3 ALGEBRA OF TIMED RELATIONS

Ne içindeyim zamanın, ne de büsbütün dışında.
Yekpare, geniş bir anın parçalanmaz akışında.

I am neither inside time, nor am I completely outside.
In the indivisible flow of a moment, atomic and wide.

— Poems, Ahmet Hamdi Tanpinar

This chapter introduces the algebra of timed relations. We define *timed relations* to be finitely representable subsets of the set of all time periods. We are interested in various algebraic (boolean, sequential, and temporal) operations on timed relations represented by symbolic and geometric means.

The study of the algebra of logic and relations began with Boole and DeMorgan in the nineteenth century and continued by Peirce and Schröder [103]. Geometric representations of relations can be traced back to Tarski's calculus of relations [105]. Other related classical works include Blake's canonical expressions in boolean algebras [21], Stone's representation theorem [104], and boolean algebras with operators [57]. The latter is closely related to modal (temporal) logics [19, 20], thus it completes a cycle of mathematical results.

We focus on concrete geometric representations specific to timed relations in this chapter. We propose algorithms to compute boolean, sequential, and temporal operations over such representations based on techniques from computational geometry. In short, we establish a computational framework here for the following chapters, which relies on classical results of the algebra of logic and relations.

3.1 DEFINITIONS AND NORMAL FORMS

We start by defining the underlying time domain we will use throughout the thesis, which is a set of time points on a dense, linear, and bounded time line.

Definition 3.1 (Time domain)**.** *A (dense, linear, bounded) time domain* T *is a dense interval of time points with rational bounds admitting a strict linear order* $<$.

We consider $T = (0, d)$ to be an interval of real or rational numbers denoting time points on the time axis. Although we have just mentioned time points, we now abandon them in favor of time periods defined as follows.

Definition 3.2 (Time period)**.** *A time period* (t, t') *is a pair of begin and end boundaries on a time domain* T *such that* $t < t'$ *with a non-zero duration of* $t' - t$. *We denote the set of all time periods over* T *by* $\Omega(T)$.

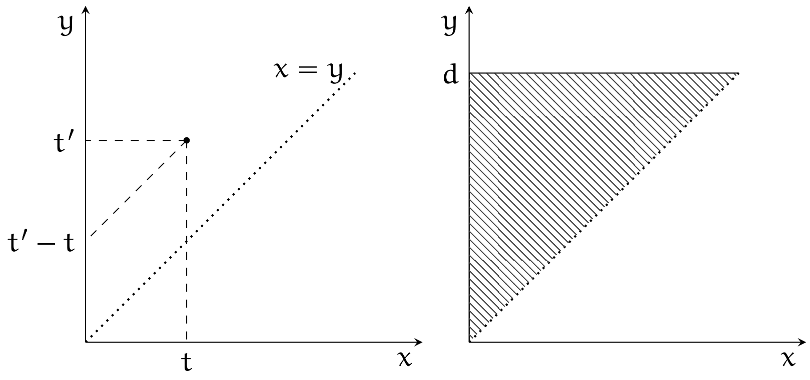

Figure 4: Geometric representations of a time period (t, t') and the set $\Omega(T)$.

We say that a time period (t_1, t_1') meets another time period (t_2, t_2') if the end of the first equals to the beginning of the second such that $t_1' = t_2$. A sequence $S = (t_0, t_1), (t_1, t_2), \ldots, (t_{n-1}, t_n)$ of meeting time periods is simply called a sequence of time periods. We say that the sequence S begins at t_0, ends at t_n, and has a duration of $t_n - t_0$ and a length of n.

Geometrically speaking, a time period (t, t') can be viewed as a single point on the standard two-dimensional xy-plane. On the left of Figure 4, we illustrate attributes of a time period (t, t'), its beginning t, its end t', and its duration $t' - t$. Then we are interested in a very specific set of linear inequalities that correspond to constraints on beginnings, endings, and durations of time periods, often called vertical, horizontal, and diagonal half-planes. In particular, the set of all time periods $\Omega(T)$ can be seen as a (triangular) set of points $\{(t, t') \mid t \geqslant 0,\ t' \leqslant d,\ t' - t > 0\}$ on the plane as depicted on the right of Figure 4.

In the following, we closely relate sets of time periods to Boolean functions [24, 34]. Boolean (set-theoretic) operations of union ($\cup$), intersection ($\cap$), and complementation ($\overline{\ \ }$) as well as the inclusion relation ($\subseteq$) over sets of time periods are defined as usual with the empty set $\varnothing$ and the universal set $\Omega(T)$. We call a set of time periods a *timed relation* if and only if it can be expressed as a finite boolean formula over vertical, horizontal, and diagonal half-planes.

Definition 3.3 (Timed Relation)**.** *A timed relation $Z \subseteq \Omega(T)$ is a (possibly uncountable) set of time periods that can be represented on the xy-plane by a boolean combination of finitely many half-planes having one of six forms (1) $x \prec c$, (2) $y \prec c$, (3) $y - x \prec c$, (4) $c \prec y - x$, (5) $c \prec y$, and (6) $c \prec x$ with rational constants where $\prec \in \{<, \leqslant\}$.*

