchange [4], data exchange between graph databases [6], and relational to RDF data exchange [7]; an overview of the main results in this area can be found in the monograph [3]. Temporal databases constitute a mature area of research that has been studied in depth over several decades; for overviews, see, e.g., the book [13] or the book chapter [8]. Data exchange and temporal databases have advanced independently and, rather surprisingly, their paths did not cross until very recently, when Golshanara and Chomicki [11] published the first paper on *temporal data exchange*, that is, data exchange between temporal databases.

Data exchange is formalized using *schema mappings*, i.e., tuples of the form $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ is the source schema, $\mathbf{T}$ is the target schema, and Σ is a finite set of constraints in some suitable logical formalism that describe the relationship between source and target. Every fixed schema mapping $\mathcal{M}$ gives rise to the *data exchange problem with respect to* $\mathcal{M} = (\boldsymbol{S}, \boldsymbol{T}, \Sigma)$: given a source instance I, find a *solution* for I, that is, a target instance J so that $(I, J) \models \Sigma$. In general, no solution for I may exist or multiple solutions for I may exist.

27th International Symposium on Temporal Representation and Reasoning (TIME 2020).
Editors: Emilio Muñoz-Velasco, Ana Ozaki, and Martin Theobald; Article No. 8; pp. 8:1–8:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In [9], the concept of a *universal* solution was introduced and a case was made that universal solutions are the "best" solutions to materialize, provided solutions exist. In a precise sense (formalized using homomorphisms), a universal solution is a most general solution, thus it embodies no more and no less information than what the constraints in Σ specify. By now, universal solutions have been widely adopted as the preferred semantics in data exchange; furthermore, a concerted research effort has been dedicated to discovering when universal solutions exist and how to compute them. The main tool for computing universal solutions is the *chase* algorithm [9] and its variants (see [12] for a survey).

In temporal databases, there are two different models of time, namely, *concrete* time and *abstract* time; in the first model, time is represented by time intervals, while in the second by time points [8, 15]. Concrete temporal databases can be converted to abstract temporal databases using the *semantic function*[1] $[\![.]\!]$, which takes as input a concrete temporal database D and returns as output the abstract temporal database $[\![D]\!]$ where intervals of time in D are replaced by all points of time in them. The semantic function is often deployed to transfer results about concrete temporal databases to results about abstract temporal databases.

As already mentioned, Golshanara and Chomicki [11] are the first to investigate temporal data exchange. Specifically, they considered *temporal* schema mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$, where Σ_{st} is a set of temporal source-to-target tuple-generating dependencies (temporal s-t tgds) and Σ_t is a set of temporal target equality-generating dependencies (temporal target egds) with the restriction that each such constraint contains exactly one temporal variable. This means that each constraint in Σ_{st} is of the form $\forall \mathbf{x} \forall t(\varphi(\mathbf{x}, t) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}, t))$, where t is the only temporal variable, $\varphi(\mathbf{x}, t)$ is a conjunction of source atoms, and $\psi(\mathbf{x}, \mathbf{y}, t)$ is a conjunction of target atoms. Also, each constraint in Σ_t is of the form $\forall \mathbf{x} \forall t(\theta(\mathbf{x}, t) \rightarrow x_k = x_l)$, where t is the only temporal variable and $\theta(\mathbf{x}, t)$ is a conjunction of target atoms.

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ be a temporal schema mapping as above. The main result in [11] is the discovery of a variant of the chase algorithm that has the following properties: (a) it runs in polynomial time; (b) given a concrete source instance I, it detects if I has a solution with respect to $\mathcal{M}$; and (c) if I has such a solution, then it produces a concrete target instance J such that J is *semantically adequate* for I, i.e., the abstract target instance $[\![J]\!]$ is a universal solution for the abstract source instance $[\![I]\!]$. In the sequel, we call *normalizing* chase the variant of the chase used in [11]. It is a natural extension of the chase algorithm to temporal dependencies, but with the twist that first a *normalization* step is performed on the given concrete source instance I and then the temporal s-t tgds are applied to the resulting normalized instance $\mathcal{N}(I)$; after this, a second normalization step is performed on the resulting concrete target instance and then the temporal target egds are applied.

Summary of Results. Our investigation began when we noticed that Golshanara and Chomicki [11] do not address the question of whether or not the normalizing chase always produces a universal solution for a given concrete source instance, provided a solution exists (in fact, the notion of a universal solution for a concrete source instance is never introduced in [11]). We first show that the normalizing chase need *not* produce a universal solution for a given concrete source instance. Actually, we establish a stronger negative result: there is a temporal schema mapping $\mathcal{M}^* = (\mathbf{S}, \mathbf{T}, \Sigma^*_{st}, \Sigma^*_t)$ as above and a concrete source instance I^* that has a solution with respect to $\mathcal{M}^*$, but there is *no* concrete universal solution J for I^* or for the normalized instance $\mathcal{N}(I^*)$ that is semantically adequate for I^* (in particular, the result of the normalizing chase on I^* cannot be a universal solution for I^*).

[1] In the temporal databases literature, $[\![.]\!]$ is called the *semantic mapping*. Here, we chose to call it the *semantic function* to avoid confusion with the term *schema mapping*, which will be used repeatedly throughout this paper.

The preceding state of affairs motivates the following question: which temporal schema mappings admit semantically adequate concrete universal solutions? We make progress towards answering this question by identifying sufficient conditions that guarantee the existence of semantically adequate concrete universal solutions. To this effect, we show that if the temporal target egds have at most one temporal atom in their left-hand side (and any number of non-temporal atoms), then the output of the normalizing chase on a given concrete instance I is a concrete universal solution for $\mathcal{N}(I)$ and is also semantically adequate for I. In a sense, this is an optimal result because the temporal schema mapping $\mathcal{M}^*$ above contains a temporal target egd with two temporal atoms in its left-hand side, hence this result cannot be extended to the class of schema mappings studied by Golshanara and Chomicki [11].

All aforementioned results concern temporal schema mappings in which each constraint contains at most one temporal variable. Here, we embark on an investigation of temporal data exchange using schema mappings specified by constraints that may contain several different temporal variables. Such constraints may also contain comparisons between temporal variables using the well known Allen's relations, thus they can capture richer data exchange scenarios. This expansion of the landscape, however, comes with a number of complications, since, among other things, constraints in the concrete model of time need to be carefully translated into constraints in the abstract model of time (constraints with at most one temporal variable do not change, only the interpretation of the temporal variables does).

In the setting of multiple temporal variables, we consider temporal *full* schema mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$, i.e., schema mappings in which no existential quantifiers occur in the consequent of constraints in Σ_{st}. We show that if each temporal target egd has at most one temporal atom in its left-hand side, then we can produce concrete target instances that are both universal solutions and semantically adequate, provided solutions exist. Finally, we introduce another variant of the chase, which we call the *coalescing* chase, and show that for arbitrary temporal full schema mappings, the coalescing chase on concrete source instances always produces semantically adequate solutions, provided solutions exist.

2 Preliminaries

This section contains the definitions of the basic concepts and some background material.

Models of Time. Let $\mathbb{N} = \{1, 2, \ldots\}$ be the set of all natural numbers. In the *abstract* model of time, natural numbers represent *time points*. In the *concrete* model of time, closed-open intervals $[s, e) = \{t \in \mathbb{N} : s \leq t < e\}$, where s and e are natural numbers with $s < e$, represent *time intervals*. Unbounded time intervals of the form $[s, \infty)$ are also allowed.

Temporal Databases. A relational schema is a finite collection $\mathbf{R}$ of relation symbols of the form $R(A_1, \ldots, A_k)$, where $A_1, \ldots, A_k$ are the *attributes* of R and k is its arity. An $\mathbf{R}$-*instance* I is a finite collection of finite relations R^I, one for each relation symbol R in $\mathbf{R}$ and such that the arity of R^I matches that of R.

A *temporal* relation symbol is a relation symbol R in which one or more of its attributes are designated as temporal attributes, i.e., they can only take temporal values. In this paper, we assume that every temporal relation symbol has exactly one temporal attribute, which, without loss of generality, is the last attribute in the list. A *temporal* relational schema is a relational schema $\mathbf{R}$ containing at least one temporal relation symbol. For such a schema $\mathbf{R}$, an *abstract* $\boldsymbol{R}$-*instance* is an $\mathbf{R}$-instance in which the values of the temporal attributes are time points. A *concrete* $\boldsymbol{R}$-*instance* is an $\mathbf{R}$-instance in which the values of the temporal attributes are time intervals. We will use the term *temporal database* to refer to both abstract instances and concrete instances.

