Journal of Universal Computer Science, vol. 26, no. 2 (2020), 268-292
submitted: 29/7/19, accepted: 12/2/20, appeared: 28/2/20

Formal Chronicle Analyses and Comparisons: How to Deal with Negative Behaviors

Yannick Pencolé
(LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
pencole@laas.fr)

Audine Subias
(LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
subias@laas.fr)

Abstract: The overall context of this paper is the event-based behavior analysis and focuses on modeling and analyzing behaviors of interest involving time information. Any behavior of interest from any time event system is concisely defined as a set of time constrained events that must occur (positive behavior) and a set of time constrained events that must not occur (negative behavior). This article proposes a formal extension of the chronicle formalism that allows for the concise description of positive and negative behaviors. Based on this new formalism, several criteria are introduced, they formally characterize and compare a set of chronicles. A fully proved implementation of the proposed criteria is then described; it relies on the use of polyhedron techniques to solve systems of linear inequalities.

Key Words: Behavior Analysis, Negative Behavior, Chronicle, Simple Temporal Problem, Polyhedron
Category: I.6.4

1 Introduction

Many applications take benefit from the analysis of specific behaviors, workflows, processes or activities involving time information: customer behaviors in the e-commerce, patient behaviors in health and medical centers, product flows in smart manufacturing systems, inhabitant activities in smart home, to name but a few. Designing and monitoring such predefined behaviors, workflows, processes, activities have several outcomes. For instance it can be used to perform temporal predictions about why kind of behaviors is going to happen. On the other hand, it can also lead to determine a temporal and causal explanation, a diagnosis, of the current situation by determining the set of events that are the causes of the situation.

Several formalisms have been proposed to represent such behaviors stressing on time information. Among them, the chronicle model [Dousson et al(1993)] is a formalism that aims at concisely representing behaviors and provides efficient tools for behavior recognition [Artikis et al(2012)]. Informally speaking, chronicles are temporal patterns defined by a set of events and time constraints between the occurrence date of the events (i.e. time points); they implicitly represent any event flow where the pattern is present (i.e. any event flow where the chronicle is recognized). [Ghallab(1996)], [Dousson et al(1993)] initially developed a chronicle model to represent an evolution scheme of a situation (partial or complete) for planning reconfiguration. The proposed model relies on a reified logical formalism; it is defined as a set of predicates and a set of temporal constraints between them. Temporal constraints are then defined as inequalities between the time variables defined in the predicates. This set of temporal constraints actually defines a temporal constraint network. Once the behaviors of the system are described this way, one can use a *chronicle recognition engine* to actually assert whether a chronicle has occurred within the event flow produced by the system at operating time [Dousson et al(1993)], [Dousson(2002)], [Carle et al(1998)]. Chronicle recognition consists in identifying in an observable flow of events all possible matchings between the event flow and the chronicle. An instance of the chronicle is a full set of instantiated events and is recorded in the set of instances.

Nowadays large systems generate a large amount of data and the problem is not only to recognize the occurrence of a given behavior but rather to detect the non-occurrence of a specific behavior. The non-occurrence of behaviors also called *negative behaviors* is present in a large spectrum of problems and can assert relevant situations. For instance, in the medical field, if a patient misses an appointment with a specialist, some important prescriptions might be missing which might aggravate the health state of the patient. In the field of consuming, consumer trends such as the boycott of certain products are typical negative behaviors whose detection might be of interest. One can also consider the case where a negative behavior (lack of a limit sensor for instance) reveals a physical failure on a system [Cao et al(2015)]. Negative patterns also play an important role to deeply understand business applications.

Obviously, whatever the type of applications, behavioral analyses depend on the quality of the modeling. Models, such as chronicles, have an impact on analysis results and therefore are crucial. How to evaluate the capacity of a chronicle to model a relevant situation? Then, a key problem is how to define which properties are associated with a chronicle. How can we define whether two chronicles are similar or dissimilar? capture the same positive/negative behavior?

This paper presents three contributions. The first one is about a formal definition of the chronicle model that handles negative behaviors: the prohibition constraints extend the negative behaviors initially introduced in the chronicle model by the *noevent* predicate [Dousson(2002)]. The second contribution is about the definition of a set of formal criteria to actually compare the available chronicles at design time before their use within a recognition engine. As opposed to the contributions of Dousson that mainly focus on the methods to efficiently recognize chronicles online, our claim is that a pre-analysis of the available set of chronicles (whatever they come from: expertise, modeling, automated extractions or learning) can also improve the online recognition by pruning/modifying chronicles from the initial set. The last contribution is about an effective set of algorithms and their implementation that perform the criteria analyses on any type of chronicles that can be described with our proposed formal definition. The proposed algorithms notably rely on polyhedron.

The paper is organized as follows. The next section of the article focuses on some related work. The chronicle model dealing with negative behaviors is presented in Section 3. Then, several criteria for chronicle evaluation and comparison are introduced in 4. Section 5 presents a set of algorithms that implement these criteria. Finally a conclusion and some perspectives are given in Section 6.

2 Related work

Chronicles have been initially introduced to supervise the execution of plans and then it has been used in a wide spectrum of applications: for telecommunication systems to manage alarms [Dousson(1996)],[Cordier and Dousson(2000)] or in production domain to monitor gas turbines [Aguilar et al(1994)]. In the medical field also, it has been used for cardiac arrhythmia detection, where electrocardiograms are modeled by chronicles [Carrault et al(1999)]: a symbolic description with time constraints is associated with pathological situations. In [Laborie and Krivine(1997)] chronicles are used to alarm processing in power distribution systems. More recently chronicles have been used in intrusion detection system. In [Morin and Debar(2003)] a chronicle approach for alarm correlation is proposed. In video understanding a formalism very closed to chronicles is proposed to detect suspect human behavior operators [Rota and Thonnat(2000)]. In the field of Unmanned Aircraft Systems (UAS), chronicles are introduced for handling breakdowns and to check the consistency between the activities in UAS [Carle et al(2013)] but also for the successful deployment of a fully autonomous unnamed aerial vehicle operating over road and traffic networks [Heintz(2001)]. In the context of high level architecture simulations [Bertrand et al(2008)], chronicle recognition is integrated into the development of a simulation as a component to analyze on line the data. [Cram et al(2009)] propose to use chronicles to assist users of a smart-kitchen in a recipe realization. Another important field of application of chronicle recognition is collaborative systems notably web services [Cordier et al(2007)]. In this case the main challenge is the distribution of the chronicles into sub-chronicles and the communication or synchronization mechanisms between the chronicles [Boufaied et al(2004)], [Guillou et al(2008)], [Vizcarrondo et al(2013)].

A number of other formalisms exist in the literature to represent situations stressing on time information. [McCarthy and Hayes(1969)] introduces the situation calculus that formally models reasoning about actions and changes. A situation represents a snapshot of the world i.e a view of the world at an instant of time, and the world is a sequence of global situations connected by actions. Situation calculus is not suited to represent concurrency moreover only properties that change as the effect of an action can be represented. The event calculus [Kowalski and Sergot(1986)] overcomes these limitations. It is based upon the notions of events, relationships and the periods they start and end, formulated within a logic programming framework. Event calculus aims to determine the value of logical proposition over time. The main difference is that the event calculus deals with local events and time periods. In other words instead of considering actions in a given situation actions are represented in an explicit moment of time. In this sense it is closely related to the Allen's Interval-based temporal logic [Allen(1983)], [Allen(1984)] providing an explicit and intuitive representation of time information based also on time periods to reason about actions and changes. This temporal reasoning framework is based on a nonempty set of time intervals and a set of thirteen basic qualitative temporal relations that hold between two intervals and are mutually exclusive. It allows to consider arbitrary complex relationships between events and effects. In [Bauer et al(2011)], situations stressing on time information are represented as Timed Linear-time Temporal Logic (TLTL) specifications that are checked on-line.

The expressive framework provided by temporal logic is not always efficient for reasoning because of the computational complexity of reasoning. Therefore, the use of constraint satisfaction problems is an efficient alternative. These techniques allow to formalize as a constraint satisfaction problem the possibly indefinite or incomplete knowledge about temporal relations between temporal objects like time points. Moreover, they provide efficient algorithms based on constraint propagation for various temporal reasonings. Among these techniques *Temporal Constraint Networks* and particularly according to the quantitative nature of the constraints the *Simple Temporal Problems* (STP) introduced by [Dechter et al(1991)] are of particular interest. A STP is defined by a set of variables representing time points and constraints between these time points defined by a set of time intervals. Chronicles are part of this type of approaches considering their associated constraint networks [Dousson et al(1993)] where intervals with numerical bounds quantify event orders.

To the best of our knowledge, the problem of chronicle analysis has not been widely studied. In [Saddem et al(2010)] the authors consider this issue by checking the consistency of a similar formalism called Causal Temporal Signatures. The aim is to check if there exist input events leading to the recognition of several signatures. In [Sahuguède et al(2018)] the authors propose to characterize a chronicle with no negative behaviors by a directed vector evaluated from the chronicle projection in a k-dimensional Euclidean space to define a similarity distance between chronicles.

3 A formal chronicle model handling negative behaviors

3.1 Background on simple temporal problem

[Dechter et al(1991)] proposes a general framework called *Temporal Constraint Satisfaction Problem* (TCSP) for constraints based temporal reasoning. A TCSP is defined by a set of variables representing time points and constraints between these time points defined by a set of time intervals. A *Simple Temporal Problem* (STP) is one particular instance of a TCSP where each constraint is defined by one time interval. More formally a STP [Dechter et al(1991)] is a finite set of variables $\mathcal{X} = \{x_0, x_1, \ldots, x_n\}$[1] with continuous ranges and a finite set of intervals $\mathcal{T}$ representing the temporal constraints between these variables: each interval $\mathcal{T}_{ij} = [a_{ij}, b_{ij}] \in \mathcal{T}, a_{ij}, b_{ij} \in \mathbb{Q}$ represents the constraint on the admissible value for the distance $x_j - x_i$. Such a constraint can also be expressed as a set of inequalities $x_j - x_i \leq b_{ij}$ and $a_{ij} \leq x_j - x_i$. A STP can be represented by a constraint graph $\mathcal{G} = (\mathcal{X}, \mathcal{A})$ where the nodes $\mathcal{X}$ are the variables $\{x_0, \ldots, x_n\}$ and where $\mathcal{A}$ is a set of edges: the edge $x_i \rightarrow x_j$ is associated with the $\mathcal{T}_{ij}$ constraint. A tuple $T = (t_0, \ldots, t_n)$ is a solution of the STP if the assignment $\{x_0 = t_0, \ldots, x_n = t_n\}$ satisfies all the constraints. The graph is said to be consistent if it exists at least one solution.

