Temporal Tableau Queries

Jef Wijsen and Alexis Bès
Université de Mons-Hainaut, Belgium

Abstract

The tableau construct plays a very important role in relational database theory. It is shown how this construct can be extended for tuple-timestamped relations. The expressive power of temporal tableau queries is compared with that of a temporal algebra. A temporal extension of the Homomorphism Theorem is given.

1. Introduction

Several algebras for temporal relational databases have been published since the early eighties [3, 7]. 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 [6, 8, 11]. The *theoretical* foundations of these temporal query languages—including issues like expressiveness, complexity, computability, genericity, and optimization—have not been systematically explored. Some intriguing theoretical questions are raised by the following example that initially aroused our interest in this field; the example also illustrates some of the notations used later on.

Consider the following temporal relation inspired by [2]:

S	$SNAME$	$STATUS$	$CITY$	$FROM$	TO
	Blake	10	Athens	1	3
	Blake	20	Athens	4	7
	Smith	10	Paris	1	4
	Smith	20	London	5	9

The natural numbers are used to denote time. The first tuple means that supplier Blake lived in Athens with status 10 at time instants 1, 2, and 3. Consider the query:

> *Give the names of suppliers who stayed within the same city throughout the entire time period* $[3, 6]$.

To answer this query, we first perform a temporal projection on $SNAME$ and $CITY$. We use an unnamed algebra perspective and denote $SNAME$ and $CITY$ as columns 1 and 3 respectively.

$\pi_{1,3}(S)$	$SNAME$	$CITY$	$FROM$	TO
	Blake	Athens	1	7
	Smith	Paris	1	4
	Smith	London	5	9

Note that the two rows about Blake that existed in S, have been "coalesced" into a single one. Next we select the rows that include the interval $[3, 6]$. This selection is denoted $\sigma_{[3,6]}(\pi_{1,3}(S))$ and retrieves only the row about Blake (result not shown). Finally we project on the first column $SNAME$:

$\pi_1(\sigma_{[3,6]}(\pi_{1,3}(S)))$	$SNAME$	$FROM$	TO
	Blake	1	7

Importantly, in the subquery $\sigma_{[3,6]}(\pi_{1,3}(S))$ we cannot push the selection through the inner projection: the result of the query $\sigma_{[3,6]}(S)$ is empty.

Now it is well-known that every (non-temporal) SPC algebra query can be equivalently rewritten in the normal form $\pi_{\vec{j}}(\sigma_F(R_1 \times \ldots \times R_k))$ [1, p. 55]. This normal form has probably been the original motivation for the *select-from-where* syntax of SQL queries, and allows a direct translation of SPC queries into tableau queries and *vice versa* [1, p. 118]. The *tableau* construct is crucial in relational database theory; it has led to general techniques applied in query optimization and dependency theory. As the temporal select operator $\sigma_{[3,6]}(\cdot)$ in the above example cannot be pushed through the projection, it turns out that the normal form cannot be attained in the temporal algebra under consideration. When we first conceived this example, we felt somewhat uneasy about the absence of the normal form in the temporal algebra. It was the incentive for raising the following related questions:

1. Considering that SQL's *select-from-where* syntax is founded on the normal form of SPC algebra queries, how do temporal SQL extensions meet the absence of this normal form in the temporal algebra?
2. Can the construct of tableau query be generalized for temporal relations? Does the absence of the normal form in the temporal algebra hinder the "temporalizing" of tableau queries?

Both questions are of practical and theoretical importance. Interestingly, TSQL2 [8], which is probably the best-known

0-7695-1107-4/01 $10.00 © 2001 IEEE

temporal SQL extension, addresses the first question by a construct called "restructuring," which is stated to be "one of the most powerful constructs of TSQL2" [10, p. 133]. Loosely speaking, restructuring allows a pre-projection in the *from* clause, nested within the outer projection of the *select* clause. However, a temporal algebra query may contain more than two projections none of which can be removed by rewriting; an example is $\pi_1(\sigma_{[1,4]}(\pi_{1,2}(\sigma_{[2,3]}(\pi_{1,2,3}(I)))))$, where I contains at least four non-temporal attributes. Consequently, restructuring alone is generally insufficient and more powerful constructs are needed, in particular the ability to nest queries in the *from* clause. The second question is at the center of this paper, whose main contribution is the introduction of *temporal tableau query*. This new construct is introduced in intimate relationship with a "kernel" temporal algebra called $\mathrm{SPC}^{\mathrm{time}}$—kernel, in the sense that it is difficult to imagine a full-fledged temporal algebra that cannot express all $\mathrm{SPC}^{\mathrm{time}}$ queries. We first introduce $\mathrm{SPC}^{\mathrm{time}}$, and then we introduce temporal tableau queries that can express an important class of $\mathrm{SPC}^{\mathrm{time}}$ queries, called *rooted* $\mathrm{SPC}^{\mathrm{time}}$ queries.

