On Query Optimization in a Temporal SPC Algebra

Jef Wijsen and Alexis Bès

December 12, 2001

TR-67

A TIMECENTER Technical Report

Title	On Query Optimization in a Temporal SPC Algebra
Author(s)	Jef Wijsen and Alexis Bès
Publication History	June 2001. A preliminary version was presented at the Eighth International Symposium on Temporal Representation and Reasoning [14]. December 2001. A TIMECENTER Technical Report.

Abstract

Tuples of a temporal relation are equipped with a valid time period. A simple extension of the SPC (Selection-Projection-Cross product) algebra for temporal relations is defined, which conforms to primitives in existing temporal query languages. In particular, temporal projection involves coalescing of time intervals, which results in non-monotonic queries. Also the "select-from-where" normal form is no longer available in this temporal extension. In view of these temporal peculiarities, it is natural and significant to ask whether query optimization techniques for the SPC algebra still apply in the temporal case. To this extent, we provide a temporal extension of the classical tableau formalism, and show its use and limits for temporal query optimization.

1 Introduction

Several algebras for temporal relational databases have been published since the early eighties [3, 8]. In more recent years, there has been a growing interest in *practical* languages for temporal databases. This has led to a number of temporal extensions of SQL [7, 9, 11]. The *theoretical* foundations of these temporal query languages—including issues like expressiveness, complexity, computability, genericity, and optimization—have not been systematically explored. In this paper, we study query equivalence and optimization in a temporal SPC algebra, containing temporal extensions of selection, projection, and cross product.

Consider the temporal relation C of celebrities shown in Figure 1. The first row means that Piaf lived in Paris from 15 to 63 (the twentieth century is assumed). FR (from) and TO are special timestamping attributes. To answer the query:

Which celebrities lived during the entire time period 45–70?

we first perform a temporal projection on the first column, which results in the table $\pi_1(C)$ of Figure 1. Note that the two rows about Brel that existed in C, have been "coalesced" into a single one. Next we select the rows that include the interval $[45, 70]$. This selection is denoted $\sigma_{[45,70]}(\pi_1(C))$ and retrieves only the row about Brel. This simple temporal query already shows two fundamental theoretical divergences from the non-temporal SPC algebra:

- The query $\sigma_{[45,70]}(\pi_1(C))$ is non-monotonic in strict sense: $C \subseteq C'$ does not imply $\sigma_{[45,70]}(\pi_1(C)) \subseteq \sigma_{[45,70]}(\pi_1(C'))$. For example, for $C' = C \cup \{\langle\text{Brel,London,79,90}\rangle\}$, the answer $\sigma_{[45,70]}(\pi_1(C'))$ is the singleton temporal relation $\{\langle\text{Brel,29,90}\rangle\}$, which is not a superset of the temporal relation $\sigma_{[45,70]}(\pi_1(C))$ of Figure 1. Adding a temporal tuple to C results in an extension of the time interval in the answer set from $[29, 78]$ to $[29, 90]$.

  Let $\mathrm{SPC}^{\leq}$ denote the non-temporal SPC algebra extended with inequality selections $\sigma_{A\theta a}(\cdot)$ and $\sigma_{A\theta B}(\cdot)$ where $\theta \in \{=, \neq, <, >, \leq, \geq\}$. Is $\sigma_{[45,70]}(\pi_1(C))$ expressible in $\mathrm{SPC}^{\leq}$?[1] The answer is "no," since all $\mathrm{SPC}^{\leq}$ queries are monotonic, while strictly speaking, $\sigma_{[45,70]}(\pi_1(C))$ is non-monotonic. So although coalescing is known to be first order expressible [2], it turns out to be non-monotonic.

- In the subquery $\sigma_{[45,70]}(\pi_1(C))$, we cannot push the selection through the inner projection: The result of the query $\sigma_{[45,70]}(C)$ is empty. It is correct to conclude that the normal form $\pi_{\vec{j}}(\sigma_F(R_1 \times \ldots \times R_k))$ [1, p. 55] that exists in the non-temporal SPC algebra, is not always attainable in the temporal case.

Figure 1 also illustrates the temporal cross product $\times$. Joining two tuples involves concatenating the non-temporal values and intersecting the time intervals. The algebra containing the temporal versions of selection, projection, and cross product will be called $\mathrm{SPC}^{\mathrm{time}}$. Query equivalence and optimization in the $\mathrm{SPC}^{\mathrm{time}}$ algebra is at the center of this paper. For example, consider the query:

$$Q(C) = \pi_1(\sigma_{1=2}(\overbrace{\pi_1(\sigma_{[14,18]}(\sigma_{2=\mathrm{Paris}}(C)))}^{Q_1} \times \overbrace{\pi_1(\sigma_{[40,45]}(\sigma_{2=\mathrm{Paris}}(C)))}^{Q_2})) \ .$$

The subqueries Q_1 and Q_2 retrieve names of celebrities living in Paris during the first (14–18) and the second (40–45) world war respectively. So the overall query $\pi_1(\sigma_{1=2}(Q_1 \times Q_2))$ asks for names of celebrities staying in Paris

[1] To make this question meaningful, we implicitly assume that the columns FR and TO are addressed by their positions in C, i.e., 3 for FR and 4 for TO.

C	1	2	FR	TO
	Piaf	Paris	15	63
	Brel	Brussels	29	52
	Brel	Paris	53	78

$\pi_1(C)$	1	FR	TO
	Piaf	15	63
	Brel	29	78

$\sigma_{[45,70]}(\pi_1(C))$	1	FR	TO
	Brel	29	78

$\pi_1(C) \times \pi_1(C)$	1	2	FR	TO
	Piaf	Piaf	15	63
	Piaf	Brel	29	63
	Brel	Piaf	29	63
	Brel	Brel	29	78

Figure 1: Temporal projection and selection.

during both world wars. Under the given temporal semantics, celebrities in the answer set did not leave Paris during the interbellum period. So the query Q happens to be equivalent to the simpler query:

$$\pi_1(\sigma_{[14,45]}(\sigma_{2=\mathrm{Paris}}(C))) \quad .$$

Which techniques can be used for query optimization in the $\mathrm{SPC}^{\mathrm{time}}$ algebra? One may think of the following approach: First translate the query into an equivalent $\mathrm{SPC}^{\leq}$ query, and then optimize this $\mathrm{SPC}^{\leq}$ query. This approach may not be appropriate for two reasons, however. Firstly, the approach clearly fails for $\mathrm{SPC}^{\mathrm{time}}$ queries that are not expressible in $\mathrm{SPC}^{\leq}$. Such queries exist, as shown above. Secondly, the approach may be needlessly expensive for $\mathrm{SPC}^{\mathrm{time}}$ queries that can be translated into $\mathrm{SPC}^{\leq}$. One should know that query optimization in $\mathrm{SPC}^{\leq}$ is considerably more difficult than in SPC [6, 12]: Query optimization in SPC is based on an elegant Homomorphism Theorem [1], which unfortunately fails when inequalities are added. We may hope, however, that query optimization in $\mathrm{SPC}^{\mathrm{time}}$ is easier than in $\mathrm{SPC}^{\leq}$, because it can profit from the syntactic restrictions on the ways in which time points can be compared for inequality. So our approach to query optimization in $\mathrm{SPC}^{\mathrm{time}}$ is not to simply "de-temporalize" queries and then apply non-temporal optimization techniques. Instead, we are going to "temporalize" the tableau formalism [1] and investigate its use for temporal query optimization. In this way, we want to take profit of temporal semantics.

Several authors, for example in [11], advocate that timestamping by time points is conceptually cleaner than timestamping by time intervals. We agree that there are good arguments for using point-based timestamping *at the conceptual level*. However, even if point-based timestamping is used at the conceptual level, efficiency considerations may dictate the use of time intervals or constraints at the storage level. Hence, investigating query optimization for interval-based timestamping is significant.

The organization of the paper is as follows. Sections 2 and 3 formalize the constructs of temporal relation and $\mathrm{SPC}^{\mathrm{time}}$ algebra. Sections 4 through 8 provide the tools that will be used for query optimization in the $\mathrm{SPC}^{\mathrm{time}}$ algebra. Section 4 introduces the construct of temporal tableau query (ttq). The semantics of ttqs relies on a property called *domain independence*, which is similar in nature to domain independence in the relational calculus. Domain independence of ttqs is decidable, as shown in Section 5. Section 6 concerns the composition of ttqs. *Dichotomic* temporal tableau queries, introduced in Section 7, are a syntactically restricted class of ttqs with nice properties. In particular, query containment for dichotomic ttqs can be decided by a small temporal extension of the homomorphism technique, as shown in Section 8. Finally, Sections 9 and 10 discuss the transformation of $\mathrm{SPC}^{\mathrm{time}}$ queries into equivalent ttqs, which can then be simplified by using the tools introduced in earlier sections. Section 11 relates the main results to existing work.

2 Temporal relation

This section introduces temporal relations and related constructs.

Definition 1 We use the set $\mathbb{Z}$ of integer numbers to denote time. We define $\mathbb{Z}^{2,\leq} := \{(p,q) \mid p,q \in \mathbb{Z} \text{ and } p \leq q\}$. We define for each $p,q \in \mathbb{Z}$, the *interval* $[p,q] := \{x \in \mathbb{Z} \mid p \leq x \leq q\}$. The empty interval is denoted $\{\}$. The set of all intervals is denoted $\mathbb{I}$. Two intervals are *unifiable* if their set union is again an interval. □

Lemma 1 *The set* $\mathbb{I}$*, ordered by set inclusion, is a lattice satisfying:*

Infimum. $[p,q] \wedge [p',q'] = [p,q] \cap [p',q']$.

Supremum. *If $p \leq q$ and $p' \leq q'$ then $[p, q']$ and $[p', q]$ are unifiable and $[p, q] \vee [p', q'] = [p, q'] \cup [p', q]$.*

PROOF. The proof of the *infimum* part is straightforward. For the *supremum* part, assume $p \leq q$ and $p' \leq q'$. Further assume $p \leq p'$ without loss of generality. $[p, q']$ and $[p', q]$ are unifiable, or else $p' > q'$, a contradiction. $[p, q], [p', q'] \subseteq [p, q'] \cup [p', q]$ follows immediately. Since $[p, q] \vee [p', q']$ must necessarily include $[p, q']$ and $[p', q]$, it follows that $[p, q'] \cup [p', q]$ constitutes a least upper bound. This concludes the proof. □

Intervals of $\mathbb{I}$ are used to timestamp tuples; the timestamping attribute is denoted $\mathbf{T}$. A temporal relation is a coalesced set of temporal tuples with non-empty timestamps. For example, the temporal relations I, J both of arity 2 ($\mathbf{T}$ does not add to the arity):

I	1	2	$\mathbf{T}$
	a	b	$[1, 3]$
	a	b	$[5, 8]$
	a	c	$[2, 4]$

and

J	1	2	$\mathbf{T}$
	a	b	$[0, 9]$
	a	c	$[2, 5]$

.

We write $\langle a, b, [1, 3]\rangle \sqsubseteq J$ because the temporal tuple $\langle a, b, [1, 3]\rangle$ is "contained" in the first tuple of J. The relationship $\sqsubseteq$ between a temporal tuple and a temporal relation is extended in a natural way to a relationship $\sqsubseteq$ among temporal relations. In the above example, $I \sqsubseteq J$.

Definition 2 We assume a countably infinite set $\mathbf{dom}$ of *constants*. Let $n \in \mathbb{N}$. A *temporal tuple* of arity n is an element of $\mathbf{dom}^n \times \mathbb{I}$. If t is a temporal tuple of arity n, then the i^{th} coordinate of t ($i \in \{1, \ldots, n\}$) is denoted $t(i)$, and the $(n+1)^{\text{th}}$ coordinate is denoted $t(\mathbf{T})$. $t(\mathbf{T})$ is also called the *timestamp* of t. Two tuples t and s of arity n are *value-equal*, denoted $t \asymp s$, iff $t(i) = s(i)$ for each $i \in \{1, \ldots, n\}$.

Let t be a temporal tuple and T, S sets of temporal tuples, all of the same arity. We write $t \sqsubseteq T$ iff either $t(\mathbf{T}) = \{\}$ or T contains a temporal tuple s with $s \asymp t$ and $t(\mathbf{T}) \subseteq s(\mathbf{T})$. We write $T \sqsubseteq S$ iff $t \sqsubseteq S$ for every temporal tuple $t \sqsubseteq T$.

A *temporal relation* of arity n is a finite set I of temporal tuples of arity n such that for each $t, s \in I$: (i) $t(\mathbf{T}) \neq \{\}$, and (ii) if $t(\mathbf{T})$ and $s(\mathbf{T})$ are unifiable and $t \asymp s$, then $t = s$.