Observe that the empty set, $\Omega(T)$, and all finite sets of time periods are timed relations. A timed relation that can be formed only by intersections is called a convex timed relation. We denote by $\mathcal{Z}$ and $\mathcal{Z}^{\cap}$ the set of all timed relations and the set of all convex timed relations over $\Omega(T)$, respectively. We differentiate the six types of half-planes in the definition by annotating superscripted numbers $(1-6)$ such as h^1. The complement of an open [closed] half-plane h^k is a closed [open] half-plane h^{7-k} with the same constant c. Intersections or unions of any number of half-planes $h^k_1, \dots, h^k_n$ of the same type would be implied by one of the half-planes h^k_i, $i \in 1, \dots, n$. Therefore, every convex timed relation $z \in \mathcal{Z}^\cap$ can be formed by an intersection of six half-planes $h^1, h^2, h^3, h^4, h^5, h^6$ of each type such that

$$\bigcap_{k=1\dots6} h^k = \{(x, y) \mid c_6 \prec x \prec c_1 \ \land \ c_5 \prec y \prec c_2 \ \land \ c_4 \prec y - x \prec c_3\}$$

where $c_1, \dots, c_6 \in \mathbb{R} \cup \{-\infty, +\infty\}$. Consequently, we can represent a convex timed relation as a six-tuple $(h^1, h^2, h^3, h^4, h^5, h^6)$ of half-planes. Notice that the inequalities defining these half-planes are not totally independent of each other and the arithmetic addition of other two certain inequalities may imply a tighter constraint than the one in the representation. Every non-empty convex timed relation $z \in \mathcal{Z}^\cap$ has a unique normal representation such that all constraints are tight. More precisely, given a representation $(h^1, h^2, h^3, h^4, h^5, h^6)$ of z that the tight representation of z can be computed as follows:

$$\begin{aligned}\textsc{tighten}(z) = \Big(& h^1 \cap (h^2 + h^4), \\ & h^2 \cap (h^1 + h^3), \\ & h^3 \cap (h^2 + h^6), \\ & h^4 \cap (h^1 + h^5), \\ & h^5 \cap (h^4 + h^6), \\ & h^6 \cap (h^3 + h^5)\Big)\end{aligned}$$

where $+$ denotes arithmetic addition of inequalities. In Figure 5, we illustrate the most general (hexagon) case for a convex timed relation where one can see, for example, that two half-planes $h^1 : x < c_1$ and $h^3 : y - x \leqslant c_3$ imply a constraint $h^1 + h^3 : y < c_1 + c_3$, which can imply or be implied by h^2. Importantly, an inclusion test between two convex timed relations z_1 and z_2 can be performed over their tight representations such that

$$z_1 \subseteq z_2 \longleftrightarrow \bigwedge_{i=1,\dots,6} h^i_1 \subseteq h^i_2$$

From now on, we consider all representations of convex timed relations to be tightened according to the definition above and we use the term *zone* both for a convex timed relation and its tight representation. We say that a zone z_1 is implied by another zone z_2 if $z_1 \subseteq z_2$. Note that zones (possibly in higher dimensions) are commonly employed for timed systems research and admit efficient data structures called difference bound matrices (DBMs) [22, 37]. Here we use many two-dimensional zones to represent time periods rather than one or a few high-dimensional zones to represent clock valuations.

We represent a timed relation as a finite union of zones similar to the disjunctive normal form of Boolean functions. By definition and by DeMorgan's laws, every timed relation Z can be represented by a union over a finite set of non-empty zones $R_Z = \{z_1, z_2, \dots, z_n\}$ such

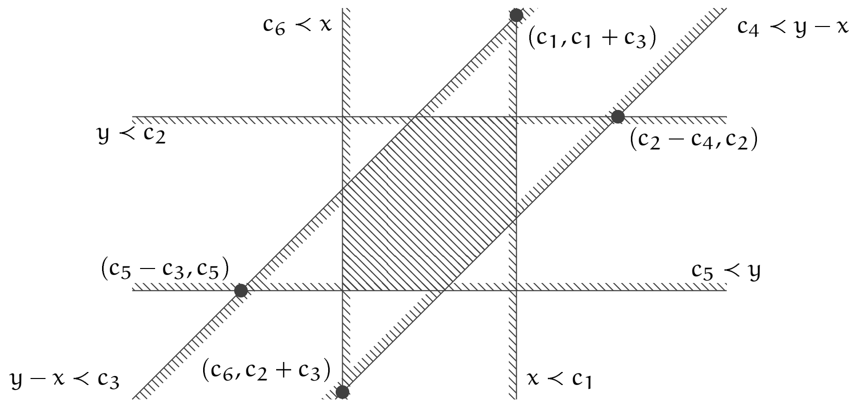

Figure 5: Dependencies between constraints of a convex timed relation.

that $Z = z_1 \cup z_2 \cup \cdots \cup z_n$. Colloquially, we say *timed relation* when we want to emphasize semantic aspects whereas *union of zones* to emphasize syntactic aspects. Now we define some important properties of representations for timed relations.

Definition 3.4 (Absorption)**.** *A union of zones R_Z is absorptive if and only if no zone in R_Z is implied by any other zone in R_Z.*

For any union of zones R_Z, we can obtain an equivalent absorptive union of zones, denoted by $\textsc{absorb}(R_Z)$, by removing all absorbed zones from the representation. Obviously, there may be different representations of a timed relation Z but an absorptive representation $\textsc{absorb}(R_Z)$ for a given R_Z is unique. Unless specified otherwise, we consider all unions of zones to be absorptive throughout the thesis. We then adapt Blake's syllogistic theory of Boolean functions [21, 24] to the case of timed relations in a direct fashion in the following.

Definition 3.5 (Syllogism)**.** *A union of zones R_Z is syllogistic if and only if every zone $z \subseteq Z$ is included in some zone in R_Z.*

Let us now define a syntactic inclusion test between two unions of zones R_{Z_1} and R_{Z_2} as follows:

$$R_{Z_1} \sqsubseteq R_{Z_2} \longleftrightarrow \forall z_1 \in R_{Z_1}.\ \exists z_2 \in R_{Z_2}.\ z_1 \subseteq z_2$$

An important result for syllogistic sets of zones is that syntactic inclusion is implied by semantic inclusion. That is to say, an inclusion test $Z_1 \subseteq Z_2$ between two timed relations Z_1 and Z_2 can be replaced by a syntactic inclusion test between their representations if R_{Z_2} is syllogistic.