Constraints and Schema Mappings. Let $\mathbf{S}$ and $\mathbf{T}$ be two relational schemas, called, respectively, the *source* schema and the *target* schema, where $\mathbf{S}$ and $\mathbf{T}$ have no relation symbols in common. Data exchange from $\mathbf{S}$ to $\mathbf{T}$ is formalized using constraints in some logical formalism that describe the relationship between these two schemas [9]. The most widely used such constraints are *source-to-target tuple-generating dependencies* (s-t tgds) and *target equality-generating dependencies* (target egds). A s-t tgd is a first-order sentence of the form $\forall\mathbf{x}(\varphi(\mathbf{x}) \rightarrow \exists\mathbf{y}\psi(\mathbf{x},\mathbf{y}))$, where $\varphi(\mathbf{x})$ is a conjunction of source atoms, and $\psi(\mathbf{x},\mathbf{y})$ is a conjunction of target atoms. Such constraints can express a variety of data transformation tasks, including copying a relation, projecting a relation, augmenting a relation with an extra column, and joining two or more relations, where, in each case, the result of the transformation is moved to the target [14]. A target egd is a first-order sentence of the form $\forall\mathbf{x}(\theta(\mathbf{x}) \rightarrow x_k = x_l)$, where $\theta(\mathbf{x})$ is a conjunction of target atoms and x_k, x_l are variables occurring in $\mathbf{x}$. Target egds include target key constraints as an important special case.

The first step in formalizing data exchange between temporal relational schemas is to extend the concepts of s-t tgds and target egds to incorporate time. As stated in Section 1, Golshanara and Chomicki [11] initiated the study of temporal data exchange by considering temporal s-t tgds of the form $\forall\mathbf{x}\forall t(\varphi(\mathbf{x},t) \rightarrow \exists\mathbf{y}\psi(\mathbf{x},\mathbf{y},t))$ and temporal target egds of the form $\forall\mathbf{x}\forall t(\theta(\mathbf{x},t) \rightarrow x_k = x_l)$, where t is the only temporal variable that occurs in these formulas (in particular, the consequent of temporal s-t tgds contains no existentially quantified temporal variables).

In Section 4, we will explore a much richer framework for temporal data exchange in which the constraints considered may contain multiple temporal variables. We introduce the basic notions for this richer framework in this section (of course, these notions apply to the framework studied by Golshanara and Chomicki [11] as well). Specifically, we consider temporal s-t tgds of the form $\forall\mathbf{x}\forall\mathbf{t}(\varphi(\mathbf{x},\mathbf{t}) \rightarrow \exists\mathbf{y}\psi(\mathbf{x},\mathbf{y},\mathbf{t}))$ and temporal target egds of the form $\forall\mathbf{x}\forall\mathbf{t}(\theta(\mathbf{x},\mathbf{t}) \rightarrow x_k = x_l)$, where $\mathbf{t}$ is a (possibly empty) tuple of temporal variable; all other variables are non-temporal, thus the consequent of such temporal s-t tgds contains no existentially quantified temporal variables. We regard s-t tgds and target egds as the special cases of their temporal counterparts in which no temporal variable occurs (i.e., the tuple $\mathbf{t}$ is empty). In what follows, we will use the term *temporal schema mapping* for a tuple $\mathcal{M} = (\mathbf{S},\mathbf{T},\Sigma_{st},\Sigma_t)$, where $\mathbf{S}$ and $\mathbf{T}$ are disjoint temporal relational schemas, Σ_{st} is a finite set of temporal s-t tgds, and Σ_t is a finite set of temporal target egds, as above.

Values in Source and Target Instances. In data exchange between relational schemas, the source instances contain values from a countable domain Const of objects, called *constants*, while the target instances may contain values from the union Const ∪ Null, where Null is a countable set of distinct *labelled nulls* $N_1, N_2, \ldots$, which are typically used to witness the existentially quantified variables in the right-hand sides of s-t tgds. Thus, a labelled null represents some unknown value. In temporal data exchange, the values occurring in source and target instances may also be time points or time intervals, depending on the model of time used. Furthermore, the use of null values in target instances requires delicate handling because such null values may need to take into account the temporal context in which they are introduced. For this reason, temporal target instances may contain values that are constants, time points in the abstract model of time (or time intervals in the concrete model of time), labelled nulls $N_1, N_2, \ldots$, and *time-stamped* nulls, that is, null values of the form $N_1^{\mathbf{t}}, N_2^{\mathbf{t}}, \ldots$, where $\mathbf{t}$ is a finite sequence of time points (or a finite sequence of time intervals). Two such time-stamped nulls are equal if and only if they have the same subscript (label) and the same time-stamp. Intuitively, a time-stamped null represents unknown values in the context of its time-stamp. For example, a time-stamped null $N_j^{[2,5)}$ represents three unknown values, one at time-point 2, one at time-point 3, and one at time-point 4.

Homomorphisms, Solutions, and Universal Solutions. Let $\mathbf{T}$ be a temporal target schema and let J and J' be two temporal target databases over the same model of time (i.e., both are abstract or both are concrete). As discussed above, the relations in J and J' may contain constants, labelled nulls, and time-stamped nulls as values.

A *homomorphism* from J to J' is a function h from the active domain[2] of J to the active domain of J' such that: (a) if v is a constant or a time value (time point or time interval), then $h(v) = v$; (b) if v is a labelled null N_j, then $h(v)$ is either a constant or a labelled null N_k; (c) if v is a time-stamped null $N_j^{\mathbf{t}}$, then $h(N_j^{\mathbf{t}})$ is a constant or a null $N_k^{\mathbf{t}}$ with the same time-stamp or a labelled null N_k (without a time-stamp); (d) if a tuple $(v_1, \ldots, v_m)$ belongs to a relation R^J of J, then $(h(v_1), \ldots, h(v_m))$ belongs to the relation $R^{J'}$ of J'.

The intuition behind this definition is that if there is a homomorphism from J to J', then J is "more general" than J'. Time-stamped nulls are "more general" than labelled nulls, since the latter represent a single unknown value, while the former may represent multiple unknown values, depending on the time-stamp used. This explains the different treatment of labelled nulls and time-stamped nulls in conditions (b) and (c), respectively, in the definition.

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ be a temporal schema mapping and I a concrete source instance. A concrete target instance J is a *solution for* I w.r.t. $\mathcal{M}$ if the following conditions hold:

- If $\forall \mathbf{x}(\varphi(\mathbf{x}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}))$ is a (non-temporal) s-t tgd in Σ_{st} and if $\mathbf{a}$ is a tuple from the active domain of I such that $I \models \varphi(\mathbf{a})$, then there is a tuple $\mathbf{b}$ that consists of constants and/or labelled nulls such that $J \models \psi(\mathbf{a}, \mathbf{b})$.
- If $\forall \mathbf{x} \forall \mathbf{t}(\varphi(\mathbf{x}, \mathbf{t}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}, \mathbf{t}))$ is a temporal s-t tgd in Σ_{st} and if $\mathbf{a}$ is a tuple of constants and $\mathbf{i}$ is a tuple of intervals such that $I \models \varphi(\mathbf{a}, \mathbf{i})$, then there is a tuple $\mathbf{b}$ that consists of constants, labelled nulls, and time-stamped nulls such that every time-stamped null in $\mathbf{b}$ has $\mathbf{i}$ as its time-stamp and $J \models \psi(\mathbf{a}, \mathbf{b}, \mathbf{i})$.
- If $\forall \mathbf{x} \forall \mathbf{t}(\theta(\mathbf{x}, \mathbf{t}) \to x_k = x_l)$ is a temporal target egd in Σ_t and if $\mathbf{a}$ and $\mathbf{i}$ are tuples such that $J \models \theta(\mathbf{a}, \mathbf{i})$, then $a_k = a_l$, which means that a_k and a_l are the same constant or the same labelled null N_j or the same time-stamped null $N_j^{\mathbf{i}}$.

A concrete target instance J is a *universal solution for* I w.r.t. $\mathcal{M}$ if J is a solution for I w.r.t. $\mathcal{M}$ and, for every solution J' for I w.r.t. $\mathcal{M}$, there a homomorphism from J to J'.

The Chase and its Variants. In the case of (standard) data exchange, universal solutions are produced using the chase procedure [9]. Intuitively, given a source instance I, the chase procedure attempts to produce a target instance J by starting with the empty target instance, repeatedly applying the constraints of the given schema mapping, and generating new tuples in the current target instance as needed, so that eventually either the current target instance satisfies all the constraints of the schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ or a conflict arises in which case there is no solution for I w.r.t. $\mathcal{M}$. We now describe at a high level how the chase algorithm can be adapted to the setting of temporal data exchange.

Let K be the current concrete target instance in the run of the chase.