Section 2 takes off with a note on coalescing. Temporal relations are defined in Section 3. The $\mathrm{SPC}^{\mathrm{time}}$ algebra is introduced in Section 4, and the subclass of rooted $\mathrm{SPC}^{\mathrm{time}}$ queries in Section 5. Sections 6 and 7 introduce temporal tableau queries and their composition. $\mathrm{SPC}^{\mathrm{time}}$ queries and temporal tableau queries are compared for expressiveness in Section 8. In particular, we develop an effective method for translating rooted $\mathrm{SPC}^{\mathrm{time}}$ queries into temporal tableau queries. Section 9 introduces a temporal generalization of the famous *Homomorphism Theorem* [1, p. 117], and illustrates how it may be applied for optimization of temporal queries. Finally, Section 10 lists some open problems.

2. A note on coalescing

Coalescing is a well-known concept in temporal databases, and was already illustrated in Section 1. It is not our aim to contribute here to the ongoing debate whether coalescing is beneficial or harmful; rather we want to shed some light on the expressive power needed for coalescing. The query coalesce defined next captures the essence of this operator. It works on sets of intervals, i.e., temporal relations of arity 0 according to Definition 3 introduced later; the extension to temporal relations of higher arity is straightforward.

Definition 1 Let $\mathbb{N}^{\leq} := \{\langle i,j\rangle \mid i,j \in \mathbb{N} \text{ and } i \leq j\}$. Let I be a finite subset of $\mathbb{N}^{\leq}$. The output of the query coalesce on input I, denoted $\mathrm{coalesce}(I)$, is a minimal (w.r.t. cardinality) subset of $\mathbb{N}^{\leq}$ satisfying:

Complete: if $\langle i,j\rangle \in I$ then $\mathrm{coalesce}(I)$ contains a pair $\langle k,l\rangle$ with $k \leq i$ and $j \leq l$, and

Sound: if $\langle k,l\rangle \in \mathrm{coalesce}(I)$ and $m \in \mathbb{N}$ such that $k \leq m \leq l$, then I contains a pair $\langle i,j\rangle$ such that $i \leq m \leq j$.

The query coalesce is expressible in the SPCUD algebra (= the SPCU-algebra [1, p. 62] extended with Difference) extended with selections of the form $\sigma_{i<j}(Q)$, equipped with the natural semantics. An SQL version has appeared in [10, p. 105]. The following result is not hard to prove but nevertheless important.

Theorem 1 *The query* coalesce *is not expressible in the SPCU algebra extended with selections of the form* $\sigma_{i<j}(Q)$.

Consequently, any temporal algebra that incorporates coalescing, as the algebra proposed in Section 4 does, allows certain queries that are outside the scope of the SPCU algebra. This suggests that certain theoretical questions may not be easy.

3. Temporal relation

We assume a discrete linear order to represent time. In all examples throughout this paper, natural numbers are used to denote time points.

Definition 2 We assume a discrete, linearly ordered set $(\mathbb{P},<)$ of *time points*. We define for each $p,q \in \mathbb{P}$ the *interval* $[p,q] := \{x \in \mathbb{P} \mid p \leq x \leq q\}$. The set of all intervals is denoted $\mathbb{I}$. Two intervals are *unifiable* if their set union is again an interval.

Theorem 2 *The set* $\mathbb{I}$, *ordered by set inclusion, is a lattice satisfying:*

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

Supremum. *If* $p \leq q$ *and* $r \leq s$ *then* $[p,q] \vee [r,s] = [p,s] \cup [r,q]$.

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):

$$
\begin{array}{c|ccc}
I & 1 & 2 & \mathbf{T} \\
\hline
 & a & b & [1,3] \\
 & a & b & [5,8] \\
 & a & c & [2,4]
\end{array}
\quad \text{and} \quad
\begin{array}{c|ccc}
J & 1 & 2 & \mathbf{T} \\
\hline
 & a & b & [0,9] \\
 & a & c & [2,5]
\end{array}
.
$$

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 containment relationship between these two tuples is written $\langle a,b,[1,3]\rangle \preceq \langle a,b,[0,9]\rangle$. 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 3 We assume a countably infinite set $\mathbf{dom}$ of *constants*. A *(temporal) tuple* of arity n ($n \geq 0$) is an element of $\mathbf{dom}^n \times \mathbb{I}$. If t is a 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 $t(\mathbf{T})$. Two tuples t and s of the same arity (say n) are *value-equal*, denoted $t \asymp s$, iff $t(i) = s(i)$ for each $i \in \{1, \ldots, n\}$. We write $t \preceq s$ iff $t \asymp s$ and $t(\mathbf{T}) \subseteq s(\mathbf{T})$.