Let I be a temporal relation of arity n. We define:

$$\begin{aligned} aidom(I) &:= \bigcup\{t(\mathbf{T}) \mid t \in I\}, \text{the active interval-domain of } I\text{; and} \\ adom(I) &:= \{t(i) \mid t \in I, i \in \{1, \ldots, n\}\}, \text{the active domain of } I. \end{aligned}$$

□

We now define the coalescing operator $\lceil \cdot \rceil$ which takes as its argument a set S of temporal tuples, all of the same arity. The operator turns S into a temporal relation by removing tuples with empty timestamp and by merging value-equal temporal tuples with unifiable timestamps..

Definition 3 Let S be a finite set of temporal tuples, all of the same arity. We write $\lceil S \rceil$ for the smallest (w.r.t. $\sqsubseteq$) temporal relation satisfying: if $t \in S$ then $t \sqsubseteq \lceil S \rceil$. S is called *coalesced* iff $S = \lceil S \rceil$. □

3 A temporal SPC algebra

We introduce a basic temporal extension of the SPC algebra; examples were already given in Section 1. The operators conform to common primitives in existing temporal query languages [9, 10]. A selection of the form $\sigma_{[p,q]}(I)$ retrieves each tuple of I whose timestamp includes the interval $[p, q]$. A projection automatically performs coalescing on the result. Joining two tuples involves concatenating the values for the non-temporal attributes and intersecting the timestamps.

Definition 4 Let I and J be temporal relations of arity n and m respectively. Let $i, j, j_1, j_2, \ldots, j_k \in \{1, \ldots, n\}$ ($k \geq 0$), $a \in \mathbf{dom}$, and $(p, q) \in \mathbb{Z}^{2,\leq}$.

Selection.

- $\sigma_{i=j}(I) := \{t \in I \mid t(i) = t(j)\}$,
- $\sigma_{i=a}(I) := \{t \in I \mid t(i) = a\}$, and
- $\sigma_{[p,q]}(I) := \{t \in I \mid t(\mathbf{T}) \supseteq [p,q]\}$.

Projection.

- $\pi_{j_1,\ldots,j_k}(I) := \lceil\{\langle t(j_1),\ldots,t(j_k),t(\mathbf{T})\rangle \mid t \in I\}\rceil$.

Cross product.

- $I \times J := \lceil\{\langle t(1),\ldots,t(n),s(1),\ldots,s(m),t(\mathbf{T}) \cap s(\mathbf{T})\rangle \mid t \in I, s \in J\}\rceil$.

□

The coalescing operator in the definition of cross product serves to eliminate temporal tuples with empty timestamp.

Definition 5 For every $n \in \mathbb{N}$, we assume the existence of denumerably many *relation variables* $R, R_1, R_2, \ldots$ of arity n. $\mathrm{SPC}^{\mathrm{time}}$ queries and their associated arities are recursively defined as follows:

Base. Every relation variable of arity n is an $\mathrm{SPC}^{\mathrm{time}}$ query of arity n.

Select. If Q is an $\mathrm{SPC}^{\mathrm{time}}$ query of arity n, and $i, j \in \{1,\ldots,n\}$, $a \in \mathbf{dom}$, and $(p,q) \in \mathbb{Z}^{2,\leq}$, then $\sigma_{i=j}(Q)$, $\sigma_{i=a}(Q)$, and $\sigma_{[p,q]}(Q)$ are $\mathrm{SPC}^{\mathrm{time}}$ queries of arity n.

Project. If Q is an $\mathrm{SPC}^{\mathrm{time}}$ query of arity n and $j_1,\ldots,j_k \in \{1,\ldots,n\}$, then $\pi_{j_1,\ldots,j_k}(Q)$ is an $\mathrm{SPC}^{\mathrm{time}}$ query of arity k ($k \geq 0$).

Cross product. If Q_1 and Q_2 are $\mathrm{SPC}^{\mathrm{time}}$ queries of arities n_1 and n_2 respectively, then $Q_1 \times Q_2$ is an $\mathrm{SPC}^{\mathrm{time}}$ query of arity $n_1 + n_2$.

We write $Q(R_1,\ldots,R_l)$, where $R_1,\ldots,R_l$ are distinct relation variables, to indicate that Q is a query containing the relation variables $R_1,\ldots,R_l$. The *semantics* of $Q(R_1,\ldots,R_l)$ is relative to an interpretation function that maps each R_i to a temporal relation of the same arity as R_i ($i \in \{1,\ldots,l\}$). This *semantics* is defined in the natural manner (not elaborated here). In the remainder of this paper, we focus on $\mathrm{SPC}^{\mathrm{time}}$ queries $Q(R)$ involving a single relation variable.

Let $Q_1(R), Q_2(R)$ be queries of the same arity. We write $Q_1 \sqsubseteq Q_2$ iff $Q_1(I) \sqsubseteq Q_2(I)$ for each temporal relation I of the same arity as R. Q_1 and Q_2 are *equivalent*, denoted $Q_1 \equiv Q_2$, iff $Q_1 \sqsubseteq Q_2$ and $Q_2 \sqsubseteq Q_1$. An $\mathrm{SPC}^{\mathrm{time}}$ query $Q(R)$ is *unsatisfiable* iff $Q(I) = \{\}$ for each temporal relation I of the same arity as R. □

It can be easily proved that unsatisfiability arises if two constants are required to be equal, as in $\sigma_{1=a}(\sigma_{1=b}(R))$. Henceforth, we will assume that all $\mathrm{SPC}^{\mathrm{time}}$ queries considered are satisfiable.

Importantly, as argued in Section 1, we cannot push the selection through the projection in the $\mathrm{SPC}^{\mathrm{time}}$ query $\sigma_{[p,q]}(\pi_{\vec{j}}(Q))$. Nor can we push the cross product through the selections in the $\mathrm{SPC}^{\mathrm{time}}$ query $\sigma_{[p,q]}(Q_1) \times \sigma_{[r,s]}(Q_2)$. As a consequence, the normal form for SPC algebra expressions [1, p. 55] does not apply to $\mathrm{SPC}^{\mathrm{time}}$ queries.

4 Temporal tableau query

The notion of temporal tableau is defined exactly as was the notion of temporal relation, except that both variables and constants may occur. Importantly, only two temporal variables are introduced, and their usage is syntactically restricted: $\mathbf{f}$ (from) can only occur as the left coordinate of an interval, and $\mathbf{t}$ (to) only as the right coordinate. No interval can contain non-temporal variables. A temporal tableau query consists of a temporal tableau followed by a summary temporal tuple with timestamp $[\mathbf{f},\mathbf{t}]$.

Definition 6 We assume a set $\mathbf{var}$ of *non-temporal variables*. We assume two *temporal variables* $\mathbf{f}$ and $\mathbf{t}$ not in $\mathbf{var}$. We define:

$$\mathbb{V} := \{[p,q] \mid (p,q) \in \mathbb{Z}^{2,\leq}\} \cup \{[\mathbf{f},q] \mid q \in \mathbb{Z}\} \cup \{[p,\mathbf{t}] \mid p \in \mathbb{Z}\} \cup \{[\mathbf{f},\mathbf{t}]\} \ .^2$$

A *tableau tuple* of arity n ($n \geq 0$) is an element of $(\mathbf{var} \cup \mathbf{dom})^n \times \mathbb{V}$. If t is a tableau tuple of arity n, then the i^{th} coordinate of t ($i \in \{1,\ldots,n\}$) is denoted $t(i)$, and the $(n+1)^{\text{th}}$ coordinate is denoted $t(\mathbf{T})$.

A *temporal tableau* of arity n is a finite set of tableau tuples of arity n. A *temporal tableau query* (ttq, plural: ttqs) is a pair (T,t) where T is a temporal tableau and t is a tableau tuple (called *summary*) such that $t(\mathbf{T}) = [\mathbf{f},\mathbf{t}]$ and each variable in t also occurs in T.

Let $\tau = (T,t)$ be a ttq. The *active point-domain* of τ, denoted $apdom(\tau)$, is the smallest set of time points containing all $p \in \mathbb{Z}$ that appear as the first or the second coordinate of $t(\mathbf{T})$ for some tableau tuple t of T.

A non-temporal variable that occurs in τ is called *free* (w.r.t. τ) if it occurs in t; otherwise it is *bounded*. □

We now define the semantics of ttqs. Let $\tau = (T,t)$ be a ttq and I a temporal relation of the same arity as T. The idea is to proceed along the lines of non-temporal tableaux [1, p. 43]: Consider a valuation ν for the variables in τ; if $\nu(T)$ is contained in I, then $\nu(t)$ belongs to the query answer $\tau(I)$.[3]

Example 1 Consider the ttq $\tau = (T,t)$ where $T = \{t_1, t_2\}$:

τ	1	2	$\mathbf{T}$	
	Brel	x	$[40,\mathbf{t}]$	(t_1)
	Brel	x	$[\mathbf{f},45]$	(t_2)
	x		$[\mathbf{f},\mathbf{t}]$	(t)

The valuation $\nu = \{(x, \text{Brussels}), (\mathbf{f}, 29), (\mathbf{t}, 52)\}$ results in:

$\lceil \nu(T) \rceil$	1	2	$\mathbf{T}$
	Brel	Brussels	$[29,52]$

Consider the temporal relation C of Figure 1. Since $\lceil \nu(T) \rceil \sqsubseteq C$, we conclude that $\nu(t) = \langle \text{Brussels}, [29,52] \rangle$ is in the answer set $\tau(C)$. It can be verified that the query τ renders the city in which Brel stayed during the period 40–45.

Some caution is in order when fixing the domain of interpretation for the variables occurring in a ttq, as shown by the next example.

Example 2

τ	1	$\mathbf{T}$
	x	$[\mathbf{f},4]$
	x	$[6,\mathbf{t}]$
	a	$[2,8]$
	x	$[\mathbf{f},\mathbf{t}]$

and

I	1	$\mathbf{T}$
	a	$[2,8]$

Consider a valuation ν for the variables in τ with $\nu(\mathbf{f}) = \nu(\mathbf{t}) = 5$. We obtain:

$\lceil \nu(T) \rceil$	1	$\mathbf{T}$
	a	$[2,8]$

Then $\lceil \nu(T) \rceil \sqsubseteq I$ independent of $\nu(x)$. That is, the non-temporal constants appearing in the answer set $\tau(I)$ are not restricted to constants appearing in I, which is unnatural.

Obviously, if I is a temporal relation and S is set of temporal tuples of the same arity as I, then $\lceil S \rceil \sqsubseteq I$ if and only if $S \sqsubseteq I$. We are now ready to formalize the semantics of a ttq.

[2] If $[p,q]$ occurs in a temporal tableau, it is considered as an element of $\mathbb{V}$ (Definition 6), i.e., as an ordered pair of time points. On the other hand, if $[p,q]$ occurs in a temporal relation, it is considered as an element of $\mathbb{I}$, i.e., as a convex set of time points. This double use does not result into any confusion.

[3] A *valuation* is a mapping from variables to constants extended to be the identity on constants. A *substitution* is a mapping from variables to variables and constants, extended to be the identity on constants. [1]

Definition 7 Let $\tau = (T, t)$ be a ttq, and I a temporal relation of the same arity as T. Let $\mathbf{d} \subseteq \mathbf{dom}$ and $\mathbf{p} \subseteq \mathbb{Z}$. We write $I \vDash^{\tau}_{\mathbf{d},\mathbf{p}} s$ iff there exists a valuation ν for the variables occurring in τ such that:

1. for every $v \in \mathbf{var}$, $\nu(v) \in adom(I) \cup \mathbf{d}$;
2. $(\nu(\mathbf{f}), \nu(\mathbf{t})) \in \mathbb{Z}^{2,\leq}$ and $[\nu(\mathbf{f}), \nu(\mathbf{t})] \subseteq aidom(I) \cup \mathbf{p}$;
3. $\lceil \nu(T) \rceil \sqsubseteq I$;[4] and
4. $\nu(t) = s$.

τ is *domain independent* iff for every temporal relation I, for every pair $\mathbf{d}, \mathbf{d}' \subseteq \mathbf{dom}$, for every pair $\mathbf{p}, \mathbf{p}' \subseteq \mathbb{Z}$, for every temporal tuple s, $I \vDash^{\tau}_{\mathbf{d},\mathbf{p}} s$ iff $I \vDash^{\tau}_{\mathbf{d}',\mathbf{p}'} s$. If τ is domain independent, we write $I \vDash^{\tau} s$ instead of $I \vDash^{\tau}_{\{\},\{\}} s$. $\square$

Obviously, the ttq τ of Example 2 is not domain independent. From now on, ttqs that are not domain independent will be considered erroneous. The output of a ttq can now be defined.