A distance graph noted $\mathcal{G}_d = (\mathcal{X}, \mathcal{A}_d)$ can also be associated with a STP. To each edge $x_i \rightarrow x_j$ a linear inequality $x_j - x_i \leq c_{ij} = b_{ij}$ is associated if it exists in the constraint graph an edge $x_i \rightarrow x_j$ associated with $\mathcal{T}_{ij} = [a_{ij}, b_{ij}]$. On the contrary if there exists $x_j \rightarrow x_i$ associated with $\mathcal{T}_{ji} = [a_{ji}, b_{ji}]$ then $x_j - x_i \leq c_{ij} = -a_{ji}$. Each path $x_i = x_{k_0} \rightarrow \cdots \rightarrow x_{k_m} = x_j$ from x_i to x_j in $\mathcal{G}_d$ induces the following constraint on the $x_j - x_i$ distance: $x_j - x_i \leq \sum_{l=1}^{m} c_{k_{l-1}k_l}$. Consider now a cycle C of nodes $x_i = x_{k_0} \rightarrow \cdots \rightarrow x_{k_m} = x_i$, such a cycle is said to be negative if summing the inequalities along C yields to $x_i - x_i < 0$. A STP is proven to be consistent iff its distance graph has no negative cycles [Dechter et al(1991)]. In this case, *the shortest path* between each pair of nodes (d_{ij}) can be defined. Thus, an important result is that for the $\mathcal{G}_d$ distance graph of a consistent STP, two consistent scenarios are given by $S_1 = (0, d_{01}, \ldots, d_{0n})$ and $S_2 = (0, -d_{10}, \ldots, -d_{n0})$. If the time origin x_0 is set to 0 then S_1 (resp. S_2) assigns to each variable $\{x_1, \ldots, x_n\}$ its latest possible time value (resp. its earliest possible time value).

If there exist more than one path from x_i to x_j then it can be easily verified that all the path constraints induce: $x_j - x_i \leq d_{ij}$ Therefore each STP can

[1] In [Dechter et al(1991)], the problem is defined over $\{x_1, \ldots, x_n\}$ and x_0 is set to be the time origin 0. Throughout this paper, x_0 still denotes the time origin but x_0 is not fixed as time origin may change.

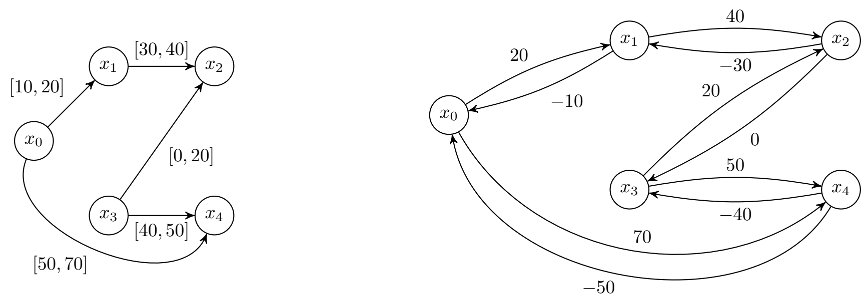

Figure 1: Simple Temporal Problem: constraint graph (on the left) and distance graph (on the right).

be specified by a complete directed graph called d-graph where each transition $x_i \rightarrow x_j$ is labeled by the shortest path length d_{ij} from the distance graph $\mathcal{G}_d$. The d-graph yields to an explicit representation of a STP.

Figure 1 gives an example of a STPwith the constraint and distance graphs. From the distance graph it is easy to generate the complete d-graph after the determination of all the lengths of the shortest paths (see Figure 2). For this, the Floyd-Warshall's algorithm [Floyd(1962)] can be applied to the distance graph. The complexity of the algorithm is $\Theta(n^3)$ with n the number of vertices. Finally, one last important result to recall about STP*s* is the theorem of decomposability.

Theorem 1. *Any consistent* STP *is decomposable relatively to the constraints in its directed graph.*

The decomposability of an STP means that, whatever the selected variable x_i is, it is always possible to initiate a solution of a consistent STP by starting with $x_i = 0$ and then sequentially assign other variables with values satisfying the constraints of the directed graph. Decomposability ensures that, regardless of the assignment order, any partial assignment is always part of a complete solution. The result then provides an efficient way to find solutions of an STP.

3.2 Chronicle: concepts and definition

We propose in this section a formal definition for chronicle that handles negative behaviors. This chronicle model is based on the framework of Simple Temporal Problems and negative behaviors are represented by specific constraints for forbidden events.

An event is a pair (e, t) where $e \in \mathbb{E}$ is an event type and t a real denoting the event occurrence date. An observable evolution of any time event-based system

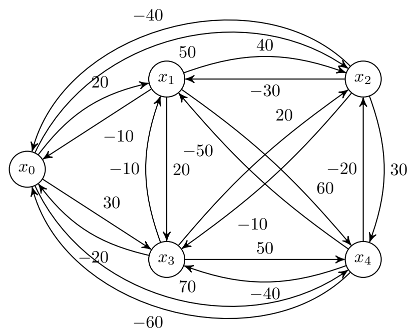

Figure 2: Simple Temporal Problem [Dechter et al(1991)]: d-graph

is characterized by an event sequence. An event sequence on $\mathbb{E}$ is an ordered set of events denoted $\mathcal{S} = \langle (e_1, t_1) \ldots (e_l, t_l) \rangle$ where $t_i < (t_{i+1})$, $i \in \mathbb{N}$, and $i = 1, \ldots, l-1$ with l the size (i.e the number of events) of the event sequence $\mathcal{S}$. We first introduce a formal definition for chronicles that do not contain negative behaviors: such a chronicle is called a positive chronicle.

Definition 2 Positive Chronicle. A positive chronicle is a 5-tuple, $\mathcal{C} = (\mathcal{X}, \mathcal{A}, \mathcal{T}, \mathcal{E}, \mathcal{M})$, where:

- $\mathcal{X}$ is a finite set of temporal variables;
- $\mathcal{A} \subseteq \mathcal{X} \times \mathcal{X}$ is a finite set of edges;
- $\mathcal{T} : \mathcal{A} \to \mathbb{I}$ is the application that associates a temporal interval to each edge $x_i \to x_j$: for short $\mathcal{T}(x_i, x_j)$ is denoted $\mathcal{T}_{ij}$;
- $\mathcal{E}$ is a finite set of event types;
- $\mathcal{M} : \mathcal{X} \to \mathcal{E}$ is a surjective function that associates to each temporal variable of $\mathcal{X}$ an event type of $\mathcal{E}$.

Note that this definition allows that several temporal variable may have the same event type. The triplet $(\mathcal{X}, \mathcal{A}, \mathcal{T})$ of a chronicle corresponds to a Simple Temporal Problem (STP) that we call the underlying STP of a chronicle.

The semantics of a chronicle is defined by the set of event sequences in which the chronicle is recognized at least once by a chronicle recognition engine. To assert that a chronicle is recognized, the engine produces a chronicle instance as an output: a chronicle instance is the result of a temporal matching between an event flow (i.e an event sequence) and the expected events modeled in the chronicle. An instance can then be interpreted as a chronicle occurrence in an event sequence.

Definition 3 Instance of a positive chronicle. An instance $i_{\mathcal{C}}$ of a positive chronicle $\mathcal{C}$ over an event sequence $\mathcal{S} = \langle (e_1, t_1) \ldots (e_l, t_l) \rangle$ is a set of couples $i_{\mathcal{C}} = \{(e_{i_1}, t_{i_1}) \ldots (e_{i_{|\mathcal{X}|}}, t_{i_{|\mathcal{X}|}})\}$ such that there exists a one-to-one correspondence $f : \mathcal{X} \to \{t_{i_1}, \ldots, t_{i_{|\mathcal{X}|}}\}$ such that:

1. for every $x \in \mathcal{X}$, $(e, f(x)) \in i_{\mathcal{C}}$ and $e = \mathcal{M}(x)$;
2. the set $\{x = f(x)\}_{x \in \mathcal{X}}$ is solution of the underlying STP of $\mathcal{C}$.

The set of instances of a positive chronicle $\mathcal{C}$ over $\mathcal{S}$ is denoted $\mathcal{I}_{\mathcal{C}}(\mathcal{S})$. As in [Pencolé and Subias(2009)], we can associate with a positive chronicle $\mathcal{C}$ a recognition language $L^{\mathcal{C}}_{\Sigma}$ over a finite alphabet Σ: suppose $\mathcal{E} \subseteq \Sigma$ then $L^{\mathcal{C}}_{\Sigma}$ is the set of event sequences $\mathcal{S}$ over the event types in Σ such that $\mathcal{I}_{\mathcal{C}}(\mathcal{S}) \neq \varnothing$.

Now we propose to extend positive chronicles with negative behaviors. In this article negative behaviors are represented by prohibition constraints. A prohibition constraint denotes the mandatory absence of any event of type e during one or several temporal intervals what is also called a noevent constraint as introduced in [Dousson(2002)]. The bounds of these intervals depend on the time variables of the constraint graph. Given two non necessarily distinct variables x_i and x_j these intervals are defined by $J = [x_i + \alpha, x_j + \beta]$. The upper and lower bounds represent the prohibition starting time and ending time respectively, with $\alpha, \beta \in \mathbb{Q}$. A chronicle including negative behaviors is then defined as follows:

Definition 4 Chronicle. A chronicle is a 6-tuple, $\mathcal{C} = (\mathcal{X}, \mathcal{A}, \mathcal{T}, \mathcal{E}, \mathcal{M}, \mathcal{F})$ where:

- $(\mathcal{X}, \mathcal{A}, \mathcal{T}, \mathcal{E}^+, \mathcal{M})$ is a positive chronicle and $\mathcal{E}^+ \subseteq \mathcal{E}$.
- $\mathcal{F} : \mathcal{E}^- \to 2^{\{\mathcal{X} \times \mathcal{X} \times \mathbb{Q}^2 \times \{[,]\}^2\}}$ is a function that associates to each event type of $\mathcal{E}^-$ ($\mathcal{E} = \mathcal{E}^+ \cup \mathcal{E}^-$) a set of prohibition constraints (see below for details).