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 $t \preceq s$ for some $s \in T$. We write $T \sqsubseteq S$ iff $t \sqsubseteq S$ for every tuple $t \in T$.

A *(temporal) relation* of arity n ($n \geq 0$) is a set I of temporal tuples of arity n such that for each $t, s \in I$:

1. $t(\mathbf{T}) \neq \{\}$, and
2. if $t(\mathbf{T})$ and $s(\mathbf{T})$ are unifiable and $t \asymp s$ then $t = s$.

4. A simple temporal algebra

We introduce a basic temporal extension of the SPC algebra; an example was already given in Section 1. 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 taking the intersection of the timestamps. These temporal operators are common; see for example [9]. The temporal algebra $\mathrm{SPC}^{\mathrm{time}}$ will be used later on for studying the expressive power of temporal tableau queries.

4.1. Temporal selection

Definition 4 Let I be a temporal relation of arity n, and $i, j \in \{1, \ldots, n\}$, $a \in \mathbf{dom}$, and $p, q \in \mathbb{P}$ with $p \leq q$.

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

4.2. Temporal projection

Definition 5 Let I be a temporal relation of arity n, and $j_1, j_2, \ldots, j_m \in \{1, \ldots, n\}$ ($m \geq 0$). $\pi_{j_1,\ldots,j_m}(I)$ is the smallest (w.r.t. $\sqsubseteq$) temporal relation J of arity m satisfying for every $t \in I$, $\langle t(j_1), \ldots, t(j_m), t(\mathbf{T})\rangle \sqsubseteq J$.

4.3. Temporal cross product

Definition 6 Let I, J be temporal relations of arities n and m respectively. $I \times J := \{\langle t(1), \ldots, t(n), s(1), \ldots, s(m), t(\mathbf{T}) \cap s(\mathbf{T})\rangle \mid t \in I \text{ and } s \in J \text{ and } t(\mathbf{T}) \cap s(\mathbf{T}) \neq \{\}\}$. It can be verified that $I \times J$ is a temporal relation of arity $n + m$.

4.4. $\mathrm{SPC}^{\mathrm{time}}$

Definition 7 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 $p, q \in \mathbb{P}$ with $p \leq q$, 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 $j_1, \ldots, j_m \in \{1, \ldots, n\}$, then $\pi_{j_1,\ldots,j_m}(Q)$ is an $\mathrm{SPC}^{\mathrm{time}}$ query of arity m ($m \geq 0$).

Join. 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 $Q(R_1, \ldots, R_n)$, where $R_1, \ldots, R_n$ are distinct relation variables, to indicate that Q is a query containing the relation variables $R_1, \ldots, R_n$. The *semantics* of $Q(R_1, \ldots, R_n)$ is relative to an interpretation function that maps each R_i to a temporal relation of the same arity as R_i ($i \in \{1, \ldots, n\}$). These *semantics* is defined in the natural manner (not elaborated here). In the remainder of this paper, we only consider $\mathrm{SPC}^{\mathrm{time}}$ 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.

Unsatisfiability arises if two constants are required to be equal, as in $\sigma_{1=a}(\sigma_{1=b}(R))$. $\mathrm{SPC}^{\mathrm{time}}$ queries are monotonic:

Theorem 3 *Let $Q(R)$ be an $\mathrm{SPC}^{\mathrm{time}}$ query, and I, J temporal relations of the same arity as R. If $I \sqsubseteq J$ then $Q(I) \sqsubseteq Q(J)$.*

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.

5. Rooted $\mathrm{SPC}^{\mathrm{time}}$ queries

The tableau construct is very important in relational database theory. It can be used, among others, for query optimization based on the *Homomorphism Theorem* [1]. It is interesting to ask whether this construct can be extended to the temporal domain.

Loosely speaking, a tableau query consists of a relation T, called the tableau of the query, followed by a tuple u, called the summary. The tableau T and the summary u can contain constants as well as variables. Intuitively, the tableau T lists the tuples that are both sufficient and necessary to include the summary tuple u in the answer to the query.

The formalism of tableau query does not carry over immediately to the temporal domain. One problem is with projection and the involved coalescing. Consider the $\mathrm{SPC}^{\mathrm{time}}$ query $\pi_1(Q(R))$ where the arity of Q is 2. Let I be a relation of the same arity as R. What tuples should be in $Q(I)$ in order to include the tuple $u = \langle x, [p, q] \rangle$ with $p \leq q$ in the answer $\pi_1(Q(I))$? According to the definition of projection, $u \sqsubseteq \pi_1(Q(I))$ if and only if $Q(I)$ contains a number $k > 0$ of tuples $\langle x, y_1, [p_1, q_1] \rangle, \ldots, \langle x, y_k, [p_k, q_k] \rangle$ such that $[p, q] \subseteq \bigcup_{j=1}^{k} [p_j, q_j]$. The problem arising is that the number k is not fixed. This suggests that the classical notion of tableau needs to be adapted to accommodate a variable number of tuples.