Definition 8 Let $\tau = (T, t)$ be a domain independent ttq, and I a temporal relation of the same arity as T. The *output* of τ on input I, denoted $\tau(I)$, is the temporal relation:

$$\tau(I) := \lceil \{ s \mid I \vDash^{\tau} s \} \rceil \ .$$

The relations $\sqsubseteq$ and $\equiv$ on ttqs are defined as in Definition 4. $\square$

Note incidentally that $s \in \tau(I)$ or $s \sqsubseteq \tau(I)$ does not imply $I \vDash^{\tau} s$, which is illustrated by Example 3 and gives rise to Definition 9. Inversely, $I \vDash^{\tau} s$ implies $s \sqsubseteq \tau(I)$.

Example 3 For the temporal relation C of Figure 1:

τ	1	2	$\mathbf{T}$
	x	v	$[\mathbf{f}, \mathbf{t}]$
	x		$[\mathbf{f}, \mathbf{t}]$

yields

$\tau(C)$	1	$\mathbf{T}$
	Piaf	$[15, 63]$
	Brel	$[29, 78]$

.

The valuation $\nu_1 = \{(x, \text{Brel}), (v, \text{Brussels}), (\mathbf{f}, 29), (\mathbf{t}, 52)\}$ shows $C \vDash^{\tau} \langle \text{Brel}, [29, 52] \rangle$, and the valuation $\nu_2 = \{(x, \text{Brel}), (v, \text{Paris}), (\mathbf{f}, 53), (\mathbf{t}, 78)\}$ shows $C \vDash^{\tau} \langle \text{Brel}, [53, 78] \rangle$. It is correct to conclude $\langle \text{Brel}, [29, 78] \rangle \in \tau(C)$. However, $C \nvDash^{\tau} \langle \text{Brel}, [29, 78] \rangle$. Note incidentally that τ is equivalent to the $\mathrm{SPC}^{\mathrm{time}}$ query $\pi_1(R)$, where R is a relation variable of arity 2.

Definition 9 Let $\tau = (T, t)$ be a (domain independent) ttq. We say that τ is *coalescing-free* (*cfree* for short) iff for every temporal relation I of the same arity as T, if $s \sqsubseteq \tau(I)$ then $I \vDash^{\tau} s$. $\square$

So the ttq of Example 3 is not cfree. To conclude this section, we show how a tableau can be viewed as a temporal relation, and we provide an operator $[\![\cdot]\!]$ which merges tableau tuples in a ttq.

Definition 10 Let $\tau = (T, t)$ be a ttq, and $(p, q) \in \mathbb{Z}^{2,\leq}$. Let T' be the temporal tableau obtained from T by substituting p and q for $\mathbf{f}$ and $\mathbf{t}$ respectively. Then $T_{\mathbf{f} \to p, \mathbf{t} \to q}$ denotes the temporal relation $\lceil T' \rceil$, where it is understood that distinct variables are interpreted as new distinct constants.[5] We write $T_{\mathbf{f}, \mathbf{t} \to p}$ as a shorthand for $T_{\mathbf{f} \to p, \mathbf{t} \to p}$.

We write $[\![\tau]\!]$ for the ttq obtained from τ by repeatedly executing one of the following modifications until no more changes can be made: For all distinct $s, s' \in T$ such that $s \asymp s'$,

1. If $s(\mathbf{T})$ and $s'(\mathbf{T})$ are unifiable intervals of $\mathbb{I}$ (say $s(\mathbf{T}) \cup s'(\mathbf{T}) = [p, q]$), then replace s and s' by the single tableau tuple $s_{\mathbf{T} \to [p,q]}$.[6]

[4] Instead of $\lceil \nu(T) \rceil$, we will often simply write $\nu(T)$, where the coalescing is implicitly understood.

[5] To be precise, one should introduce a one-to-one valuation ν from non-temporal variables to constants, mapping each variable x to a new, distinct constant, such that $\nu^{-1}(\nu(x)) = x$. By a little abuse of notation, we assume that distinct variables can be interpreted as new, distinct constants. Sometimes we make this assumption explicit and write $\tilde{x}$ for the variable x treated as a constant.

[6] If f is a function, then $f_{x \to a}$ denotes the function satisfying: $f_{x \to a}(x) = a$ and $f_{x \to a}(y) = f(y)$ for each y distinct from x. Temporal tuples of arity n are total functions with domain $\{1, \ldots, n\} \cup \{\mathbf{T}\}$. So $s_{\mathbf{T} \to [p,q]}$ is the temporal tuple u such that $u \asymp s$ and $u(\mathbf{T}) = [p, q]$.

2. If $s(\mathbf{T}) = [p, \mathbf{t}]$, $s'(\mathbf{T}) = [p', \mathbf{t}]$, and $p \leq p'$, then remove s' from T ($p, p' \in \mathbb{Z}$).

3. If $s(\mathbf{T}) = [\mathbf{f}, q]$ and $s'(\mathbf{T}) = [\mathbf{f}, q']$, and $q \leq q'$, then remove s from T ($q, q' \in \mathbb{Z}$).

4. If $s(\mathbf{T}) = [p, \mathbf{t}]$, $s'(\mathbf{T}) = [p', q']$, and $p \in [p', q'+1]$, then replace s by $s_{\mathbf{T} \to [p', \mathbf{t}]}$, while leaving s' unaffected ($p, p', q' \in \mathbb{Z}$).

5. If $s(\mathbf{T}) = [\mathbf{f}, q]$, $s'(\mathbf{T}) = [p', q']$, and $q \in [p'-1, q']$, then replace s by $s_{\mathbf{T} \to [\mathbf{f}, q']}$, while leaving s' unaffected ($p', q, q' \in \mathbb{Z}$).

A ttq is called *stretched* iff $[\![\tau]\!] = \tau$. □

Example 4 Let $\tau = (T, t)$ as indicated below. Then:

τ	1	2	$\mathbf{T}$
	x	u	$[\mathbf{f}, 3]$
	x	u	$[2, 6]$
	x	v	$[2, 6]$
	x	v	$[5, \mathbf{t}]$
	x		$[\mathbf{f}, \mathbf{t}]$

and

$[\![\tau]\!]$	1	2	$\mathbf{T}$
	x	u	$[\mathbf{f}, 6]$
	x	u	$[2, 6]$
	x	v	$[2, 6]$
	x	v	$[2, \mathbf{t}]$
	x		$[\mathbf{f}, \mathbf{t}]$

and

$T_{\mathbf{f} \to 1, \mathbf{t} \to 7}$	1	2	$\mathbf{T}$
	$\tilde{x}$	$\tilde{u}$	$[1, 6]$
	$\tilde{x}$	$\tilde{v}$	$[2, 7]$

.

Lemma 2 *For every ttq* τ, $\tau \equiv [\![\tau]\!]$.

PROOF. Straightforward. □

5 Testing domain independence

We show how to decide domain independence of ttqs.

Lemma 3 *A ttq* $\tau = (T, t)$ *is domain independent iff for each* $(p, q) \in \mathbb{Z}^{2,\leq}$, *(i)* $[p, q] \subseteq aidom(T_{\mathbf{f} \to p, \mathbf{t} \to q})$, *and (ii) every free non-temporal variable* x *of* τ *occurs in* $T_{\mathbf{f} \to p, \mathbf{t} \to q}$.

PROOF. *Only-if part.* Argumentation by contradiction. Suppose $[p, q] \not\subseteq aidom(T_{\mathbf{f} \to p, \mathbf{t} \to q})$. Let $r \in [p, q] \setminus aidom(T_{\mathbf{f} \to p, \mathbf{t} \to q})$. Then T contains no tableau tuple with timestamp $[\mathbf{f}, \mathbf{t}]$; furthermore, whenever T contains a tableau tuple with timestamp $[\mathbf{f}, q']$, then $q' < r$, and whenever T contains a tableau tuple with timestamp $[p', \mathbf{t}]$, then $p' > r$. Then $T_{\mathbf{f}, \mathbf{t} \to r}$ can be obtained from T by simply removing tableau tuples that contain $\mathbf{f}$ or $\mathbf{t}$. Then obviously, $T_{\mathbf{f} \to p, \mathbf{t} \to q} \models^{\mathcal{T}}_{\{\},\{r\}} t_{\mathbf{T} \to [r,r]}$, but $T_{\mathbf{f} \to p, \mathbf{t} \to q} \not\models^{\mathcal{T}}_{\{\},\{\}} t_{\mathbf{T} \to [r,r]}$, hence τ is not domain independent.

Next suppose that the free non-temporal variable x of τ does not occur in $T_{\mathbf{f} \to p, \mathbf{t} \to q}$. Let $a \in \mathbf{dom}$ be a constant not occurring in τ. Let t' be the tableau tuple obtained from t by substituting a for each occurrence of x in t. Then $T_{\mathbf{f} \to p, \mathbf{t} \to q} \models^{\mathcal{T}}_{\{a\},[p,q]} t'_{\mathbf{T} \to [p,q]}$, but $T_{\mathbf{f} \to p, \mathbf{t} \to q} \not\models^{\mathcal{T}}_{\{\},[p,q]} t'_{\mathbf{T} \to [p,q]}$, hence τ is not domain independent.
If part. Assume $\mathbf{d}, \mathbf{d}' \subseteq \mathbf{dom}$ and $\mathbf{p}, \mathbf{p}' \subseteq \mathbb{Z}$. Assume $I \models^{\mathcal{T}}_{\mathbf{d},\mathbf{p}} s$. Hence, there exists a valuation ν for the variables in τ such that:

1. $\nu(v) \in adom(I) \cup \mathbf{d}$ for every non-temporal variable v,

2. $(\nu(\mathbf{f}), \nu(\mathbf{t})) \in \mathbb{Z}^{2,\leq}$ and $[\nu(\mathbf{f}), \nu(\mathbf{t})] \subseteq aidom(I) \cup \mathbf{p}$,

3. $\nu(T) \sqsubseteq I$, and

4. $\nu(t) = s$.

We show that $adom(I) \neq \{\}$. Suppose on the contrary $adom(I) = \{\}$, hence $I = \{\}$. Since $[\nu(\mathbf{f}), \nu(\mathbf{t})] \subseteq aidom(T_{\mathbf{f} \to \nu(\mathbf{f}), \mathbf{t} \to \nu(\mathbf{t})})$ by the premise, it follows $T_{\mathbf{f} \to \nu(\mathbf{f}), \mathbf{t} \to \nu(\mathbf{t})} \neq \{\}$. But then $\nu(T) \not\sqsubseteq I$, a contradiction. We consider the valuation ν for each variable of τ in turn.

Temporal variables. By the premise, $[\nu(\mathbf{f}), \nu(\mathbf{t})] \subseteq aidom(T_{\mathbf{f} \to \nu(\mathbf{f}), \mathbf{t} \to \nu(\mathbf{t})})$. From $\nu(T) \sqsubseteq I$, it follows $aidom(\nu(T)) \subseteq aidom(I)$. Obviously, $aidom(\nu(T)) = aidom(T_{\mathbf{f} \to \nu(\mathbf{f}), \mathbf{t} \to \nu(\mathbf{t})})$. It follows $[\nu(\mathbf{f}), \nu(\mathbf{t})] \subseteq aidom(I)$. Hence, $[\nu(\mathbf{f}), \nu(\mathbf{t})] \subseteq aidom(I) \cup \mathbf{p}'$.

Bounded non-temporal variables. Let v be a bounded (w.r.t. τ) non-temporal variable. If $\nu(v) \in adom(I)$, then $\nu(v) \in adom(I) \cup \mathbf{d}'$. Next assume $\nu(v) \notin adom(I)$. Since $\nu(T) \sqsubseteq I$, v cannot occur in $T_{\mathbf{f} \to \nu(\mathbf{f}), \mathbf{t} \to \nu(\mathbf{t})}$. Then the value assigned to v by ν does not matter. More precisely, the above properties (1) through (4) still hold after replacing ν by $\nu_{v \to a}$, for any $a \in adom(I) \neq \{\}$.

Free non-temporal variables. Let v be a free non-temporal variable. By the premise, x occurs in $T_{\mathbf{f} \to \nu(\mathbf{f}), \mathbf{t} \to \nu(\mathbf{t})}$. Since $\nu(T) \sqsubseteq I$, $\nu(v) \in adom(I)$. Hence, $\nu(v) \in adom(I) \cup \mathbf{d}'$.

It follows $I \models^{T}_{\mathbf{d}', \mathbf{p}'} s$. It is correct to conclude that τ is domain independent. □

It can be easily verified that in order to test condition (i) of Lemma 3, there is no need to try all $(p, q) \in \mathbb{Z}^{2,\leq}$; it suffices to verify whether $[m, M] \subseteq aidom(T_{\mathbf{f} \to m, \mathbf{t} \to M})$, for some $m, M \in \mathbb{Z}$ with m strictly smaller than every time point in $apdom(\tau)$, and M strictly greater than every time point in $apdom(\tau)$.