- If $\forall \mathbf{x}(\varphi(\mathbf{x}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}))$ is a (non-temporal) s-t tgd in Σ_{st} and if $\mathbf{a}$ is a tuple from the active domain of I such that $I \models \varphi(\mathbf{a})$, but $K \not\models \exists \mathbf{y} \psi(\mathbf{a}, \mathbf{y})$, then the chase generates a tuple $\mathbf{b}$ of distinct labelled nulls for the variables in $\mathbf{y}$ and adds tuples to the relations in K so that the resulting instance K' satisfies $\psi(\mathbf{a}, \mathbf{b})$. (Same as in standard chase.)
- If $\forall \mathbf{x} \forall \mathbf{t}(\varphi(\mathbf{x}, \mathbf{t}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}, \mathbf{t}))$ is a temporal s-t tgd in Σ_{st} and if $\mathbf{a}$ and $\mathbf{i}$ are such that $I \models \varphi(\mathbf{a}, \mathbf{i})$, but $K \not\models \exists \mathbf{y} \psi(\mathbf{a}, \mathbf{y}, \mathbf{i})$, then the chase generates a tuple $\mathbf{b}$ of distinct time-stamped labelled nulls for the variables in $\mathbf{y}$ all of which have the same time-stamp $\mathbf{i}$ and adds tuples to the relations in K so that the resulting instance K' satisfies $\psi(\mathbf{a}, \mathbf{b}, \mathbf{i})$.

[2] The *active domain* of a database is the set of all values occurring in the relations of that database.

- After the concrete source instance I has been chased with the constraints in Σ_{st}, then the concrete target instance K produced thus far is chased with the constraints in Σ_t. Specifically, if $\forall \mathbf{x} \forall \mathbf{t}(\theta(\mathbf{x}, \mathbf{t}) \to x_k = x_l)$ is a temporal target egd in Σ_t and $\mathbf{a}$ and $\mathbf{i}$ are tuples such that $K \models \theta(\mathbf{a}, \mathbf{i})$, then the following cases are considered: (1) if both a_k and a_l are labelled nulls or both are time-stamped nulls with the same time-stamp, then one of the two is replaced by the other throughout K; (2) if one of a_k and a_l is a constant and the other is a labelled null or a time-stamped null, then the labelled null or the time-stamped null is replaced by the constant throughout K; (3) if one of a_k, a_l is a labelled null and the other is a time-stamped null, then the time-stamped null is replaced by the labelled null throughout K; (4) if a_k and a_l are time-stamped nulls with different time-stamps or if a_k and a_l are different constants, then the chase fails.

In what follows, we will use the term the *concrete chase algorithm* to refer to the algorithm just described. In their study of temporal data exchange, Golshanara and Chomicki [11] considered a variant of the chase algorithm, which here we will call the *concrete n-chase algorithm*. There are two main differences between these two algorithms:

- In [11], all temporal schema mappings have s-t tgds with exactly one temporal variable, which implies that (standard) s-t tgds are not allowed. As a result, the target instances produced by the concrete n-chase algorithm contain no labelled nulls, but, of course, they may contain time-stamped nulls in which the time-stamp is a single interval.
- The concrete n-chase algorithm performs a *normalization* step before the constraints in Σ_{st} are applied and another normalization step before the constraints in Σ_t are applied. In particular, the concrete n-chase algorithm does not chase the given concrete source instance I with Σ_{st}, but, instead, chases the normalized instance $\mathcal{N}(I)$ with Σ_{st}. We refer the reader to Section 4.2 in [11] for the definition of normalization.

In what follows, if $\mathcal{M}$ is a temporal schema mapping and I is a concrete source instance, we will write c-chase$_{\mathcal{M}}(I)$ and n-chase$_{\mathcal{M}}(I)$ to denote the concrete target instance produced by the concrete chase algorithm and, respectively, the concrete n-chase algorithm on I.

Semantic Functions and Semantic Adequacy. As mentioned in Section 1, concrete instances are converted to abstract instances using the semantic function $[\![.]\!]$.

- If $\mu = (c_1, \ldots, c_m, [s, e))$ is a tuple in which each c_k is a constant and $[s, e)$ is an interval, then $[\![\mu]\!] = \{(c_1, \ldots, c_m, t) : s \le t < e\}$.
- If $I = (R_1, \ldots, R_n)$ is a concrete source instance, then $[\![I]\!]$ is the abstract source instance $[\![I]\!] = ([\![R_1]\!], \ldots, [\![R_n]\!])$, where $[\![R_l]\!] = \bigcup_{\mu \in R_l} [\![\mu]\!]$, for $1 \le l \le n$.
- We say that a tuple $\nu = (a_1, \ldots, a_m, [s, e))$ is *compatible* if each a_k is a constant or a labelled null or a time-stamped null $N_j^{[s_1,e_1),\ldots,[s_p,e_p)}$ such that $[s, e)$ is one of the intervals in the time-stamp, and all time-stamped nulls in ν have the same time-stamp. If ν is a compatible tuple, then $[\![\nu]\!]$ is the set of all tuples $(b_1, \ldots, b_m, t)$ such that the following conditions hold: if a_l is a constant or a labelled null, then $b_l = a_l$; if a_l is a time-stamped null $N_j^{[s_1,e_1),\ldots,[s_p,e_p)}$, then b_j is a time-stamped null $N_j^{t_1,\ldots,t_p}$, where $s_1 \le t_1 < e_1, \ldots, s_p \le t_p < e_p$; and, finally, $s \le t < e$.
- Let $J = (T_1, \ldots, T_m)$ be the concrete target instance produced by the concrete chase algorithm or by the concrete n-chase algorithm on a source instance I. It is easy to verify that every tuple occurring in one of the relations of J is compatible. Then $[\![J]\!]$ is the abstract target instance $[\![J]\!] = ([\![T_1]\!], \ldots, [\![T_m]\!])$, where $[\![T_l]\!] = \bigcup_{\nu \in T_l} [\![\nu]\!]$, for $1 \le l \le m$.
- Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ be a temporal schema mapping with exactly one temporal variable per constraint and let I be a concrete source instance. We say that a concrete target instance J is *semantically adequate* for I if the abstract target instance $[\![J]\!]$ is a universal solution for $[\![I]\!]$ w.r.t. $\mathcal{M}$.

We are now ready to state the main result in [11].

▶ **Theorem 1.** (Theorem 19 in [11]) *Let $\mathcal{M} = (\boldsymbol{S}, \boldsymbol{T}, \Sigma_{st}, \Sigma_t)$ be a temporal schema mapping, such that each relational symbol in* **S** *and* **T** *has one temporal attribute and each constraint in $\Sigma_{st} \cup \Sigma_t$ has* exactly *one temporal variable. If I is a concrete source instance, then the following statements are true.*

- *If the concrete n-chase algorithm on I fails, then there is no solution for I w.r.t. $\mathcal{M}$.*
- *If the concrete n-chase algorithm on I does not fail, then the concrete target instance n-chase$_{\mathcal{M}}(I)$ produced by the algorithm is semantically adequate for I.*

We note that the normalization steps in the concrete n-chase algorithm guarantee that there is a homomorphism from the left-hand side of a constraint in Σ_{st} or in Σ_t to a concrete instance K, provided there is a homomorphism from the left-hand side of that constraint to the abstract instance $[\![K]\!]$.

3 Temporal Data Exchange with a Single Temporal Variable

In this section, we explore aspects of data exchange for temporal schema mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ in which each constraint in $\Sigma_{st} \cup \Sigma_t$ contains at most one temporal variable. In what follows, we will also assume that all concrete source instances I are *coalesced*, that is, if $c_1, \ldots, c_m$ are constants and i, i' are intervals such that $(c_1, \ldots, c_m, i)$ and $(c_1, \ldots, c_m, i')$ belong to the same relation of I, then i and i' are disjoint intervals. Clearly, every concrete source instance can be easily transformed to an "equivalent" coalesced one [8].

3.1 No Semantically Adequate Concrete Universal Solutions

We begin by focusing more narrowly on schema mappings in the setting of Golshanara and Chomicki [11], that is, temporal schema mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ such that each relational symbol in **S** and **T** has one temporal attribute and each constraint in $\Sigma_{st} \cup \Sigma_t$ has *exactly* one temporal variable (hence, this variable occurs in every atom of the consequent of every s-t tgd). This class of schema mappings does not contain standard (non-temporal) schema mappings as a special case. Several remarks are in order now.

1. Such a schema mapping $\mathcal{M}$ is meaningful in both the concrete model of time and the abstract model of time without changing the constraints in $\Sigma_{st} \cup \Sigma_t$. In the first case, the temporal variable is ranging over time intervals and in the second over time points.
2. Every abstract source instance can be viewed as a sequence of *snapshots*, that is, as a sequence of non-temporal source instances parameterized by time points. One can then drop the temporal variable from the constraints in $\Sigma_{st} \cup \Sigma_t$, chase each snapshot with the resulting standard schema mapping, produce a universal solution for each snapshot (if a solution exists for each snapshot), and then consolidate the resulting target snapshots into an abstract target instance, which is an abstract universal solution for the given abstract source instance[3] - see [11] for formal details.
3. Let I be a concrete source instance. The concrete chase algorithm described in Section 2 produces a concrete universal solution for I w.r.t. $\mathcal{M}$, if a solution exists; if the concrete chase fails, no solution for I w.r.t. $\mathcal{M}$ exists. This follows from Theorem 5 in Section 4.

