Towards the use of Situation Hierarchies for supporting Decision Making: A Formal Lattice-Based Approach

Franco Giustozzi$^{[0000-0002-2709-2625]}$, Julien Saunier$^{[0000-0002-7385-4395]}$, and Cecilia Zanni-Merk$^{[0000-0002-5189-9154]}$

Normandie Université, INSA Rouen, LITIS, Rouen 76000, France

Abstract. Monitoring is a critical task to manage modern applications using cyber-physical systems such as in healthcare, environmental monitoring, water and energy distribution or Industry 4.0. Its main goal is the optimization of the equipment use, the prediction or avoidance of failures, and the change in the operating modes of the system according to both the system goals and its current condition. The abnormal situations that could lead to failures can have different levels of severity, and can be nested in different ways. In this context, this paper proposes a method to build a lattice, ordering those situations depending on the constraints they rely on. This lattice represents a road-map of all the situations that can be reached from a given one, desirable or undesirable. This helps in decision support, by allowing the identification of the actions that can be taken to correct the abnormality avoiding in this way the interruption of the system processes.

Keywords: Knowledge Representation · Lattice Theory · Industry 4.0

1 Introduction

Monitoring is a critical task to manage cyber-physical systems in different application fields such as health and environmental monitoring, or water and energy distribution. Its goal is to analyze an ongoing process to find out whether it behaves according to expectations [12]. This allows the avoidance or prediction of undesirable behaviors or situations, such as failures, and the change in the operating modes of the system according to both the system goals and its current condition. Such is also the case of Industry 4.0, whose main objective is to improve production and associated services through the digitization and automation of manufacturing processes. Several fields and technologies, such as the Internet of Things, Robotics, Cyber-Physical Systems, are fundamental to Industry 4.0 in order to build intelligent machines, storage systems and production facilities capable of exchanging information in an autonomous and smart way.

Early detection of situations that may lead to failures in the Industry 4.0 scenario requires the integration of data from heterogeneous data sources in real-time. There are several works that propose the use of different technologies to deal with those issues.

The approach presented in [13] exploits big data technologies for data-driven anomaly detection in manufacturing processes. In [6], the authors propose an approach that uses stream reasoning to face these issues dynamically, while applying classical reasoning approaches to overcome the limitations of stream reasoning when needed. This proposal is used to detect relevant situations, where a situation is a combination of one or several sensor measurements linked through spatio-temporal relationships. Others approaches are surveyed in [9].

Manufacturing processes are not always executed in optimal conditions, without being stopped completely. Expert knowledge enables to describe these *"intermediate"* manufacturing conditions. The associated abnormal situations can have different levels of severity, and be nested in different ways: they can impact other processes or resources that participate in these processes, they can trigger other situations that represent a risk of major interruption of the manufacturing process or a risk of accident.

Once a situation that may lead to failures is detected, decisions must be taken, such as deciding whether the process should be interrupted or continued under sub-optimal conditions. In order to help in decision making and choose the most appropriate action, it is relevant to consider which other situations can be reached according to the possible actions. Thus, this paper proposes an approach to establish an order among the situations, depending on how their constraints are correlated. This order represents a road-map of all the situations, desirable or undesirable, that can be reached from a given one. In this way, it is possible to identify the actions that can be taken to correct the abnormality, considering that certain actions can modify the value of a property and thus change the state of the system, either by satisfying another constraint or, on the contrary, by not satisfying constraints anymore.

The proposed approach uses the lattice theory [8,4,7] to provide an expressive formalization to order the situations by their level of generality or specificity. In this way the hierarchy of situations is formally extracted from the situations definitions.

The remainder of the paper is structured as follows: in section 2 the related work is presented. In section 3, the general approach is introduced, providing the definitions for the construction of the hierarchy of situations. In section 4, the interpretation and exploitation of the lattice to support the decision making is detailed. In Section 5, we present some concluding remarks, including some perspectives for future work.

2 Related Work

As mentioned in the introduction, the definition of abnormal situations requires the integration of data from different data sources, with different underlying meanings and different temporal resolutions. Furthermore, these situations may share constraints and involve similar resources. These challenges have been addressed from different areas of research.

Most of the proposals coming from the Complex Event Processing (CEP) community [3,1,17] offer relatively simple languages that allow to describe how information from different sources should be processed. These languages do not include complete support for reuse of patterns to form hierarchies of events. To overcome this limitation, TESLA [2], an event specification language for CEP, supports content-based event filtering and allows to capture relations among temporally related patterns of events. In the same direction, in [15] the authors present a syntax for Description Logic Event Processing (DELP) to represent both simple events and more complex events built on simple events.

Other approaches to structurally define the event type composition in cyber-physical systems (CPS) adopt the theory of concept lattice [16]. Concept lattice has been used in machine learning, knowledge discovery and software engineering. In [14], a concept lattice-based event model for CPS is presented. With this model, a CPS event is uniformly represented by three components: the event type, its internal attributes, and its external attributes. The internal and external attributes together characterize the event type. The model allows events to be composed across different components and devices within and among both the cybernetic and physical domains.

Another approach is the use of rule-based models such as Adaptive Neuro Fuzzy Interference Systems (ANFIS) models for monitoring wind turbine SCADA (Supervisory Control and Data Acquisition) signals [10,11]. In order to obtain turbine condition statements, the authors implement rules given by an expert who is familiar with the behavior of the turbine, typical faults and their root causes. There are two types of rules: generic rules used to highlight anomalies, and specific rules providing specific condition or potential root cause. In this case, it is possible to determine whether one antecedent is contained in another or not. However, there is not a direct way to determine that an antecedent or situation is partially occurring or that other violations must be satisfied for another situation to occur.

In most of the solutions mentioned above, it is possible to observe a hierarchy of events in the sense that complex events are composed of simple ones, but this hierarchy does not directly provide the information that certain complex events may occur partially in other complex events or that they share certain simple events or not. To the best of our knowledge, no works have been done that attempt to establish a formal representation of a hierarchy among (abnormal) situations. For this, establishing an order among situations permits taking into account the relations that exist among each other and to support the actions to take to overcome the abnormal situations when they happen.

The contribution of the approach presented in this paper is a method that allows to order not only the complex events already defined based on the simple events that compose them, but also to order certain combinations of simple events that do not completely compose a complex event. Additionally, it is possible to consider occurrences of simple events that imply the occurrence of other simple events for the construction of the hierarchy of situations.