For testing condition (ii) of Lemma 3, we consider each free non-temporal variable x of τ in turn. If x occurs in a tableau tuple t with $t(\mathbf{T}) = [\mathbf{f}, \mathbf{t}]$ or $t(\mathbf{T}) = [p, q]$ with $(p, q) \in \mathbb{Z}^{2,\leq}$, then the condition is satisfied for x. Otherwise, construct a set S of inequalities as follows:

- Add $\mathbf{f} \leq \mathbf{t}$ to S.
- For every tableau tuple $t \in T$ such that t contains x and $t(\mathbf{T}) = [\mathbf{f}, q']$, add $\mathbf{f} > q'$ to S ($q' \in \mathbb{Z}$).
- For every tableau tuple $t \in T$ such that t contains x and $t(\mathbf{T}) = [p', \mathbf{t}]$, add $\mathbf{t} < p'$ to S ($p' \in \mathbb{Z}$).

Obviously, the set S of inequalities has an integer solution in $\mathbf{f}$ and $\mathbf{t}$ if and only if condition (ii) of Lemma 3 is falsified for x.

A natural question is whether every ttq is equivalent to some $\mathrm{SPC}^{\mathrm{time}}$ query. The answer is "no," as shown by Example 5. The inverse question (i.e., is every $\mathrm{SPC}^{\mathrm{time}}$ query equivalent to some ttq?) will be thoroughly addressed from Section 6 on.

Example 5 Consider the domain independent ttq τ and the result $\tau(I)$:

τ:

1	**T**
a	$[\mathbf{f}, 2]$
a	$[8, \mathbf{t}]$
b	$[2, 8]$
	$[\mathbf{f}, \mathbf{t}]$

and I:

1	**T**
b	$[2, 8]$

and $\tau(I)$:

T
$[3, 7]$

.

It can be readily seen (by induction on the structure of Q) that for every $\mathrm{SPC}^{\mathrm{time}}$ query Q, if $s \in Q(I)$, then $s(\mathbf{T}) = [2, 8]$. It follows that τ is equivalent to no $\mathrm{SPC}^{\mathrm{time}}$ query.

6 Composition of temporal tableau queries

We now address the following natural question: Given two ttqs τ and σ, is there a single ttq that is equivalent to τ followed by σ?

Definition 11 Let $\tau = (T, t)$ and $\sigma = (S, s)$ be two ttqs, where t and S have the same arity. We write $\sigma \circ \tau$ for the query satisfying for every temporal relation I of the same arity as T, $(\sigma \circ \tau)(I) = \sigma(\tau(I))$. □

Clearly, if $\sigma \circ \tau$ is unsatisfiable, i.e., $(\sigma \circ \tau)(I) = \{\}$ for each temporal relation I, then $\sigma \circ \tau$ is equivalent to no single ttq, as each ttq is obviously satisfiable. Unfortunately, even if $\sigma \circ \tau$ is satisfiable, it may not correspond to a single ttq:

Lemma 4 *There exist ttqs τ and σ such that $\sigma \circ \tau$ is satisfiable but equivalent to no ttq.*

PROOF. Let τ, σ be ttqs, and I, J_k (for each $k \in \mathbb{Z}$) temporal relations, as follows:

$$
\begin{array}{c|cc}
\tau & 1 & \mathbf{T} \\ \hline
 & v & [\mathbf{f}, \mathbf{t}] \\ \hline
 & & [\mathbf{f}, \mathbf{t}]
\end{array}
\quad \text{and} \quad
\begin{array}{c|c}
\sigma & \mathbf{T} \\ \hline
 & [\mathbf{f}, 0] \\
 & [0, \mathbf{t}] \\ \hline
 & [\mathbf{f}, \mathbf{t}]
\end{array}
\quad \text{and} \quad
\begin{array}{c|cc}
J_k & 1 & \mathbf{T} \\ \hline
 & 0 & [0,0] \\
 & 1 & [1,1] \\
 & -1 & [-1,-1] \\
 & & \dots \\
 & k & [k,k] \\
 & -k & [-k,-k]
\end{array}
\quad \text{and} \quad
\begin{array}{c|cc}
I & 1 & \mathbf{T} \\ \hline
 & a & [0,0] \\
 & a & [2,2]
\end{array}
\ .
$$

Obviously, $\sigma(\tau(J_k)) = \tau(J_k) = \{\langle[-|k|, |k|]\rangle\}$, where $|\cdot|$ denotes the absolute value. Furthermore, $\sigma(\tau(I)) = \{\langle[0,0]\rangle\}$. Assume that the ttq $\rho = (R, \langle[\mathbf{f}, \mathbf{t}]\rangle)$ is equivalent to $\sigma \circ \tau$. Suppose that the tableau R, of arity 1, contains a tableau tuple r with $r(\mathbf{T}) = [p, \mathbf{t}]$, for some $p \in \mathbb{Z}$. Then for no valuation ν, $\nu_{\mathbf{t} \to p+1}(R) \sqsubseteq J_{p+1}$, hence $\langle[p+1, p+1]\rangle \not\in \rho(J_{p+1})$, a contradiction. Hence, R cannot contain a tableau tuple r with $r(\mathbf{T}) = [p, \mathbf{t}]$. Likewise, R cannot contain a tableau tuple r with $r(\mathbf{T}) = [\mathbf{f}, q]$, for some $q \in \mathbb{Z}$. Since $\mathbf{f}$ and $\mathbf{t}$ occur in R, R must contain a tableau tuple r with $r(\mathbf{T}) = [\mathbf{f}, \mathbf{t}]$. If R contains a tableau tuple r with $r(\mathbf{T}) = [p, q]$, then $p = q = 0$, or else for no valuation ν, $\nu_{\mathbf{f} \to 0, \mathbf{t} \to 0}(R) \sqsubseteq J_0$, hence $\langle[0,0]\rangle \not\in \rho(J_0)$, a contradiction. R cannot contain constants from $\mathbf{dom}$ for obvious reasons. Since $\rho \not\equiv \tau$, ρ must be equivalent to either:

$$
\begin{array}{c|cc}
\rho_1 & 1 & \mathbf{T} \\ \hline
 & u & [0,0] \\
 & v & [\mathbf{f}, \mathbf{t}] \\ \hline
 & & [\mathbf{f}, \mathbf{t}]
\end{array}
\quad \text{or} \quad
\begin{array}{c|cc}
\rho_2 & 1 & \mathbf{T} \\ \hline
 & v & [0,0] \\
 & v & [\mathbf{f}, \mathbf{t}] \\ \hline
 & & [\mathbf{f}, \mathbf{t}]
\end{array}
\ .
$$

Then $\rho(I) = \{\langle[0,0]\rangle, \langle[2,2]\rangle\} \neq \sigma(\tau(I))$. We conclude by contradiction that for no ρ, $\rho \equiv \sigma \circ \tau$. $\square$

It is natural to ask under which conditions $\sigma \circ \tau$ corresponds to a ttq. In the remainder of this section, we show how to construct a ttq equivalent to $\sigma \circ \tau$ when τ is cfree (cfree was defined in Definition 9). The construction uses composition of ttqs, which generalizes the composition of typed tableaux [1, p. 226]. We assume that the reader is familiar with the composition of typed tableaux; the composition of ttqs is slightly more complicated, as illustrated next. Let τ and σ be the ttqs:

$$
\begin{array}{c|cccc}
\tau & 1 & 2 & & \mathbf{T} \\ \hline
 & a & x & & [\mathbf{f}, 4] \\
 & x & v & & [1, \mathbf{t}] \\ \hline
 & a & x & x & [\mathbf{f}, \mathbf{t}]
\end{array}
\ (t)
\quad \text{and} \quad
\begin{array}{c|cccc}
\sigma & 1 & 2 & 3 & \mathbf{T} \\ \hline
 & w_1 & y_1 & y_2 & [\mathbf{f}, 4] \\
 & w_2 & w_1 & y_3 & [1, \mathbf{t}] \\ \hline
 & y_1 & y_2 & y_3 & [\mathbf{f}, \mathbf{t}]
\end{array}
\begin{array}{c}
\\ (s_1) \\ (s_2) \\ \\
\end{array}
\ .
$$

We want to construct the ttq $\tau \bullet \sigma$ corresponding to the composition of τ followed by σ. Proceeding along the lines of typed tableaux, one looks for a substitution mapping t to s_1, and another one mapping t to s_2. The difficulty is that there is no substitution that maps t to s_1; for such substitution to exist, we must have $w_1 = a$ and $y_1 = y_2$. Likewise, the existence of a substitution that maps t to s_2, requires $w_2 = a$ and $w_1 = y_3$. From $w_1 = a$ and $w_1 = y_3$, it follows $y_3 = a$. Applying these equalities to σ, we obtain a new ttq σ':

$$
\begin{array}{c|cccc}
\sigma' & 1 & 2 & 3 & \mathbf{T} \\ \hline
 & a & y_1 & y_1 & [\mathbf{f}, 4] \\
 & a & a & a & [1, \mathbf{t}] \\ \hline
 & y_1 & y_1 & a & [\mathbf{f}, \mathbf{t}]
\end{array}
\begin{array}{c}
\\ (s'_1) \\ (s'_2) \\ \\
\end{array}
\ .
$$

Now there exist substitutions from t to s'_1, and from t to s'_2.

Definition 12 Let $\tau = (T, t)$ and $\sigma = (S, s)$ be two ttqs, where t and S are both of arity n. Let $S = \{s_1, \dots, s_m\}$.

A *unifier* for t and σ is a substitution θ defined as follows. Compute $\equiv$, the equivalence relation on $\mathbf{var} \cup \mathbf{dom} \cup \{\mathbf{f}, \mathbf{t}\}$ defined as the reflexive, transitive closure of:

1. $s_i(j) \equiv a$ if $t(j) = a$ for some constant a (for each $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$);

2. $s_i(j) \equiv s_i(k)$ if $t(j) = t(k)$, for each $i \in \{1, \dots, m\}$ and $j, k \in \{1, \dots, n\}$.

The unifier of t and σ does not exist if two distinct constants are in the same equivalence class. Otherwise their unifier is the substitution θ such that:

1. if $x \equiv a$ for some constant a, $\theta(x) = a$;
2. otherwise $\theta(x) = x'$, where x' is the smallest (under a fixed ordering on $\mathbf{var}$) such that $x \equiv x'$.

$\square$

Definition 13 Let $\tau = (T, t)$ and $\sigma = (S, s)$ be two ttqs, where t and S are of the same arity. Let $S = \{s_1, \ldots, s_m\}$. Assume θ is a unifier for t and σ; if no such unifier exists, the composition is undefined. For each $i \in \{1, \ldots, m\}$, let θ_i be a substitution that maps t to $\theta(s_i)$ and maps each other variable of T to a new, distinct variable not used elsewhere in the construction. Then:

$$\tau \bullet \sigma := (\bigcup_{i=1}^{m} \theta_i(T), \theta(s)) \ ,$$

where it is understood that θ is extended to be the identity on constants. $\square$

One can easily verify that the unifier of t and σ is defined in Definition 13 if and only if $\sigma \circ \tau$ is satisfiable.

Example 6 For τ and σ as introduced at the beginning of this section:

$\tau \bullet \sigma$	1	2		**T**
	a	y_1		$[\mathbf{f}, 4]$
	y_1	v_1		$[1, 4]$
	a	a		$[1, 4]$
	a	v_2		$[1, \mathbf{t}]$
	y_1	y_1	a	$[\mathbf{f}, \mathbf{t}]$

Theorem 1 *Let $\tau = (T, t)$ and $\sigma = (S, s)$ be two ttqs, where t and S are of the same arity. If $\sigma \circ \tau$ is satisfiable and τ is cfree, then $\sigma \circ \tau \equiv \tau \bullet \sigma$.*

PROOF. Let $S = \{s_1, \ldots, s_m\}$, θ and θ_i be as in Definition 13 ($i \in \{1, \ldots, m\}$). $\sigma \circ \tau$ satisfiable implies that $\tau \bullet \sigma$ is defined. We first show $\sigma \circ \tau \sqsubseteq \tau \bullet \sigma$. Assume $\tau(I) \models^{\sigma} u$ for some temporal relation I of the same arity as T. It suffices to show $I \models^{\tau \bullet \sigma} u$. $\tau(I) \models^{\sigma} u$ implies the existence of a valuation ν such that $\nu(S) \sqsubseteq \tau(I)$ and $\nu(s) = u$. Since τ is cfree, $I \models^{\tau} \nu(s_i)$ for each $i \in \{1, \ldots, m\}$. Hence, for each $i \in \{1, \ldots, m\}$, there exists a valuation μ_i such that $\mu_i(T) \sqsubseteq I$ and $\mu_i(t) = \nu(s_i)$. Clearly, $\nu(s_i) = \nu(\theta(s_i))$ for (i) if $t(j) = t(k)$ then $\mu_i(t(j)) = \mu_i(t(k))$, hence $\nu(s_i(j)) = \nu(s_i(k))$, and (ii) if $t(j) = a$ then $\mu_i(t(j)) = \nu(s_i(j)) = a$. Let η be a valuation for the variables occurring in $\tau \bullet \sigma$ such that:

1. if v is a new, distinct variable introduced in $\theta_i(T)$, then $\eta(v) = \mu_i({\theta_i}^{-1}(v))$; and
2. if v occurs in S, then $\eta(v) = \nu(v)$.