Lemma 3.1. *Let R_{Z_1} and R_{Z_2} be two unions of zones. If R_{Z_2} is syllogistic, we have the equivalence $Z_1 \subseteq Z_2 \longleftrightarrow R_{Z_1} \sqsubseteq R_{Z_2}$.*

Proof. Assume R_{Z_2} is syllogistic. One direction ($\leftarrow$) of the equivalence is trivial. In the other direction, assume a zone z of R_{Z_1} is not included in any zone of R_{Z_2} and then it implies $z \nsubseteq Z_2$ by definition. Hence we have ($\rightarrow$) by contraposition. $\square$

It is also easily seen that ABSORB(R_Z) is syllogistic if and only if R_Z is syllogistic. Below we denote by $R_{Z_1} \cap R_{Z_2}$ a union of zones developed by intersecting zones from both sets using the distributive laws.

Lemma 3.2. *Let $R_{Z_1}, \ldots, R_{Z_n}$ be syllogistic unions of zones. Then we have that $R_{Z_1} \cap \cdots \cap R_{Z_n}$ is syllogistic.*

Proof. Let z be a zone of $R_{Z_1} \cap \cdots \cap R_{Z_n}$. Then $z \subseteq Z_i$ for $i = 1 \ldots n$. Since R_{Z_i} are syllogistic, there exists a zone $z'_i \in R_{Z_i}$ such that $z \subseteq z'_i$ and then we have $z \subseteq z'_1 \cap \cdots \cap z'_n$. See that $z'_1 \cap \cdots \cap z'_n$ is a zone of $R_{Z_1} \cap \cdots \cap R_{Z_n}$, hence $R_{Z_1} \cap \cdots \cap R_{Z_n}$ is syllogistic. □

Next we define a canonical representation for timed relations, which is the analogue of Blake's canonical form for Boolean functions. The maximal normal form of a timed relation Z is defined to be a union of all maximal zones as follows.

Definition 3.6 (Maximal Zone)**.** *Let Z be a timed relation. We say that a zone $z \subseteq Z$ is a maximal zone of Z if there is no other zone z' that satisfies $z \subset z' \subseteq Z$.*

Definition 3.7 (Maximal Normal Form)**.** *A union of zones R_Z is in the maximal normal form if it is the union of all maximal zones of Z.*

Lemma 3.3. *A union of zones R_Z is in the maximal normal form of Z if and only if it is absorptive and syllogistic.*

Proof. The maximal normal form of Z is absorptive and syllogistic by definition. For the other direction, suppose R_Z is syllogistic and absorptive. Let z be a maximal zone of Z. Then z has to be in R_z since there is no zone z' such that $z \subset z' \subseteq Z$. Since R_Z is absorptive, it does not contain any zone z'' such that $z'' \subset z$. Hence R_Z contains all maximal zones, and no other zones. □

3.2 OPERATIONS ON TIMED RELATIONS

When describing operations on timed relations, we first give the semantic definition. Then we demonstrate the operation over single zones and extend it towards unions of zones. We overload operators for both semantic and syntactic operations if there is no ambiguity.

Duration restriction. We consider that the operation of duration restriction is a first-class operation for the algebra of timed relations. A timed relation Z whose elements restricted to have a duration value within an interval $[a, b]$ of non-negative reals is denoted by $\langle Z \rangle_{[a,b]}$ and defined as follows.

$$\langle Z \rangle_{[a,b]} = \{(x, y) \mid (x, y) \in Z \text{ and } y - x \in [a, b]\}$$

Over a zone, we directly apply duration restriction by letting

$$\langle z \rangle_{[a,b]} = \text{TIGHTEN}(h^1, h^2, h^3 \cap (y - x \leqslant b), h^4 \cap (a \leqslant y - x), h^5, h^6)$$

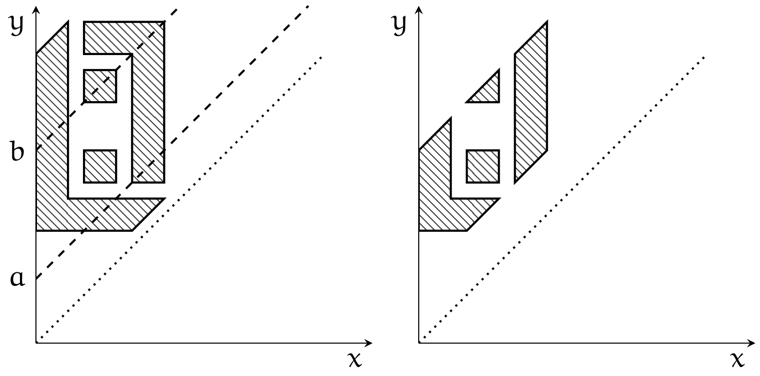

Figure 6: Duration restriction.

and extend it towards unions of zones as

$$\langle R_Z \rangle_{[a,b]} = \{\langle z \rangle_{[a,b]} \mid z \in R_Z\}$$

In Figure 6, we depict a timed relation Z on the left and $\langle Z \rangle_{[a,b]}$ on the right, which can be viewed as a diagonal slice of Z on the plane.