3 Proposed Approach

The idea driving this approach is to formally represent a hierarchy among situations that may lead to failures. Therefore, through the use of this hierarchy of situations, decisions can be made based on which others situation can be reached according to the constraints that can be satisfied. This allows to adapt the maintenance schedule or to take further measures to prevent unexpected downtime.

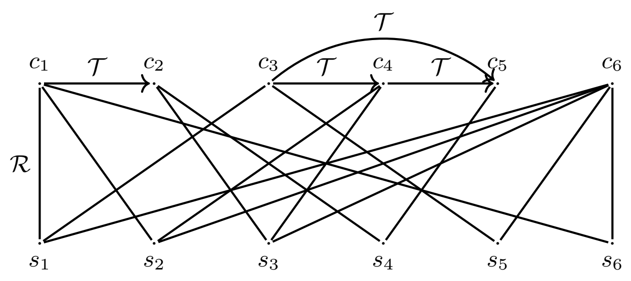

Fig. 1: Situations with the constraints ($\mathcal{R}$) and implications among the constraints ($\mathcal{T}$).

In this section, we firstly introduce the definitions that are necessary for the construction of the hierarchy of situations, and secondly, we describe the steps for building the hierarchy.

3.1 Structure definitions

In order to formally represent a hierarchy of situations, let us consider the following structure $\langle \mathcal{S}, \mathcal{C}, \mathcal{R}, \mathcal{T} \rangle$ where:

- $\mathcal{S} = \{s_1, s_2, s_3, \ldots, s_n\}$ is the set of all the situations,
- $\mathcal{C} = \{c_1, c_2, c_3, \ldots, c_m\}$ is the set of all the constraints,
- $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{C}$ is a binary relation that links a situation with a constraint and
- $\mathcal{T} \subseteq \mathcal{C} \times \mathcal{C}$ is a binary relation that links a constraint with another constraint.

A situation defines an abstract state of affairs that represents a particular scenario of interest and involves observations linked through spatio-temporal relationships, resources and processes. Let us note that situations are abstract, meaning that there may be different instances of a given situation. Instances of the same situation can happen in different periods of time, but they all satisfy the same constraints.

The set $\mathcal{C}$ contains all the constraints that are associated with one or more situations in the set of situations $\mathcal{S}$. These constraints concern different properties of the processes, machines, resources or even the environment in which the tasks are executed. For example, if we consider the variables `MC1_Temp` and `MC2_Temp`, corresponding to the temperature of a component of a machine and the temperature of another component of the same machine, then `MC1_Temp` $< 40°C$ and `MC1_Temp`$>$`MC2_Temp` are constraints defined on them.

The binary relation $\mathcal{R}$ is used to establish that a constraint is involved in a situation. We write $s_1 \mathcal{R} c_1$ to indicate that the constraint c_1 is involved in the situation s_1. The $\mathcal{R}$ relation is therefore built from the relationships between the situations and the constraints that are extracted from expert knowledge.

For example, consider the following set of constraints and set of situations, with six constraints and six situations:

$$C = \{c_1, c_2, c_3, c_4, c_5, c_6\} \text{ and } S = \{s_1, s_2, s_3, s_4, s_5, s_6\}.$$

Situations	Constraints	Constraints ($\mathcal{T}$)
s_1	c_1, c_3, c_6	c_1, $\mathbf{c_2}$, c_3, $\mathbf{c_4}$, $\mathbf{c_5}$, c_6
s_2	c_1, c_4, c_6	c_1, $\mathbf{c_2}$, c_4, $\mathbf{c_5}$, c_6
s_3	c_2, c_4, c_6	c_2, c_4, $\mathbf{c_5}$, c_6
s_4	c_2, c_5	c_2, c_5
s_5	c_3, c_6	c_3, $\mathbf{c_4}$, $\mathbf{c_5}$, c_6
s_6	c_1, c_6	c_1, $\mathbf{c_2}$, c_6

Table 1: Situations and the constraints concerned by them (in bold face the constraints that are implied by other constraints)

The corresponding $\mathcal{R}$ relation, built from the set of situations $\mathcal{S}$ and the set of constraints $\mathcal{C}$, is shown in Table 1 (columns 1 and 2). For example, the 4th line in this table indicates that if constraints c_2 and c_5 are satisfied (2nd column) then situation s_4 happens (1st column).

Another way to show relation $\mathcal{R}$ is depicted in Figure 1. In this figure, we can observe how different situations share a part of the constraints in their definition, for example s_1 and s_2 share constraints c_1 and c_6.

Some constraints can be more general than others, *i.e.* include others. Such is the case with, for example, the constraints c_1 and c_2 defined as $\texttt{MC1\_Temp} < 40°C$ and $\texttt{MC1\_Temp} < 60°C$, respectively. If c_1 is satisfied, then c_2 is necessarily also satisfied. Furthermore, some constraints may imply other constraints due to physical properties extracted from expert knowledge or observations. For this reason, the $\mathcal{T}$ relation is defined to indicate that if a constraint is satisfied, then another one is also satisfied. We write $c_1 \mathcal{T} c_2$.

For example, let us consider the implications among the constraints from the set of constraints $\mathcal{C}$, presented before, shown in Figure 1 (arrows labeled with $\mathcal{T}$). These implications are inferred either from mathematical properties, or from observations, or extracted from expert knowledge. Taking into account these relations, the constraints associated with each situation are established as shown in Table 1 (third column). This allows to associate more constraints than those that are explicitly concerned by the situations.

In order to formalize the relations between situations and constraints, two operators are defined below, based on the use of both $\mathcal{R}$ and $\mathcal{T}$ relations. These operators are defined on a set of situations or constraints because each situation can involve several constraints, and several situations can have several constraints in common.

The first operator $\lceil \mathcal{X} \rceil$ enables the retrieval of the set of constraints associated to a set of situations.

Definition 1 *For a situation set $\mathcal{X}$, $\mathcal{X} \subseteq \mathcal{S}$, let*

$$\lceil \mathcal{X} \rceil := \{c \in \mathcal{C} | \forall x \in \mathcal{X} : x\mathcal{R}c \vee \exists c' \in \mathcal{C} : x\mathcal{R}c' \wedge c'\mathcal{T}c\}$$

Considering the situations s_1 and s_2 of the example presented above, the constraints involved in both situations are $\lceil \{s_1, s_2\} \rceil = \{c_1, c_2, c_4, c_5, c_6\}$.