Interestingly, there are situations where the use of the projection operator does not lead to this kind of difficulty. Assume in the previous example $Q(R) = \sigma_{[r,s]}(Q'(R))$. Clearly, $t \in \sigma_{[r,s]}(Q'(I))$ implies $t(\mathbf{T}) \supseteq [r, s]$. Then it is correct to conclude that, in order to have $u = \langle x, [p, q] \rangle \sqsubseteq \pi_1(\sigma_{[r,s]}(Q'(I)))$, it is sufficient and necessary to have two tuples $\langle x, y_1, [p, s] \rangle, \langle x, y_2, [r, q] \rangle \sqsubseteq \sigma_{[r,s]}(Q'(I))$. The outer projection in $\pi_1(\sigma_{[r,s]}(Q'(I)))$ will coalesce these two tuples into a single tuple $\langle x, [p, q] \vee [r, s] \rangle$ (cf. Theorem 2). Hence, the number k reduces to 2 as a result of the fact that every tuple of $\sigma_{[r,s]}(Q'(I))$ must necessarily include $[r, s]$. This observation leads to the subclass of "rooted" queries defined next.

Definition 8 An interval $[p, q]$ is called *ubiquitous* for an $\mathrm{SPC}^{\mathrm{time}}$ query $Q(R)$ iff for every temporal relation I of the same arity as R, $t \in Q(I)$ implies $t(\mathbf{T}) \supseteq [p, q]$.

$Q(R)$ is called *rooted* iff (1) it is satisfiable and (2) for each subquery of Q of the form $\pi_{\vec{j}}(Q')$, there is a non-empty, ubiquitous interval for Q'; otherwise $Q(R)$ is *unrooted*.

Example 1 Each query $Q(R)$ without projection is rooted. $\pi_1(\sigma_{[1,4]}(R))$ is rooted, but $\sigma_{[1,4]}(\pi_1(R))$ is not.

There exist unrooted $\mathrm{SPC}^{\mathrm{time}}$ queries that are equivalent to rooted ones. Obviously, the unrooted query $\pi_{1,2,\ldots,n}(R)$, where R is of arity n, is equivalent to the rooted query R. It is slightly harder to verify that the unrooted query $\pi_{1,2,\ldots,n}(\sigma_{[1,4]}(R) \times R)$, where R is of arity n, is equivalent to the rooted query $\sigma_{[1,4]}(R)$.

6. 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. The summary tuple t in a temporal tableau query (T, t) must satisfy $t(\mathbf{T}) = [\mathbf{f}, \mathbf{t}]$. The semantics of temporal tableau queries is a natural extension of non-temporal tableau query semantics [1, p. 43].

Definition 9 We assume a set $\mathbf{var}$ of *non-temporal variables*. We assume two *temporal variables* $\mathbf{f}$ and $\mathbf{t}$ not in $\mathbf{var}$. An $\mathbf{f}$-$\mathbf{t}$-*valuation* is a mapping μ from $\{\mathbf{f}, \mathbf{t}\}$ to $\mathbb{P}$ such that $\mu(\mathbf{f}) \leq \mu(\mathbf{t})$ extended to be the identity on $\mathbf{var} \cup \mathbf{dom}$. We define:

$$\mathbb{V} := \{[x, y] \mid x \in \mathbb{P} \cup \{\mathbf{f}\} \text{ and } y \in \mathbb{P} \cup \{\mathbf{t}\}\} .$$

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 ($n \geq 0$) is a set of tableau tuples of arity n. A *(temporal) tableau query* is a pair (T, t) where T is a temporal tableau and t is a tableau tuple such that $t(\mathbf{T}) = [\mathbf{f}, \mathbf{t}]$ and each variable in t also occurs in T; moreover, (T, t) should satisfy the following *safety* requirement: for every $\mathbf{f}$-$\mathbf{t}$-valuation μ, for every non-temporal variable x occurring in t, there must be a tableau tuple $s \in T$ such that x occurs in s and $\mu(s(\mathbf{T})) \neq \{\}$.

Let $\tau = (T, t)$ be a temporal tableau query and I a temporal relation of the same arity as T. An *embedding* of temporal tableau T into I is a valuation ν for the variables occurring in T such that:

- $\nu(x) \in \mathbf{dom}$ if x is a variable in $\mathbf{var}$,
- $\nu(\mathbf{f}), \nu(\mathbf{t}) \in \mathbb{P}$ and $\nu(\mathbf{f}) \leq \nu(\mathbf{t})$, and

- $\nu(T) \sqsubseteq I$, where it is understood that ν is extended to be the identity on constants.