Let $\tau \bullet \sigma$ be the tableau $(R, \theta(s))$. Then $\eta(R) \sqsubseteq I$, hence $I \models^{\tau \bullet \sigma} \eta(\theta(s)) = u$.

We next show $\tau \bullet \sigma \sqsubseteq \sigma \circ \tau$. Suppose $I \models^{\tau \bullet \sigma} u$. Hence, there exists a valuation ν such that $\nu(\theta_i(T)) \sqsubseteq I$ for each $i \in \{1, \ldots, m\}$, and $\nu(\theta(s)) = u$. Since $\nu(\theta_i(T)) \sqsubseteq I$, it follows $\nu(\theta_i(t)) = \nu(\theta(s_i)) \sqsubseteq \tau(I)$, for each $i \in \{1, \ldots, m\}$. Hence, $(\nu \circ \theta)(S) = \nu(\theta(S)) \sqsubseteq \tau(I)$. Consequently, $(\nu \circ \theta)(s) = \nu(\theta(s)) = u \sqsubseteq \sigma(\tau(I))$. This concludes the proof. $\square$

7 Dichotomic temporal tableau queries

Definition 9 provides no effective procedure for deciding whether a ttq is cfree. The family of dichotomic ttqs, as defined next, is a subset of the family of cfree ttqs; moreover, it can be easily verified whether a given ttq is dichotomic. Dichotomic ttqs possess some interesting properties that will be useful later on.

Definition 14 Let $\tau = (T, t)$ be a ttq. Define $\sim$, the equivalence relation on the bounded non-temporal variables of τ defined as the reflexive, transitive closure of:

$u \sim v$ iff some tableau tuple of T contains both u and v.

We define $\mathbf{from}(\tau)$ as the smallest set containing each bounded non-temporal variable u of τ for which there exists a temporal tableau tuple in T that contains both $\mathbf{f}$ and a variable $v \sim u$. The set $\mathbf{to}(\tau)$ of bounded non-temporal variables is defined similarly (replace $\mathbf{f}$ by $\mathbf{t}$).

A ttq τ is called *dichotomic* iff $\mathbf{from}(\tau) \cap \mathbf{to}(\tau) = \{\}$.[7] □

Example 7 Let τ be the following ttq:

τ	1	2	3	**T**
	x	a	u	$[\mathbf{f}, 4]$
	x	v	w_1	$[2, 4]$
	v	a	w_2	$[2, \mathbf{t}]$
	z_1	z_2	b	$[3, 6]$
	x			$[\mathbf{f}, \mathbf{t}]$

.

The bounded non-temporal variables are $\{u, v, w_1, w_2, z_1, z_2\}$. We have $v \sim w_1$, $v \sim w_2$, and $z_1 \sim z_2$. Hence, the equivalence classes of the relation $\sim$ are $\{u\}$, $\{v, w_1, w_2\}$, and $\{z_1, z_2\}$. $\mathbf{from}(\tau) = \{u\}$, $\mathbf{to}(\tau) = \{v, w_1, w_2\}$. The ttq τ is dichotomic.

Lemma 5 *Every dichotomic ttq is cfree.*

PROOF. Let $\tau = (T, t)$ be a ttq such that $\mathbf{from}(\tau) \cap \mathbf{to}(\tau) = \{\}$. Let I be a temporal relation such that $I \models^{\mathcal{T}} \langle \vec{a}, [p, q] \rangle$ and $I \models^{\mathcal{T}} \langle \vec{a}, [r, s] \rangle$, where $[p, q]$ and $[r, s]$ are unifiable. Hence, there exists a valuation ν_1 such that $\nu_1(T) \sqsubseteq I$ and $\nu_1(t) = \langle \vec{a}, [p, q] \rangle$, and there exists a valuation ν_2 such that $\nu_2(T) \sqsubseteq I$ and $\nu_2(t) = \langle \vec{a}, [r, s] \rangle$. It suffices to show that $I \models^{\mathcal{T}} \langle \vec{a}, [p, q] \cup [r, s] \rangle$. That is, it suffices to show that there exists a valuation μ such that $\mu(T) \sqsubseteq I$ and $\mu(t) = \langle \vec{a}, [p, q] \cup [r, s] \rangle$. This is obvious if $[p, q] \subseteq [r, s]$ or $[r, s] \subseteq [p, q]$. Next assume $[p, q] \not\subseteq [r, s]$ and $[r, s] \not\subseteq [p, q]$. Then $[p, q] \cup [r, s]$ equals either $[p, s]$ or $[r, q]$. Without loss of generality, assume $[p, q] \cup [r, s] = [p, s]$. Obviously, for every free non-temporal variable x of τ, $\nu_1(x) = \nu_2(x) = \mu(x)$. For every bounded non-temporal variable v of τ, define $\mu(v)$ as follows:

$$\mu(v) := \begin{cases} \nu_1(v) & \text{if } v \in \mathbf{from}(\tau) \\ \nu_2(v) & \text{if } v \in \mathbf{to}(\tau) \\ \nu_1(v) & \text{if } v \notin \mathbf{from}(\tau) \text{ and } v \notin \mathbf{to}(\tau) \end{cases}$$

Moreover, let $\mu(\mathbf{f}) = p$ and $\mu(\mathbf{t}) = s$. Since $\mu(t) = \langle \vec{a}, [p, s] \rangle$, it suffices to show that $\mu(T) \sqsubseteq I$. Consider each tableau tuple u of T in turn.

$u \equiv \langle \vec{y}, [\mathbf{f}, \mathbf{t}] \rangle$. Since τ is dichotomic, $\vec{y}$ contains no bounded non-temporal variables, thus $\mu(\vec{y}) = \nu_1(\vec{y}) = \nu_2(\vec{y})$. Since $\langle \nu_1(\vec{y}), [p, q] \rangle \sqsubseteq I$ and $\langle \nu_2(\vec{y}), [r, s] \rangle \sqsubseteq I$, it follows $\langle \mu(\vec{y}), [p, s] \rangle \sqsubseteq I$.

$u \equiv \langle \vec{y}, [\mathbf{f}, l] \rangle$. We have $\mu(\vec{y}) = \nu_1(\vec{y})$. Since $\langle \nu_1(\vec{y}), [p, l] \rangle \sqsubseteq I$, it follows $\langle \mu(\vec{y}), [p, l] \rangle \sqsubseteq I$.

$u \equiv \langle \vec{y}, [k, \mathbf{t}] \rangle$. Analogous to the previous case.

$u \equiv \langle \vec{y}, [k, l] \rangle$. Two cases can occur:

1. u contains a variable of $\mathbf{to}(\tau)$. We have $\mu(\vec{y}) = \nu_2(\vec{y})$. Since $\langle \nu_2(\vec{y}), [k, l] \rangle \sqsubseteq I$, it follows $\langle \mu(\vec{y}), [k, l] \rangle \sqsubseteq I$.
2. u contains no variable of $\mathbf{to}(\tau)$. We have $\mu(\vec{y}) = \nu_1(\vec{y})$. Since $\langle \nu_1(\vec{y}), [k, l] \rangle \sqsubseteq I$, it follows $\langle \mu(\vec{y}), [k, l] \rangle \sqsubseteq I$.

This concludes the proof. □

Dichotomy is preserved by ttq composition:

[7] Note that either $\mathbf{from}(\tau) \cap \mathbf{to}(\tau) = \{\}$ or $\mathbf{from}(\tau) \cap \mathbf{to}(\tau) = \mathbf{from}(\tau) = \mathbf{to}(\tau)$.

Lemma 6 *If τ and σ are dichotomic ttqs and $\tau \bullet \sigma$ is defined, then $\tau \bullet \sigma$ is dichotomic.*

PROOF. Let $\tau = (T, t)$ and $\sigma = (S, s)$ with $S = \{s_1, \ldots, s_m\}$ be as in Definition 13. Assume τ and σ dichotomic. Suppose $\tau \bullet \sigma$ is not dichotomic. Then there is a sequence of tableau tuples of $\tau \bullet \sigma$, where each θ_{i_j} captures its meaning from Definition 13 ($i_j \in \{1, \ldots, m\}$, overbraces added for clarity):

$$\overbrace{\theta_{i_1}(t_1^{i_1})\theta_{i_1}(t_2^{i_1})\ldots\theta_{i_1}(t_{k_1}^{i_1})}^{\text{tableau tuples in } \theta_{i_1}(T)}\ \overbrace{\theta_{i_2}(t_1^{i_2})\theta_{i_2}(t_2^{i_2})\ldots\theta_{i_2}(t_{k_2}^{i_2})}^{\text{tableau tuples in } \theta_{i_2}(T)}\ldots\overbrace{\theta_{i_l}(t_1^{i_l})\theta_{i_l}(t_2^{i_l})\ldots\theta_{i_l}(t_{k_l}^{i_l})}^{\text{tableau tuples in } \theta_{i_l}(T)}$$

where:

- $l \geq 1$,
- $i_1 \neq i_2 \neq \ldots \neq i_l \in \{1, \ldots, m\}$,
- $t_1^{i_1}, \ldots, t_{k_l}^{i_l}$ are tableau tuples of T,
- every two tuples that are neighbors in the sequence have a bounded (w.r.t. $\tau \bullet \sigma$) non-temporal variable in common, and
- $\theta_{i_1}(t_1^{i_1})$ contains the temporal variable $\mathbf{f}$ and $\theta_{i_l}(t_{k_l}^{i_l})$ contains $\mathbf{t}$.

Then the temporal tableau tuple s_{i_1} of S contains $\mathbf{f}$, s_{i_l} contains $\mathbf{t}$, and s_{i_j} and $s_{i_{j+1}}$ have a bounded (w.r.t. σ) non-temporal variable in common, for each $j \in [1, l-1]$. Two cases can occur.

1. If $l > 1$ then σ is not dichotomic.
2. If $l = 1$ then $s_{i_1}(\mathbf{T}) = [\mathbf{f}, \mathbf{t}]$. If s_{i_1} contains a bounded non-temporal variable, then σ is not dichotomic. If s_{i_1} contains no bounded non-temporal variable, then τ is not dichotomic.

The desired result follows by contradiction. □

8 Homomorphisms and dichotomic temporal tableau queries

For conjunctive queries, the problem of query containment is settled by an elegant Homomorphism Theorem [1]. When conjunctive queries are extended with inequality [6, 13] or linear [4] constraints, the homomorphism technique fails and the query containment problem becomes considerably more difficult. In this section, we show that query containment for dichotomic ttqs can be decided by applying the homomorphism technique a finite number of times. This finite number is linear in the number of distinct time points occurring in the ttqs under consideration.

Definition 15 Let $\tau = (T, t)$ and $\sigma = (S, s)$ be two ttqs such that T and S have the same arity, as well as t and s. A *homomorphism* from σ to τ under $(p, q) \in \mathbb{Z}^{2,\leq}$ is a substitution θ for the non-temporal variables in S such that $\theta(S_{\mathbf{f}\to p,\mathbf{t}\to q}) \sqsubseteq T_{\mathbf{f}\to p,\mathbf{t}\to q}$ and $\theta(s) = t$. □

We now provide a number of lemmas that will be used in the main theorem to follow.

Lemma 7 *Let $\tau = (T, t)$ be a ttq. Let $\sigma = (S, s)$ be a cfree ttq such that T and S have the same arity, as well as t and s. Then $\tau \sqsubseteq \sigma$ iff for each $(p, q) \in \mathbb{Z}^{2,\leq}$, there exists a homomorphism from σ to τ under (p, q).*

PROOF. *Only-if part.* Assume $\tau \sqsubseteq \sigma$. Since $T_{\mathbf{f}\to p,\mathbf{t}\to q} \models^{\tau} t_{\mathbf{T}\to[p,q]}$ is obvious, $t_{\mathbf{T}\to[p,q]} \in \sigma(T_{\mathbf{f}\to p,\mathbf{t}\to q})$. Since σ is cfree, $T_{\mathbf{f}\to p,\mathbf{t}\to q} \models^{\sigma} t_{\mathbf{T}\to[p,q]}$. Hence, there exists a valuation ν such that $\nu(S) \sqsubseteq T_{\mathbf{f}\to p,\mathbf{t}\to q}$ and $\nu(s) = t_{\mathbf{T}\to[p,q]}$. ν restricted to the non-temporal variables in S is a homomorphism from σ to τ under (p, q).