The second operator $\lfloor \mathcal{Y} \rfloor$ conversely enables the retrieval of the set of situations involving a set of constraints $\mathcal{Y}$.

Definition 2 *For a constraint set $\mathcal{Y}$, $\mathcal{Y} \subseteq \mathcal{C}$, let*

$$\lfloor \mathcal{Y} \rfloor := \{s \in \mathcal{S} | \forall y \in \mathcal{Y} : s\mathcal{R}y \vee \exists c' \in \mathcal{C} : s\mathcal{R}c' \wedge c'\mathcal{T}c\}$$

This operator retrieves the situations involving at least all the constraints in the set $\mathcal{Y}$. Therefore, considering the example presented above, if the constraints c_2 and c_5 are chosen then the situations involving those constraints are $\lfloor \{c_2, c_5\} \rfloor = \{s_1, s_2, s_3, s_4\}$. Let us note that these situations may involve other constraints, *e.g.* s_3 with c_4 and c_6.

Using the elements of the structure $\langle \mathcal{S}, \mathcal{C}, \mathcal{R}, \mathcal{T} \rangle$ and the two operators $\lceil . \rceil$ and $\lfloor . \rfloor$ previously defined, the construction of the lattice representing the hierarchy of situations is detailed below.

3.2 Building the lattice

First the nodes of the lattice are described. As highlighted above, situations have common constraints, nested in different ways: inclusion, non-null intersections, etc. Therefore, we propose to group them considering the constraints they share through the operators defined in the previous section.

Algorithm 1 Calculate all the pairs $(\mathcal{X}, \mathcal{Y})$ where $\mathcal{X} \subseteq \mathcal{S}$, $\mathcal{Y} \subseteq \mathcal{C}$, $\lceil \mathcal{X} \rceil = \mathcal{Y}$ and $\lfloor \mathcal{Y} \rfloor = \mathcal{X}$ $(setofPairs)$

```
Require: a set of Situations $\mathcal{S}$ and a set of Constraints $\mathcal{C}$
Ensure: $\{(\mathcal{X},\mathcal{Y}) | \mathcal{X} \subseteq \mathcal{S} \wedge \mathcal{Y} \subseteq \mathcal{C} \wedge \lceil \mathcal{X} \rceil = \mathcal{Y} \wedge \lfloor \mathcal{Y} \rfloor = \mathcal{X}\}$
 1: $consSet \leftarrow \{\}$ $//consSet\ is\ a\ set\ of\ constraint\ sets$
 2: $setofPairs \leftarrow \{\}$
 3: for all $s \in \mathcal{S}$ do
 4:     $consSet \leftarrow consSet \cup \{\lceil \{s\} \rceil\}$
 5: end for
 6: for all $\mathcal{O}_1 \in consSet$ do
 7:     for all $\mathcal{O}_2 \in consSet$ do
 8:         if $\mathcal{O}_1 \cap \mathcal{O}_2 \notin consSet$ then
 9:             $constSet \leftarrow consSet \cup \{\mathcal{O}_1 \cap \mathcal{O}_2\}$
10:         end if
11:     end for
12: end for
13: if $C \notin consSet$ then
14:     $consSet \leftarrow consSet \cup \{\mathcal{C}\}$
15: end if
16: if $\{\} \notin consSet$ then
17:     $consSet \leftarrow consSet \cup \{\{\}\}$
18: end if
19: for all $\mathcal{O} \in consSet$ do
20:     $setofPairs \leftarrow setofPairs \cup \{(\lfloor \mathcal{O} \rfloor, \mathcal{O})\}$
21: end for
```

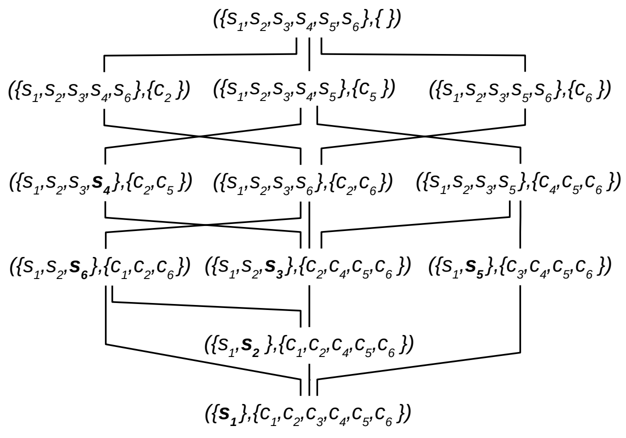

Fig. 2: Lattice representing the hierarchy of situations (situations having exactly all the constraints of the second component of the pair in the node are shown in bold face)

In our approach, we group situations and their constraints as pairs $(\mathcal{X}, \mathcal{Y})$ where $\mathcal{X}$ is a set of situations and $\mathcal{Y}$ is a set of constraints such that $\lceil \mathcal{X} \rceil = \mathcal{Y}$ and $\lfloor \mathcal{Y} \rfloor = \mathcal{X}$. The first component of the pair is a set with all the situations that share all the constraints of the second component. The second component is the set with all the constraints in common among the situations from the first component.

In order to find all the pairs and thus the nodes of the lattice, given a set of situations $\mathcal{S}$, a set of constraints $\mathcal{C}$, and relations $\mathcal{R}$ and $\mathcal{T}$, Algorithm 1 is followed. Firstly, $\lceil \{s\} \rceil$ is computed for each situation $s \in \mathcal{S}$ (lines 3-5). Then, for any two sets in this set of sets ($consSet$), their intersection is calculated. If this intersection is not yet contained in the set, it is added (lines 6-12). This step is repeated until no new sets are generated. If the set $\mathcal{C}$ of all the constraints and the empty set ($\{\}$) are not in $consSet$, they are added (lines 13-18). These sets yield the minimum and maximum nodes of the lattice, respectively. Finally, for every set $\mathcal{O}$ in $consSet$, $\lfloor \mathcal{O} \rfloor$ is computed (lines 19-21). By the end, the set ($setofPairs$) of all the possible $(\mathcal{X}, \mathcal{Y})$ pairs that satisfy $\lceil \mathcal{X} \rceil = \mathcal{Y}$ and $\lfloor \mathcal{Y} \rfloor = \mathcal{X}$ are obtained.