The *output* of τ on input I, denoted $\tau(I)$, is the temporal relation of the same arity as t satisfying: $s \sqsubseteq \tau(I)$ iff there exists an embedding ν of T into I such that $\nu(t) = s$.

Example 2 Consider the temporal tableau query $\tau = (T, t)$ and the temporal relation I:

τ	1	2	**T**	
	a	x	$[\mathbf{f}, 4]$	(T)
	a	x	$[1, \mathbf{t}]$	
		x	$[\mathbf{f}, \mathbf{t}]$	(t)

and

I	1	2	**T**
	a	b	$[0, 9]$
	a	c	$[2, 4]$

.

Consider the valuation $\nu = \{(x, b), (\mathbf{f}, 6), (\mathbf{t}, 7)\}$. Then $\nu(T)$ is the following set of tuples:

$\nu(T)$	1	2	**T**
	a	b	$[6, 4]$
	a	b	$[1, 7]$

$= \{\}$.

Since $\nu(T) \sqsubseteq I$, $\nu(t) = \langle b, [6, 7]\rangle \sqsubseteq \tau(I)$. It can be verified that τ is equivalent to the SPC$^{\text{time}}$ query $\pi_2(\sigma_{[1,4]}(\sigma_{1=a}(R)))$.

Example 3 The following table $\tau = (T, t)$ falsifies the safety requirement of Definition 9 and hence is not a valid temporal tableau query:

τ	1	**T**
	x	$[\mathbf{f}, 3]$
	a	$[6, \mathbf{t}]$
	x	$[\mathbf{f}, \mathbf{t}]$

Consider an embedding ν with $\nu(\mathbf{f}) = 4$ and $\nu(\mathbf{t}) = 5$. Then $\nu(T) \sqsubseteq I$ for every temporal relation I of arity 1, *independent* of the value of $\nu(x)$. Similar to safety in the relational calculus, the safety requirement in Definition 9 ensures that query answers are finite and depend on the content of the input temporal relation.

Note that $\nu(T)$ may not be a temporal relation, as no coalescing is required. It can be easily verified that $t \sqsubseteq \tau(I)$ and $s \preceq t$ imply $s \sqsubseteq \tau(I)$. Temporal tableau queries are monotonic:

Theorem 4 *Let $\tau = (T, t)$ be a temporal tableau query, and I, J temporal relations of the same arity as T. If $I \sqsubseteq J$ then $\tau(I) \sqsubseteq \tau(J)$.*

7. Composition of temporal tableau queries

In Section 8 we will develop an approach that inductively constructs temporal tableau queries corresponding to subexpressions of an SPC$^{\text{time}}$ query. The construction uses composition of temporal tableau queries, 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 temporal tableau queries is slightly more complicated, as illustrated next. Let τ and σ be the temporal tableau queries:

τ	1	2		**T**	
	a	x_1		$[\mathbf{f}, 4]$	
	x_1	v_1		$[1, \mathbf{t}]$	
	a	x_1	x_1	$[\mathbf{f}, \mathbf{t}]$	(t)

and

σ	1	2	3	**T**	
	w_1	y_1	y_2	$[\mathbf{f}, 4]$	(s_1)
	w_2	w_1	y_3	$[1, \mathbf{t}]$	(s_2)
	y_1	y_2	y_3	$[\mathbf{f}, \mathbf{t}]$	

We want to construct the temporal tableau query $\tau \bullet \sigma$ 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 tableau query σ':

σ'	1	2	3	**T**	
	a	y_1	y_1	$[\mathbf{f}, 4]$	(s'_1)
	a	a	a	$[1, \mathbf{t}]$	(s'_2)
	y_1	y_1	a	$[\mathbf{f}, \mathbf{t}]$	

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

Definition 10 Let $\tau = (T, t)$ and $\sigma = (S, s)$ be two temporal tableau queries, where t and S are both of arity n. Let $S = \{s_1, \ldots, s_m\}$.

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

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 $x \equiv x'$.

Definition 11 Let $\tau = (T, t)$ and $\sigma = (S, s)$ be two temporal tableau queries, where t and S are of the same arity. Let $S = \{s_1, \dots, s_m\}$. Assume θ is a unifier for t and σ; if no such unifier exists, the composition is undefined. For each $i \in \{1, \dots, m\}$, 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:

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

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

Example 4 For τ and σ as introduced at the beginning of this section:

$$\begin{array}{c|cccc}
\tau \bullet \sigma & 1 & 2 & & \mathbf{T} \\ \hline
 & a & y_1 & & [\mathbf{f}, 4] \\
 & y_1 & v_{11} & & [1, 4] \\
 & a & a & & [1, 4] \\
 & a & v_{12} & & [1, \mathbf{t}] \\ \hline
 & y_1 & y_1 & a & [\mathbf{f}, \mathbf{t}]
\end{array} .$$