In a chronicle, a prohibition constraint for an event type $e \in \mathcal{E}^-$ is a 6-tuple $(x_i, x_j, \alpha, \beta, \lfloor, \rceil) \in \mathcal{F}(e)$. Such a constraint means that any event of type e is forbidden in the interval $\lfloor x_i + \alpha, x_j + \beta \rceil$ where $\alpha, \beta \in \mathbb{Q}$ and where symbols $\lfloor$ and $\rceil$ are either the bracket [or the bracket].

Example 1. Let us consider a system at least composed of a motor and a switch command and the chronicle represented in Figure 3. The system may be rather complex and contains other components and may produce many types of events but the chronicle focuses on a specific temporal pattern involving observable events of type *on* (switch on), *off* (switch off), *start* (the motor starts), *stop* (the motor stops). The chronicle represents a pattern that expects an event *on* at any time followed by the starting of the motor either simultaneously or before 5 time units. Then a switch-off is expected between time 50 and time 100 after the motor starts. Finally, the motor is expected to stop in the time interval [0,5] after the switch-off. Two prohibition constraints are also defined in this chronicle. The chronicle expects that if *on* occurs at time t_0 then no event *off* should have occurred later than 2 time units before the event *on*. The same prohibition constraint also ensures that the expected *off* event of node x_2 is the first one after *on*. A second prohibition constraint expects that there is no event *on* between the event *off* and 3 time units after the event *stop*. The complete formal definition of this chronicle is given below.

$\mathcal{C}_1 = (\mathcal{X}_1, \mathcal{A}_1, \mathcal{T}_1, \mathcal{E}_1, \mathcal{M}_1, \mathcal{F}_1)$ where:

- $\mathcal{X}_1 = \{x_0, x_1, x_2, x_3\}$; $\mathcal{A}_1 = \{(x_0, x_1), (x_1, x_2), (x_2, x_3)\}$;
- $\mathcal{T}_{1_{01}} = [0, 5], \mathcal{T}_{1_{12}} = [50, 100], \mathcal{T}_{1_{23}} = [0, 5]$;
- $\mathcal{E}_1^+ = \{on, off, start, stop\}$; $\mathcal{E}_1^- = \{on, off\}$; $\mathcal{E}_1 = \mathcal{E}_1^+ \cup \mathcal{E}_1^-$;
- $\mathcal{M}_1(x_0) = on, \mathcal{M}(x_1) = start, \mathcal{M}(x_2) = off, \mathcal{M}(x_3) = stop$;
- $\mathcal{F}_1(off) = \{(x_0, x_2, -2, 0, [, [)\}$, $\mathcal{F}_1(on) = \{(x_2, x_3, 0, 3, [,])\}$

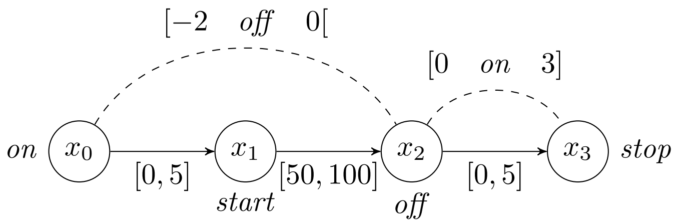

Figure 3: Chronicle $\mathcal{C}_1$ motor start/stop with time delays.

The semantics of a chronicle with negative behaviors still rely on the notion of chronicle instance that is extended as follows.

Definition 5 Chronicle instance. An instance $i_{\mathcal{C}}$ of a chronicle $\mathcal{C}$ over an event sequence $\mathcal{S} = \langle (e_1, t_1) \dots (e_l, t_l) \rangle$ is a pair $(i_{\mathcal{C}}^+, i_{\mathcal{C}}^-)$ such that:

1. $i_{\mathcal{C}}^+$ is a set of couples $i_{\mathcal{C}}^+ = \{(e_{i_1}, t_{i_1}) \dots (e_{i_{|\mathcal{X}|}}, t_{i_{|\mathcal{X}|}})\}$ such that there exists a one-to-one correspondence $f : \mathcal{X} \to \{t_{i_1}, \dots, t_{i_{|\mathcal{X}|}}\}$ such that:

(a) for every $x \in \mathcal{X}$, $(e, f(x)) \in i^+_{\mathcal{C}}$ and $e = \mathcal{M}(x)$;

(b) the set $\{x = f(x)\}_{x \in \mathcal{X}}$ is solution of the underlying STP of $\mathcal{C}$;

(c) $\forall e \in \mathcal{E}^+ \cap \mathcal{E}^-$ and every $(e, f(x)) \in i^+_{\mathcal{C}}$, $\forall (x_i, x_j, \alpha, \beta, \lfloor, \rceil) \in \mathcal{F}(e)$, $f(x) \notin \lfloor f(x_i) + \alpha, f(x_j) + \beta \rceil$.

2. $i^-_{\mathcal{C}}$ is the set of n couples $(n \geq 0)$ $i^-_{\mathcal{C}} = \{(e_{j_1}, t_{j_1}) \ldots (e_{j_n}, t_{j_n})\}$ such that:

(a) for every $(e, t) \in (S)$ such that $e \in \mathcal{E}^-$, $(e, t) \in i^-_{\mathcal{C}}$ iff $(e, t) \notin i^+_{\mathcal{C}}$

(b) $\forall e \in \mathcal{E}^-$ and for every $(e, t) \in i^-_{\mathcal{C}}$, $\forall (x_i, x_j, \alpha, \beta, \lfloor, \rceil) \in \mathcal{F}(e)$, $t \notin \lfloor f(x_i) + \alpha, f(x_j) + \beta \rceil$.

Informally speaking, $i^+_{\mathcal{C}}$ is the set of events matching with the underlying positive chronicle and that also satisfies the prohibition constraints. $i^-_{\mathcal{C}}$ is the set of events that are not involved in the positive chronicle but that satisfy at a prohibition constraint. The set of instances of a chronicle $\mathcal{C}$ over $\mathcal{S}$ is still denoted by $\mathcal{I}_{\mathcal{C}}(\mathcal{S})$ and the definition of the recognition language $L^{\mathcal{C}}_{\Sigma}$ is as for a positive chronicle.

Example 2. Consider an observed behavior from the system in Figure 3 given by the event sequence $\mathcal{S} = \langle (on, 2), (off, 3), (low\_temp, 4), (on, 6), (start, 7), (low\_temp, 20), (off, 58), (high\_temp, 60), (stop, 62), (on, 66) \rangle$. There is a chronicle instance in $\mathcal{S}$ that is $\mathcal{I}_{\mathcal{C}_1}(\mathcal{S}) = \{i_{\mathcal{C}_1} = (i^+_{\mathcal{C}_1}, i^-_{\mathcal{C}_1})\}$ where $i^+_{\mathcal{C}_1} = \{(on, 6), (start, 7), (off, 58), (stop, 62)\}, i^-_{\mathcal{C}_1} = \{(on, 2), (off, 3), (on, 66)\}$. Suppose now that event sequence $\mathcal{S}'$ is as $\mathcal{S}$ but without event $(off, 3)$ so $\mathcal{I}_{\mathcal{C}_1}(\mathcal{S}')$ has two instances: $\{(i^+_1, i^-_1), (i^+_2, i^-_2)\}$ where $i^+_1 = i^+_{\mathcal{C}_1}, i^-_1 = \{(on, 2), (on, 66)\}$, and $i^+_2 = \{(on, 2), (start, 7), (off, 58), (stop, 62)\}, i^-_2 = \{(on, 6), (on, 66)\}$.

Obviously any useful chronicle must be recognized in some situations so it has to be consistent: it must exist at least one event sequence leading to the generation of one instance of $\mathcal{C}$. The results on STP*s* [Dechter et al(1991)] allow to conclude that the underlying triplet $(\mathcal{X}, \mathcal{A}, \mathcal{T})$ associated with a chronicle $\mathcal{C}$ is consistent if and only if its distance graph has no negative cycle. By adding prohibition constraints to STP*s* the consistency property is different as a prohibition constraint may have an influence on the consistency of the underlying STP*s* by preventing the instantiation of one or more temporal variables.

3.3 Prohibition constraints

For a prohibition constraint denoted $(x_i, x_j, \alpha, \beta, \lfloor, \rceil) \in \mathcal{F}(e)$, $(x_i + \alpha)$ corresponds to the lower bound of the time interval where events of type e are forbidden and $(x_j + \beta)$ corresponds to the upper bound of this time interval. To ensure that the prohibition constraint interval is not empty, it is necessary that $(x_i + \alpha) \leq (x_j + \beta)$ if the interval is closed and $(x_i + \alpha) < (x_j + \beta)$ if it is open.

The possible values of the variables x_i and x_j depend on the whole constraints of the chronicle. It is then possible that for some values of x_i and x_j, the non-emptiness condition holds and for others it does not. This leads to the following definition that ensures that the prohibition constraint is indeed useful.

Definition 6 Well-formed prohibition constraint. A closed (resp. open /semi-open) interval $(x_i, x_j, \alpha, \beta, \lfloor, \rceil) \in \mathcal{F}(e)$ is well-formed if $\alpha \leq (d_{ij} + \beta)$ (resp. $\alpha < (d_{ij} + \beta)$) where d_{ij} is the length of the shortest path from x_i to x_j in the underlying d-graph.

Example 3. Back to Figure 3, any prohibition constraint is well-formed. The prohibition constraint $(x_0, x_2, -2, 0, [, [) \in \mathcal{F}(\mathit{off})$ is such that $\alpha = -2, \beta = 0, d_{02} = 50$ and $-2 < 50 + 0$. Suppose now that this constraint would be replaced by $(x_0, x_2, 110, 0, [, [) \in \mathcal{F}(\mathit{off})$, then it would lead to $110 \not< 50 + 0$, the prohibition constraint would not be well-formed: the first event *off* would be expected between time 50 and 105 from the event *on* so the prohibition constraint would forbid any *off* event within the time interval $[x0 + 110, x_0 + 105[$ which is invalid.

The aim of a well-formed prohibition constraint is to effectively express that a given type of event $e \in \mathcal{E}^-$ cannot occur within a time interval. As long as e is not involved as an event type of a temporal variable ($e \notin \mathcal{E}^+$), the prohibition constraint does not have any influence on the constraints between temporal variables. However, if $e \in \mathcal{E}^+$, the prohibition constraint is influential if it forbids a temporal variable x_k such that $\mathcal{M}(x_k) = e$ to be instantiated.