Considering the example in Figure 1 and Table 1, if the set of situations $\mathcal{S}$ and the set of constraints $\mathcal{C}$ are given as input to the algorithm, then the output are the nodes of the lattice shown in Figure 2. Let us note that since the situations are predefined, the search for all pairs is done offline. If a situation is added, it is either added to a node or a new node is created.

Once all the pairs are found, the next step is to order those pairs in a lattice to build the hierarchy of situations. A **lattice** is an algebraic structure that consists of a partially ordered set in which every two elements have a unique supremum (also called a least upper bound or join) and a unique infimum (also called a greatest lower bound or meet).

A **partial order** is a pair $(\mathcal{P}, \preceq)$ where $\mathcal{P}$ is a set and $\preceq$ is binary relation over $\mathcal{P}$ so that it is reflexive, anti-symmetric and transitive.

For our lattice, we consider the set:

$$\mathcal{L} = \{(\mathcal{X}, \mathcal{Y}) | \mathcal{X} \subseteq \mathcal{S} \wedge \mathcal{Y} \subseteq \mathcal{C} \wedge \lceil \mathcal{X} \rceil = \mathcal{Y} \wedge \lfloor \mathcal{Y} \rfloor = \mathcal{X}\}$$

and the following binary relation defined over it:

Definition 3 *Let $(\mathcal{X}, \mathcal{Y})$ and $(\mathcal{X}', \mathcal{Y}')$ be two pairs where $\mathcal{X}$, $\mathcal{X}'$ are sets of situations and $\mathcal{Y}$, $\mathcal{Y}'$ are sets of constraints. The situations in $\mathcal{X}'$ are reachable from $\mathcal{X}$ if the constraints in $\mathcal{Y} \cap \mathcal{Y}'$ are satisfied, noted as $(\mathcal{X}, \mathcal{Y}) \preceq (\mathcal{X}', \mathcal{Y}') \Leftrightarrow \mathcal{X} \subseteq \mathcal{X}' \wedge \mathcal{Y}' \subseteq \mathcal{Y}$.*

3.3 Lattice proof

Theorem 1. *$(\mathcal{L}, \preceq)$ is a lattice.*

Proof. The proof is done in two parts. In the first part it is proved that $(\mathcal{L}, \preceq)$ is a partially ordered set. In the second part, it is proved that for any two elements of the lattice they have a unique supremum and a unique infimum.

Part 1. First it is proven that $\preceq$ is reflexive, i.e. $\forall (\mathcal{X}, \mathcal{Y}) \in \mathcal{L} \mid (\mathcal{X}, \mathcal{Y}) \preceq (\mathcal{X}, \mathcal{Y})$.
By the definition of $\preceq$, $(\mathcal{X}, \mathcal{Y}) \preceq (\mathcal{X}, \mathcal{Y}) \Leftrightarrow \mathcal{X} \subseteq \mathcal{X} \wedge \mathcal{Y} \subseteq \mathcal{Y}$ It is true since each set is self-contained. Then, $\preceq$ is reflexive.

Now we prove that $\preceq$ is anti-symmetric, i.e. $\forall (\mathcal{X}, \mathcal{Y}), (\mathcal{X}', \mathcal{Y}') \in \mathcal{L} | (\mathcal{X}, \mathcal{Y}) \preceq (\mathcal{X}', \mathcal{Y}') \wedge (\mathcal{X}', \mathcal{Y}') \preceq (\mathcal{X}, \mathcal{Y}) \Rightarrow (\mathcal{X}, \mathcal{Y}) = (\mathcal{X}', \mathcal{Y}')$.
Let two sets $(\mathcal{X}, \mathcal{Y})$, $(\mathcal{X}', \mathcal{Y}')$ belong to $\mathcal{L}$, and $(\mathcal{X}, \mathcal{Y}) \preceq (\mathcal{X}', \mathcal{Y}')$ and $(\mathcal{X}', \mathcal{Y}') \preceq (\mathcal{X}, \mathcal{Y})$. Applying the definition of $\preceq$ on both we obtain: $\mathcal{X} \subseteq \mathcal{X}' \wedge \mathcal{Y}' \subseteq \mathcal{Y}$ and $\mathcal{X}' \subseteq \mathcal{X} \wedge \mathcal{Y} \subseteq \mathcal{Y}'$, respectively. Thus, on the one hand $\mathcal{X} \subseteq \mathcal{X}'$ and $\mathcal{X}' \subseteq \mathcal{X}$ means that $\mathcal{X} = \mathcal{X}'$, and on the other hand $\mathcal{Y}' \subseteq \mathcal{Y}$ and $\mathcal{Y} \subseteq \mathcal{Y}'$ means that $\mathcal{Y} = \mathcal{Y}'$. Since $(\mathcal{X}, \mathcal{Y}) = (\mathcal{X}', \mathcal{Y}')$, $\preceq$ is anti symmetric.

Finally, we show that $\preceq$ is transitive, i.e. $(\mathcal{X}, \mathcal{Y}) \preceq (\mathcal{X}', \mathcal{Y}') \wedge (\mathcal{X}', \mathcal{Y}') \preceq (\mathcal{X}'', \mathcal{Y}'') \Rightarrow (\mathcal{X}, \mathcal{Y}) \preceq (\mathcal{X}'', \mathcal{Y}'')$.
Let $(\mathcal{X}, \mathcal{Y})$, $(\mathcal{X}', \mathcal{Y}'), (\mathcal{X}'', \mathcal{Y}'')$ belong to $\mathcal{L}$, and consider that $(\mathcal{X}, \mathcal{Y}) \preceq (\mathcal{X}', \mathcal{Y}')$ and $(\mathcal{X}', \mathcal{Y}') \preceq (\mathcal{X}'', \mathcal{Y}'')$. Now, if we apply the definition of $\preceq$ on both we get, $\mathcal{X} \subseteq \mathcal{X}' \wedge \mathcal{Y}' \subseteq \mathcal{Y}$ and $\mathcal{X}' \subseteq \mathcal{X}'' \wedge \mathcal{Y}'' \subseteq \mathcal{Y}'$, respectively. Considering sets $\mathcal{X}$, $\mathcal{X}'$, and $\mathcal{X}''$ $\mathcal{X} \subseteq \mathcal{X}'$ and $\mathcal{X}' \subseteq \mathcal{X}''$ means that $\mathcal{X} \subseteq \mathcal{X}''$ because $\subseteq$ is transitive. Similarly, $\mathcal{Y}'' \subseteq \mathcal{Y}'$ and $\mathcal{Y}' \subseteq \mathcal{Y}$ means that $\mathcal{Y}'' \subseteq \mathcal{Y}$. Then, $(\mathcal{X}, \mathcal{Y}) \preceq (\mathcal{X}'', \mathcal{Y}'')$. Therefore, $\preceq$ is transitive.