As mentioned in Section 1, Golshanara and Chomicki [11] do not address the question of whether or not their concrete n-chase algorithm produces a concrete universal solution. In fact, the notion of a concrete universal solution is not introduced in [11]. Our first result provides a strong negative answer to this question.

[3] If the chase fails on one of the snapshots, then no solution for the given abstract source instance exists.

▶ **Theorem 2.** *There is a temporal schema mapping* $\mathcal{M}^* = (\boldsymbol{S}, \boldsymbol{T}, \Sigma^*_{st}, \Sigma^*_t)$ *with one temporal variable in each constraint in* $\Sigma^*_{st} \cup \Sigma^*_t$ *and there is a concrete source instance* I^* *such that the following properties hold:*

1. *The concrete target instance* n-chase$_{\mathcal{M}^*}(I^*)$ *returned by the concrete n-chase algorithm on* I^* *is neither a solution for* I^* *nor for the normalized instance* $\mathcal{N}(I^*)$ *w.r.t.* $\mathcal{M}^*$*.*
2. *There is a concrete universal solution for* I^* *w.r.t.* $\mathcal{M}^*$*, but there is no concrete universal solution for* I^* *w.r.t.* $\mathcal{M}^*$ *that is semantically adequate for* I^**.*
3. *There is a concrete universal solution for* $\mathcal{N}(I^*)$ *w.r.t.* $\mathcal{M}^*$*, but there is no concrete universal solution for* $\mathcal{N}(I^*)$ *w.r.t.* $\mathcal{M}^*$ *that is semantically adequate for* $\mathcal{N}(I^*)$*.*

Proof. Let $\mathcal{M}^* = (\mathbf{S}, \mathbf{T}, \Sigma^*_{st}, \Sigma^*_t)$ be the schema mapping where Σ^*_{st} consists of the constraints

$$\forall n, s, c, t(E(n, c, t) \wedge S(n, s, t) \rightarrow Emp(n, c, s, t))$$

$$\forall n, c, p, t(P(n, p, t) \rightarrow \exists c EmpPos(n, c, p, t))$$

and Σ^*_t consists of the constraint

$$\forall n, c_1, c_2, s, p, t(Emp(n, c_1, s, t) \wedge EmpPos(n, c_2, p, t) \rightarrow c_1 = c_2).$$

Let I^* be the concrete source instance whose relations are depicted in Table 1. After normalizing I^* w.r.t. Σ^*_{st} (see [11] for the precise definition of normalization), we obtain the normalized instance $\mathcal{N}(I^*)$ whose relations are depicted in Table 2.

Table 1 The relations E, S, and P of the concrete source instance I^*.

(a) E.

Name	Company	Time
Ada	IBM	[2013, 2018)
Bob	IBM	[2012, 2015)

(b) S.

Name	Salary	Time
Ada	18000	[2014, 2018)
Bob	13000	[2013, 2015)

(c) P.

Name	Position	Time
Ada	Manager	[2015, 2017)
Bob	Consultant	[2012, 2015)

Table 2 The relations E, S, and P of the normalized instance $\mathcal{N}(I^*)$.

(a) E.

Name	Company	Time
Ada	IBM	[2013, 2014)
Ada	IBM	[2014, 2018)
Bob	IBM	[2012, 2013)
Bob	IBM	[2013, 2015)

(b) S.

Name	Salary	Time
Ada	18000	[2014, 2018)
Bob	13000	[2013, 2015)

(c) P.

Name	Position	Time
Ada	Manager	[2015, 2017)
Bob	Consultant	[2012, 2015)

Let n-chase$_{\mathcal{M}^*}(I^*)$ be the concrete target instance produced by the concrete n-chase algorithm on I^*; its relations are depicted in Table 3. It is easy to see that n-chase$_{\mathcal{M}^*}(I^*)$ is neither a solution for I^* nor a solution for $\mathcal{N}(I^*)$. This proves the first part of the theorem.

Table 3 The relations Emp and $EmpPos$ of the concrete target instance n-chase$_{\mathcal{M}^*}(I^*)$.

(a) Emp.

Name	Company	Salary	Time
Ada	IBM	18000	[2014, 2015)
Ada	IBM	18000	[2015, 2017)
Ada	IBM	18000	[2017, 2018)
Bob	IBM	13000	[2013, 2015)

(b) $EmpPos$.

Name	Company	Position	Time
Ada	IBM	Manager	[2015, 2017)
Bob	$N_2^{[2012,2013)}$	Consultant	[2012, 2013)
Bob	IBM	Consultant	[2013, 2015)

Let c-chase$_{\mathcal{M}^*}(I^*)$ and c-chase$_{\mathcal{M}^*}(\mathcal{N}(I^*))$ be the concrete target instances produced by the concrete chase algorithm on I^* and on $\mathcal{N}(I^*)$. The relations of c-chase$_{\mathcal{M}^*}(I^*)$ are depicted in Table 4, and those of c-chase$_{\mathcal{M}^*}(\mathcal{N}(I^*))$ in Table 5. Note that c-chase$_{\mathcal{M}^*}(I^*)$ is a universal solution for I^*, while c-chase$_{\mathcal{M}^*}(\mathcal{N}(I^*))$ is a universal solution for $\mathcal{N}(I^*)$.

Table 4 The relations *Emp* and *EmpPos* of the concrete target instance c-chase$_{\mathcal{M}^*}(I^*)$.

(a) *Emp*.

Name	Company	Salary	Time

(b) *EmpPos*.

Name	Company	Position	Time
Ada	$N_1^{[2015,2017)}$	Manager	$[2015, 2017)$
Bob	$N_2^{[2012,2015)}$	Consultant	$[2012, 2015)$

Table 5 The relations *Emp* and *EmpPos* of the concrete target instance c-chase$_{\mathcal{M}^*}(\mathcal{N}(I^*))$.

(a) *Emp*.

Name	Company	Salary	Time
Ada	IBM	18000	$[2014, 2018)$
Bob	IBM	13000	$[2013, 2015)$

(b) *EmpPos*.

Name	Company	Position	Time
Ada	$N_1^{[2015,2017)}$	Manager	$[2015, 2017)$
Bob	$N_2^{[2012,2015)}$	Consultant	$[2012, 2015)$

Let a-chase$_{\mathcal{M}^*}([\![I^*]\!])$ be the abstract target instance produced by chasing the snapshots of $[\![I^*]\!]$; its relations are depicted in Table 6.

Table 6 The relations *Emp* and *EmpPos* of the abstract target instance a-chase$_{\mathcal{M}^*}([\![I^*]\!])$.

(a) *Emp*.

Name	Company	Salary	Time
Ada	IBM	18000	2014
Ada	IBM	18000	2015
Ada	IBM	18000	2016
Ada	IBM	18000	2017
Bob	IBM	13000	2013
Bob	IBM	13000	2014

(b) *EmpPos*.

Name	Company	Position	Time
Ada	IBM	Manager	2015
Ada	IBM	Manager	2016
Bob	N_3^{2012}	Consultant	2012
Bob	IBM	Consultant	2013
Bob	IBM	Consultant	2014

As shown in [11], a-chase$_{\mathcal{M}^*}([\![I^*]\!])$ is a universal solution for $[\![I^*]\!]$ w.r.t. $\mathcal{M}^*$. It is now easy to verify that $[\![$c-chase$_{\mathcal{M}^*}(I^*)]\!]$ is *not* homomorphically equivalent to a-chase$_{\mathcal{M}^*}([\![I^*]\!])$. It follows that c-chase$_{\mathcal{M}^*}(I^*)$ is *not* semantically adequate for I^*. Furthermore, it is not hard to show that if J and J' are universal solutions for I^* w.r.t. $\mathcal{M}^*$, then $[\![J]\!]$ and $[\![J']\!]$ are homomorphically equivalent. Therefore, no concrete universal solution for I^* is semantically adequate for I^*. This proves the second part of the theorem. A similar argument with c-chase$_{\mathcal{M}^*}(\mathcal{N}(I^*))$ in place of c-chase$_{\mathcal{M}^*}(I^*)$ proves the third part of the theorem. ◀

3.2 Semantically Adequate Concrete Universal Solutions

Theorem 2 tells that in the temporal data exchange setting studied in [11], there are rather simple temporal schema mappings and temporal source instances for which no concrete universal solution is semantically adequate for these instances or for their normalized versions. A close scrutiny of the proof of Theorem 2 reveals that the root cause for this state of affairs appears to be the presence of two temporal atoms in the antecedent of the temporal target egd in Σ_t^*. Our next result tells that if the temporal target egds contain at most one temporal atom in the antecedent, then normalized instances have concrete universal solutions that are also semantically adequate concrete. Moreover, this result holds if each temporal constraint has *at most* one temporal variable, instead of *exactly* one temporal variable as in [11]; such constraints contain standard (non-temporal) s-t tgds and target egds as a special case.