Theorem 7. *A prohibition constraint $(x_i, x_j, \alpha, \beta, \lfloor, \rceil) \in \mathcal{F}(e)$ is not influential in regards to x_k with $\mathcal{M}(x_k) = e$ iff $0 \notin \lfloor d_{ki} + \alpha, d_{kj} + \beta \rceil$ and $0 \notin \lfloor -d_{ik} + \alpha, -d_{jk} + \beta \rceil$ and $(d_{ki} + \alpha), (d_{kj} + \beta), (-d_{ik} + \alpha)$ and $(-d_{jk} + \beta)$ are of the same sign.*

Proof. Suppose $\lfloor = [$ and $\rceil =]$. Based on Theorem 1, by assigning $x_k = 0$, we have $-d_{ik} + \alpha \leq x_i + \alpha \leq d_{ki} + \alpha$ and $-d_{jk} + \beta \leq x_j + \beta \leq d_{kj} + \beta$. Two consistent solutions can be distinguished: the earliest one $S_1 : (x_i = d_{ki}, x_j = d_{kj})$ and the latest one $S_2 : (x_i = -d_{ik}, x_j = -d_{jk})$. $x_k = 0$ is not influential so it means it cannot forbid neither S_1 nor S_2 nor any other solution between S_1 and S_2, the result follows for $\lfloor = [$ and $\rceil =]$. The same reasoning can then be applied for any other combination of symbols $\lfloor$ and $\rceil$.

Influential prohibition constraints should be avoided in chronicles as they interfere with the other constraints. In the following we suppose that a chronicle does not contain any influential constraint.

4 Criteria for chronicles comparison

The section focuses on the definition and the formal analysis of chronicle criteria. Given a set of available chronicles we aim at analyzing them in order to filter out some of the available chronicles at design stage. For instance for the sake of performance of the chronicle recognition engine we would consider that a set of chronicles must be concise, that is any chronicle must be consistent, there must be no equivalent chronicles in the set and a chronicle should not be covered by another chronicle. However, for the sake of fault diagnosis or specific situation recognition, it might be interesting to filter out covering chronicles as they are not specific enough (too ambiguous). In the following, we do not make any assumption about how the set of chronicles has been built (either designed by experts or by machine learning techniques) but we claim that the following criteria and resulting algorithms can be used during any modeling/learning process to reach a relevant set of chronicles.

4.1 Chronicle consistency

Let us start with chronicle consistency. Basically a chronicle is consistent if it is recognized on at least one sequence $\mathcal{S}$.

Definition 8 Consistency. A chronicle $\mathcal{C}$ is consistent if there exists a sequence $\mathcal{S}$ such that $\mathcal{I}_{\mathcal{C}}(\mathcal{S}) \neq \varnothing$.

We have here a straightforward result about the consistency of a positive chronicle (noted in the following $\mathcal{C}_+$).

Theorem 9. *A positive chronicle $\mathcal{C}_+$ is consistent iff its underlying* STP *is consistent.*

Proof. By definition, there exists a sequence $\mathcal{S} = \langle(\mathcal{M}(x_0), t_0), \ldots, (\mathcal{M}(x_n), t_n)\rangle$ such that $\mathcal{I}_{\mathcal{C}}(\mathcal{S}) \neq \varnothing$ iff $\{x_0 = t_0, \ldots, x_n = t_n\}$ is a solution of the underlying STP.

Consider now a chronicle $\mathcal{C}$ with prohibition constraints. Any minimal sequence where the positive chronicle $\mathcal{C}_+$ extracted from $\mathcal{C}$ is recognized is also a sequence where $\mathcal{C}$ is recognized. [2] Hence the following result.

Corollary 10. *A chronicle $\mathcal{C}$ is consistent iff its underlying* STP *is consistent.*

To summarize, checking consistency consists in checking the consistency of the underlying STP that can be performed for instance by searching for the presence of a negative cycle in the distance graph of the underlying STP (see Section 3.1). Adding prohibition constraints does not generate more difficulties for this criteria.

[2] This is true as we assume that there are no influential prohibition constraints.

4.2 Chronicle equivalence

Two chronicles are equivalent if they are always recognized at the same time on any event sequence $\mathcal{S}$.

Definition 11 Equivalence. Two chronicles $\mathcal{C}$ and $\mathcal{C}'$ are equivalent (denoted $\mathcal{C} \equiv \mathcal{C}'$) if $\mathcal{C}$ and $\mathcal{C}'$ have the same set of instances whatever the observable input flow $\mathcal{S}$ is: $\forall \mathcal{S}, \mathcal{I}_{\mathcal{C}}(\mathcal{S}) = \mathcal{I}_{\mathcal{C}'}(\mathcal{S})$.

Here also, in the case of positive chronicles, equivalence checking is a well known problem that can be solved using *d*-graphs [Dousson et al(1993)].

Theorem 12. *Two positive chronicles $\mathcal{C}_+$ and $\mathcal{C}'_+$ are equivalent if their underlying* Stp*s lead to the same d-graph.*

Proof. Any instance of $\mathcal{I}_{\mathcal{C}}(\mathcal{S})$ corresponds to a solution of the underlying Stp (see Definition 3). $\mathcal{I}_{\mathcal{C}}(\mathcal{S}) = \mathcal{I}_{\mathcal{C}'}(\mathcal{S})$ means that the underlying Stp*s* of $\mathcal{C}_+$ and $\mathcal{C}'_+$ have the same set of solutions. The result follows.

In this case, a way to check the equivalence is to compute the *d*-graph of both chronicles (Floyd-Warshall algorithm) and check that both graphs are isomorphic. When negative behaviors are introduced the problem is more tricky. The previous result still holds in the case where both chronicles have the same *d*-graph and the same prohibition constraints. Nevertheless, two chronicles may be equivalent while the prohibition constraints are not the same and their *d*-graphs are identical: chronicles $\mathcal{C}_2$ and $\mathcal{C}_3$ illustrate such a case (see Figure 4).

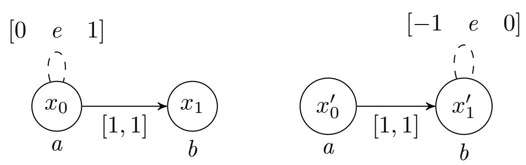

Figure 4: Equivalent chronicles $\mathcal{C}_2$ and $\mathcal{C}_3$ with different prohibition constraints.

It follows that the generic test for chronicle equivalence cannot be based on *d*-graphs, another computation technique must be used. We present in the remainder a test for chronicle equivalence based on systems of linear inequalities. Let $\mathcal{C} = (\mathcal{X}, \mathcal{A}, \mathcal{T}, \mathcal{E}, \mathcal{M}, \mathcal{F})$ be a chronicle. We denote by $Ine^+(\mathcal{C})$ the set of inequalities defined as follows: $x_j - x_i \leq b_{ij}$ and $x_j - x_i \geq a_{ij}$ for every edge (x_i, x_j) from $\mathcal{A}$ with $\mathcal{T}_{ij} = [a_{ij}, b_{ij}]$. The set of prohibition constraints of $\mathcal{C}$ is $\{c \in \mathcal{F}(e), e \in \mathcal{E}^-\}$ so it is the image of $\mathcal{F}$. In the following we will simply denote this set by $\mathcal{F}$. Let $c = (x_i, x_j, \alpha, \beta, \lfloor, \rceil)$ be a prohibition constraint from $\mathcal{F}(e)$, $e \in \mathcal{E}^-$. Let us denote by x_e^- a temporal variable such that $x_e^- \notin \mathcal{X}$, we denote by $Ine^<(c)$ the inequality set $\{x_e^- - x_i < \alpha\}$ if $\lfloor$ is closed or $Ine^<(c)$ is $\{x_e^- - x_i \leq \alpha\}$ if $\lfloor$ is open. Similarly $Ine^>(c)$ denotes the inequality set $\{x_e^- - x_i > \beta\}$ if $\rceil$ is closed and denotes $\{x_e^- - x_i \geq \beta\}$ if $\rceil$ is open. Finally $Ine(c) = Ine^<(c) \cup Ine^>(c)$. We consider here that the prohibition constraints are well-formed so it means that the system $Ine^+(\mathcal{C}) \cup Ine(c)$ cannot have a solution: the variable x_e^- cannot be lower than $x_i + \alpha$ and greater than $x_j + \beta$ at the same time.

Let *op* denote the set of comparison operators $op = \{<, >\}$. In the following, we say that a configuration CF is the assignment of an operator from *op* to each prohibition constraint c from $\mathcal{C}$. For a given configuration CF among the set of the possible configurations, we denote the set of inequalities involving the prohibition constraints of the chronicle $\mathcal{C}$: $Ine^-(\mathcal{C}, CF) = \bigcup_{c \in \mathcal{F}} Ine^{CF(c)}(c)$.

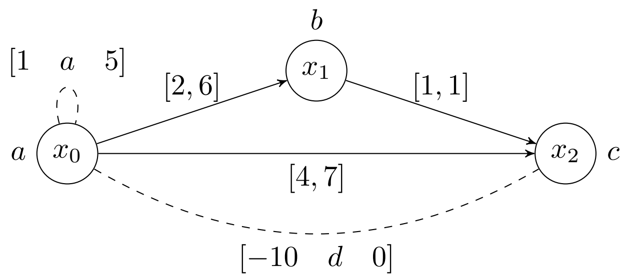

Figure 5: A chronicle $\mathcal{C}_4$ with two prohibition constraints.

Example 4. Figure 5 presents a chronicle with 3 temporal variables x_0, x_1 and x_2. The set of inequalities $Ine^+(\mathcal{C}_4)$ is $x_1 - x_0 \geq 2$ and $x_1 - x_0 \leq 6$; $x_2 - x_0 \geq 4$ and $x_2 - x_0 \leq 7$; $x_2 - x_1 \geq 1$ and $x_2 - x_1 \leq 1$; Chronicle $\mathcal{C}_4$ also has two prohibition constraints. The first one is $c_d = (x_0, x_2, -10, 0, [,]) \in \mathcal{F}(d), d \in \mathcal{E}^-$. To define the associated set of inequalities $Ine(c_d)$, we first introduce a new variable x_d^- that represents the date of the occurrence of an event of type d with $Ine^<(c_d) = \{x_d^- - x_0 < -10\}$ and $Ine^>(c_d) = \{x_d^- - x_2 > 0\}$ and finally $Ine(c_d) = Ine^>(c_d) \cup Ine^<(c_d)$. Similarly for the second prohibition constraint $c_a = (x_0, x_0, 1, 5, [,]) \in \mathcal{F}(a), a \in \mathcal{E}^-$, we have: $Ine^<(c_a) = \{x_a^- - x_0 < 1\}$ and $Ine^>(c_a) = \{x_a^- - x_0 > 5\}$ and finally $Ine(c_a) = Ine^>(c_a) \cup Ine^<(c_a)$; In this example, there are 4 possible configurations CF. For instance, suppose that the configuration CF_1 is such that $CF_1(c_a) = \;<$ and $CF_1(c_d) = \;>$ then $Ine^-(\mathcal{C}_4, CF_1) = Ine^{CF1(c_a)}(c_a) \cup Ine^{CF1(c_d)}(c_d) = Ine^<(c_a) \cup Ine^>(c_d)$ and then $Ine^-(\mathcal{C}_4, CF_1) = \{x_a^- - x_0 < 1, x_d^- - x_2 > 0\}$.