Part 2. Firstly, two lemmas are introduced:

Lemma 1. *Let $\mathcal{Y}$ be a set of constraints and $\mathcal{X}$ a set of situations, then*

$$\mathcal{Y} \subseteq \lceil \lfloor \mathcal{Y} \rfloor \rceil \text{ and } \mathcal{X} \subseteq \lfloor \lceil \mathcal{X} \rceil \rfloor.$$

Proof. $\lfloor \mathcal{Y} \rfloor$ has all the situations involving the constraints in the set $\mathcal{Y}$ and the situations involving constraints that are included by the constraints in the set $\mathcal{Y}$. Then, $\lceil \lfloor \mathcal{Y} \rfloor \rceil$ has all the constraints associated with the set of situations $\lfloor \mathcal{Y} \rfloor$. Therefore, the set of constraints $\mathcal{Y}$ is contained in $\lceil \lfloor \mathcal{Y} \rfloor \rceil$. The reasoning is the same for the set of situations$\mathcal{X}$.

Lemma 2. *Let $\mathcal{Y}, \mathcal{Y}'$ be sets of constraints and $\mathcal{X}, \mathcal{X}'$ be sets of situations, then*

$$\mathcal{Y} \subseteq \mathcal{Y}' \Rightarrow \lfloor \mathcal{Y}' \rfloor \subseteq \lfloor \mathcal{Y} \rfloor \text{ and } \mathcal{X} \subseteq \mathcal{X}' \Rightarrow \lceil \mathcal{X}' \rceil \subseteq \lceil \mathcal{X} \rceil.$$

Proof. We prove that $\lceil \mathcal{X}' \rceil \subseteq \lceil \mathcal{X} \rceil$. By definition of $\lceil . \rceil$ on $\mathcal{X}'$, $\lceil \mathcal{X}' \rceil = \{c \in \mathcal{C} | \forall x \in \mathcal{X}' : x\mathcal{R}c \vee \exists c' \in \mathcal{C} : x\mathcal{R}c' \wedge c'\mathcal{T}c\}$ This is for all $b \in \mathcal{X}'$, and by hypothesis $\mathcal{X} \subseteq \mathcal{X}'$. Therefore,

$$\begin{aligned} \lceil \mathcal{X}' \rceil &= \{c \in \mathcal{C} | \forall x \in \mathcal{X}' : x\mathcal{R}c \vee \exists c' \in \mathcal{C} : x\mathcal{R}c' \wedge c'\mathcal{T}c\} \\ &\subseteq \{c \in \mathcal{C} | \forall x \in \mathcal{X} : x\mathcal{R}c \vee \exists c' \in \mathcal{C} : x\mathcal{R}c' \wedge c'\mathcal{T}c\} = \lceil \mathcal{X} \rceil \end{aligned}$$

The same reasoning can be applied to prove $\mathcal{Y} \subseteq \mathcal{Y}' \Rightarrow \lfloor \mathcal{Y}' \rfloor \subseteq \lfloor \mathcal{Y} \rfloor$. This lemma shows that the more constraints are required, the fewer situations involve all of them. Conversely, the more situations we consider, the fewer constraints they have in common.

Having defined the two lemmas and their respective proofs, we continue with the general proof of the theorem.

For any two pairs $(\mathcal{X}, \mathcal{Y})$ and $(\mathcal{X}', \mathcal{Y}')$ we obtain:

- the infimum (greatest common pair) of $(\mathcal{X}, \mathcal{Y})$ and $(\mathcal{X}', \mathcal{Y}')$ as
$$(\mathcal{X}, \mathcal{Y}) \wedge (\mathcal{X}', \mathcal{Y}') := (\mathcal{X} \cap \mathcal{X}', \lceil \lfloor \mathcal{Y} \cup \mathcal{Y}' \rfloor \rceil)$$
- the supremum (least common pair) of $(\mathcal{X}, \mathcal{Y})$ and $(\mathcal{X}', \mathcal{Y}')$ as
$$(\mathcal{X}, \mathcal{Y}) \vee (\mathcal{X}', \mathcal{Y}') := (\lfloor \lceil \mathcal{X} \cup \mathcal{X}' \rceil \rfloor, \mathcal{Y} \cap \mathcal{Y}')$$

To prove the existence of the unique infimum, we have to show that $(\mathcal{X}, \mathcal{Y}) \wedge (\mathcal{X}', \mathcal{Y}')$ is smaller than both $(\mathcal{X}, \mathcal{Y})$ and $(\mathcal{X}', \mathcal{Y}')$, and any other common child of $(\mathcal{X}, \mathcal{Y})$ and $(\mathcal{X}', \mathcal{Y}')$ is also a child of $(\mathcal{X}, \mathcal{Y}) \wedge (\mathcal{X}', \mathcal{Y}')$.
First we prove that $(\mathcal{X} \cap \mathcal{X}', \lceil \lfloor \mathcal{Y} \cup \mathcal{Y}' \rfloor \rceil) \preceq (\mathcal{X}, \mathcal{Y})$. By the definition of $\preceq$, we have $\mathcal{X} \cap \mathcal{X}' \subseteq \mathcal{X}$ and $\mathcal{Y} \subseteq \lceil \lfloor \mathcal{Y} \cup \mathcal{Y}' \rfloor \rceil$.
The proof of $\mathcal{X} \cap \mathcal{X}' \subseteq \mathcal{X}$ is trivial. For the other part of the conjunction we start from $\mathcal{Y} \subseteq \mathcal{Y} \cup \mathcal{Y}'$ applying Lemma 2 twice we obtain $\lceil \lfloor \mathcal{Y} \rfloor \rceil \subseteq \lceil \lfloor \mathcal{Y} \cup \mathcal{Y}' \rfloor \rceil$. In addition, Lemma 1 expresses that $\mathcal{Y} \subseteq \lceil \lfloor \mathcal{Y} \rfloor \rceil$. Thus, combining the last two statements we conclude that $\mathcal{Y} \subseteq \lceil \lfloor \mathcal{Y} \rfloor \rceil \subseteq \lceil \lfloor \mathcal{Y} \cup \mathcal{Y}' \rfloor \rceil$. Therefore, $\mathcal{Y} \subseteq \lceil \lfloor \mathcal{Y} \cup \mathcal{Y}' \rfloor \rceil$.