Intersection. The operation of intersection between timed relations is defined as usual.

$$Z_1 \cap Z_2 = \{(x, y) \mid (x, y) \in Z_1 \text{ and } (x, y) \in Z_2\}$$

Observe that duration restriction is a special case of intersection with a (constant) timed relation $C = \langle \Omega(T) \rangle_{[a,b]}$ such that $\langle Z \rangle_{[a,b]} = Z \cap C$. We intersect two zones by letting

$$z_1 \cap z_2 = \textsc{tighten}(h_1^1 \cap h_2^1, h_1^2 \cap h_2^2, h_1^3 \cap h_2^3, h_1^4 \cap h_2^4, h_1^5 \cap h_2^5, h_1^6 \cap h_2^6)$$

and extend it towards union of zones as

$$R_{Z_1} \cap R_{Z_2} = \{z_1 \cap z_2 \mid z_1 \in R_{Z_1} \text{ and } z_2 \in R_{Z_2}\}$$

which indicates a quadratic complexity for the worst case.

Complementation. The complement $\overline{Z}$ of a timed relation Z with respect to the universal set $\Omega(T)$ is given as follows:

$$\overline{Z} = \{(x, y) \in \Omega(T) \mid (x, y) \notin Z\}$$

See that the complement of a timed relation Z is unique and the double complement of Z is equivalent to Z. The statements $Z \cap \overline{Z} = \varnothing$ and $Z \cup \overline{Z} = \Omega(T)$ hold for every timed relation Z. Since both DeMorgan laws hold for timed relations, we complement a zone z and a union of zones R_Z, respectively as follows.

$$\overline{R_z} = \{\overline{h_z^1}, \overline{h_z^2}, \overline{h_z^3}, \overline{h_z^4}, \overline{h_z^5}, \overline{h_z^6}\} \quad (2) \qquad \overline{R_Z} = \bigcap_{z \in R_Z} \overline{R_z} \quad (3)$$

Observe that the complement of a single zone as in Equation 2 is a syllogistic union of zones. Then, by Lemma 3.2, we also have that the complement of a timed relation as in Equation 3 are syllogistic. Moreover we can obtain the maximal normal form of $\overline{Z}$ by removing absorbed zones from $\overline{R_Z}$. It follows that double complementation can be used to obtain the maximal normal form of any timed relation.

Finally we note that Equation 3 can be developed into a union of zones in an incremental and more efficient manner such that

$$\overline{R^i_Z} = \text{ABSORB}\left(\overline{R^{i-1}_Z} \cap \overline{R_{z_i}}\right)$$

where $\overline{R^i_Z}$ is the complement of the subset $\{z_1, z_2, \ldots, z_i\}$ of a union of zones with n elements for $1 \leqslant i \leqslant n$ and $\overline{R^0_Z} = \Omega(T)$.

Union. Given two timed relations Z_1 and Z_2, the union of two timed relations is defined as usual.

$$Z_1 \cup Z_2 = \{(x, y) \mid (x, y) \in Z_1 \text{ or } (x, y) \in Z_2\}$$

The class of zones (convex timed relations) is not closed under union and the operation $R_{Z_1} \cup R_{Z_2}$ simply corresponds to that the union of member zones of both representations. We note, however, timed relations of $R_{Z_1} \cup R_{Z_2}$ and $\overline{\overline{R_{Z_1}} \cap \overline{R_{Z_2}}}$ are equivalent but they may have different representations. The latter would produce the maximal normal form of $Z_1 \cup Z_2$ whereas the former does not necessarily.

Composition. Given two timed relations Z_1 and Z_2, their (sequential) composition, equivalently their concatenation, is defined by

$$Z_1 \circ Z_2 = \{(x, y) \mid \exists r.\ (x, r) \in Z_1 \text{ and } (r, y) \in Z_2\}$$

In Figure 7 we illustrate the composition of two singleton timed relations $\{(t, t'')\} \circ \{(t'', t')\} = \{(t, t')\}$ on the left side. Notice that a time period (t_1, t_1') can be sequentially composed (concatenated) with another time period (t_2, t_2') only if the end of the first meets the beginning of the second such that $t_1' = t_2$. Otherwise the composition $\{(t_1, t_1')\} \circ \{(t_2, t_2')\} = \varnothing$ for $t_1' \neq t_2$. Next we have that the class of zones (convex timed relations) is closed under composition.

Proposition 3.4. *The composition of two zones is a zone.*

Proof. Following the semantics of composition we have

$$(t, t') \in z_1 \circ z_2 \text{ iff } \exists t''.\ t < t'' < t',\ (t, t'') \in z_1, \text{ and } (t'', t') \in z_2$$

which translates to $\exists t'' \in (t, t')$ such that

$$\left\{\begin{array}{ccccc} \underline{b}_1 & \prec & t & \prec & \overline{b}_1 \\ \underline{e}_1 & \prec & t'' & \prec & \overline{e}_1 \\ \underline{d}_1 & \prec & t'' - t & \prec & \overline{d}_1 \end{array}\right\} \text{ and } \left\{\begin{array}{ccccc} \underline{b}_2 & \prec & t'' & \prec & \overline{b}_2 \\ \underline{e}_2 & \prec & t' & \prec & \overline{e}_2 \\ \underline{d}_2 & \prec & t' - t'' & \prec & \overline{d}_2 \end{array}\right\}$$

By eliminating the quantifier, we obtain that $z_1 \circ z_2$ equals to a zone

$$\left\{\begin{array}{ccccc} \max(\underline{b}_1,\ \underline{b}_2 - \overline{d}_1) & \prec & t & \prec & \min(\overline{b}_1,\ \overline{b}_2 - \underline{d}_1) \\ \max(\underline{e}_2,\ \underline{e}_1 + \underline{d}_2) & \prec & t' & \prec & \min(\overline{e}_2,\ \overline{e}_1 + \overline{d}_2) \\ \underline{d}_1 + \underline{d}_2 & \prec & t' - t & \prec & \overline{d}_1 + \overline{d}_2 \end{array}\right\}$$

if $\underline{e}_1 \prec \overline{b}_2$ and $\underline{b}_2 \prec \overline{e}_1$, the empty set otherwise. □