Let $\mathcal{C} = (\mathcal{X}, \mathcal{A}, \mathcal{T}, \mathcal{E}, \mathcal{M}, \mathcal{F})$ and $\mathcal{C}' = (\mathcal{X}', \mathcal{A}', \mathcal{T}', \mathcal{E}', \mathcal{M}', \mathcal{F}')$ be two chronicles. The set of event types $\mathcal{E}$ is decomposed as $\mathcal{E} = \mathcal{E}^+ \cup \mathcal{E}^-$ and the set of event types $\mathcal{E}'$ is decomposed as $\mathcal{E}' = \mathcal{E}'^+ \cup \mathcal{E}'^-$ (see Definition 4).

Theorem 13. *Two chronicles $\mathcal{C}$ and $\mathcal{C}'$ are equivalent iff*

1. $\mathcal{E}^- = \mathcal{E}'^-$;
2. *there exists a one-to-one correspondence m between $\mathcal{X}$ and $\mathcal{X}'$ such that*
   (a) $\mathcal{M}(x) = \mathcal{M}'(m(x))$; *and*
   (b) *for every $e \in \mathcal{E}^-$, $\{x = t_x\}_{x\in\mathcal{X}} \cup \{x_e^- = t_e\}$ is solution of $Ine^+(\mathcal{C}) \cup Ine^-(\mathcal{C}, CF)$ for at least a given configuration CF of the prohibition constraints of $\mathcal{C}$ iff $\{m(x) = t_x\}_{x\in\mathcal{X}} \cup \{x_e^- = t_e\}$ is solution of $Ine^+(\mathcal{C}') \cup Ine^-(\mathcal{C}', CF')$ for at least a given configuration CF' of the prohibition constraints of $\mathcal{C}'$.*

Proof. ($\Rightarrow$) Let $\mathcal{C}$ and $\mathcal{C}'$ be two equivalent chronicles, so for every sequence S, $\mathcal{I}_\mathcal{C}(S) = \mathcal{I}_{\mathcal{C}'}(S)$. Suppose that $\mathcal{E}^- \neq \mathcal{E}'^-$, it is easy to design a sequence S such that $\mathcal{I}_\mathcal{C}(S) \neq \mathcal{I}_{\mathcal{C}'}(S)$, so condition 1 must hold. Let $i_\mathcal{C} = (i_\mathcal{C}^+, i_\mathcal{C}^-) = \{(e_{i_1}, t_{i_1}) \dots (e_{i_n}, t_{i_n})\}$ denote an instance of $\mathcal{I}_\mathcal{C}(S) = \mathcal{I}_{\mathcal{C}'}(S)$. Definition 5 ensures there exists a couple of one-to-one correspondences $f : \mathcal{X} \to \{t_{i_1}, \dots, t_{i_n}\}$ and $f' : \mathcal{X}' \to \{t_{i_1}, \dots, t_{i_n}\}$ with for every $x \in \mathcal{X}$, $(e, f(x)) \in i_\mathcal{C}^+$ and $e = \mathcal{M}(x)$ and for every $x \in \mathcal{X}'$, $(e, f'(x)) \in i_\mathcal{C}^+$ and $e = \mathcal{M}'(x)$. Condition 2.a thus holds. By construction of the inequality system, $i_\mathcal{C} = (i_\mathcal{C}^+, i_\mathcal{C}^-)$ is an instance of $\mathcal{I}_\mathcal{C}(S)$ iff there exists a one-to-one correspondence f such that for every $e \in \mathcal{E}^-$, $\{x = f(x)\}_{x\in\mathcal{X}} \cup \{x_e^- = t_e\}$ is solution of $Ine^+(\mathcal{C}) \cup Ine^-(\mathcal{C}, CF)$ for at least a given configuration CF of the prohibition constraints of $\mathcal{C}$. As m exists by condition 2.a, it follows that condition 2.b finally holds.

($\Leftarrow$) Consider an instance $i_\mathcal{C}$ of $\mathcal{C}$. By construction of the inequality system, for every $e \in \mathcal{E}^-$ involved in $i_\mathcal{C}^-$, there exists a configuration CF such that $\{x = f(x)\}_{x\in\mathcal{X}} \cup \{x_e^- = t_e\}$ is a solution of $Ine^+(\mathcal{C}) \cup Ine^-(\mathcal{C}, CF)$. As conditions 2.a and 2.b hold, it implies that $i_\mathcal{C}$ is also an instance of $\mathcal{C}'$. As m is a one-to-one correspondence, we can apply the same reasoning to show that any instance $i_{\mathcal{C}'}$ of $\mathcal{C}'$ is also an instance of $\mathcal{C}$. $\mathcal{C}$ and $\mathcal{C}'$ are therefore equivalent.

Example 5. Figure 6 presents a couple of chronicles that are equivalent. Let us denote $\mathcal{C}_5 = (\mathcal{X}_5, \mathcal{A}_5, \mathcal{T}_5, \mathcal{E}_5^+ \cup \mathcal{E}_5^-, \mathcal{M}_5, \mathcal{F}_5)$ and $\mathcal{C}_6 = (\mathcal{X}_6, \mathcal{A}_6, \mathcal{T}_6, \mathcal{E}_6^+ \cup \mathcal{E}_6^-, \mathcal{M}_6, \mathcal{F}_6)$. The events involved in prohibition constraints in both chronicles are the same: $\mathcal{E}_5^- = \{c\} = \mathcal{E}_6^-$. Consider now the one-to-one correspondence $m : \mathcal{X}_5 \to \mathcal{X}_6$ such that $m(x_0) = y_0$, $m(x_1) = y_1$ and $m(x_2) = y_2$. The set of inequalities $Ine^+(\mathcal{C}_5)$ is $\{x_1 - x_0 \geq 2, x_1 - x_0 \leq 6, x_2 - x_0 \geq 6, x_2 - x_0 \leq 20, x_2 - x_1 \geq 5, x_2 - x_1 \leq 5\}$. By transitivity over the set of inequalities, it can be noticed that $\{x_2 - x_0 \geq$

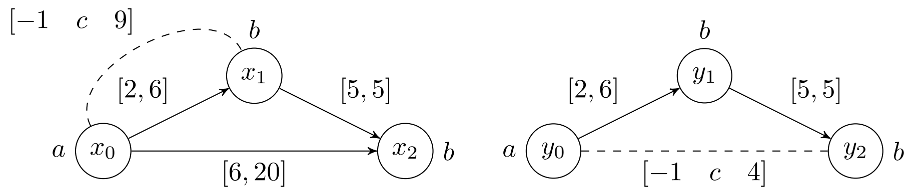

Figure 6: Two equivalent chronicles ($\mathcal{C}_5$ (left) and $\mathcal{C}_6$ (right)).

$6, x_2 - x_0 \leq 20\}$ can be equivalently restricted to $\{x_2 - x_0 \geq 7, x_2 - x_0 \leq 11\}$. Now consider the set of inequalities $Ine^+(\mathcal{C}_6) = \{y_1 - y_0 \geq 2, y_1 - y_0 \leq 6, y_2 - y_1 \geq 5, y_2 - y_1 \leq 5\}$. Also by transitivity, $Ine^+(\mathcal{C}_6)$ can be rewritten as $\{y_1 - y_0 \geq 2, y_1 - y_0 \leq 6, y_2 - y_0 \geq 7, y_2 - y_0 \leq 11, y_2 - y_1 \geq 5, y_2 - y_1 \leq 5\}$ so by definition of m: $Ine^+(\mathcal{C}_6) = \{m(x_1) - m(x_0) \geq 2, m(x_1) - m(x_0) \leq 6, m(x_2) - m(x_0) \geq 7, m(x_2) - m(x_0) \leq 11, m(x_2) - m(x_1) \geq 5, m(x_2) - m(x_1) \leq 5\}$. As the prohibition constraint is not influential, it means that $\{x = t_x\}_{x \in \mathcal{X}_2}$ is a solution of $Ine^+(\mathcal{C}_5)$ iff $\{m(x) = t_x\}_{x \in \mathcal{X}_2}$ is solution of $Ine^+(\mathcal{C}_6)$. Finally consider the variable x_c^-, we have $Ine^-(\mathcal{C}_5) = \{x_c^- - x_0 \leq -1, x_c^- - x_1 \geq 9\}$ and $Ine^-(\mathcal{C}_6) = \{x_c^- - m(x_0) \leq -1, x_c^- - m(x_2) \geq 4\}$. There are two configurations in $Ine^-(\mathcal{C}_5)$ and in $Ine^-(\mathcal{C}_6)$. Obviously, $\{x = t_x\}_{x \in \mathcal{X}_2} \cup \{x_c^- = t_c\}$ is solution of $Ine^+(\mathcal{C}_5) \cup \{x_c^- - x_0 \leq -1\}$ iff $\{m(x) = t_x\}_{x \in \mathcal{X}_5} \cup \{x_c^- = t_c\}$ is solution of $Ine^+(\mathcal{C}_6) \cup \{x_c^- - m(x_0) \leq -1\}$. Similarly, $\{x = t_x\}_{x \in \mathcal{X}_5} \cup \{x_c^- = t_c\}$ is solution of $Ine^+(\mathcal{C}_5) \cup \{x_c^- - x_1 \geq 9\}$ iff $\{m(x) = t_x\}_{x \in \mathcal{X}_5} \cup \{x_c^- = t_c\}$ is solution of $Ine^+(\mathcal{C}_6) \cup \{x_c^- - m(x_1) \geq 9\}$. As $Ine^+(\mathcal{C}_6)$ ensures that $m(x_2) - m(x_1) = 5$, it follows it must also be solution of $Ine^+(\mathcal{C}_6) \cup \{x_c^- - m(x_2) \geq 4\}$. Hence the equivalence between chronicles $\mathcal{C}_5$ and $\mathcal{C}_6$.