▶ **Theorem 3.** *Let $\mathcal{M} = (\boldsymbol{S}, \boldsymbol{T}, \Sigma_{st}, \Sigma_t)$ be a temporal schema mapping such that (a) each s-t tgd contains at most one temporal variable; (b) if a s-t tgd contains a temporal variable, then that temporal variable occurs in every atom of its consequent; (c) each target egd contains at most one temporal atom in its antecedent. If I is a concrete source instance, then the following statements hold:*

1. *If a solution for I w.r.t. $\mathcal{M}$ exists, then n-chase$_{\mathcal{M}}(I)$ = c-chase$_{\mathcal{M}}(\mathcal{N}(I))$, that is, the concrete target instance returned by the concrete n-chase algorithm coincides with the concrete target instance returned by the concrete chase algorithm on $\mathcal{N}(I)$. Consequently, $\mathcal{N}(I)$ has a semantically adequate concrete universal solution.*
2. *If the concrete chase algorithm fails on $\mathcal{N}(I)$, then there is no solution for $[\![I]\!]$ w.r.t. $\mathcal{M}$.*

Proof. (*Hint*) The key observation is that if every constraint in Σ_t contains at most one temporal atom in its antecedent, then the second normalization step in the concrete n-chase algorithm does not change the temporal target instance produced by chasing $\mathcal{N}(I)$ with the constraints in Σ_{st}. It follows that n-chase$_{\mathcal{M}}(I)$ = c-chase$_{\mathcal{M}}(\mathcal{N}(I))$. It can also be shown that n-chase$_{\mathcal{M}}(I)$ is semantically adequate, even in this setting where each constraint in $\Sigma_{st} \cup \Sigma_t$ contains at most one temporal variable (instead of exactly one such variable as in [11]). ◀

It should be pointed out that there are a schema mapping $\mathcal{M}'$ that satisfies the hypothesis in Theorem 3 and a concrete source instance I' such that no semantically adequate concrete universal solution for I' w.r.t. $\mathcal{M}'$ exists. This is shown in the next proposition.

▶ **Proposition 4.** *There is a temporal schema mapping $\mathcal{M}' = (\boldsymbol{S}, \boldsymbol{T}, \Sigma'_{st}, \Sigma'_t)$ where each constraint in $\Sigma'_{st} \cup \Sigma'_t$ contains at most one temporal variable and each constraint in Σ'_t contains at most one temporal atom in its antecedent, and there is a concrete source instance I', such that there exists a concrete universal solution for I' w.r.t. $\mathcal{M}'$, but there is no concrete universal solution for I' w.r.t. $\mathcal{M}'$ that is semantically adequate for I'.*

Proof. Let $\mathcal{M}' = (\mathbf{S}, \mathbf{T}, \Sigma'_{st}, \Sigma'_t)$ be the schema mapping where Σ'_{st} consists of the constraints

$$\forall n, s, c, t(E(n, c, t) \land S(n, s, t) \rightarrow Emp(n, c, s, t))$$

$$\forall n, c, p(P(n, p) \rightarrow \exists c\; EmpPos(n, c, p))$$

and Σ'_t consists of the constraint

$$\forall n, c_1, c_2, s, p, t(Emp(n, c_1, s, t) \land EmpPos(n, c_2, p) \rightarrow c_1 = c_2).$$

Let I' be the concrete source instance whose relations are depicted in Table 7. After applying the semantic function on I', we obtain the abstract source instance $[\![I']\!]$ whose relations are depicted in Table 8.

Table 7 The relations E, S, and P of the concrete source instance I'.

(a) E.

Name	Company	Time
Ada	IBM	[2013, 2018)
Bob	IBM	[2012, 2015)

(b) S.

Name	Salary	Time
Ada	18000	[2014, 2018)
Bob	13000	[2013, 2015)

(c) P.

Name	Position
Ada	Manager
Bob	Consultant

Table 8 The relations E, S, and P of the abstract source instance $[\![I']\!]$.

(a) E.

Name	Company	Time
Ada	IBM	2013
Ada	IBM	2014
Ada	IBM	2015
Ada	IBM	2016
Ada	IBM	2017
Bob	IBM	2012
Bob	IBM	2013
Bob	IBM	2014

(b) S.

Name	Salary	Time
Ada	18000	2014
Ada	18000	2015
Ada	18000	2016
Ada	18000	2017
Bob	13000	2013
Bob	13000	2014

(c) P.

Name	Position
Ada	Manager
Bob	Consultant

Let c-chase$_{\mathcal{M}'}(I')$ be the concrete target instances produced by the concrete chase algorithm on I'. The relations of c-chase$_{\mathcal{M}'}(I')$ are depicted in Table 9. Note that c-chase$_{\mathcal{M}'}(I')$ is a universal solution for I'.

Table 9 The relations *Emp* and *EmpPos* of the concrete target instance c-chase$_{\mathcal{M}'}(I')$.

(a) *Emp.*

Name	Company	Salary	Time

(b) *EmpPos.*

Name	Company	Position
Ada	N_1	Manager
Bob	N_2	Consultant

Let a-chase$_{\mathcal{M}'}(\llbracket I' \rrbracket)$ be the abstract target instance produced by chasing the snapshots of $\llbracket I' \rrbracket$; its relations are depicted in Table 10.

Table 10 The relations *Emp* and *EmpPos* of the abstract target instance a-chase$_{\mathcal{M}'}(\llbracket I' \rrbracket)$.

(a) *Emp.*

Name	Company	Salary	Time
Ada	IBM	18000	2014
Ada	IBM	18000	2015
Ada	IBM	18000	2016
Ada	IBM	18000	2017
Bob	IBM	13000	2013
Bob	IBM	13000	2014

(b) *EmpPos.*

Name	Company	Position
Ada	IBM	Manager
Bob	IBM	Consultant

As shown in [11], a-chase$_{\mathcal{M}'}(\llbracket I' \rrbracket)$ is a universal solution for $\llbracket I' \rrbracket)$ w.r.t. $\mathcal{M}'$. It is now easy to verify that $\llbracket$c-chase$_{\mathcal{M}'}(I')\rrbracket$ is *not* homomorphically equivalent to a-chase$_{\mathcal{M}'}(\llbracket I' \rrbracket)$. From this, it follows that c-chase$_{\mathcal{M}'}(I')$ is *not* semantically adequate for I'. Furthermore, it is not hard to show that if J and J' are universal solutions for I' w.r.t. $\mathcal{M}'$, then $\llbracket J \rrbracket$ and $\llbracket J' \rrbracket$ are homomorphically equivalent. Consequently, no concrete universal solution for I' is semantically adequate for I'. This completes the proof of the proposition. ◀

4 Temporal Data Exchange with Multiple Temporal Variables

In this section, we initiate the study of temporal data exchange for schema mappings whose constraints may contain multiple temporal variables. Such constraints make it possible to model more complex transformations of temporal data. In the presence of multiple temporal variables, it is natural to also allow comparisons between different temporal variables. In the concrete model of time, this means that the antecedents of the s-t tgds and the target egds may also contain Boolean combinations of the well known Allen's relations between time intervals [1, 2], such as m (meets), o (overlaps), $\prec$ (before) , $\succ$ (after), and $=$. Thus, in this section, we consider temporal schema mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ in which each constraint in Σ_{st} is of the form $\forall \mathbf{x} \forall \mathbf{t}(\varphi(\mathbf{x}, \mathbf{t}) \wedge \pi(\mathbf{t}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}, \mathbf{t}))$, where the only temporal variables are those in $\mathbf{t}$; $\varphi(\mathbf{x}, \mathbf{t})$ is a conjunction of source atoms; $\pi(\mathbf{t})$ is a Boolean combination of Allen's relations involving variables from $\mathbf{t}$; and $\psi(\mathbf{x}, \mathbf{y}, \mathbf{t})$ is a conjunction of target atoms (in particular, no temporal variable is existentially quantified). By the same token, each constraint in Σ_t is of the form $\forall \mathbf{x} \forall \mathbf{t}(\theta(\mathbf{x}, \mathbf{t}) \wedge \rho(\mathbf{t}) \rightarrow x_k = x_l)$, where the only temporal variables are those in $\mathbf{t}$; $\theta(\mathbf{x}, \mathbf{t})$ is a conjunction of target atoms; $\rho(\mathbf{t})$ is a Boolean combination of Allen's relations involving variables from $\mathbf{t}$; and x_k, x_l are among the variables in $\mathbf{x}$.

The next result extends Theorem 3.3 in [9] from the case of (standard) data exchange to a restricted case of temporal data exchange.