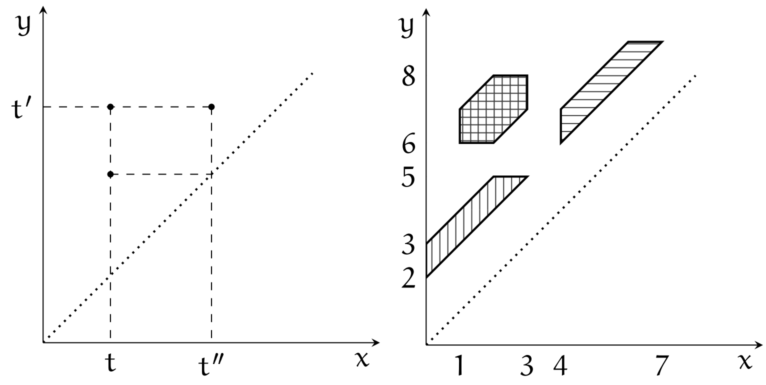

Figure 7: Composition of timed relations.

Following the constructive proof above, we have an algorithm to compute concatenation between two zones and extend it towards unions of zones as

$$R_{Z_1} \circ R_{Z_2} = \{z_1 \circ z_2 \mid z_1 \in R_{Z_1} \text{ and } z_2 \in R_{Z_2}\}$$

In Figure 7, we illustrate the composition operation

$$z_1 \circ z_2 = \{(x,y) \mid 1 \leqslant x \leqslant 3 \ \cap\ 6 \leqslant y \leqslant 8 \ \cap\ 4 \leqslant y - x \leqslant 6\}$$

of two zones $z_1 = \{(x,y) \mid 0 \leqslant x \leqslant 3 \ \cap\ 2 \leqslant y \leqslant 5 \ \cap\ 2 \leqslant y - x \leqslant 3\}$ and $z_2 = \{(x,y) \mid 4 \leqslant x \leqslant 7 \ \cap\ 6 \leqslant y \leqslant 9 \ \cap\ 2 \leqslant y - x \leqslant 3\}$ on the right side.

Transitive Closure. The i-th power of a timed relation for $i \geqslant 1$ and transitive closure Z^+ are defined, respectively as follows.

$$Z^i = \underbrace{Z \circ Z \circ \cdots \circ Z}_{i \text{ times}} \qquad\qquad Z^+ = \bigcup_{i \geqslant 1} Z^i$$

Now we show that the transitive closure of any timed relation is representable by a finite number of composition operation, thus timed relations are closed under transitive closure operation.

Proposition 3.5. *Over a bounded time domain* $T = (0, d)$, *for all* $Z \in \mathcal{Z}$, *there exists an integer* k *that satisfies the equality* $Z^+ = \bigcup_{1 \leqslant i \leqslant k} Z^i$.

Proof. Let H be a finite partition of $\Omega(T)$ with a form of right-triangular grid generated by integer multiples ($t_i = ic$ for $i = 0, \ldots, m = d/c$) of the greatest common divisor c and constant parts of inequalities defining Z and $\Omega(T)$. Each cell in H defines a set of equivalent time segments with respect to Z^+ membership such that $\forall (x,y), (x',y') \in C.\ (x,y) \in Z^+ \leftrightarrow (x',y') \in Z^+$. In particular we use the equivalence in cells $t_i < x < y < t_{i+1}$ to show the original claim in the following.

Assume a segment $(r_0, r_k) \in Z^{k+1}$ and then there exists a sequence of time segments $(r_0, r_1), (r_1, r_2), \ldots (r_k, r_{k+1})$ such that each segment $(r_i, r_{i+1}) \in Z$. When $k > 3m - 2$, by the pigeonhole principle, there are two consecutive segments, denoted by $(r_{i-1}, r_i), (r_i, r_{i+1})$, within the same cell $t_i < x < y < t_{i+1}$. By the equivalence inside a cell, the sequence $(r_{i-1}, r_i), (r_i, r_{i+1})$ can be replaced by (r_{i-1}, r_{i+1}) thus $(r_0, r_k) \in Z^k$, and we conclude that the test $Z^{k+1} \subseteq Z^k$ holds and we have that $Z^+ = \bigcup_{1 \leqslant i \leqslant k} Z^i$. $\square$

Temporal modalities. An important extension for the algebra of timed relations is a set of temporal modalities based on Allen's relations [2] on time periods and introduced by Halpern and Shoham in [48]. Here we use the compass notation introduced by Venema in [112] with slight modifications since it has nice geometric connotations on the two-dimensional plane. The basic set consists of six existential modalities (diamonds) denoted by $\Diamond \in \{\Diamond^{\uparrow}, \Diamond^{\downarrow}, \Diamond^{\leftarrow}, \Diamond^{\rightarrow}, \Diamond^{\triangleleft}, \Diamond^{\triangleright}\}$. We talk in more detail about the meaning of these modalities in Chapter 4.3. Now we introduce metric compass operators $\Diamond_I$ on timed relations obtained by constraining the range of quantification in an amount specified by an interval I as follows:

$$
\begin{aligned}
\Diamond^{\downarrow}_I Z &= \{(x, y) \in \Omega(T) \mid \exists r.\ x < r < y,\ y - r \in I, \text{ and } (x, r) \in Z\} \\
\Diamond^{\uparrow}_I Z &= \{(x, y) \in \Omega(T) \mid \exists r.\ x < y < r,\ r - y \in I, \text{ and } (x, r) \in Z\} \\
\Diamond^{\rightarrow}_I Z &= \{(x, y) \in \Omega(T) \mid \exists r.\ x < r < y,\ r - x \in I, \text{ and } (r, y) \in Z\} \\
\Diamond^{\leftarrow}_I Z &= \{(x, y) \in \Omega(T) \mid \exists r.\ r < x < y,\ x - r \in I, \text{ and } (r, y) \in Z\} \\
\Diamond^{\triangleright}_I Z &= \{(x, y) \in \Omega(T) \mid \exists r.\ x < y < r,\ r - y \in I, \text{ and } (y, r) \in Z\} \\
\Diamond^{\triangleleft}_I Z &= \{(x, y) \in \Omega(T) \mid \exists r.\ r < x < y,\ x - r \in I, \text{ and } (r, x) \in Z\}
\end{aligned}
$$

In Figure 8, we illustrate each diamond operator over a timed relation containing a single time period. Intuitively speaking, a diamond operator shifts a time period in the specific direction on the plane by an allowed amount. This operation, called back-shifting (of time points) [72, 83], is used to evaluate the timed eventually modality of metric temporal logic. Notice that the shift of time points can be viewed as a degenerate case of that of timed periods and there are surely more directions to move in two dimensions. Next we note that metric compass operators possess two important algebraic properties of normality and additivity such that

$$
\begin{aligned}
\Diamond_I \varnothing &= \varnothing \\
\Diamond_I (Z_1 \cup Z_2) &= \Diamond_I Z_1 \cup \Diamond_I Z_2
\end{aligned}
$$

Therefore, the algebra of timed relations with compass operators forms a boolean algebra with operators in the sense of [57] that provides an alternative (algebraic) semantics for various modal logics including temporal logics. The close connection between modal logics and boolean algebras with operators is extensively studied in several monographs [19, 20, 75].

In the following, we show the class of zones is also closed under metric compass operators. Consequently, we have that applying a metric compass operation on a zone results in another zone whose bounds are shifted according to the direction given by the type $\Diamond$ and the metric constraint I of the compass operation $\Diamond_I$.

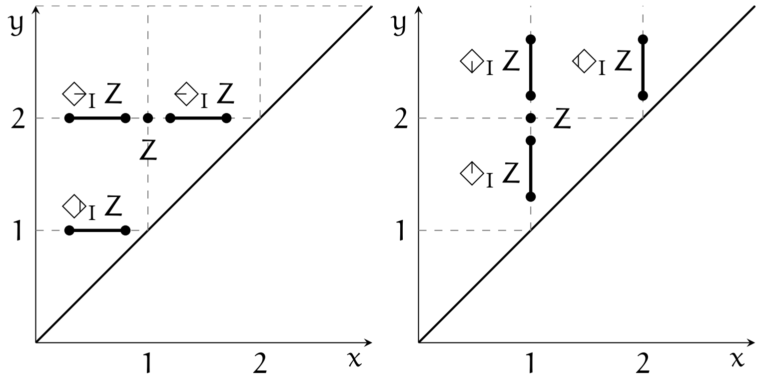

Figure 8: Metric compass operations $\Diamond_I$ on a timed relation $Z = \{(1,2)\}$ for a temporal constraint $I = [0.2, 0, 7]$.

Proposition 3.6. *Given a zone z, the timed relation $\Diamond_I z$ is a zone.*

Proof. We only show the case of $\Diamond_{[m,n]}$ as other cases are symmetric and we assume $T = (0, d)$ without loss of generality. Following the semantics of $\Diamond_{[m,n]}$,

$$(x, y) \in \Diamond_{[m,n]} \text{ iff } \exists r \in (x, y).\ r - x \in [m, n], \text{ and } (r, y) \in z$$

which translates to $\exists r \in (x, y)$ such that

$$\left\{ \begin{array}{ccccc} \underline{b} & \prec & r & \prec & \overline{b} \\ \underline{e} & \prec & y & \prec & \overline{e} \\ \underline{d} & \prec & y - r & \prec & \overline{d} \end{array} \right\}$$

By eliminating the quantifier, we obtain that $\Diamond_{[m,n]}$ equals to a zone

$$\left\{ \begin{array}{ccccc} \underline{b} - n & \prec & x & \prec & \overline{b} - m \\ \underline{e} & \prec & y & \prec & \overline{e} \\ \underline{d} + m & \prec & y - x & \prec & \overline{d} + n \end{array} \right\}$$

$\square$

Following the proof above, we apply metric compass operations over zones directly and extend it towards unions of zones as

$$\Diamond_I R_Z = \{\Diamond_I z \mid z \in R_Z\}$$

Finally, we illustrate the application of metric compass operations in Figure 9. Consider the zone $z = \{(x, y) \mid 3 \leqslant x \leqslant 5 \ \cap\ 5 \leqslant y \leqslant 7 \ \cap\ 3 \leqslant y - x \leqslant 5\}$ on the left of the figure. Then we have $\Diamond_{[1,2]} z = \{(x, y) \mid 1 \leqslant x \leqslant 4 \ \cap\ 5 \leqslant y \leqslant 7 \ \cap\ 4 \leqslant y - x \leqslant 7\}$ on the right, which is obtained by shifting z to the left accordingly. Equivalently, this operation can be viewed as a Minkowski sum $z \oplus S_{\mathtt{left}}$ of the zone z and a left-shifting set $S_{\mathtt{left}} = \{(-t, 0) \mid t \in [1, 2]\}$ with respect to $\Omega(T)$.