4.3 Chronicle covering

A chronicle $\mathcal{C}'$ covers a chronicle $\mathcal{C}$ if whatever the input event sequence considered ($\forall \mathcal{S}$) an instance of $\mathcal{C}$ is recognized each time an instance of $\mathcal{C}'$ is also recognized. This notion is important as it asserts that as soon as a chronicle is recognized, the other one will also be.

Definition 14 Covering of positive chronicles. Let $\mathcal{C}$ and $\mathcal{C}'$ be two positive chronicles, $\mathcal{C}'$ covers $\mathcal{C}$ (denoted $\mathcal{C}' \succ \mathcal{C}$), if for every input flow $\mathcal{S}$, for every instance $i_{\mathcal{C}} \in \mathcal{I}_{\mathcal{C}}(S)$ there exists $i_{\mathcal{C}'} \in \mathcal{I}_{\mathcal{C}'}(S)$ such that $i_{\mathcal{C}'} \subseteq i_{\mathcal{C}}$.

Example 6. Figure 7 depicts two positive chronicles $\mathcal{C}_7$ and $\mathcal{C}_8$ such that $\mathcal{C}_7 \succ \mathcal{C}_8$. Consider one input flow S, any instance of $i_{\mathcal{C}_8} \in \mathcal{I}_{\mathcal{C}_8}(S)$ is such that $i_{\mathcal{C}_8} = \{(a, t_0), (c, t_1), (b, t_2)\}$ with $t_1 - t_0 \in [2, 3]$ and $t_2 - t_1 \in [4, 5]$ which follows that

Figure 7: Chronicle $\mathcal{C}_7$ (left) covers chronicle $\mathcal{C}_8$ (right).

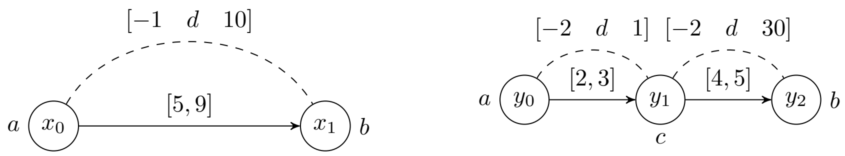

Figure 8: Chronicle $\mathcal{C}_9$ (left) covers chronicle $\mathcal{C}_{10}$ (right).

$t_2 - t_0 \in [6, 8] \subset [5, 9]$. Therefore $i_{\mathcal{C}_7} = \{(a, t_0), (b, t_2)\}$ is also an instance of $\mathcal{I}_{\mathcal{C}_7}(S)$: $i_{\mathcal{C}_7} \subseteq i_{\mathcal{C}_8}$.

Theorem 15. *A positive chronicle $\mathcal{C}_+$ covers a positive chronicle $\mathcal{C}'_+$ iff the d-graph of $\mathcal{C}_+$ covers the d-graph of $\mathcal{C}'_+$.*

Proof. We say that a d-graph dg covers another d-graph dg' if the set of variables of dg is a subset of the variables of dg' and for every edge from x_i to x_j in dg it holds a distance that is greater or a equal to the distance between x_i to x_j in dg'. It follows that given a solution of dg', a solution of dg is obtained by only keeping the values of the variables involved in dg. Theorem 15 follows.

Proposition 16. *Let $\mathcal{C}$ and $\mathcal{C}'$ be two positive chronicles, $\mathcal{C}$ and $\mathcal{C}'$ are equivalent iff $\mathcal{C} \succ \mathcal{C}'$ and $\mathcal{C}' \succ \mathcal{C}$.*

Proof. Straightforward from Definition 14.

Definition 17 Covering. A chronicle $\mathcal{C}'$ covers a chronicle $\mathcal{C}$: $\mathcal{C}' \succ \mathcal{C}$, if for every input flow $\mathcal{S}$, for every $i_{\mathcal{C}} = (i^+_{\mathcal{C}}, i^-_{\mathcal{C}}) \in \mathcal{I}_{\mathcal{C}}(\mathcal{S})$ there exists an instance $i_{\mathcal{C}'} = (i^+_{\mathcal{C}'}, i^-_{\mathcal{C}'}) \in \mathcal{I}_{\mathcal{C}'}(\mathcal{S})$ such that:

1. $i^+_{\mathcal{C}'} \subseteq i^+_{\mathcal{C}}$;
2. $\mathcal{E}'^- \subseteq \mathcal{E}^-$
3. for every $e \in \mathcal{E}'^-$, $\{(e, t) \in i^-_{\mathcal{C}}\} \subseteq \{(e, t) \in i^-_{\mathcal{C}'}\}$

Example 7. Figure 8 shows a chronicle $\mathcal{C}_9$ (on the left) that covers a chronicle $\mathcal{C}_{10}$ (on the right): $\mathcal{C}_9 \succ \mathcal{C}_{10}$. Chronicle $\mathcal{C}_9$ is an extension of chronicle $\mathcal{C}_7$ with a prohibition constraint for event type d (see Figure 7). Chronicle $\mathcal{C}_{10}$

is an extension of chronicle $\mathcal{C}_8$ added with a couple of prohibition constraints for the same event type d. Now consider an input flow $\mathcal{S}$ and an instance $i_{\mathcal{C}_{10}} = (i^+_{\mathcal{C}_{10}}, i^-_{\mathcal{C}_{10}}) \in \mathcal{I}_{\mathcal{C}_{10}}(\mathcal{S})$, we show there exists $i_{\mathcal{C}_9} = (i^+_{\mathcal{C}_9}, i^-_{\mathcal{C}_9}) \in \mathcal{I}_{\mathcal{C}_9}(\mathcal{S})$ as defined in Definition 5. As prohibition constraints are not influential, it is obvious that $i^+_{\mathcal{C}_9} = \{(a,t_0),(b,t_2)\} \subseteq i^+_{\mathcal{C}_{10}} = \{(a,t_0),(c,t_1),(b,t_2)\}$ (see Figure 7). Now regarding the prohibition constraints, they are on the same event type d in $\mathcal{C}_9$ and $\mathcal{C}_{10}$. Suppose now that on input flow $\mathcal{S}$, we have $i^-_{\mathcal{C}_{10}} = \{(d,t^d_0),\dots(d,t^d_n)\}$ then we know that $\forall t^d_i, i \in \{0,\dots,n\}$ either $t^d_i < t_0-2$ or $t^d_i > t_2+30$ so $t^d_i < t_0-1$ or $t^d_i > t_2+10$ which means that $\{(d,t^d_0),\dots(d,t^d_n)\} \subseteq i^-_{\mathcal{C}_9}$.

Theorem 18. *Let $\mathcal{C}$ and $\mathcal{C}'$ be two chronicles, the chronicle $\mathcal{C}'$ covers the chronicle $\mathcal{C}$ (i.e $\mathcal{C}' \succ \mathcal{C}$) iff*

1. $\mathcal{E}'^- \subseteq \mathcal{E}^-$;
2. *there exists a one-to-one correspondence m between $\mathcal{X}'$ and a subset $\mathcal{X}_m$ of $\mathcal{X}$, $\forall x \in \mathcal{X}'$ such that*
   - (a) $\mathcal{M}'(x) = \mathcal{M}(m(x))$; *and*
   - (b) *for every $e \in \mathcal{E}'^-$, if $\{m(x) = t_x\}_{x\in\mathcal{X}'} \cup \{x^-_e = t_e\}$ belongs to a solution of $Ine^+(\mathcal{C}) \cup Ine^-(\mathcal{C}, CF)$ for at least a given configuration of the prohibition constraints of $\mathcal{C}$ then $\{x = t_x\}_{x\in\mathcal{X}'} \cup \{x^-_e = t_e\}$ is also a solution of $Ine^+(\mathcal{C}') \cup Ine^-(\mathcal{C}', CF')$ for at least a given configuration CF' of the prohibition constraints of $\mathcal{C}'$.*

Proof. ($\Rightarrow$)Condition 1 holds from Definition 17. Let $i_\mathcal{C} = (i^+_\mathcal{C}, i^-_\mathcal{C})$ denote an instance of $\mathcal{I}_\mathcal{C}(S)$ and consider that $i^+_\mathcal{C} = \{(e_{i_1},t_{i_1})\dots(e_{i_n},t_{i_n})\}$. As $\mathcal{C}'$ covers $\mathcal{C}$, there must exist an instance $i_{\mathcal{C}'} = (i^+_{\mathcal{C}'}, i^-_{\mathcal{C}'})$ such that $i^+_{\mathcal{C}'} = \{(e_{j_1},t_{j_1})\dots(e_{j_m},t_{j_m})\} \subseteq \{(e_{i_1},t_{i_1})\dots(e_{i_n},t_{i_n})\}$. By applying the same type of reasoning as in the proof ($\Rightarrow$) of Theorem 13, conditions 2.a and 2.b hold (the only difference is that m is a one-to-one correspondence between a subpart of $\mathcal{X}$ and $\mathcal{X}'$).
($\Leftarrow$) Let $i_\mathcal{C}$ be an instance of $\mathcal{C}$, by applying the same type of reasoning as in the proof ($\Leftarrow$) of Theorem 13, we can show that $i_\mathcal{C}$ is also an instance of $\mathcal{C}'$, hence the result.

Example 8. The set of inequalities $Ine^+(\mathcal{C}_9)$ is $\{x_1 - x_0 \geq 5, x_1 - x_0 \leq 9\}$. Chronicle $\mathcal{C}_9$ has only one prohibition constraint c^9_d that leads to the set of inequalities $Ine^<(\mathcal{C}_9) = \{x^-_d - x_0 < -1\}$ and $Ine^>(\mathcal{C}_9) = \{x^-_d - x_1 > 10\}$. The set of inequalities $Ine^+(\mathcal{C}_{10})$ is $\{y_1 - y_0 \geq 2, y_1 - y_0 \leq 3, y_2 - y_1 \geq 4, y_2 - y_1 \leq 5\}$. Chronicle $\mathcal{C}_{10}$ has two prohibition constraints. The first one, denoted c^{101}_d, leads to the set of inequalities $Ine^<(c^{101}_d) = \{x^-_d - y_0 < -2\}$ and $Ine^>(c^{101}_d) = \{x^-_d - y_1 > 1\}$. The second one, denoted c^{102}_d leads to the set of inequalities $Ine^<(c^{102}_d) = \{x^-_d - y_1 < -2\}$ and $Ine^>(c^{102}_d) = \{x^-_d - y_2 > 30\}$. So for chronicle $\mathcal{C}_{10}$ there are four possible configurations:

1. $Ine^{-}(\mathcal{C}_{10}, CF_1) = Ine^{<}(c_d^{101}) \cup Ine^{<}(c_d^{102}) = \{x_d^{-} - y_0 < -2, x_d^{-} - y_1 < -2\}$;
2. $Ine^{-}(\mathcal{C}_{10}, CF_2) = Ine^{<}(c_d^{101}) \cup Ine^{>}(c_d^{102}) = \{x_d^{-} - y_0 < -2, x_d^{-} - y_2 > 30\}$;
3. $Ine^{-}(\mathcal{C}_{10}, CF_3) = Ine^{>}(c_d^{101}) \cup Ine^{<}(c_d^{102}) = \{x_d^{-} - y_1 > 1, x_d^{-} - y_1 < -2\}$;
4. $Ine^{-}(\mathcal{C}_{10}, CF_4) = Ine^{>}(c_d^{101}) \cup Ine^{>}(c_d^{102}) = \{x_d^{-} - y_1 > 1, x_d^{-} - y_2 > 30\}$.