Composition is well-defined:

Proposition 1 *Let $\tau = (T, t)$ and $\sigma = (S, s)$ be two temporal tableau queries, where t and S are of the same arity. Let I be a temporal relation of the same arity as T. If $\tau \bullet \sigma$ is defined then $\tau \bullet \sigma(I) = \sigma(\tau(I))$; otherwise $\sigma(\tau(I)) = \{\}$.*

8. Temporal tableaux of rooted queries

We give an inductive definition that translates a rooted SPC$^{\text{time}}$ query into a temporal tableau query. The crux lies in the translation of temporal selection $\sigma_{[p,q]}(\cdot)$ and projection, where the introduction of two tuples is needed.

Definition 12 For each rooted SPC$^{\text{time}}$ query $Q(R)$, $tab(Q)$ is recursively defined as follows. Assume $Q(R)$ is of arity n in what follows.

1. $tab(R) :=$ $\begin{array}{|ccc c} 1 & \dots & n & \mathbf{T} \\ \hline x_1 & \dots & x_n & [\mathbf{f}, \mathbf{t}] \\ \hline x_1 & \dots & x_n & [\mathbf{f}, \mathbf{t}] \end{array}$, where R is a relation variable of arity n.

2.

$$tab(\sigma_{[p,q]}(Q)) := tab(Q) \bullet \begin{array}{|cccc} 1 & \dots & n & \mathbf{T} \\ \hline x_1 & \dots & x_n & [\mathbf{f}, q] \\ x_1 & \dots & x_n & [p, \mathbf{t}] \\ \hline x_1 & \dots & x_n & [\mathbf{f}, \mathbf{t}] \end{array} .$$

3. $tab(\sigma_{i=j}(Q)) := tab(Q)\bullet$

$$\begin{array}{|cccccccc} 1 & \dots & i-1 & i & i+1 & \dots & n & \mathbf{T} \\ \hline x_1 & \dots & x_{i-1} & x_j & x_{i+1} & \dots & x_n & [\mathbf{f}, \mathbf{t}] \\ \hline x_1 & \dots & x_{i-1} & x_j & x_{i+1} & \dots & x_n & [\mathbf{f}, \mathbf{t}] \end{array} .$$

4. $tab(\sigma_{i=a}(Q)) := tab(Q)\bullet$

$$\begin{array}{|cccccccc} 1 & \dots & i-1 & i & i+1 & \dots & n & \mathbf{T} \\ \hline x_1 & \dots & x_{i-1} & a & x_{i+1} & \dots & x_n & [\mathbf{f}, \mathbf{t}] \\ \hline x_1 & \dots & x_{i-1} & a & x_{i+1} & \dots & x_n & [\mathbf{f}, \mathbf{t}] \end{array} .$$

5. $tab(\pi_{j_1,\dots,j_m}(Q)) := tab(Q)\bullet$

$$\begin{array}{|cccccccccl} 1 & 2 & \dots & j_1 & \dots & j_m & \dots & n & \mathbf{T} & \\ \hline y_1 & y_2 & \dots & x_{j_1} & \dots & x_{j_m} & \dots & y_n & [\mathbf{f}, q] & (u) \\ z_1 & z_2 & \dots & x_{j_1} & \dots & x_{j_m} & \dots & z_n & [p, \mathbf{t}] & (v) \\ \hline & & & x_{j_1} & \dots & x_{j_m} & & & [\mathbf{f}, \mathbf{t}] & (w) \end{array} ,$$

where $[p, q]$ is ubiquitous for Q (typically the maximal ubiquitous interval). More precisely, the arity of w is m; $u(j_i) = v(j_i) = w(i)$ for each $i \in \{1, 2, \dots, m\}$, and all constants are pairwise distinct otherwise.

6. $Q = Q_1 \times Q_2$; assume Q_1 and Q_2 are of arity n_1 and n_2 respectively. Let $tab(Q_1) = (T_1, \langle x_1, \dots, x_{n_1}, [\mathbf{f}, \mathbf{t}]\rangle)$ and $tab(Q_2) = (T_2, \langle y_1, \dots, y_{n_2}, [\mathbf{f}, \mathbf{t}]\rangle)$. Assume without loss of generality that T_1 and T_2 have no variables in common except for $\mathbf{f}$ and $\mathbf{t}$. Then $tab(Q) := (T_1 \cup T_2, \langle x_1, \dots, x_{n_1}, y_1, \dots, y_{n_2}, [\mathbf{f}, \mathbf{t}]\rangle)$.