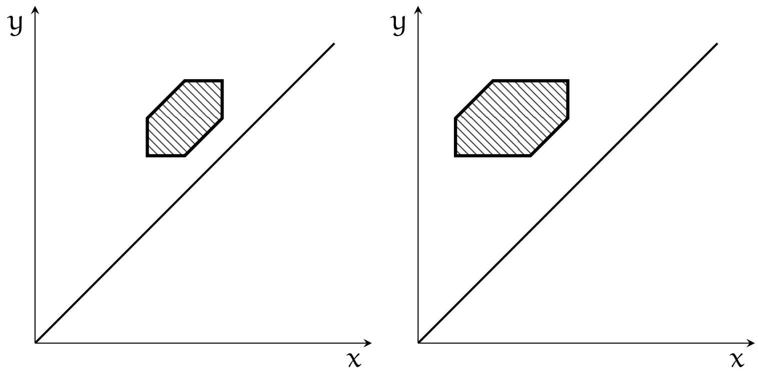

Figure 9: A zone z (left) and its left-shift $\Diamond_{[1,2]}\, z$ (right).

3.3 ALGORITHMS AND COMPLEXITY

In this section, we present our algorithms to compute operations on timed relations represented as unions of zones. Unary operations of duration restriction and temporal modalities as well as union operation are implemented in a straightforward way followed by an absorption operation. Binary operations of intersection, concatenation, and absorption (as it is an operation on two copies of the same set) on unions of zones with n elements requires $\mathcal{O}(n^2)$ in the worst case. Operations of transitive closure and complementation are more expensive; the worst case complexity of transitive closure can be bounded from above by $\mathcal{O}(n^3)$ and that of complementation by $\mathcal{O}(n^4)$ using simple arguments. Although tighter bounds can be shown for these operations, our focus here is to develop algorithms that exploit intrinsic relations and achieve linear or quasi-linear (time and space) complexity in practice. The key assumption we make is that time periods of interest and zones representing them would be dispersed on the time axis and their (maximum) durations are much smaller than the entire time domain. Therefore, most operations between two zones would be redundant since they are far away from each other (e.g. their intersection/concatenation are trivially empty) and can be avoided by sorting zones and applying the operations to zones that are temporally closer to each other according to the sort order. Since we already view time as a space and timed relations as geometric objects, we propose the use of efficient computational geometry techniques to perform operations between unions of zones in the following. In particular, we employ the plane sweep technique [85] since it performed better in our initial tests than other spatial join techniques surveyed in [56] such as R-tree-based spatial queries.

Plane Sweep Technique. Given two sets of geometric objects in an Euclidean space, the plane sweep is one of spatial join techniques that finds all pairs of objects satisfying a given relation between their spatial components. The rough idea behind the technique is to move a virtual line across the plane, keep track of objects that are in relation with this sweeping line, and perform an spatial operation on these

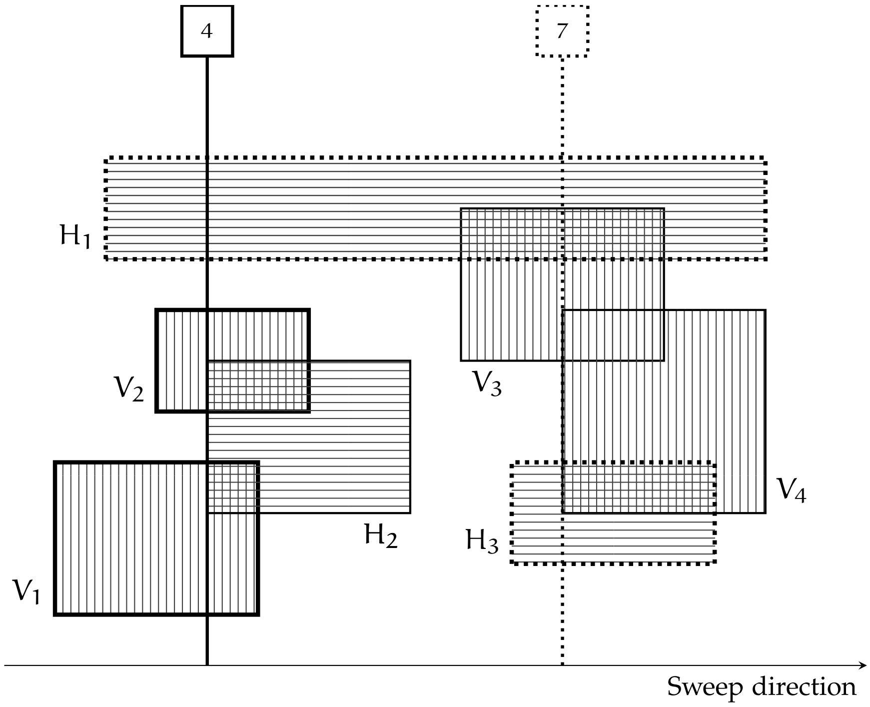

Figure 10: Illustration of plane sweep technique.

objects. More precisely, assuming that objects are sorted according to the sweep direction, a plane sweep algorithm sweeps the virtual line through the sorted list from left to right, pauses at each object, performs the actual operation between the current object and other objects in relation (usually intersects) with the sweep line.