Now consider the one-to-one correspondence $m : \mathcal{X}_9 \to \{y_0, y_2\} \subset \mathcal{X}_{10}$ such that $m(x_0) = y_0, m(x_1) = y_2$ where $\mathcal{X}_9$ and $\mathcal{X}_{10}$ are the set of variables of $\mathcal{C}_9$ and $\mathcal{C}_{10}$ respectively. Suppose now that $\{m(x_0) = t_0, m(x_1) = t_1\} \cup \{x_d^{-} = t_d\}$ belongs to a solution in $Ine^{+}(\mathcal{C}_{10}) \cup Ine^{-}(\mathcal{C}_{10}, CF_1)$ so it means that $\{y_1 - m(x_0) \geq 2, y_1 - m(x_0) \leq 3, m(x_1) - y_1 \geq 4, m(x_1) - y_1 \leq 5, x_d^{-} - m(x_0) < -2, x_d^{-} - y_1 < -2\}$. As $2 + m(x_0) \leq y_1 \leq 3 + m(x_0)$ it means that if $x_d^{-} - m(x_0) < -2$ then $x_d^{-} - y_1 < -2$ also holds and can be removed from the inequality set. Moreover, we also have $m(x_0) + 6 \leq m(x_1) \leq m(x_0) + 8$. So $\{m(x_0) = t_0, m(x_1) = t_1\} \cup \{x_d^{-} = t_d\}$ is also a solution of $Ine^{+}(\mathcal{C}_9) \cup Ine^{<}(\mathcal{C}_9)$. Noticing that $Ine^{-}(\mathcal{C}_{10}, CF_2)$ and $Ine^{-}(\mathcal{C}_{10}, CF_3)$ are both inconsistent, it follows that, if $\{m(x_0) = t_0, m(x_1) = t_1\} \cup \{x_d^{-} = t_d\}$ belongs to a solution that is not in $Ine^{+}(\mathcal{C}_{10}) \cup Ine^{-}(\mathcal{C}_{10}, CF_1)$ then it must be in $Ine^{+}(\mathcal{C}_{10}) \cup Ine^{-}(\mathcal{C}_{10}, CF_4)$ which implies that $m(x_0) + 6 \leq m(x_1) \leq m(x_0) + 8$ and $x_d^{-} - m(x_1) > 30$ so it is a solution of $Ine^{+}(\mathcal{C}_9) \cup Ine^{>}(\mathcal{C}_9)$. Theorem 18 then ensures that $\mathcal{C}_9 \succ \mathcal{C}_{10}$.

5 Implementation of the chronicle comparison criteria

All the results presented here above have been fully implemented within a C++ toolkit called TiPaDiag (Time Patterns for Diagnosis) and all the presented examples have been generated with this tool. The aim of this toolkit is to provide a tool-chain for modeling, analyzing and learning chronicles [Subias et al(2014)]. In particular, TiPaDiag also embeds a chronicle recognizer, called Yacre (Yet Another Chronicle Recognition Engine) that is a C++ clone of the initial CRS engine from [Dousson et al(1993), Dousson(2002)] that is able to recognize the chronicles as defined in Definition 4. Both engines rely on clock propagations to compute on-the-fly the set of chronicle instances by managing a set of partial instances (partially assigned instances). When the engine receives an event, it checks for every partial instance whether the date of the received event is consistent with the time constraints involved in the instance by updating their clock. If the chronicle is positive, this operation is in $O(|\mathcal{X}|^2)$ where $|\mathcal{X}|$ is the number of nodes in the chronicle [Dousson et al(1993)]. In the general case, the engine also needs to check the consistency with the prohibition constraints so the operation is in $O(|\mathcal{X}|^2 + |\mathcal{F}|)$ where $|\mathcal{F}|$ is the number of prohibition constraints. From a complexity viewpoint, the advantage of using prohibition constraints is that it aims at reducing the set of current partial instances to handle by the engine.

Regarding the chronicle comparison criteria, their implementation is detailed below. As explained in Section 4.1 for the consistency criteria, we build the distance graph of its underlying STP and check for negative cycles (see Corollary 10). To implement the equivalence and covering tests, we always first distinguish the type of chronicles that are involved in the tests for performance issues. If both chronicles are positive, all the tests then consist in building the d-graphs. If both chronicles involve prohibition constraints, it is then also preferred to start by building the d-graphs to analyze the positive part of the chronicle and then structurally analyze the prohibition constraints if it is possible. The last case is the generic one that can implement all the tests by the use of inequality systems. We implement the different inequality systems of the chronicles involved in these criteria with not necessarily closed polyhedra (NNC for short) as defined in [Bagnara et al(2002)]. Polyhedra are mostly used to analyze software programs efficiently. We have used the up-to-date C++ Parma Polyhedra Library (PPL).[3]

Algorithm 1 describes a sketch of the implementation of the equivalence criteria in TiPaDiag. In the case of positive chronicles (lines 1-4), the equivalence test is implemented by building both d-graphs and checking whether they are isomorphic (see Theorems 12) which relies on the Floyd-Warshall algorithm that is in $O(n^3)$ with n the number of variables in a chronicle. The function *checkIsomorphicDGraphs* performs this analysis and returns the set of possible one-to-one correspondences between the two graphs. This set must be non-empty if the equivalence holds. Then, in the general case, we also first attempt to solve the problem with *checkIsomorphicDGraphs* and a structural analysis of the prohibition constraints to check whether they are all isomorphic (a prohibition constraint (x_1, x_2, t_1, t_2) of $\mathcal{C}$ is isomorphic to a prohibition constraint (x_3, x_4, t_3, t_4) of $\mathcal{C}'$ iff $m(x_1) = x_3, m(x_2) = x_4, t_1 = t_3, t_2 = t_4$), this analysis is in $O(|\mathcal{F}|^2)$ (lines 5-8). If, however, these isomorphisms do not hold, it is not sufficient to conclude and a further investigation with the help of polyhedra is required as in the latter case. In the general case (lines 9-22), the algorithm basically searches for a correspondence m between the variables of $\mathcal{C}$ and $\mathcal{C}'$ so that the set of possible instances of $\mathcal{C}$ is the set of possible instances of $\mathcal{C}'$ (see Theorem 13). To do so, for each chronicle and for any of its configurations CF, the algorithm converts the corresponding inequality system to a polyhedron and stores this polyhedron in a specific polyhedron set of PPL (see Algorithm 2). Each polyhedron set then implicitly represents the set of possible instances of each chronicle. Then we take advantage of a PPL operator, called *geometrically_equals* that checks whether two polyhedra sets represent the same solution space. To look for such a correspondence m, either we test the ones in the set of compatible correspondences M if *checkIsomorphicDGraphs* has previously been called (lines 10-14) or we search for it directly (lines 16-21). Back to the complexity analysis for this last case, let

[3] `http://www.cs.unipr.it/Software/`

n_{max} be the maximal number of variables that are labeled with the same event type in $\mathcal{C}$, the number of correspondences m to deal with is then in $O(n_{max}!)$. The complexity of Algorithm 2 is the complexity of *geometrically_equals* (line 12) that is in $O(2^n)$ where n is the maximum between the number of variables involved in the inequality systems of $\mathcal{C}$ and $\mathcal{C}'$.

```
Data: Chronicles C, C'
Result: true iff C, C' are equivalent
if Both chronicles are positive then
    M ← checkIsomorphicDGraphs(C, C');
    return M ≠ ∅
end
M ← checkIsomorphicDGraphs(C, C');
if ∃m ∈ M, isomorphicProhibitionConstraints(C, C', m) then
    return true;
end
if M ≠ ∅ then
    for m ∈ M do
        if polyhedra_equals(C, C', m) then
            return true
        end
    end
else
    for every one-to-one correspondence m of C, C' do
        if polyhedra_equals(C, C', m) then
            return true
        end
    end
end
return false
```

Algorithm 1: Algorithm of the equivalence criteria

As far as the implementation of the covering criteria is concerned, to check whether $\mathcal{C} \succ \mathcal{C}'$ or not, the implementation follows exactly the same sketch as Algorithm 1. There are only few differences. *checkIsomorphicDGraphs*$(\mathcal{C}, \mathcal{C}')$ is replaced by *checkCoveringDGraphs*$(\mathcal{C}, \mathcal{C}')$ that searches for correspondence such that the d-graph of $\mathcal{C}$ covers the one of $\mathcal{C}'$ (see Theorem 15). The call of the function *isomorphicProhibitionConstraints* is replaced by the call of the function *coveringProhibitionConstraints* that checks whether a prohibition con-

1 **Function** $polyhedra\_equals(\mathcal{C}, \mathcal{C}', m)$;
Data: Chronicles $\mathcal{C}$, $\mathcal{C}'$
Data: one-to-one correspondence m of $\mathcal{C}$, $\mathcal{C}'$
Result: true if the test succeeds
// $phs1$ is a polyhedra set of PPL
2 $phs1 \leftarrow \varnothing$;
3 **for** every configuration CF of $\mathcal{C}$ **do**
4 $\quad ph \leftarrow getPolyhedron(Ine^{+}(\mathcal{C}) \cup Ine^{-}(\mathcal{C}, CF))$;
5 $\quad phs1 \leftarrow phs1 \cup \{ph\}$;
6 **end**
// $phs2$ is a polyhedra set of PPL
7 $phs2 \leftarrow \varnothing$;
8 **for** every configuration CF of $\mathcal{C}'$ **do**
9 $\quad ph \leftarrow getPolyhedron(Ine^{+}(\mathcal{C}') \cup Ine^{-}(\mathcal{C}', CF))$;
10 $\quad phs2 \leftarrow phs2 \cup \{ph\}$;
11 **end**
12 **return** $geometrically\_equals(phs1, phs2)$;

Algorithm 2: Polyhedra equivalence test.

straint (x_1, x_2, t_1, t_2) of $\mathcal{C}$ covers a prohibition constraint (x_3, x_4, t_3, t_4) of $\mathcal{C}'$, (i.e. $m(x_1) = x_3, m(x_2) = x_4, t_1 \geq t_3, t_2 \leq t_4$). And the call of *polyhedra_equals* is replaced by *polyhedra_covers*, the function *polyhedra_covers* is defined as *polyhedra_equals* (Algorithm 2) where the call of *geometrically_equals* (line 12) is replaced by the PPL operator *geometrically_covers*.