If part. Assume $I \models^{\tau} u$ with $u(\mathbf{T}) = [p, q]$, for some temporal relation I. Hence, there exists a valuation ν such that $\nu(T) \sqsubseteq I$ and $\nu(t) = u$, i.e., $\nu(\mathbf{f}) = p$ and $\nu(\mathbf{t}) = q$. By the premisse, we can assume the existence of a homomorphism θ such that $\theta(S_{\mathbf{f}\to p,\mathbf{t}\to q}) \sqsubseteq T_{\mathbf{f}\to p,\mathbf{t}\to q}$ and $\theta(s) = t$. Then $\nu(\theta(S_{\mathbf{f}\to p,\mathbf{t}\to q})) \sqsubseteq I$ and $\nu(\theta(s)) = u$. It follows $I \models^{\sigma} u$. This concludes the proof. □

The following example shows that, in general, the "homomorphism property" does not hold for ttqs.

Example 8

$$
\begin{array}{c|ccc}
\tau' & 1 & 2 & \mathbf{T} \\
\hline
 & x & u & [\mathbf{f}, 6] \\
 & x & u & [2, 6] \\
 & x & v & [2, 6] \\
 & x & v & [2, \mathbf{t}] \\
\hline
 & x & & [\mathbf{f}, \mathbf{t}]
\end{array}
\quad (t_2) \quad \text{and} \quad
\begin{array}{c|ccc}
\sigma & 1 & 2 & \mathbf{T} \\
\hline
 & x & u & [\mathbf{f}, 6] \\
 & x & v & [2, 6] \\
 & x & v & [2, \mathbf{t}] \\
\hline
 & x & & [\mathbf{f}, \mathbf{t}]
\end{array}
$$

Note incidentally that $\tau' = [\![\tau]\!]$ where τ as in Example 4. σ is obtained from τ' by removing the tableau tuple t_2. For each $(p, q) \in \mathbb{Z}^{2,\leq}$, if $p \leq 2$ then the identity is a homomorphism from τ' to σ under (p, q), and if $p > 2$ then $\{(x, x),(u, v),(v, v)\}$ is a homomorphism from τ' to σ under (p, q). Hence, for each $(p, q) \in \mathbb{Z}^{2,\leq}$, there exists a homomorphism from τ' to σ under (p, q). The "homomorphism property" [6] does not hold: There exists no substitution θ such that for each $(p, q) \in \mathbb{Z}^{2,\leq}$, θ is a homomorphism from τ' to σ under (p, q): We must sometimes map u to u, and sometimes to v.

We now show that the necessary and sufficient condition for $\tau \sqsubseteq \sigma$ of Lemma 7 can be simplified if σ is dichotomic.

Lemma 8 *Let $\tau = (T, t)$ be a ttq. Let $\sigma = (S, s)$ be a dichotomic ttq such that T and S have the same arity, as well as t and s. If for every $p \in \mathbb{Z}$, there exists a homomorphism from σ to τ under (p, p), then for every $(p, q) \in \mathbb{Z}^{2,\leq}$, there exists a homomorphism from σ to τ under (p, q).*

PROOF. The tableau tuples of S can be partitioned in four disjoint sets:

S^1**:** tableau tuples with timestamp $[\mathbf{f}, \mathbf{t}]$.

S^2**:** tableau tuples containing no variable of $\mathbf{from}(\tau) \cup \mathbf{to}(\tau) \cup \{\mathbf{f}, \mathbf{t}\}$.

S^3**:** tableau tuples containing at least one variable of $\mathbf{from}(\tau)$ as well as tableau tuples with timestamp $[\mathbf{f}, l]$ containing no bounded non-temporal variables ($l \in \mathbb{Z}$).

S^4**:** tableau tuples containing at least one variable of $\mathbf{to}(\tau)$ as well as tableau tuples with timestamp $[k, \mathbf{t}]$ containing no bounded non-temporal variables ($k \in \mathbb{Z}$).

Importantly, since σ is dichotomic, the same bounded non-temporal variable u cannot occur in two tableau tuples that belong to distinct partitions of $\{S^1, S^2, S^3, S^4\}$. Let $(p, q) \in \mathbb{Z}^{2,\leq}$. Assume that for every $p \in \mathbb{Z}$, there exists a homomorphism from σ to τ under (p, p). Consequently,

- All non-temporal variables in S^1 are free. Let θ be a substitution such that $\theta(s) = t$. Whenever θ' is a homomorphism from σ to τ, then for each free non-temporal variable x of σ, $\theta'(x) = \theta(x)$. Since $\theta(S^1{}_{\mathbf{f},\mathbf{t}\to r}) \sqsubseteq T_{\mathbf{f},\mathbf{t}\to r}$ for every $r \in \mathbb{Z}$, it follows $\theta(S^1{}_{\mathbf{f}\to p,\mathbf{t}\to q}) \sqsubseteq T_{\mathbf{f}\to p,\mathbf{t}\to q}$.
- There exists a homomorphism θ_2 from σ to τ under (p, p). Since neither $\mathbf{f}$ nor $\mathbf{t}$ appears in S^2, $\theta_2(S^2{}_{\mathbf{f}\to p,\mathbf{t}\to q}) \sqsubseteq T_{\mathbf{f}\to p,\mathbf{t}\to q}$.
- There exists a homomorphism θ_3 from σ to τ under (p, p). Since $\mathbf{t}$ does not appear in S^3, $\theta_3(S^3{}_{\mathbf{f}\to p,\mathbf{t}\to q}) \sqsubseteq T_{\mathbf{f}\to p,\mathbf{t}\to q}$.
- There exists a homomorphism θ_4 from σ to τ under (q, q). Since $\mathbf{f}$ does not appear in S^4, $\theta_4(S^4{}_{\mathbf{f}\to p,\mathbf{t}\to q}) \sqsubseteq T_{\mathbf{f}\to p,\mathbf{t}\to q}$.

Let θ be the substitution satisfying for each non-temporal variable u of τ:

$$
\theta(u) = \begin{cases} \theta_2(u) = \theta_3(u) = \theta_4(u) \text{ if } u \text{ is a free variable of } \sigma \\ \theta_2(u) \quad \text{if } u \text{ is a bounded variable appearing in } S^2 \\ \theta_3(u) \quad \text{if } u \text{ is a bounded variable appearing in } S^3 \\ \theta_4(u) \quad \text{if } u \text{ is a bounded variable appearing in } S^4 \end{cases}
$$

Obviously, θ is a homomorphism from σ to τ under (p, q). This concludes the proof. □

As a corollary, a necessary and sufficient condition for $\tau \sqsubseteq \sigma$, with σ dichotomic, is the existence of a homomorphism from σ to τ under (p, p), for each $p \in \mathbb{Z}$. Next it is intuitively clear that it should be sufficient to look at a finite number of distinguished time points, as captured next.

Definition 16 Let P be a (possibly empty) finite set of time points. We define $\approx_P$ as the equivalence relation on $\mathbb{Z}$ defined as the reflexive, symmetric, transitive closure of:

$p \approx_P q$ iff $p < q$ and for no point $k \in P$, $p \leq k \leq q$.

□

For example, if $P = \{2, 6\}$ then the equivalence classes of $\approx_P$ are: $\{p \in \mathbb{Z} \mid p < 2\}$, $\{2\}$, $[3, 5]$, $\{6\}$, and $\{p \in \mathbb{Z} \mid p > 6\}$.

Lemma 9 *Let σ and τ be two ttqs whose tableaux and summary rows are pairwise of the same arity. Let $P = apdom(\sigma) \cup apdom(\tau)$. Let $p, q \in \mathbb{Z}$ such that $p \approx_P q$. Each homomorphism from σ to τ under (p, p) is a homomorphism from σ to τ under (q, q).*

PROOF. Easy. □

Theorem 2 *Let $\tau = (T, t)$ be a ttq. Let $\sigma = (S, s)$ be a dichotomic ttq such that T and S have the same arity, as well as t and s. Let $P = apdom(\tau) \cup apdom(\sigma)$. Let B be a set of time points containing one representative from each equivalence class of the relation $\approx_P$. Then $\tau \sqsubseteq \sigma$ iff for each $p \in B$, there exists a homomorphism from σ to τ under (p, p).*

PROOF. Immediate from Lemmas 7, 8, and 9. □

To conclude, a necessary and sufficient condition for $\tau \sqsubseteq \sigma$, with σ dichotomic, is the existence of a homomorphism from σ to τ under (p, p), for one arbitrarily chosen representative p of each equivalence class of $\approx_P$, where $P = apdom(\tau) \cup apdom(\sigma)$. Note that $\approx_P$ has at most $2 \times |P| + 1$ equivalence classes.

Example 9 Consider the queries $\tau' = (T', t')$ and $\sigma = (S, s)$ of Example 8. Let $P = apdom(\tau') \cup apdom(\sigma) = \{2, 6\}$. $B = \{1, 2, 4, 6, 7\}$ contains one time point from each equivalence class of $\approx_P$. Since τ' is dichotomic, $\sigma \sqsubseteq \tau'$ if and only if for each $p \in B$, there exists a homomorphism from τ' to σ under (p, p). Consider, for example, the time point 7:

$$T'_{\mathbf{f},\mathbf{t}\to 7} \begin{array}{|ccc} 1 & 2 & \mathbf{T} \\ \hline x & u & [2,6] \\ x & v & [2,7] \end{array} \text{ and } S_{\mathbf{f},\mathbf{t}\to 7} \begin{array}{|ccc} 1 & 2 & \mathbf{T} \\ \hline x & v & [2,7] \end{array} .$$

Clearly, $\{(x, x), (u, v), (v, v)\}$ is a homomorphism from τ' to σ under $(7, 7)$.

For query optimization reasons, we are interested in the following problem: Given a ttq τ, find a minimal (w.r.t. tableau size) ttq equivalent to τ. It is natural to ask whether any dichotomic ttq can be minimized by simply removing tableau tuples. This is known to be true for non-temporal conjunctive queries [1], but false for conjunctive queries containing inequalities [12]. It does not generally hold for dichotomic ttqs, as illustrated by the following example.

Example 10 The query τ of Example 4 is equivalent to both σ and ρ:

$$\sigma \begin{array}{|ccc} 1 & 2 & \mathbf{T} \\ \hline x & u & [\mathbf{f},6] \\ x & u & [2,6] \\ x & v & [2,\mathbf{t}] \\ \hline x & & [\mathbf{f},\mathbf{t}] \end{array} \text{ and } \rho \begin{array}{|ccc} 1 & 2 & \mathbf{T} \\ \hline x & u & [\mathbf{f},6] \\ x & u & [2,\mathbf{t}] \\ \hline x & & [\mathbf{f},\mathbf{t}] \end{array} .$$

ρ is minimal, but not dichotomic, and hence does not possess some nice properties of dichotomic ttqs (Theorems 1 and 2). σ cannot be obtained by simply removing tableau tuples from τ. It can be obtained, however, by removing tableau tuples from $[\![\tau]\!]$. We have no examples of minimization *within the class of dichotomic ttqs* that do not amount simply to removing tableau tuples from stretched ttqs (stretched ttqs were defined in Definition 10).

9 From $\mathrm{SPC}^{\mathrm{time}}$ to dichotomic temporal tableau queries

We now apply the results obtained so far in the optimization of $\mathrm{SPC}^{\mathrm{time}}$ queries. The overall strategy proceeds in two steps: First, translate a given $\mathrm{SPC}^{\mathrm{time}}$ query Q into an equivalent dichotomic ttq τ, and then rely on Theorem 2 to remove redundant rows from τ. The first step raises a number of issues:

- In the non-temporal case, the normal form of SPC queries allows direct mappings between SPC and tableau queries. Recall from Section 1 that this normal form is no longer available in $\mathrm{SPC}^{\mathrm{time}}$, and hence the mapping from $\mathrm{SPC}^{\mathrm{time}}$ queries to ttqs will be more complicated.
- Although each $\mathrm{SPC}^{\mathrm{time}}$ query can be easily written as a composition of ttqs $\tau_1 \circ \tau_2 \circ \ldots \circ \tau_n$ (cf. Definition 17 below), this composition may be equivalent to no single ttq (cf. Lemma 4 and Theorem 1).
- The ttq obtained must be dichotomic if we want to rely on Theorem 2 for the detection of redundant tableau tuples.

We develop an approach that inductively constructs ttqs corresponding to subexpressions of an $\mathrm{SPC}^{\mathrm{time}}$ query.