We illustrate the technique for the intersection of rectangles and consider two sets of rectangles, $H = \{H_1, H_2, H_3\}$ and $V = \{V_1, V_2, V_3, V_4\}$, of horizontal and vertical stripes in Figure 10. The plane sweep algorithm first sorts all the rectangle according to their leftmost points so that it processes the list $(V_1, H_1, V_2, H_2, V_3, H_3, V_4)$ in order and moves a line from the leftmost point of one rectangle to that of the next at each step. Rectangles from the other set that intersect the line are considered active and we know that these are the only rectangle that may intersect with the current rectangle and processed earlier. Therefore the algorithm computes an intersection operation between these rectangles and the current rectangle and proceeds. For example, at the fourth step, the current rectangle is H_2 and the sweep line (depicted as a solid line) intersects rectangles V_1 and V_2 from the other set, and the algorithm computes $H_2 \cap V_1$ and $H_2 \cap V_2$, which are non-empty sets. Similarly the algorithm only computes $V_4 \cap H_1$ and $V_4 \cap H_3$ at the seventh and final step where the sweep line is depicted as dotted. □

Now we explain our algorithms for operations of absorption, intersection, and concatenation that use the plane sweep technique. For the absorption operation, Algorithm 1 keeps an initially empty set A of active zones and selects the next zone z from the union of zones of R sorted by h^6. Thus we sweep the plane starting from the zone with

Algorithm 1 ABSORB(R) *assume* R *sorted by* h^6

$A := R' := \varnothing$
foreach $z \in R$ **and not** $\bigcup_{z' \in A} z \subseteq z'$ **loop**
  $A := A \backslash \{z' \in A \mid z' \subseteq z\} \cup \{z\}$
  foreach $z' \in A$ **loop**
    Move z' from A to R' **if** $h^6_z \subset h^6_{z'}$
  end loop
end loop
return $R' \cup A$

the earliest begin point. The algorithm skips to the next zone if z is included in some zone in A, removes zones in A that are included in z, and adds Z to A otherwise. Then it moves zones in the active set A to the output set R' if their earliest begin value is less than the earliest begin value of z (thus the sweep line) since upcoming zones cannot include those. Once every zone in the original set has been processed, the algorithm yields the remaining zones in A as output zones in addition to previously yielded zones.

For the intersection operation, Algorithm 2 similarly sorts R and R' by h^6 and keeps two active sets of zones A and A' for zones in R and R', respectively. Zones are successively moved to their corresponding active lists and are removed from them when it is clear they will not participate in further non-empty intersections. Note that the function

Algorithm 2 INTERSECT(R, R') *assume* R, R' *sorted by* h^6

$A := A' := Y := \varnothing$
while $R \neq \varnothing$ **or** $R' \neq \varnothing$ **do**
  $z :=$ FIRST(R)
  $c :=$ CONST(h^6_z)
  $z' :=$ FIRST(R')
  $c' :=$ CONST($h^6_{z'}$)
  if $c < c'$ **then**
    Move z from R to A
    $A' := \{a' \in A' \mid \text{CONST}(h^1_{a'}) \geqslant c\}$
    foreach $a' \in A'$ **loop**
      $z'' := z \cap a'$
      $Y := Y \cup \{z''\}$
    end loop
  else
    Move z' from R' to A'
    $A := \{a \in A \mid \text{CONST}(h^1_a) \geqslant c'\}$
    foreach $a \in A$ **loop**
      $z'' := a \cap z'$
      $Y := Y \cup \{z''\}$
    end loop
  end if
end while
return ABSORB(Y)

Algorithm 3 CONCATENATE(R, R') *assume* R *sorted by* h^5
assume R' *sorted by* h^6

```
A := A' := Y := ∅
while R ≠ ∅ or R' ≠ ∅ do
  z := FIRST(R)
  c := CONST(h^5_z)
  z' := FIRST(R')
  c' := CONST(h^6_{z'})
  if c < c' then
    Move z from R to A
    A' := {a' ∈ A' | CONST(h^1_{a'}) ⩾ c}
    foreach a' ∈ A'  loop
      z'' := z ∘ a'
      Y := Y ∪ {z''}
    end loop
  else
    Move z' from R' to A'
    A := {a ∈ A | CONST(h^2_a) ⩾ c'}
    foreach a ∈ A  loop
      z'' := a ∘ z'
      Y := Y ∪ {z''}
    end loop
  end if
end while
return  ABSORB(Y)
```

FIRST(R) denotes the the first element of R and CONST(h) denotes the constant part of the inequality that defines the half-plane h.

For the concatenation operation, Algorithm 3 is very similar to the intersection except that the first set R is sorted by h^5 (that is, the earliest end) and the second set R' by h^6 (that is, the earliest begin). Therefore plane sweep technique finds all pairs of zones from two sets such that the end interval of the first one intersects with the begin interval of the second, which is a necessary condition for a non-empty concatenation.

Our complementation procedure presented in Algorithm 4 follows the definition given in the previous section. However, notice that we perform an absorption operation for each step while developing the complement. This reduces the number of zones yielded in the intermediate stages; and therefore, it is more efficient than multiplying out all complemented zones and then performing the absorption.

Algorithm 4 COMPLEMENT(R)

```
Y := Ω
foreach z ∈ R  loop
  Y := ABSORB(Y ∩ COMPLEMENT(z))
end loop
return  Y
```

Algorithm 5 CLOSURE(R)

$Y := R$
$X :=$ CONCATENATE(R, R)
while $X \nsubseteq Y$ **do**
  $Y :=$ ABSORB$(X \cup Y)$
  $X :=$ CONCATENATE(X, R)
end while
return Y

Algorithm 6 CLOSURE2(R)

$Y := R$
$X :=$ CONCATENATE(R, R)
while $X \nsubseteq Y$ **do**
  $Y :=$ ABSORB$(X \cup Y \cup$ CONCATENATE$(X, Y))$
  $X :=$ CONCATENATE(X, X)
end while
return Y

Finally, we present two algorithms for the transitive closure, one incremental and one based on so-called squaring, given in Algorithm 5 and Algorithm 6, respectively. Our tests did not show a clear winner between two approaches, however, it seems that the squaring approach performs better for sets whose fixed point index is large and the incremental approach is better when the index is small.