▶ **Theorem 5.** *Let $\mathcal{M} = (\boldsymbol{S}, \boldsymbol{T}, \Sigma_{st}, \Sigma_t)$ be a temporal schema mapping, such that one of the following two conditions holds: (a) Every s-t tgd is full (i.e., its consequent contains no existential quantifiers); (b) If a s-t tgd is not full and if it contains a temporal variable, then this is the only temporal variable in that s-t tgd, and it occurs in every atom of the consequent of the s-t tgd; moreover, every target egd contains at most one temporal variable. If I is a concrete source instance, then the following statements hold:*

1. *If the concrete chase algorithm does not fail on I, then the concrete target instance c-chase$_{\mathcal{M}}(I)$ returned by this algorithm is a concrete universal solution for I w.r.t. $\mathcal{M}$.*
2. *If the concrete chase algorithm fails on I, there is no solution for I w.r.t. $\mathcal{M}$.*

The running time of the concrete chase algorithm is bounded by a polynomial in the size of I.

Next, we explore the interplay between the concrete and the abstract models of time with focus on the existence of semantically adequate concrete universal solutions. In the presence of multiple temporal variables, concrete s-t tgds and concrete target egds must be converted to "essentially equivalent" abstract s-t tgds and to abstract target egds, respectively, because the concrete ones involve Allen's relations while the abstract ones involve suitable formulas of first-order logic over time points compared with the $<$ relation. Due to space limitations, we do not include here the precise definition of this conversion. Instead, we describe the precise sense in which this conversion transforms concrete constraints to "essentially equivalent" abstract constraints, and also illustrate this conversion in the proof of Proposition 7.

We will use the terms *concrete schema mapping* and *abstract schema mapping* for a schema mapping consisting of concrete constrains and, respectively, of abstract constraints. If σ is a concrete s-t tgd or a concrete target egds, then we write $a(\sigma)$ for the abstract s-t tgd or the abstract target egd resulting from σ via the aforementioned conversion. Every concrete schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ gives rise to an abstract schema mapping $\mathcal{M}^a = (\mathbf{S}, \mathbf{T}, \Sigma^a_{st}, \Sigma^a_t)$, where $\Sigma^a_{st} = \{a(\sigma) : \sigma \in \Sigma_{st}\}$ and $\Sigma^a_t = \{a(\sigma) : \sigma \in \Sigma_t\}$.

Let $\mathbf{x} = (x_1, \ldots, x_m)$ be a tuple of non-temporal variables and let $\mathbf{t} = (t_1, \ldots, t_k)$ be a tuple of temporal variables. A *concrete* (respectively, an *abstract*) assignment to the tuple $(\mathbf{x}, \mathbf{t})$ is a function p defined on the set $\{x_1, \ldots, x_m, t_1, \ldots, t_k\}$ such that $p(x_i) = c_i$ is a constant, $1 \leq i \leq m$, and $p(t_j) = [s_j, e_j)$ is an interval (respectively, $p(t_j) = \alpha_j$ is a time point), $1 \leq j \leq k$. If p is a concrete assignment as above, we will use the notation $p(\mathbf{x}, \mathbf{t}) = (c_1, \ldots, c_m, [s_1, e_1), \ldots, [s_k, e_k))$ to denote it. Similarly, if p is an abstract assignment, it will be denoted as $p(\mathbf{x}, \mathbf{t}) = (c_1, \ldots, c_m, \alpha_1, \ldots, \alpha_k)$.

The semantic function $[\![.]\!]$ on concrete assignments is defined as follows: if $p(\mathbf{x}, \mathbf{t}) = (c_1, \ldots, c_m, [s_1, e_1), \ldots, [s_k, e_k))$ is a concrete assignment, then $[\![(p(\mathbf{x}, \mathbf{t}))]\!]$ is the set of all abstract assignments $q(\mathbf{x}, \mathbf{t}) = (c_1, \ldots, c_m, \alpha_1, \ldots, \alpha_k)$, where $s_j \leq \alpha_j < e_j$ and $1 \leq j \leq k$. The next proposition describes the properties of the conversion from concrete formulas to "essentially equivalent" abstract formulas.

▶ **Proposition 6.** *Assume that $\psi(\boldsymbol{x}, \boldsymbol{t})$ is a formula of the form $\psi(\mathbf{x}, \boldsymbol{t}) = \varphi(\mathbf{x}, \boldsymbol{t}) \wedge \pi(\boldsymbol{t})$, where the variables in $\boldsymbol{t}$ are the only temporal variables, $\varphi(\boldsymbol{x}, \boldsymbol{t})$ is a conjunction of atoms over a temporal schema $\boldsymbol{S}$, and $\pi(\boldsymbol{t})$ is a Boolean combination of Allen's relations involving variables from $\boldsymbol{t}$. Let $\psi^a(\boldsymbol{x}, \boldsymbol{t})$ be the formula obtained by converting $\psi(\boldsymbol{x}, \boldsymbol{t})$ from the concrete model of time to the abstract model of time. Given a coalesced concrete instance I and a concrete assignment $p(\boldsymbol{x}, \boldsymbol{t})$ taking values in I, the following statements are equivalent:*

- *$I, p(\boldsymbol{x}, \boldsymbol{t}) \models \psi(\mathbf{x}, \boldsymbol{t})$.*
- *For every abstract assignment $q(\boldsymbol{x}, \boldsymbol{t}) \in [\![p(\boldsymbol{x}, \boldsymbol{t})]\!]$, we have that $[\![I]\!], q(\boldsymbol{x}, \boldsymbol{t}) \models \psi^a(\boldsymbol{x}, \boldsymbol{t})$.*

Furthermore, for every abstract assignment $q(\boldsymbol{x}, \boldsymbol{t})$ such that $[\![I]\!], q(\boldsymbol{x}, \boldsymbol{t}) \models \psi^a(\boldsymbol{x}, \boldsymbol{t})$, there is a unique concrete assignment $p(\boldsymbol{x}, \boldsymbol{t})$ such that $q(\boldsymbol{x}, \boldsymbol{t}) \in [\![p(\boldsymbol{x}, \boldsymbol{t})]\!]$ and $I, p(\boldsymbol{x}, \boldsymbol{t}) \models \psi(\boldsymbol{x}, \boldsymbol{t})$.

Earlier, we defined the notion of semantic adequacy for temporal schema mappings in which each constraint had (at most) one temporal variable. We now extend this notion to temporal schema mappings in which constraints may have any number of temporal variables If $\mathcal{M}$ is a concrete schema mapping and I is a concrete source instance, then a concrete target instance J is *semantically adequate* for I if $[\![J]\!]$ is a universal solution for $[\![I]\!]$ w.r.t. $\mathcal{M}^a$. Ideally, we would like to have concrete universal solutions for I that are also semantically adequate for I. As we have seen in Section 3, however, this is not possible in general, even for temporal schema mappings $\mathcal{M}$ with a single temporal variable (where we have $\mathcal{M} = \mathcal{M}^a$). In what follows, we identify a sufficient condition for semantic adequacy.

A concrete s-t tgd is *full* if its consequent contains no existentially quantified variables, i.e., it is of the form $\forall \mathbf{x} \forall \mathbf{t}(\varphi(\mathbf{x}, \mathbf{t}) \land \pi(\mathbf{t}) \to \psi(\mathbf{x}, \mathbf{t}))$. A concrete schema mapping is *full* if all its concrete s-t tgds are full. Full schema mappings are also known as *Global-as-View* or GAV schema mappings, because each full s-t tgd is logically equivalent to a finite set of s-t tgds with a single atom in their consequents.

As an example of a concrete full schema mapping, let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ be the schema mapping in which Σ_{st} consists of the concrete full s-t tgds

$$\sigma^1_{st} = \forall x_1, x_2, x_3, t_1(R_1(x_1, x_2, x_3, t_1) \to T_1(x_1, x_2, t_1)),$$

$$\sigma^2_{st} = \forall x_1, x_2, x_3, x_4, t_1, t_2(R_2(x_1, x_2, x_3, t_1) \land R_3(x_1, x_4, t_2) \land (t_2 \mathsf{\ m\ } t_1) \to T_2(x_1, x_3, t_2))$$

and Σ_t consists of the concrete target egds

$$\sigma^1_t = \forall x_1, x_2, x_3, t_1(T_1(x_1, x_2, t_1) \land T_1(x_1, x_3, t_1) \to x_2 = x_3),$$

$$\sigma^2_t = \forall x_1, x_2, x_3, t_1, t_2(T_1(x_1, x_2, t_1) \land T_2(x_1, x_3, t_2) \land (t_1 \mathsf{\ o\ } t_2) \to x_2 = x_3).$$

In standard data exchange, full schema mappings have been extensively studied and have been shown to possess a variety of good structural and algorithmic properties (see, e.g., [10, 14]). Unfortunately, as our next result shows, these good properties do not include semantic adequacy.