Definition 17 For each $\mathrm{SPC}^{\mathrm{time}}$ query $Q(R)$, $tab(Q)$ is recursively defined as follows. Assume R is a relation variable of arity m, and $Q(R)$ is an $\mathrm{SPC}^{\mathrm{time}}$ query of arity n.

1. $tab(R) := \begin{array}{|cccc} 1 & \ldots & m & \mathbf{T} \\ \hline x_1 & \ldots & x_m & [\mathbf{f},\mathbf{t}] \\ \hline x_1 & \ldots & x_m & [\mathbf{f},\mathbf{t}] \end{array}$
2. $tab(\sigma_{[p,q]}(Q)) := tab(Q) \bullet \begin{array}{|cccc} 1 & \ldots & n & \mathbf{T} \\ \hline x_1 & \ldots & x_n & [\mathbf{f},q] \\ x_1 & \ldots & x_n & [p,\mathbf{t}] \\ \hline x_1 & \ldots & x_n & [\mathbf{f},\mathbf{t}] \end{array}$
3. $tab(\sigma_{i=j}(Q)) := tab(Q) \bullet \begin{array}{|cccccccc} 1 & \ldots & i-1 & i & i+1 & \ldots & n & \mathbf{T} \\ \hline x_1 & \ldots & x_{i-1} & x_j & x_{i+1} & \ldots & x_n & [\mathbf{f},\mathbf{t}] \\ \hline x_1 & \ldots & x_{i-1} & x_j & x_{i+1} & \ldots & x_n & [\mathbf{f},\mathbf{t}] \end{array}$
4. $tab(\sigma_{i=a}(Q)) := tab(Q) \bullet \begin{array}{|cccccccc} 1 & \ldots & i-1 & i & i+1 & \ldots & n & \mathbf{T} \\ \hline x_1 & \ldots & x_{i-1} & a & x_{i+1} & \ldots & x_n & [\mathbf{f},\mathbf{t}] \\ \hline x_1 & \ldots & x_{i-1} & a & x_{i+1} & \ldots & x_n & [\mathbf{f},\mathbf{t}] \end{array}$
5. $tab(\pi_{j_1,\ldots,j_k}(Q)) := tab(Q) \bullet \begin{array}{|ccccc} 1 & 2 & \ldots & n & \mathbf{T} \\ \hline x_1 & x_2 & \ldots & x_n & [\mathbf{f},\mathbf{t}] \\ \hline x_{j_1} & \ldots & x_{j_k} & [\mathbf{f},\mathbf{t}] & \end{array}$, provided that the composition results in a dichotomic ttq; otherwise $tab(\pi_{j_1,\ldots,j_k}(Q))$ is undefined. Note that x_i is bounded w.r.t. the right-hand ttq for every $i \in \{1,\ldots,n\} \setminus \{j_1,\ldots,j_k\}$.
6. Assume Q_1 and Q_2 are of arity n_1 and n_2 respectively. Let $tab(Q_1) = (T_1, t_1)$ and $tab(Q_2) = (T_2, t_2)$. Assume without loss of generality that T_1 and T_2 have no variables in common except for $\mathbf{f}$ and $\mathbf{t}$. Let $t := \langle t_1(1),\ldots,t_1(n_1), t_2(1),\ldots,t_2(n_2), [\mathbf{f},\mathbf{t}]\rangle$, a tableau tuple of arity n_1+n_2. Then $tab(Q_1 \times Q_2) := (T_1 \cup T_2, t)$.

□

Theorem 3 *Let $Q(R)$ be an $\mathrm{SPC}^{\mathrm{time}}$ query.*[8] *If $tab(Q)$ is defined, then $tab(Q) \equiv Q$.*

PROOF. Assume $tab(Q)$ is defined. We first show by induction on the structure of Q, that $tab(Q)$ is dichotomic. This is obvious for the base case $Q = R$ (item (1) in Definition 17). For the selection, since $tab(Q)$ is dichotomic by the induction hypothesis, and since the ttqs introduced in items (2), (3), and (4) of Definition 17 are all dichotomic, it follows by Lemma 6 that the result is dichotomic. For the cross product (item (6) in Definition 17), since $tab(Q_1)$

[8] As mentioned in Section 3, $Q(R)$ is assumed to be satisfiable.

and $tab(Q_2)$ are dichotomic by the induction hypothesis, and have no non-temporal variables in common, it follows that $tab(Q_1 \times Q_2)$ is dichotomic. Finally, Definition 17 explicitly requires dichotomy in the case of projection (item (5) in Definition 17).

Next we show by induction on the structure of Q, that $tab(Q) \equiv Q$. This is obvious if $Q = R$. Next we show $tab(\sigma_{[p,q]}(Q)) \equiv \sigma_{[p,q]}(Q)$. Let τ be the ttq:

$$\tau \begin{array}{|cccc} 1 & \ldots & n & \mathbf{T} \\ \hline x_1 & \ldots & x_n & [\mathbf{f}, q] \\ x_1 & \ldots & x_n & [p, \mathbf{t}] \\ \hline x_1 & \ldots & x_n & [\mathbf{f}, \mathbf{t}] \end{array} .$$

We show $\tau(I) = \sigma_{[p,q]}(I)$ for every temporal relation I of arity n. The inclusion $\sigma_{[p,q]}(I) \subseteq \tau(I)$ is straightforward. For the opposite inclusion, assume $I \vdash^{\tau} \langle \vec{a}, [p', q'] \rangle$, where $p' \leq q'$. Hence, $\langle \vec{a}, [p', q] \rangle \sqsubseteq I$ and $\langle \vec{a}, [p, q'] \rangle \sqsubseteq I$. By Lemma 1, $\langle \vec{a}, [p', q'] \vee [p, q] \rangle \sqsubseteq I$. Obviously, $\langle \vec{a}, [p', q'] \vee [p, q] \rangle \sqsubseteq \sigma_{[p,q]}(I)$, hence $\langle \vec{a}, [p', q'] \rangle \sqsubseteq \sigma_{[p,q]}(I)$. Since $tab(Q)$ is dichotomic, and hence cfree by Lemma 5, $tab(Q) \bullet \tau \equiv \tau \circ tab(Q)$ by Theorem 1. By the induction hypothesis, $tab(Q) \equiv Q$. It follows $tab(\sigma_{[p,q]}(Q)) \equiv \sigma_{[p,q]}(Q)$. The other forms of selection, as well as projection (item (5) in Definition 17) are treated analogously. For the cross product, assume $tab(Q_1)$ and $tab(Q_2)$ are as in Definition 17. We first show $tab(Q_1 \times Q_2) \sqsubseteq Q_1 \times Q_2$. Assume $I \vdash^{tab(Q_1 \times Q_2)} u$ for some temporal relation I. Hence, there exists a valuation ν such that $\nu(T_1) \sqsubseteq I$, $\nu(T_2) \sqsubseteq I$, and $u = \nu(t)$. Hence, $I \vdash^{tab(Q_1)} \nu(t_1)$ and $I \vdash^{tab(Q_2)} \nu(t_2)$. By the induction hypothesis, $\nu(t_1) \sqsubseteq Q_1(I)$ and $\nu(t_2) \sqsubseteq Q_2(I)$, hence $u \sqsubseteq (Q_1 \times Q_2)(I)$. Finally, we show $Q_1 \times Q_2 \sqsubseteq tab(Q_1 \times Q_2)$. Let $u \sqsubseteq (Q_1 \times Q_2)(I)$. Hence, there exist $u_1 \in Q_1(I)$ and $u_2 \in Q_2(I)$ such that:

$$u := \langle u_1(1), \ldots, u_1(n_1), u_2(1), \ldots, u_2(n_2), u_1(\mathbf{T}) \cap u_2(\mathbf{T}) \rangle \ .$$

Since $tab(Q_1)$ and $tab(Q_2)$ are dichotomic, $I \vdash^{tab(Q_1)} u_1$ and $I \vdash^{tab(Q_2)} u_2$ by the induction hypothesis. Hence there exists valuations ν_1 and ν_2 such that $\nu_1(T_1) \sqsubseteq I$, $\nu_2(T_2) \sqsubseteq I$, $\nu_1(t_1) = u_1$, and $\nu_2(t_2) = u_2$. Define ν as the valuation satisfying $\nu(v) = \nu_1(v)$ if v is a non-temporal variable in T_1, $\nu(v) = \nu_2(v)$ if v is a non-temporal variable in T_2, and $[\nu(\mathbf{f}), \nu(\mathbf{t})] = u(\mathbf{T})$. Since T_1 and T_2 have no non-temporal variables in common, ν is well-defined. Obviously, $\nu(T_1 \cup T_2) \sqsubseteq I$, hence $I \vdash^{tab(Q_1 \times Q_2)} \nu(t)$, and $\nu(t) = u$. This concludes the proof. □

To conclude this section, we show how the above results can be applied to simplify the query $Q(C)$ introduced in Section 1.

Example 11 Consider the SPC query $Q_1 = \pi_1(\sigma_{[14,18]}(\sigma_{2=\text{Paris}}(C)))$ introduced in Section 1. Using Definition 17, the ttq $tab(\sigma_{[14,18]}(\sigma_{2=\text{Paris}}(C)))$ is computed (call it τ):

$$\tau \begin{array}{|ccc} 1 & 2 & \mathbf{T} \\ \hline x & \text{Paris} & [\mathbf{f}, 18] \\ x & \text{Paris} & [14, \mathbf{t}] \\ \hline x & \text{Paris} & [\mathbf{f}, \mathbf{t}] \end{array} .$$

For the projection, we obtain:

$$tab(Q_1) = \tau \bullet \begin{array}{|ccc} 1 & 2 & \mathbf{T} \\ \hline x & v & [\mathbf{f}, \mathbf{t}] \\ \hline x & & [\mathbf{f}, \mathbf{t}] \end{array} = \begin{array}{|ccc} 1 & 2 & \mathbf{T} \\ \hline x & \text{Paris} & [\mathbf{f}, 18] \\ x & \text{Paris} & [14, \mathbf{t}] \\ \hline x & & [\mathbf{f}, \mathbf{t}] \end{array} ,$$

which is dichotomic and hence well-defined.

Example 12 Consider the ttq $Q(C)$ introduced in Section 1:

$$Q(C) = \pi_1(\sigma_{1=2}(\overbrace{\pi_1(\sigma_{[14,18]}(\sigma_{2=\text{Paris}}(C)))}^{Q_1} \times \overbrace{\pi_1(\sigma_{[40,45]}(\sigma_{2=\text{Paris}}(C)))}^{Q_2})) \ .$$

Using a partial result of Example 11, the ttq $tab(Q)$ is computed (call it τ):

$$\begin{array}{l|lll} \tau & 1 & 2 & \mathbf{T} \\ \hline & x & \text{Paris} & [\mathbf{f},18] \\ & x & \text{Paris} & [14,\mathbf{t}] \\ & x & \text{Paris} & [\mathbf{f},45] \\ & x & \text{Paris} & [40,\mathbf{t}] \\ \hline & x & & [\mathbf{f},\mathbf{t}] \end{array}\;.$$

The outermost projection caused no problem: The result is dichotomic. We then rely on Theorem 2 to verify that the first and last tableau tuple can be removed without changing the meaning of the query. We finally obtain:

$$Q \equiv \begin{array}{|lll} 1 & 2 & \mathbf{T} \\ \hline x & \text{Paris} & [14,\mathbf{t}] \\ x & \text{Paris} & [\mathbf{f},45] \\ \hline x & & [\mathbf{f},\mathbf{t}] \end{array}\;.$$

It can be verified (cf. Example 11) that the latter ttq is equivalent to $\pi_1(\sigma_{[14,45]}(\sigma_{2=\text{Paris}}(C)))$.