The same method can be used to prove that $(\mathcal{X} \cap \mathcal{X}', \lceil \lfloor \mathcal{Y} \cup \mathcal{Y}' \rfloor \rceil) \preceq (\mathcal{X}', \mathcal{Y}')$.

Similarly, $(\mathcal{X}, \mathcal{Y}) \vee (\mathcal{X}', \mathcal{Y}')$ is greater than both $(\mathcal{X}, \mathcal{Y})$ and $(\mathcal{X}', \mathcal{Y}')$, and it is a child of any common parent of these two pairs. Only the proof for the infimum is developed here since the proof for the supremum is similar.

4 Lattice Interpretation and Exploitation

An illustrative case study is described below and is used to highlight how the lattice can be used and the advantages it offers to support the decisions that need to be made when an abnormal situation is detected in an industrial framework.

The case study is based on a manufacturing production line composed of several machines. These machines are equipped with sensors on different components. The

Set of constraints $\mathcal{C}$

ID	Properties	Restriction	Device	ID	Properties	Restriction	Device
c_1	`Oil temp.`	$> 40°\text{C}$	M_1	c_{13}	`Power output`	< 500 kW	PL_1
c_2	`Oil temp.`	$> 60°\text{C}$	M_1	c_{14}	`Power output`	< 200 kW	PL_1
c_3	`Transformer temp.`	$> 45°\text{C}$	M_1T_1	c_{15}	`Conv. water temp.`	$> 60°\text{C}$	M_3Cv_1
c_4	`Controller temp.`	$> 40°\text{C}$	M_1Ct_1	c_{16}	`Conv. water temp.`	$> 80°\text{C}$	M_3Cv_1
c_5	`Generator curr.`	< 800 A	M_1G_1	c_{17}	`Trans. grid temp.`	$< 35°\text{C}$	M_3T_1
c_6	`Platform temp.`	$< 35°\text{C}$	PL_1	c_{18}	`Generator temp.`	$> 45°\text{C}$	M_3G_1
c_7	`Platform temp.`	$> 40°\text{C}$	PL_1	c_{19}	`Converter temp.`	$> 60°\text{C}$	M_3Cv_1
c_8	`Gearbox temp.`	$> 40°\text{C}$	M_2GB_1	c_{20}	`Converter temp.`	$> 80°\text{C}$	M_3Cv_1
c_9	`Gearbox temp.`	$> 60°\text{C}$	M_2GB_1	c_{21}	`Rotor speed`	< 200 rpm	M_4R_1
c_{10}	`Generator speed`	< 500 rpm	M_2G_1	c_{22}	`Rotor speed`	< 100 rpm	M_4R_1
c_{11}	`Environment temp.`	$< 25°\text{C}$	PL_1	c_{23}	`Rotor Pitch angle`	$< 5°$	M_4R_1
c_{12}	`Power output`	> 2000 kW	PL_1				

Table 2: Constraints definition.

sensors collect data on the properties described in Table 2. This scenario is formally represented using the manufacturing domain ontology introduced in [5]. It enables to represent a production line: the machines that compose it, the tasks they perform and the observations made by the sensors as well as the context in which the observations are measured.

Several abnormal situations that could lead to failures in this scenario are defined from expert knowledge. Each of them is expressed as a set of constraints. For reasons of space, situations are not described in detail, but the situations represented here cover the following types of failures: hydraulic oil leakage, cooling system filter obstructions, converter and rotor malfunctions and global malfunctions of the production line. The defined abnormal situations are shown in Table 3 with a brief description of what they represent; and the constraints concerned by them are described in Table 2.

Some situations represent conditions that can lead to the same potential failure, indicating different levels of severity. For example, situations s_1 and s_2 both represent situations that could lead to the same failure but s_2 indicates a higher severity since the temperature threshold is higher than the temperature threshold for s_1. In this case, the actions to be taken may be more decisive or follow a safety protocol, since the situations are of higher severity.

The resulting lattice from the scenario described above is shown in Figure 3. By observing the hierarchy of situations, it is possible to notice that in the upper part of the lattice the constraints which are involved in most of the situations are verified. The further we go down in the lattice, the more the situations are specific as they include more constraints. The stronger constraints are close to the bottom of the diagram, meaning that these situations embed those situated higher up in the lattice.

The lattice provides a structure that represents the order in which the situations may arise according to which constraints are verified. Considering that certain actions can modify the value of a property and thus change the state of the system, either by satisfying another constraint or on the contrary by not satisfying a constraint anymore,

Set of situations $\mathcal{S}$		
Situation	Constraints ($\mathcal{T}$)	Description
s_1	c_1,c_3,c_4,c_5,c_6	$\mathtt{M}_1$ oil leakage
s_2	$\mathbf{c_1}$,c_2,c_3,c_4,c_5,c_6	$\mathtt{M}_1$ oil leakage
s_3	c_6,c_8,c_{10},c_{11}	Increase $\mathtt{M}_2$ oil temp.
s_4	c_6,$\mathbf{c_8}$,c_9,c_{10},c_{11}	Increase $\mathtt{M}_2$ oil temp.
s_5	c_7,$\mathbf{c_8}$,c_9,c_{10},c_{11}	Increase $\mathtt{M}_2$ oil temp.
s_6	c_{15},c_{17},c_{18}	$\mathtt{M}_3$ filter obstruction
s_7	$\mathbf{c_{15}}$,c_{16},c_{17},c_{18}	$\mathtt{M}_3$ filter obstruction
s_8	c_6,c_{17},c_{19}	$\mathtt{M}_3\mathtt{Cv}_1$ malfunction
s_9	c_6,c_{17},$\mathbf{c_{19}}$,c_{20}	$\mathtt{M}_3\mathtt{Cv}_1$ malfunction
s_{10}	$\mathbf{c_{15}}$,c_{16},c_{17},c_{18},c_{19}	$\mathtt{M}_3\mathtt{Cv}_1$ malfunction
s_{11}	c_{12},c_{21},c_{23}	$\mathtt{M}_4$R$_1$ malfunction
s_{12}	c_{12},$\mathbf{c_{21}}$,c_{22},c_{23}	$\mathtt{M}_4$R$_1$ malfunction
s_{13}	c_{13},c_{21},c_{23}	PL global malfunction
s_{14}	$\mathbf{c_{13}}$,c_{14},c_{21},c_{23}	PL global malfunction

Table 3: Situations and their concerned constraints (the constraints that are implied by other constraints are in bold face)

the lattice allows the analysis of the actions to take. It implies, from the decision support point of view, reaching a node situated lower in the lattice if new constraints are satisfied, or higher in the other case.