▶ **Proposition 7.** *There are a concrete full schema mapping $\mathcal{M}^+ = (\boldsymbol{S}, \boldsymbol{T}, \Sigma^+_{st}, \Sigma^+_t)$ and a concrete source instance I^+ such that the following statements hold:*

1. *There is a concrete universal solution for I^+ w.r.t. $\mathcal{M}^+$.*
2. *There is no solution for $[\![I^+]\!]$ w.r.t. $\mathcal{M}^{+a}$; therefore, no concrete universal solution for I^+ w.r.t. $\mathcal{M}^+$ is semantically adequate for I^+.*

Proof. Let $\mathcal{M}^+ = (\mathbf{S}, \mathbf{T}, \Sigma^+_{st}, \Sigma^+_t)$ be a concrete schema mapping where Σ^+_{st} consists of the concrete s-t tgds

$$\sigma^1_{st} = \forall x_1, x_2, x_3, t_1(R_1(x_1, x_2, x_3, t_1) \to T_1(x_1, x_2, t_1))$$

$$\sigma^2_{st} = \forall x_1, x_2, x_3, x_4, t_1, t_2(R_2(x_1, x_2, x_3, t_1) \land R_3(x_1, x_4, t_2) \land (t_2 \mathsf{\ m\ } t_1) \to T_2(x_1, x_3, t_2))$$

and Σ^+_t consists of the concrete target egd

$$\sigma_t = \forall x_1, x_2, x_3, t_1(T_1(x_1, x_2, t_1) \land T_2(x_1, x_3, t_1) \to x_2 = x_3).$$

Let $\mathcal{M}^{+a} = (\mathbf{S}, \mathbf{T}, \Sigma^{+a}_{st}, \Sigma^{+a}_t)$ be the abstract schema mapping obtained from $\mathcal{M}^+$ by converting the concrete constraints of $\mathcal{M}$ to abstract constraints. In this case, Σ^{+a}_{st} consists of the following abstract s-t tgds $a(\sigma^1_{st})$ and $a(\sigma^2_{st})$ obtained from σ^1_{st} and σ^2_{st}, respectively:

$$a(\sigma^1_{st}) = \forall x_1, x_2, x_3, t_1(R_1(x_1, x_2, x_3, t_1) \to T_1(x_1, x_2, t_1))$$

$$a(\sigma_{st}^2) = \forall x_1, x_2, x_3, x_4, t_1, t_2 \Big(R_2(x_1, x_2, x_3, t_1) \wedge R_3(x_1, x_4, t_2) \wedge \exists t_1^-, t_1^+, t_2^-, t_2^+ \Big(R_2(x_1, x_2, x_3, t_1^-) \wedge R_3(x_1, x_4, t_2^-) \wedge R_2(x_1, x_2, x_3, t_1^+) \wedge R_3(x_1, x_4, t_2^+) \wedge (t_1^- \leq t_1 \leq t_1^+) \wedge (t_2^- \leq t_2 \leq t_2^+) \wedge \forall t_1', t_2' \Big(((t_1^- \leq t_1' \leq t_1^+) \wedge (t_2^- \leq t_2' \leq t_2^+) \rightarrow R_2(x_1, x_2, x_3, t_1') \wedge R_3(x_1, x_4, t_2')) \wedge (R_2(x_1, x_2, x_3, t_1') \wedge R_3(x_1, x_4, t_2') \rightarrow (t_1' \neq t_1^- - 1) \wedge (t_1' \neq t_1^+ + 1) \wedge (t_2' \neq t_2^- - 1) \wedge (t_2' \neq t_2^+ + 1)) \Big) \wedge (t_2^+ + 1 = t_1^-) \Big) \rightarrow T_2(x_1, x_3, t_2) \Big).$$

Moreover, Σ_t^{+a} consists of the following abstract target egd $a(\sigma_t)$ obtained from σ_t:

$$a(\sigma_t) = \forall x_1, x_2, x_3, t_1, t_2 \Big(T_1(x_1, x_2, t_1) \wedge T_2(x_1, x_3, t_2) \wedge \exists t_1^-, t_1^+, t_2^-, t_2^+ \Big(T_1(x_1, x_2, t_1^-) \wedge T_2(x_1, x_3, t_2^-) \wedge T_1(x_1, x_2, t_1^+) \wedge T_2(x_1, x_3, t_2^+) \wedge (t_1^- \leq t_1 \leq t_1^+) \wedge (t_2^- \leq t_2 \leq t_2^+) \wedge \forall t_1', t_2' \Big(((t_1^- \leq t_1' \leq t_1^+) \wedge (t_2^- \leq t_2' \leq t_2^+) \rightarrow T_1(x_1, x_2, t_1') \wedge T_2(x_1, x_3, t_2')) \wedge (T_1(x_1, x_2, t_1') \wedge T_2(x_1, x_3, t_2') \rightarrow (t_1' \neq t_1^- - 1) \wedge (t_1' \neq t_1^+ + 1) \wedge (t_2' \neq t_2^- - 1) \wedge (t_2' \neq t_2^+ + 1)) \Big) \wedge (t_1^- = t_2^- \wedge t_1^+ = t_2^+) \Big) \rightarrow x_2 = x_3 \Big).$$

Before completing the proof of the proposition, we provide some intuition about the conversion of the concrete temporal constraints of $\mathcal{M}$ to the abstract temporal constraints of $\mathcal{M}^{+a}$. To begin with, $a(\sigma_{st}^1)$ is the same as σ_{st}^1 because σ_{st}^1 has a single temporal variable and no Allen's relations. In contrast, $a(\sigma_{st}^2)$ is quite different from σ_{st}^2 because it has two temporal variables and one atomic formula involving Allen's relation m (meets). The sub-formula $\exists t_1^-, t_1^+, t_2^-, t_2^+ \Big(R_2(x_1, x_2, x_3, t_1^-) \wedge \ldots \wedge (t_2^+ + 1 = t_1^-) \Big)$ of $a(\sigma_{st}^2)$ asserts that: (i) the abstract variables t_1 and t_2 belong to intervals that meet each other (this is the purpose of the sub-formula $(t_2^+ + 1 = t_1^-)$; (ii) all temporal values t_1' and t_2' in these intervals have the property that $R_2(x_1, x_2, x_3, t_1')$ and $R_3(x_1, x_4, t_2')$ hold; and (iii) there are no bigger intervals for which (i) and (ii) hold. A similar intuition applies to the construction of the abstract target egd $a(\sigma_t)$. The correctness of this conversion (i.e., that the abstract constraints $a(\sigma_{st}^1)$, $a(\sigma_{st}^2)$, and $a(\sigma_t)$ are "essentially equivalent" to the concrete constraints σ_{st}^1, σ_{st}^2, and σ_t) uses the fact that we use coalesced concrete source instances.

Let I^+ be the concrete source instance whose relations are depicted in Table 11. By applying the semantic function on I^+, we obtain the abstract instance $[\![I^+]\!]$, whose relations are depicted in Table 12.

Let c-chase$_{\mathcal{M}^+}(I^+)$ be the target instance produced by the concrete chase algorithm on I^+. The relations of c-chase$_{\mathcal{M}^+}(I^+)$ are depicted in Table 13. According to Theorem 5, c-chase$_{\mathcal{M}^+}(I^+)$ is a universal solution for I^+ w.r.t. $\mathcal{M}^+$.

Table 11 The relations R_1, R_2, and R_3 in the coalesced concrete source instance I^+.

(a) R_1.

name	school	position	Ptime
a_1	c_1	d_1	$[1,3)$
a_1	c_1	d_2	$[2,4)$

(b) R_2.

name	address	school	Stime
a_1	b_1	c_2	$[4,6)$

(c) R_3.

name	city	Ctime
a_1	e_1	$[1,4)$

We claim that the abstract chase algorithm w.r.t. $\mathcal{M}^+$ fails on $[\![I^+]\!]$. To see this, let a-chase$_{\Sigma_{st}^{+a}}([\![I^+]\!])$ be the target instance produced by chasing $[\![I^+]\!]$ with the abstract s-t tgds in Σ_{st}^{+a}. The relations of a-chase$_{\Sigma_{st}^{+a}}([\![I^+]\!])$ are depicted in Table 14. If we now chase

Table 12 The relations R_1, R_2 and R_3 in the abstract source instance $[\![I^+]\!]$.

(a) R_1.

name	school	position	Ptime
a_1	c_1	d_1	1
a_1	c_1	d_1	2
a_1	c_1	d_2	2
a_1	c_1	d_2	3

(b) R_2.

name	address	school	Stime
a_1	b_1	c_2	4
a_1	b_1	c_2	5

(c) R_3.

name	city	Ctime
a_1	e_1	1
a_1	e_1	2
a_1	e_1	3

Table 13 The relations T_1 and T_2 in the target instance c-chase$_{\mathcal{M}^+}(I^+)$.

(a) T_1.

name	school	Ptime
a_1	c_1	$[1, 3)$
a_1	c_1	$[2, 4)$

(b) T_2.

name	school	Ctime
a_1	c_2	$[1, 4)$

Table 14 The relations T_1 and T_2 in the abstract target instance a-chase$_{\Sigma_{st}^{+a}}([\![I^+]\!])$.