10 Ubiquitousness

In this section, we extend the operator $tab(\cdot)$ to certain SPC^{time} queries left uncaptured by Definition 17. This extension is motivated by the following example. Suppose we want to build $tab(Q)$ for a query Q that contains a subquery $Q_1 = \pi_1(\sigma_{[2,6]}(R))$, where R has arity 2. By Theorem 1:

$$Q_1 \equiv \begin{array}{l|lll} tab(\sigma_{[2,6]}(R)) & 1 & 2 & \mathbf{T} \\ \hline & x & y & [\mathbf{f},6] \\ & x & y & [2,\mathbf{t}] \\ \hline & x & y & [\mathbf{f},\mathbf{t}] \end{array} \bullet \begin{array}{|lll} 1 & 2 & \mathbf{T} \\ \hline x & v & [\mathbf{f},\mathbf{t}] \\ \hline x & & [\mathbf{f},\mathbf{t}] \end{array} = \begin{array}{|lll} 1 & 2 & \mathbf{T} \\ \hline x & u & [\mathbf{f},6] \\ x & u & [2,\mathbf{t}] \\ \hline x & & [\mathbf{f},\mathbf{t}] \end{array}$$

Unfortunately, the outcome ttq is not dichotomic, and hence may not be cfree. Consequently, the outcome cannot be used later on in the inductive procedure that constructs $tab(Q)$, as we can no longer rely on Theorem 1, which tells us that $\tau \bullet \sigma \equiv \sigma \circ \tau$ *provided that* τ *is cfree*. However, a simple observation can help us to solve this problem. Whenever a tuple $\langle x, u, [\mathbf{f},\mathbf{t}]\rangle$ belongs to $\sigma_{[2,6]}(R)$, then $[\mathbf{f},\mathbf{t}]$ must necessarily contain $[2,6]$. Then in order to have $\langle x, [\mathbf{f},\mathbf{t}]\rangle$ in $\pi_1(\sigma_{[2,6]}(R))$, it is both necessary and sufficient to have $\langle x, u, [\mathbf{f},6]\rangle$ and $\langle x, v, [2,\mathbf{t}]\rangle$ in $\sigma_{[2,6]}(R)$, where u and v may be equal. These tuples will be coalesced in Q_1. Hence, it is correct to conclude:

$$Q_1 \equiv \begin{array}{l|lll} tab(\sigma_{[2,6]}(R)) & 1 & 2 & \mathbf{T} \\ \hline & x & y & [\mathbf{f},6] \\ & x & y & [2,\mathbf{t}] \\ \hline & x & y & [\mathbf{f},\mathbf{t}] \end{array} \bullet \begin{array}{|lll} 1 & 2 & \mathbf{T} \\ \hline x & u & [\mathbf{f},6] \\ x & v & [2,\mathbf{t}] \\ \hline x & & [\mathbf{f},\mathbf{t}] \end{array} = \begin{array}{|lll} 1 & 2 & \mathbf{T} \\ \hline x & u & [\mathbf{f},6] \\ x & u & [2,6] \\ x & v & [2,6] \\ x & v & [2,\mathbf{t}] \\ \hline x & & [\mathbf{f},\mathbf{t}] \end{array}$$

The latter result is dichotomic, and can be used in the inductive procedure that computes $tab(Q)$. Note incidentally that an optimization consists in removing the second or third row (see Examples 8 and 10).

Definition 18 Let $\tau = (T, t)$ be a ttq. An interval $[p,q]$ is *ubiquitous* for τ iff for every temporal relation I of the same arity as T, $s \in \tau(I)$ implies $s(\mathbf{T}) \supseteq [p,q]$.

The procedure $tab(\cdot)$ can then be extended as follows: If $tab(\pi_{j_1,\ldots,j_k}(Q))$ remains undefined by Definition 17, but Q has a non-empty ubiquitous interval (say $[p,q]$), then:

$$tab(\pi_{j_1,\ldots,j_k}(Q)) := tab(Q) \bullet \begin{array}{|lllllll|l} 1 & 2 & \ldots & j_i & \ldots & n & \mathbf{T} & \\ \hline x_1 & x_2 & \ldots & x_{j_i} & \ldots & x_n & [\mathbf{f},q] & (t_1) \\ v_1 & v_2 & \ldots & x_{j_i} & \ldots & v_n & [p,\mathbf{t}] & (t_2) \\ \hline x_{j_1} & \ldots & x_{j_k} & & & & [\mathbf{f},\mathbf{t}] & (t) \end{array}\;.$$

That is, $t_1(j_i) = t_2(j_i) = t(i)$ for each $i \in \{1,\ldots,k\}$, and all variables are pairwise distinct otherwise. Note that the right-hand ttq is dichotomic. $\square$

It is now straightforward to show that Theorem 3 remains valid under this extended definition of $tab(\cdot)$. Definition 18 raises another problem, however: Determine whether a given dichotomic ttq $\tau = tab(Q)$ has a non-empty ubiquitous interval. It can be easily verified that if $apdom(\tau) = \{\}$, then τ has no non-empty ubiquitous interval. Next, assume $apdom(\tau) \neq \{\}$. Let m and M be $\min(apdom(\tau))$ and $\max(apdom(\tau))$ respectively. It can be easily seen that if τ has a non-empty ubiquitous interval, then it has an ubiquitous interval $[p, p]$ with $p \in [m, M]$. Hence, τ has a non-empty ubiquitous interval if and only if for some $p \in [m, M]$, $\tau \sqsubseteq tab(\sigma_{[p,p]}(Q))$. The latter inclusion can be verified using Theorem 2. Finally, it can be verified that if $p_1 \approx_{apdom(\tau)} p_2$, then $\tau \sqsubseteq tab(\sigma_{[p_1,p_1]}(Q))$ if and only if $\tau \sqsubseteq tab(\sigma_{[p_2,p_2]}(Q))$. So if $p_1 \approx_{apdom(\tau)} p_2$, it suffices to verify either $\tau \sqsubseteq tab(\sigma_{[p_1,p_1]}(Q))$ or $\tau \sqsubseteq tab(\sigma_{[p_2,p_2]}(Q))$.

11 Contributions to related work

The $\mathrm{SPC}^{\mathrm{time}}$ algebra complies with common primitives in existing temporal query languages [9]. $\mathrm{SPC}^{\mathrm{time}}$ queries could be optimized by minimizing an equivalent conjunctive query with inequalities. However, this approach is expensive [6, 13] and may overlook some simplifications that apply in the temporal case, but not in general. What is more, the $\mathrm{SPC}^{\leq}$ algebra does not capture coalescing, which is non-monotonic in strict sense.[9]

Temporal tableau queries (ttqs) extend classical tableau queries with time intervals that can contain two temporal variables $\mathbf{f}$ and $\mathbf{t}$. Interestingly, if σ is dichotomic, then $\tau \sqsubseteq \sigma$ coincides with the existence of $2 \times N + 1$ (or less) distinguished homomorphisms from σ to τ, where N is the number of distinct time constants occurring in σ or τ. Consequently, testing query containment is easier for dichotomic ttqs than in $\mathrm{SPC}^{\leq}$. The procedure $tab(\cdot)$ translates $\mathrm{SPC}^{\mathrm{time}}$ queries into equivalent dichotomic ttqs, which can then be simplified by removing redundant tableau tuples. An open problem is whether a minimal (w.r.t. tableau size) equivalent dichotomic ttq can always be achieved by simply removing tableau tuples. Unfortunately, $tab(\cdot)$ cannot handle all queries with projection and associated coalescing. Nevertheless, it can detect and handle some queries that are outside the scope of $\mathrm{SPC}^{\leq}$. For example, $Q = \pi_1(\sigma_{[2,6]}(R))$, where R has arity 2, is non-monotonic in strict sense, but has an equivalent dichotomic ttq. An open problem is deciding containment of $\mathrm{SPC}^{\mathrm{time}}$ queries with projections that can result in an unbounded number of tuples being coalesced into a single one. It is unclear whether such projections can be captured by finite tableaux.

The use of "homomorphisms" for deciding containment of queries that deal with intervals, is somehow remarkable. Queries in which each variable is bounded by a constant from only one side (either left or right) have been called semiinterval queries. The homomorphism property is known to fail for semiinterval queries [5]. Left-semiinterval queries are queries where *all* inequalities are of the form $x\theta c$, where x is a variable, c is a constant, and $\theta \in \{\leq, <, =\}$, i.e., inequalities can express only upper bounds for variables. Right-semiinterval queries are defined analogously. In [6] it is shown that the homomorphism property holds for left-semiinterval queries and right-semiinterval queries. Temporal tableau queries are neither left- nor right-semiinterval. If $\mathrm{SPC}^{\mathrm{time}}$ queries are translated in $\mathrm{SPC}^{\leq}$, inequalities of the form $\mathbf{f} \leq p$, $\mathbf{f} > p$, $\mathbf{t} \geq p$, $\mathbf{t} < p$ emerge, together with $\mathbf{f} \leq \mathbf{t}$ ($p \in \mathbb{Z}$). For example, a tableau tuple $\langle \vec{x}, [\mathbf{f}, p] \rangle$ expresses: If $\mathbf{f} \leq p$ then $\langle \vec{x}, [\mathbf{f}, p] \rangle$ should be in the argument temporal relation (up to a substitution of constants for variables in $\vec{x}$), and if $\mathbf{f} > p$ then *true*.

The operator $tab(\cdot)$ maps $\mathrm{SPC}^{\mathrm{time}}$ queries to ttqs. The inverse mapping from ttqs to $\mathrm{SPC}^{\mathrm{time}}$ has not been considered. In this respect, it is interesting to note that the removal of redundant rows in $tab(Q)$ does not necessarily lead to a simplification of the $\mathrm{SPC}^{\mathrm{time}}$ query Q. For example, the query $[\![\tau]\!]$ of Example 4 equals $tab(\pi_1(\sigma_{[2,6]}(R)))$, where R is of arity 2. The tableau size of $[\![\tau]\!]$ can be reduced, as shown in Example 10. However, it is easy to see that $\pi_1(\sigma_{[2,6]}(R))$ is equivalent to no $\mathrm{SPC}^{\mathrm{time}}$ query with less operations. Recall incidentally that some ttqs are equivalent to no $\mathrm{SPC}^{\mathrm{time}}$ query, as illustrated by Example 2.

Some limits of our ttq formalism are as follows. The results are presented for queries over a single relation, but can be generalized for database schemas involving more than one relation. For multiple relations, a rule-based formalism may be more appropriate than the tableau formalism. We considered only valid time; the extension to bitemporal relations is an open issue. As already noticed in [14], many of our results rely on the assumption that time is discrete. It is interesting to investigate how the ttq formalism needs to be adapted to deal with dense time.

Finally, it is important to note that ttqs have been introduced as a tool for query optimization, and by no means as a query language for end-users. Several temporal query languages for end-users have been proposed in the literature. Such query languages may use point-based time instead of interval-based time, as in [11]. However, even if

[9] Monotonic in strict sense means that $I \subseteq J$ implies $Q(I) \subseteq Q(J)$.

point-based time is used at the conceptual level, time intervals are likely to emerge at the storage level for efficiency reasons, and hence optimization of interval-based queries remains significant.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] M. H. Böhlen, R. T. Snodgrass, and M. D. Soo. Coalescing in temporal databases. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, editors, *Proceedings of the twenty-second international Conference on Very Large Data Bases, September 3–6, 1996, Mumbai (Bombay), India*, pages 180–191, Los Altos, CA 94022, USA, 1996. Morgan Kaufmann Publishers.

[3] D. Dey, T. M. Barron, and V. C. Storey. A complete temporal relational algebra. *VLDB Journal: Very Large Data Bases*, 5(3):167–180, 1996.

[4] O. H. Ibarra and J. Su. On the containment and equivalence of database queries with linear constraints (extended abstract). In ACM, editor, *PODS '97. Proceedings of the Sixteenth ACM SIG-SIGMOD-SIGART Symposium on Principles of Database Systems, May 12–14, 1997, Tucson, Arizona*, pages 32–43, New York, NY 10036, USA, 1997. ACM Press.

[5] P. C. Kanellakis, G. M. Kuper, and P. Z. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, 51(1):26–52, 1995.

[6] A. Klug. On conjunctive queries containing inequalities. *Journal of the ACM*, 35(1):146–160, 1988.

[7] N. A. Lorentzos and Y. G. Mitsopoulos. SQL Extensions for Interval Data. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):480–499, 1997.

[8] L. E. McKenzie, Jr. and R. T. Snodgrass. Evaluation of relational algebras incorporating the time dimension in databases. *ACM Computing Surveys*, 23(4):501–543, 1991.

[9] R. T. Snodgrass, editor. *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers, 1995.

[10] R. T. Snodgrass. Overview of Temporal Databases. In C. Zaniolo, S. Ceri, C. Faloutsos, R. Snodgrass, V. S. Subrahmanian, and R. Zicari, editors, *Advanced Database Systems*, chapter 5, pages 99–126. Morgan Kaufmann, San Francisco, 1997.

[11] D. Toman. Point-Based Temporal Extensions of SQL and Their Efficient Implementation. In O. Etzion, S. Jajodia, and S. Sripada, editors, *Temporal Databases: Research and Practice*, volume 1399 of *Lecture Notes in Computer Science*, pages 211–237. Springer-Verlag, 1998.

[12] J. D. Ullman. *Principles of Database and Knowledge-Base Systems – Volume II*. Computer Science Press, Rockville, MD, 1989.

[13] R. van der Meyden. The complexity of querying indefinite data about linearly ordered domains. *Journal of Computer and System Sciences*, 54(1):113–135, Feb. 1997.

[14] J. Wijsen and A. Bès. Temporal tableaux queries. In *Proc. Eighth International Symposium on Temporal Representation and Reasoning, TIME 2001*, pages 136–143. IEEE Computer Society Press, 2001.