For example, if the constraints c_{15}, c_{17} and c_{18} are satisfied then it means that the situation s_6 is happening, *i.e.* there is a cooling filter obstruction in the machine $\mathtt{M}_3$. The lattice, and in particular the node $(\{s_6, s_7, s_{10}\}, \{c_{15}, c_{17}, c_{18}\})$; shows that situation s_7, a more critical situation than s_6, is reachable if no action is taken or if the actions taken imply the satisfaction of c_{16}. In general, the actions aim at correcting the abnormal property values causing them to return to normal values. In this case, a *filter change* action in the cooling system would make the `Converter water temp.` and `Generator temp.` properties decrease to values lower than their respective thresholds, *i.e.* the constraints c_{15} and c_{18} would no longer be satisfied. This would lead to the process at the node $(\{s_6, s_7, s_8, s_9, s_{10}\}, \{c_{17}\})$ where only the constraint c_{17} is satisfied and other situations can be reached from there. Ideally, it is always intended to go up on the lattice to the node $(\mathcal{S}, \{\})$ where none of the constraints are satisfied, meaning that no abnormal situation is (partially) present.

A particular node in the lattice represents a possible state of the system. It should be noted that the minimum of the lattice, the node $(\{\}, \mathcal{C})$, may represent an unreachable state. That is, since this node includes all the constraints ($\mathcal{C}$), it may be that two constraints cannot be satisfied at the same time because they are exclusive, such as the constraints c_6 and c_7. However, this node is necessary for the hierarchy to be a lattice. If this happens in another node of the lattice that it is not the minimum, then there is a problem in the definition of the $\mathcal{R}$ relation or of the $\mathcal{T}$ relation: it would be a situation that can never be satisfied because it concerns constraints that are mutually exclusive.

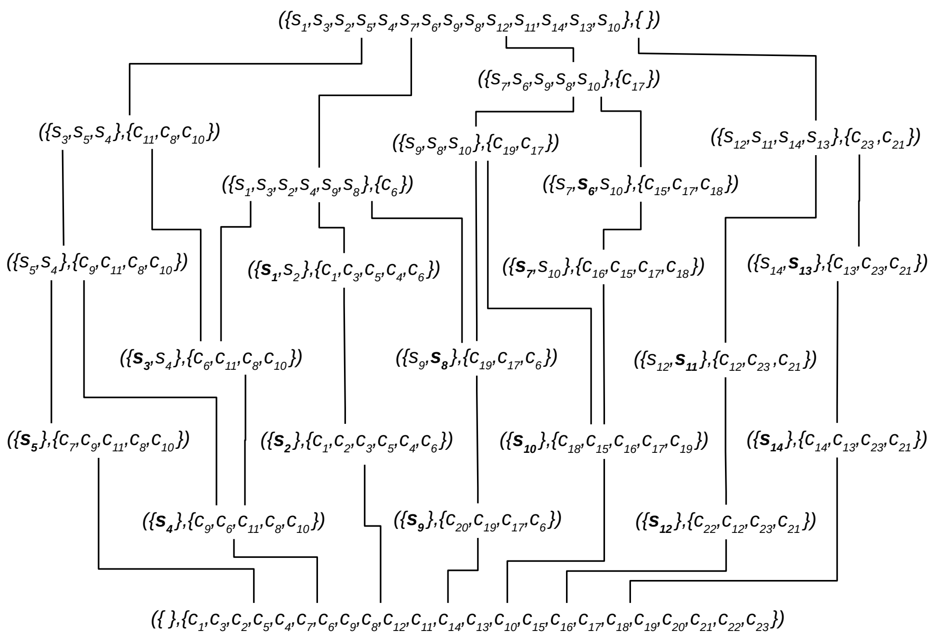
({s1,s3,s2,s5,s4,s7,s6,s9,s8,s12,s11,s14,s13,s10},{ })
({s7,s6,s9,s8,s10},{c17})
({s3,s5,s4},{c11,c8,c10})
({s9,s8,s10},{c19,c17})
({s12,s11,s14,s13},{c23,c21})
({s1,s3,s2,s4,s9,s8},{c6})
({s7,s6,s10},{c15,c17,c18})
({s5,s4},{c9,c11,c8,c10})
({s1,s2},{c1,c3,c5,c4,c6})
({s7,s10},{c16,c15,c17,c18})
({s14,s13},{c13,c23,c21})
({s3,s4},{c6,c11,c8,c10})
({s9,s8},{c19,c17,c6})
({s12,s11},{c12,c23,c21})
({s5},{c7,c9,c11,c8,c10})
({s2},{c1,c2,c3,c5,c4,c6})
({s10},{c18,c15,c16,c17,c19})
({s14},{c14,c13,c23,c21})
({s4},{c9,c6,c11,c8,c10})
({s9},{c20,c19,c17,c6})
({s12},{c22,c12,c23,c21})
({ },{c1,c3,c2,c5,c4,c7,c6,c9,c8,c12,c11,c14,c13,c10,c15,c16,c17,c18,c19,c20,c21,c22,c23})

Fig. 3: Hierarchy of the situations defined in the illustrative case study.

For each node defined as $(\mathcal{X}, \mathcal{Y})$, all situations in $\mathcal{X}$ are at least partially occurring, *i.e.* a part of their constraints is satisfied. When one of the situations in $\mathcal{X}$ involves exactly all the constraints in $\mathcal{Y}$ and no other, this means that this situation is occurring. This is formally defined in Definition 4 (these situations appear in bold face in Figures 2 and 3).

Definition 4 *For a pair* $(\mathcal{X}, \mathcal{Y})$, $\mathcal{X} \subseteq \mathcal{S}$ *and* $\mathcal{Y} \subseteq \mathcal{C}$, *let*

$$\|(\mathcal{X}, \mathcal{Y})\| := \{s \in \mathcal{X} | \forall y \in \mathcal{Y} : s\mathcal{R}y \wedge \nexists c' \in \mathcal{C}\text{-}\mathcal{Y} : s\mathcal{R}c'\}$$

For example, $\|(\{s_6, s_7, s_{10}\}, \{c_{15}, c_{17}, c_{18}\})\| = \{s_6\}$ means that the situation s_6 happens in the node $(\{\mathbf{s_6}, s_7, s_{10}\}, \{c_{15}, c_{17}, c_{18}\})$ because it involves only the constraints c_{15}, c_{17} and c_{18}.