Example 5 Let R be of arity 1. Let $Q = \pi_1(\sigma_{[1,4]}(R) \times \sigma_{[2,3]}(R))$. Since

$$tab(\sigma_{[1,4]}(R)) = \begin{array}{|cc} 1 & \mathbf{T} \\ \hline x & [\mathbf{f}, 4] \\ x & [1, \mathbf{t}] \\ \hline x & [\mathbf{f}, \mathbf{t}] \end{array}$$

and

$$tab(\sigma_{[2,3]}(R)) = \begin{array}{|cc} 1 & \mathbf{T} \\ \hline y & [\mathbf{f}, 3] \\ y & [2, \mathbf{t}] \\ \hline y & [\mathbf{f}, \mathbf{t}] \end{array} ,$$

it follows

$$tab(\sigma_{[1,4]}(R) \times \sigma_{[2,3]}(R)) = \begin{array}{|ccc} 1 & & \mathbf{T} \\ \hline x & & [\mathbf{f}, 4] \\ x & & [1, \mathbf{t}] \\ y & & [\mathbf{f}, 3] \\ y & & [2, \mathbf{t}] \\ \hline x & y & [\mathbf{f}, \mathbf{t}] \end{array} .$$

Since $[2,3]$ is ubiquitous for $\sigma_{[1,4]}(R) \times \sigma_{[2,3]}(R)$,

$$tab(Q) = tab(\sigma_{[1,4]}(R) \times \sigma_{[2,3]}(R)) \bullet \begin{array}{|ccc} 1 & 2 & \mathbf{T} \\ \hline z & u & [\mathbf{f},3] \\ z & v & [2,\mathbf{t}] \\ \hline z & & [\mathbf{f},\mathbf{t}] \end{array}$$

$$= \begin{array}{|cc} 1 & \mathbf{T} \\ \hline z & [\mathbf{f},4] \\ z & [1,3] \\ u & [\mathbf{f},3] \\ u & [2,3] \\ z & [2,4] \\ z & [1,\mathbf{t}] \\ v & [2,3] \\ v & [2,\mathbf{t}] \\ \hline z & [\mathbf{f},\mathbf{t}] \end{array}$$

The main result of this paper can now be stated.

Theorem 5 *For each rooted* $\mathrm{SPC}^{\mathrm{time}}$ *query* $Q(R)$, $tab(Q) \equiv Q$.

For the non-temporal domain, it is well-known that satisfiable SPC queries and tableau queries are equivalent. For the temporal domain, the expressive powers of satisfiable $\mathrm{SPC}^{\mathrm{time}}$ and temporal tableau queries do not coincide.

Theorem 6 *There exists a satisfiable* $\mathrm{SPC}^{\mathrm{time}}$ *query* $Q(R)$ *that is equivalent to no temporal tableau query.*

Theorem 7 *There exists a temporal tableau query that is equivalent to no* $\mathrm{SPC}^{\mathrm{time}}$ *query.*

9. The Homomorphism Theorem

We give a temporal variant of the famous Homomorphism Theorem and illustrate how it can be used for temporal query optimization.

Definition 13 Let $\tau = (T, t)$ and $\sigma = (S, s)$ be two temporal tableau queries such that T and S have the same arity, as well as t and s. A *homomorphism* from σ to τ under $\mathbf{f}$-$\mathbf{t}$-valuation μ is a substitution θ such that $\theta(\mu(S)) \sqsubseteq \mu(T)$ and $\theta(\mu(s)) = \mu(t)$.

Theorem 8 *Let* $\tau = (T, t)$ *and* $\sigma = (S, s)$ *be two temporal tableau queries such that* T *and* S *have the same arity, as well as* t *and* s. *Then* $\tau \sqsubseteq \sigma$ *iff for every* $\mathbf{f}$-$\mathbf{t}$-*valuation* μ, *there exists a homomorphism from* σ *to* τ *under* μ.

In the following example, Theorem 8 is used for temporal query optimization.

Example 6 Consider the temporal tableau queries τ and σ as follows:

$$\begin{array}{c|cc} \tau & 1 & \mathbf{T} \\ \hline & z & [\mathbf{f},4] \\ & z & [1,\mathbf{t}] \\ \hline & z & [\mathbf{f},\mathbf{t}] \end{array} \quad \text{and} \quad \begin{array}{c|cc} \sigma & 1 & \mathbf{T} \\ \hline & z & [\mathbf{f},4] \\ & z & [1,3] \\ & u & [\mathbf{f},3] \\ & u & [2,3] \\ & z & [2,4] \\ & z & [1,\mathbf{t}] \\ & v & [2,3] \\ & v & [2,\mathbf{t}] \\ \hline & z & [\mathbf{f},\mathbf{t}] \end{array}$$