(a) T_1.

name	school	Ptime
a_1	c_1	1
a_1	c_1	2
a_1	c_1	3

(b) T_2.

name	school	Ctime
a_1	c_2	1
a_1	c_2	2
a_1	c_2	3

a-chase$_{\Sigma_{st}^{+a}}([\![I^+]\!])$ with the abstract target egd in Σ_t^{+a}, then the abstract chase algorithm fails. This is because the tuple $(a_1, b_1, c_1, 1)$ in the relation T_1 and the tuple $(a_1, c_2, 1)$ in the relation T_2 of a-chase$_{\Sigma_{st}^{+a}}([\![I^+]\!])$ trigger the antecedent of the abstract target egd $a(\sigma_t)$ in Σ_t^{+a}, hence the abstract chase algorithm fails because it attempts to equate the distinct constants c_1 and c_2. It follows that there is no solution for $[\![I^+]\!]$ w.r.t. $\mathcal{M}^{+a}$. Furthermore, it is not hard to show that if J and J' are universal solutions for I^+ w.r.t. $\mathcal{M}^+$, then $[\![J]\!]$ and $[\![J']\!]$ are homomorphically equivalent. Consequently, no concrete universal solution for I^+ w.r.t. $\mathcal{M}^+$ is semantically adequate for I^+ (in particular, the concrete universal solution c-chase$_{\mathcal{M}^+}(I^+)$ of I^+ w.r.t. $\mathcal{M}^+$ is not semantically adequate for I^+). This completes the proof of the proposition. ◀

Observe that the temporal target egd σ_t of $\mathcal{M}^+$ had two temporal atoms in its antecedent. Our next result tells that semantically adequate universal solutions exist for full schema mappings whose temporal target egds have at most one temporal atom in their antecedent.

▶ **Theorem 8.** *Let $\mathcal{M} = (\boldsymbol{S}, \boldsymbol{T}, \Sigma_{st}, \Sigma_t)$ be a concrete full schema mapping such that each constraint in Σ_t contains at most one temporal atom. If I is a concrete source instance, then the following statements hold:*

1. *If a solution for I w.r.t. $\mathcal{M}$ exists, then the concrete target instance c-chase$_{\mathcal{M}}(I)$ returned by the concrete chase algorithm is semantically adequate for I.*
2. *If the concrete chase algorithm fails on I, then there is no solution for $[\![I]\!]$ w.r.t. to the abstract schema mapping $\mathcal{M}^a$.*

According to Proposition 7, if $\mathcal{M}$ is a concrete full schema mapping, then there may exist concrete source instances I for which no concrete universal solution is semantically adequate. As discussed earlier, Golshanara and Chomicki [11] used the concrete n-chase algorithm to construct semantically adequate concrete target instances in the setting of temporal schema mappings with exactly one temporal variable. It is not all clear whether or not the concrete n-chase algorithm can be extended to temporal schema mappings with multiple temporal variables. Instead, we introduce a different variant of the chase, which we call the *coalescing* chase algorithm. This algorithm proceeds along the lines of the concrete chase algorithm by introducing labelled nulls or time-stamped nulls as needed when temporal s-t tgds are considered or by equating two values when temporal target egds are considered. However, after each such chase step, the resulting target instance is transformed to a coalesced one before the next chase step is applied (in general, a chase step on a coalesced instance may produce a non-coalesced instance). Note that the concrete n-chase algorithm applies only two normalization steps, while the number of coalescing steps applied by the coalescing chase algorithm is not fixed.

Our final result asserts that the coalescing chase algorithm produces semantically adequate target instances in the setting of concrete full schema mappings.

▶ **Theorem 9.** *Let $\mathcal{M} = (\boldsymbol{S}, \boldsymbol{T}, \Sigma_{st}, \Sigma_t)$ be a concrete full schema mapping. If I is a concrete source instance, then the following statement hold:*

1. *If the coalescing chase does not fail on I, then the concrete target instance returned by the coalescing chase is semantically adequate for I.*
2. *If the coalescing chase fails on I, then there is no solution for $[\![I]\!]$ w.r.t. to the abstract schema mapping $\mathcal{M}^a$.*

5 Concluding Remarks

The work reported here contributes to the development of temporal data exchange. Our main focus was on the pursuit of semantically adequate universal solutions. We showed that such solutions may not exist even for temporal schema mappings with a single temporal variable. Nonetheless, we identified classes of schema mappings for which such solutions exist and also classes of schema mappings for which semantically adequate target instances exist. Along the way, we expanded the original framework of temporal data exchange studied in [11] by considering temporal schema mappings with multiple temporal variables and exploring some of the issues involved in the translation from the concrete model of time to the abstract.

We conclude by describing two directions for further research in this area.

- Explore temporal data exchange for schema mappings that also have target tuple-generating dependencies. Several challenges arise in this case, including the translation of the constraints from the concrete model of time to the abstract model of time, the management of time-stamped nulls, and the design of a suitable chase algorithm.
- Explore temporal data exchange for schema mappings in which the constraints have existentially quantified variables. Several challenges of different nature arise in this case, some of which are similar to challenges in answering queries over temporal data with the help of ontologies (see [5] for a comprehensive survey of that area).

References

1 James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, 1983. doi:10.1145/182.358434.
2 James F. Allen. Time and time again: The many ways to represent time. *Int. J. Intell. Syst.*, 6(4):341–355, 1991. doi:10.1002/int.4550060403.
3 Marcelo Arenas, Pablo Barceló, Leonid Libkin, and Filip Murlak. *Foundations of Data Exchange*. Cambridge University Press, 2014. URL: http://www.cambridge.org/9781107016163.
4 Marcelo Arenas and Leonid Libkin. XML data exchange: Consistency and query answering. *J. ACM*, 55(2):7:1–7:72, 2008. doi:10.1145/1346330.1346332.

5 Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. Ontology-mediated query answering over temporal data: A survey (invited talk). In Sven Schewe, Thomas Schneider, and Jef Wijsen, editors, *24th International Symposium on Temporal Representation and Reasoning, TIME 2017, October 16-18, 2017, Mons, Belgium*, volume 90 of *LIPIcs*, pages 1:1–1:37. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. `doi:10.4230/LIPIcs.TIME.2017.1`.

6 Pablo Barceló, Jorge Pérez, and Juan L. Reutter. Schema mappings and data exchange for graph databases. In Wang-Chiew Tan, Giovanna Guerrini, Barbara Catania, and Anastasios Gounaris, editors, *Joint 2013 EDBT/ICDT Conferences, ICDT '13 Proceedings, Genoa, Italy, March 18-22, 2013*, pages 189–200. ACM, 2013. `doi:10.1145/2448496.2448520`.

7 Iovka Boneva, Jose Lozano, and Slawomir Staworko. Relational to RDF data exchange in presence of a shape expression schema. In Dan Olteanu and Barbara Poblete, editors, *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management, Cali, Colombia, May 21-25, 2018*, volume 2100 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018. URL: `http://ceur-ws.org/Vol-2100/paper6.pdf`.

8 Jan Chomicki and David Toman. Temporal databases. In Michael Fisher, Dov M. Gabbay, and Lluís Vila, editors, *Handbook of Temporal Reasoning in Artificial Intelligence*, volume 1 of *Foundations of Artificial Intelligence*, pages 429–467. Elsevier, 2005. `doi:10.1016/S1574-6526(05)80016-1`.

9 Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005. `doi:10.1016/j.tcs.2004.10.033`.

10 Ronald Fagin, Phokion G. Kolaitis, Lucian Popa, and Wang Chiew Tan. Composing schema mappings: Second-order dependencies to the rescue. *ACM Trans. Database Syst.*, 30(4):994–1055, 2005. `doi:10.1145/1114244.1114249`.

11 Ladan Golshanara and Jan Chomicki. Temporal data exchange. *Inf. Syst.*, 87, 2020. `doi:10.1016/j.is.2019.07.004`.

12 Gösta Grahne and Adrian Onet. Anatomy of the chase. *Fundam. Inform.*, 157(3):221–270, 2018. `doi:10.3233/FI-2018-1627`.

13 Abdullah Uz Tansel, James Clifford, Shashi K. Gadia, Sushil Jajodia, Arie Segev, and Richard T. Snodgrass, editors. *Temporal Databases: Theory, Design, and Implementation*. Benjamin/Cummings, 1993.

14 Balder ten Cate and Phokion G. Kolaitis. Structural characterizations of schema-mapping languages. *Commun. ACM*, 53(1):101–110, 2010. `doi:10.1145/1629175.1629201`.

15 David Toman. Point vs. interval-based query languages for temporal databases. In Richard Hull, editor, *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, 1996, Montreal, Canada*, pages 58–67. ACM Press, 1996. `doi:10.1145/237661.237676`.