Regarding the nodes where $\|(\mathcal{X}, \mathcal{Y})\|$ is empty, this means that situations in $\mathcal{X}$ are partially occurring or that they are potential situations that the system could reach. The discovery of these combinations of constraints in common among certain situations can give rise to the definition of new situations that allow the early stage detection of certain *"relevant"* situations. This would allow preventive decisions to be taken.

5 Concluding remarks

This paper presents an approach that uses the lattice theory to represent a hierarchical order among certain situations that lead to potential failures. This lattice is automatically built from existing knowledge and represents a road-map of all the situations that can be reached from a given one, desirable or not. This allows the identification of the actions that can be taken to correct the abnormality.

Although the case study presented in the paper comes from the Industry 4.0 context, the proposed approach can be applied to other application domains, where real time monitoring is needed. In fact, in any monitoring application, the notion of situation (normal or abnormal) exists. Their identification thanks to expert knowledge enables to automatically build the hierarchy among the identified situations using our approach.

It is also to be remarked that in the case study presented, the $\mathcal{T}$ relation represents a hierarchy between the constraints in the sense that one constraint may be more specific than another, so that when it is satisfied, the more general constraint is also satisfied. The $\mathcal{T}$ relation can also represent another type of relationship among the constraints such as the fact that two different properties have a physical link, meaning that the values of one directly affect the values of the other.

The following research lines will be addressed in future works. Firstly, a more extensive case with more complex situations, involving more constraints, will be explored. This raises scalability and complexity issues to build the lattice. Therefore, tests will be performed with different variants of Algorithm 1 to evaluate their relative efficiency. Secondly, a more exhaustive and detailed study will be made on the constraints implications that represent dependencies among constraints for the identification of new situations or the refinement of existing ones.

Acknowledgements

These PhD works are funded by the Normandy Region (France) in the framework of the STEaMINg (Semantic Time Evolving Models for Industry 4.0) project.

References

1. Brenna, L., Demers, A., Gehrke, J., Hong, M., Ossher, J., Panda, B., Riedewald, M., Thatte, M., White, W.: Cayuga: A high-performance event processing engine (01 2007). https://doi.org/10.1145/1247480.1247620
2. Cugola, G., Margara, A.: Tesla: A formally defined event specification language. In: Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems. p. 50–61. DEBS '10, Association for Computing Machinery, New York, NY, USA (2010). https://doi.org/10.1145/1827418.1827427, https://doi.org/10.1145/1827418.1827427
3. Cugola, G., Margara, A.: Processing flows of information: From data stream to complex event processing. ACM Comput. Surv. **44**(3) (Jun 2012). https://doi.org/10.1145/2187671.2187677, https://doi.org/10.1145/2187671.2187677
4. Davey, B. A. Priestley, H.A.: Introduction to Lattices and Order. Cambridge University Press, 2 edn. (2002). https://doi.org/10.1017/CBO9780511809088
5. Giustozzi, F., Saunier, J., Zanni-Merk, C.: Context Modeling for Industry 4.0: an Ontology-Based Proposal. Procedia Computer Science **126**, 675 – 684 (2018)
6. Giustozzi, F., Saunier, J., Zanni-Merk, C.: Abnormal situations interpretation in industry 4.0 using stream reasoning. Procedia Computer Science **159**, 620 – 629 (2019)

7. Grätzer, G.: Lattice Theory: Foundation (01 2011). https://doi.org/10.1007/978-3-0348-0018-1
8. Nation, J.B.: Notes on lattice theory (1998)
9. Reis, M.S., Gins, G.: Industrial process monitoring in the big data/industry 4.0 era: From detection, to diagnosis, to prognosis. Processes **5**(3), 35 (2017)
10. Schlechtingen, M., Santos, I.F.: Wind turbine condition monitoring based on scada data using normal behavior models. part 2: Application examples. Applied Soft Computing **14**, 447 – 460 (2014). https://doi.org/https://doi.org/10.1016/j.asoc.2013.09.016, http://www.sciencedirect.com/science/article/pii/S1568494613003104
11. Schlechtingen, M., Santos, I.F., Achiche, S.: Wind turbine condition monitoring based on scada data using normal behavior models. part 1: System description. Applied Soft Computing **13**(1), 259 – 270 (2013). https://doi.org/https://doi.org/10.1016/j.asoc.2012.08.033, http://www.sciencedirect.com/science/article/pii/S1568494612003821
12. Schreiber, G., Akkermans, H., Anjewierden, A., Hoog, R., Shadbolt, N., Velde, W., Wielinga, B.: Knowledge Engineering and Management - The CommonKADS Methodology, vol. 24 (01 2001). https://doi.org/10.7551/mitpress/4073.001.0001
13. Stojanovic, L., Dinic, M., Stojanovic, N., Stojadinovic, A.: Big-data-driven anomaly detection in industry (4.0): An approach and a case study. pp. 1647–1652 (12 2016). https://doi.org/10.1109/BigData.2016.7840777
14. Tan, Y., Vuran, M.C., Goddard, S., Yu, Y., Song, M., Ren, S.: A concept lattice-based event model for cyber-physical systems. In: Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems. p. 50–60. ICCPS '10, Association for Computing Machinery, New York, NY, USA (2010). https://doi.org/10.1145/1795194.1795202, https://doi.org/10.1145/1795194.1795202
15. Tommasini, R., Bonte, P., Della Valle, E., Mannens, E., Turck, F., Ongenae, F.: Towards ontology-based event processing. pp. 115–127 (02 2017). https://doi.org/10.1007/978-3-319-54627-8_9
16. Wille, R.: Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies, pp. 1–33. Springer Berlin Heidelberg, Berlin, Heidelberg (2005). https://doi.org/10.1007/11528784_1, https://doi.org/10.1007/11528784_1
17. Wu, E., Diao, Y., Rizvi, S.: High-performance complex event processing over streams. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data. p. 407–418. SIGMOD '06, Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1142473.1142520, https://doi.org/10.1145/1142473.1142520