Computing the covering criteria is simpler in practice than computing the equivalence criteria, but from a complexity point of view in the worst case, the results are the same. Proposed algorithms have been fully implemented within the TiPaDiag platform. Table 1 presents a selection of our tests. For each test, we randomly generate a couple of chronicles with the same configuration based on three parameters: number of nodes $|\mathcal{X}|$, number of events types $|\mathcal{E}^{+}|$, and the number of prohibition constraints $|\mathcal{F}|$. Then we compute the equivalence tests between both chronicles and between the first chronicle and itself (i.e. the first test is very likely false, the second is always true). Each configuration has been tested 2000 times each. Table 1 presents the mean/minimal/maximal time for each configuration [4] (on the left are configurations for positive chronicles, on the right are the ones for chronicle with at least one prohibition constraint). While prohibition constraints clearly improve the expressivity of chronicles, the results show their impact on the computation complexity.

[4] Tests performed on a AMD Ryzen 7 1700X (3.4 GHz) with Linux.

Config.	Mean/Min/Max (ms)	Config.	Mean/Min/Max (ms)
5,3,0	0/0/1	5,5,5	60/5/210
5,5,0	0/0/1	5,3,5	90/22/2049
10,5,0	6/5/9	10,8,2	235/29/4545
10,10,0	6/5/11	10,8,3	581/82/26657
50,50,0	1675/1609/1793	10,10,5	2424/38/61121
50,25,0	1676/1607/1892	10,8,6	11736/504/455221
100,50,0	22865/22355/24226	15,15,1	21099/109/4617238
100,100,0	22865/22310/24292	16,16,1	87199/233/21138419

Table 1: Equivalence test: computation time on random chronicles.

6 Conclusions and perspectives

The context of the current work is event-based behavior analysis and more precisely of behaviors involving time information. The research presented in this article is motivated by the need for modeling and analyzing any behavior of interest by succinct pieces of information combining timed positive constraints (timed events that must occur) and timed negative constraints (timed events that must not occur). To do so, we propose a formal extension of chronicles by the use of prohibition constraints which aims at increasing the expressivity of positive chronicles to include negative behaviors. Beside its use for behavior modeling, the new chronicle formalism proposed in this paper allows the definition of several criteria to formally characterize and compare a set of chronicles. An implemented solution is also proposed allowing to check chronicle consistency and to compare chronicles with the equivalence and covering tests. The proposed implementation relies on the solving of systems of linear inequalities by polyhedra techniques. As a future work, we plan to extend the proposal by considering other criteria to compare chronicles with the main objective to improve the quality of a chronicle database and therefore the quality of their recognition in any type of application, notably in the field of diagnosis where chronicles can be used to model normal behaviors as well as faulty ones as observable timed patterns.

References

[Aguilar et al(1994)] Aguilar J, Bousson K, Dousson C, Ghallab M, Guasch A, Milne R, Nicol C, Quevedo J, Travé-Massuyès L (1994) Tiger: real-time situation assessment of dynamic systems. Technical report, LAAS-CNRS, Toulouse, France

[Allen(1983)] Allen JF (1983) Maintaining knowledge about temporal intervals. Communications of the ACM 26(11):832 – 843

[Allen(1984)] Allen JF (1984) Towards a general theory of action and time. Artificial Intelligence 23(2):123–154

[Artikis et al(2012)] Artikis A, Skarlatidis A, Portet F, Paliouras G (2012) Logic-Based Event Recognition. Knowledge Engineering Review 27(4):469–506

[Bagnara et al(2002)] Bagnara R, Ricci E, Zaffanella E, Hill PM (2002) Possibly not closed convex polyhedra and the Parma Polyhedra Library. Quaderno 286, Dipartimento di Matematica, Università di Parma, Italy

[Bauer et al(2011)] Bauer A, Leucker M, Schallhart C (2011) Runtime Verification for LTL and TLTL. ACM Transactions on Software Engineering and Methodology (TOSEM) 20(4)

[Bertrand et al(2008)] Bertrand O, Carle P, Choppy C (2008) Towards a coloured Petri nets semantics of a chronicle language for distributed simulation processing. In: In CHINA 2008 Workshop (Concurrency metHods: Issues and Applications), pp 105–119

[Boufaied et al(2004)] Boufaied A, Subias A, Combacau M (2004) Distributed fault detection with delays consideration. In: 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI'2004), Orlando,USA, pp 135–140

[Cao et al(2015)] Cao L, Yu PS, Kumar V (2015) Nonoccurring behavior analytics: A new area. IEEE Intelligent Systems 30(6):4–11

[Carle et al(1998)] Carle P, Benhamou P, Ornato M, Dolbeau F (1998) Building dynamic organizations using intentions recognition. Tech. Rep. ONERA-TP - 99-136, Office national d'études et de recherches aérospatiales, Châtillon, France

[Carle et al(2013)] Carle P, Choppy C, Kervarc R, Piel A (2013) Safety of Unmanned Aircraft Systems Facing Multiple Breakdowns. In: Choppy C, Sun J (eds) 1st French Singaporean Workshop on Formal Methods and Applications (FSFMA 2013), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, OpenAccess Series in Informatics (OASIcs), vol 31, pp 86–91

[Carrault et al(1999)] Carrault G, Cordier M, Quiniou R, Garreau M, andA Bardou JB (1999) A model-based approach for learning to identify cardiac arrhythmias. Lecture Note on Artificial Intelligence 1620:165–174

[Cordier and Dousson(2000)] Cordier MO, Dousson C (2000) Alarm driven monitoring based on chronicles. In: 4th Sumposium on Fault Detection Supervision and Safety for Technical Processes (SafeProcess), Budapest, Hungary, pp 286–291

[Cordier et al(2007)] Cordier MO, Guillou XL, Robin S, Rozé L, Vidal T (2007) Distributed chronicles for on-line diagnosis of web services. In: Biswas G, Koutsoukos X, Abdelwahed S (eds) 18th International Workshop on Principles of Diagnosis, pp 37–44

[Cram et al(2009)] Cram D, Kröner A, Mille A (2009) Using object memory patterns to make plan-driven help systems more flexible. In: Workshops Proceedings of the 5th International Conference on Intelligent Environments, Barcelona, Spain, 19th of July, 2009, pp 45–50

[Dechter et al(1991)] Dechter R, Meiri I, Pearl J (1991) Temporal constraint networks. Artificial intelligence 49(1):61–95

[Dousson(1996)] Dousson C (1996) Alarm driven supervision for telecommunication networks:ii- on line chronicle recognition. Annales des Télécommunications 51 (9-10):501–508

[Dousson(2002)] Dousson C (2002) Extending and unifying chronicle representation with event counters. In: Proceedings of the 15th Eureopean Conference on Artificial Intelligence, ECAI'2002, Lyon, France, July 2002, pp 257–261

[Dousson et al(1993)] Dousson C, Gaborit P, Ghallab M (1993) Situation recognition: representation and algorithms. In: IJCAI: International Joint Conference on Artificial Intelligence, Chambéry, France, pp 166–172

[Floyd(1962)] Floyd RW (1962) Algorithm 97: Shortest path. Commun ACM 5(6):345–, DOI 10.1145/367766.368168

[Ghallab(1996)] Ghallab M (1996) On chronicles : Representation, on-line recognition and learning. In: Proc. of the 5th International Conference on Principles of Knowledge Representation and Reasoning (KR-96), pp 597–606

[Guillou et al(2008)] Guillou XL, Cordier MO, Robin S, Rozé L (2008) Chronicles for on-line diagnosis of distributed systems. In: 18th European Conference on Artificial Intelligence, ECAI08, Patras, Greece, pp 194–198

[Heintz(2001)] Heintz F (2001) Chronicle recognition in the WITAS UAV project a preliminary report. In: Swedish AI Society Workshop (SAIS2001)

[Kowalski and Sergot(1986)] Kowalski R, Sergot M (1986) A logic-based calculus of events. New Generation Computing 4(1):67–95

[Laborie and Krivine(1997)] Laborie P, Krivine JP (1997) Automatic generation of chronicles and its application to alarm processing in power distribution systems. In: 8th international workshop of diagnosis (DX97), Mont Saint-Michel, France

[McCarthy and Hayes(1969)] McCarthy J, Hayes PJ (1969) Some philosophical problems from the standpoint of artificial intelligence. Machine Intelligence 4

[Morin and Debar(2003)] Morin B, Debar H (2003) Correlation on intrusion: an application of chronicles. In: 6th International Conference on recent Advances in Intrusion Detection RAID, Pittsburgh, USA

[Pencolé and Subias(2009)] Pencolé Y, Subias A (2009) A chronicle-based diagnosability approach for discrete timed-event systems: Application to web-services. Journal of Universal Computer Science 15(17):3246–3272

[Rota and Thonnat(2000)] Rota N, Thonnat M (2000) Activity recognition from video sequences using declarative models. In: 14th International Workshop on Principles of Diagnosis (DX00), Morelia, Michoacen, Mexico

[Saddem et al(2010)] Saddem R, Toguyeni A, Moncef T (2010) Consistency's checking of chronicles' set using time Petri nets. In: Control & Automation (MED), 2010 18th Mediterranean Conference on, IEEE, pp 1520–1525

[Sahuguède et al(2018)] Sahuguède A, Fergani S, Corronc EL, Lann ML (2018) Mapping chronicles to a k-dimensional euclidean space via random projections. In: 14th annual IEEE International Conference on Automation Science and Engineering (CASE), Munich, Germany,

[Subias et al(2014)] Subias A, Travé-Massuyès L, Corronc EL (2014) Learning chronicles signing multiple scenario instances. IFAC World Congress, Le Cap, South Africa, 26-29 August

[Vizcarrondo et al(2013)] Vizcarrondo J, Aguilar J, Subias A, Exposito E (2013) Distributed chronicles for recognition of failures in web services composition. In: XXXIX Latin American Computing Conference (CLEI 2013), Puerto Azul(Venezuela)