Then $\theta = \{(u,z),(v,z)\}$ is a homomorphism of σ to τ under every $\mathbf{f}$-$\mathbf{t}$-valuation. To see this, recall that $[\mathbf{f},4] \cup [1,\mathbf{t}] \supseteq [1,4]$ (Theorem 2). Hence, $\tau \sqsubseteq \sigma$. On the other hand, the identity is a homomorphism of τ to σ under every $\mathbf{f}$-$\mathbf{t}$-valuation. It follows $\tau \equiv \sigma$. Recall from Example 5 that $\sigma = tab(\pi_1(\sigma_{[1,4]}(R) \times \sigma_{[2,3]}(R)))$, where R is a relation variable of arity 1. Moreover, $\tau = tab(\sigma_{[1,4]}(R))$. By Theorem 5, it follows that $\pi_1(\sigma_{[1,4]}(R) \times \sigma_{[2,3]}(R))$ can be equivalently rewritten (or "optimized") as $\sigma_{[1,4]}(R)$.

10. Discussion

Recall the questions asked in Section 1. Can the construct of tableau query be generalized for temporal relations? We have shown that two temporal variables ($\mathbf{f}$ and $\mathbf{t}$) are sufficient to express all rooted $\mathrm{SPC}^{\mathrm{time}}$ queries. Does the absence of the normal form in the temporal algebra hinder the "temporalizing" of tableau queries? Somewhat surprisingly, even though a rooted $\mathrm{SPC}^{\mathrm{time}}$ query can contain multiple nested projections none of which can be removed, its equivalent temporal tableau query needs no nesting. Every projection leads to two tuples being inserted in the inductive construction of Section 8, implying that the size of the equivalent tableau query is exponential in the number of projections of the original $\mathrm{SPC}^{\mathrm{time}}$ query.

Our study revealed a number of open questions:

1. Find a syntactic characterization of rooted queries (if it exists). Characterize the class of all $\mathrm{SPC}^{\mathrm{time}}$ queries that are equivalent to temporal tableau queries. Is every $\mathrm{SPC}^{\mathrm{time}}$ query in that class equivalent to a rooted $\mathrm{SPC}^{\mathrm{time}}$ query?

2. Extend the construct of temporal tableau query so that it can also express all unrooted queries (cf. Theorem 6). This seems to imply tableaux that can accommodate a variable number of tuples.

3. Extend the $\mathrm{SPC}^{\mathrm{time}}$ algebra so that it can express all temporal tableau queries (cf. Theorem 7).

4. Find procedures to (1) determine whether a given tableau query is equivalent to some $\mathrm{SPC}^{\mathrm{time}}$ query, and (2) build the equivalent $\mathrm{SPC}^{\mathrm{time}}$ query if it exists.

5. Given two temporal tableau queries σ and τ, find an algorithm to decide whether there exists a homomorphism from σ to τ under every $\mathbf{f}$-$\mathbf{t}$-valuation.

We conclude with a number of open-ended considerations that deserve future attention.

Timestamping in temporal tableau queries is effectuated by intervals. An alternative approach would be to use point based timestamping extended with inequality constraints for representing intervals. In [5] and [4], the extension of tableau queries with inequality comparisons is studied in a general setting that does not focus on time. Interestingly, for general inequality comparisons, query containment does not imply the existence of a homomorphism. Note that Theorem 8 does not quite express the "homomorphism property" for temporal tableau queries, as our construct of homomorphism is still relative to the concept of $\mathbf{f}$-$\mathbf{t}$-valuation.

We used discrete time and closed intervals. The extension to dense time requires the introduction of open intervals. For example, consider the temporal tableau query $\tau = (T, t)$ and the temporal relation I:

$$\tau\ \begin{array}{|cc} 1 & \mathbf{T} \\ \hline x & [\mathbf{f},\mathbf{t}] \\ a & [\mathbf{f},5] \\ \hline x & [\mathbf{f},\mathbf{t}] \end{array} \quad \text{and} \quad I\ \begin{array}{|cc} 1 & \mathbf{T} \\ \hline b & [1,9] \end{array} .$$

When temporal variables range over the real numbers, the answer $\tau(I)$ should be $\{\langle b,]5,9]\rangle\}$, showing a half-open interval.

We only considered a single temporal dimension, representing valid time. The extension with transaction time seems straightforward for queries that do not compare valid time with transaction time. An interesting problem is to extend the current framework for queries where valid and transaction time are not independent.

Finally, it is interesting to ask whether temporal tableaux can serve as a unifying framework for studying temporal dependencies, in the same way as classical tableaux provide a unified understanding of most non-temporal dependencies raised in the literature.

References

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

[2] C. Date. *An Introduction to Database Systems*. Addison-Wesley, seventh edition, 2000.

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

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

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

[7] 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.

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

[9] 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.

[10] R. T. Snodgrass. TSQL2. In C. Zaniolo, S. Ceri, C. Faloutsos, R. Snodgrass, V. S. Subrahmanian, and R. Zicari, editors, *Advanced Database Systems*, chapter 6, pages 127–147. 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.