Condensed Representations for Contrast Sequential Pattern Mining in ASP

Gioacchino Sterlicchio[1,2,3,*], Francesca A. Lisi[2,4]

[1]Dept. of Mechanics, Mathematics and Management, Polytechnic University of Bari, Via G. Amendola 126/b - 70126 Bari, Italy
[2]Dept. of Computer Science, University of Bari "Aldo Moro", Via E. Orabona 4, Bari, 70125, Italy
[3]Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, 29 Av. John F. Kennedy, 1855 Kirchberg Luxembourg, Luxembourg
[4]Centro Interdipartimentale di Logica e Applicazioni (CILA), University of Bari "Aldo Moro", Via E. Orabona 4, Bari, 70125, Italy

Abstract
In this work, we address an extension of the contrast sequential pattern mining problem which aims at detecting condensed representations for contrast sequential patterns. The problem is encoded with Answer Set Programming (ASP). The efficiency and scalability of the ASP encoding are evaluated on two publicly available dataset, iPRG and UNIX.

Keywords
Declarative Pattern Mining, Contrast Sequential Pattern Mining, Condensed representations, Answer Set Programming

1. Introduction

The continuous increase in available data makes effective and efficient techniques necessary to extrapolate key information to be used to make decisions in various application contexts. An example is temporal data (e.g. system logs, banking transactions, telephone records, ...) that must be analyzed using suitable methodologies such as *Sequential Pattern Mining (SPM)* [1]. In SPM, the goal is to find frequent and non-empty temporal sequences (i.e. sequential patterns) from a sequence dataset. It also happens that available temporal data is labeled or grouped according to precise semantics. For example, in the domain of network security it is possible to label the network behavior as normal or as anomalous and in this last case an attack could be underway. The idea behind *Contrast Pattern Mining (CPM)* [2] is to find statistically significant differences between two or more disjoint datasets or portions of the same dataset. The possibility of merging the two previous concepts for finding significant differences between frequent sequences of different classes is known as *Contrast Sequential Pattern Mining (CSPM)* [3]. CSPM is the pattern mining task considered in this paper.

In recent years there has been an increasing interest in the so-called *Declarative Pattern Mining (DPM)*, a research stream in which the objective is to develop declarative approaches to pattern mining. Several encodings have been presented so far, to cover pattern mining tasks such as sequence mining [4, 5] and frequent itemset mining [6, 7]. Answer Set Programming (ASP) is widely used in DPM. The first proposal is described by *Guyet et al.* [8]. The authors explore the SPM problem with ASP and compare their method with a dedicated algorithm. *Gebser et al.* [5] use ASP for extracting condensed representations of sequential patterns. *Samet et al.* in [9] mine rare sequential patterns with ASP. In [10], *Guyet et al.* propose a real world application to ASP-based DPM investigating the possible association between hospitalization for seizure and antiepileptic drug switch from a French medico-administrative database. Guyet et al. [11] present the use of ASP to mine sequential patterns within two representations of embeddings (fill-gaps vs skip-gaps) and compare them with CP. An hybrid ASP approach is proposed

CILC 2024: 39th Italian Conference on Computational Logic, June 26-28, 2024, Rome, Italy
*Corresponding author.
g.sterlicchio@phd.poliba.it (G. Sterlicchio); FrancescaAlessandra.Lisi@uniba.it (F. A. Lisi)
http://www.di.uniba.it/~lisi/ (F. A. Lisi)
0000-0002-2936-0777 (G. Sterlicchio); 0000-0001-5414-5844 (F. A. Lisi)

by *Paramonov et al.* [12] which combines dedicated algorithms for pattern mining and ASP. In [13, 14] Guyet's ASP encodings for SPM are adapted in order to address the requirements of an application in the digital forensics domain.[1] Motivated by the same application, *Lisi and Sterlicchio* in [15] propose an ASP-based approach to CPM.[2] In [16], the same authors present the first ASP encoding for the CSPM problem, which we call *MASS-CSP* (*Mining with Answer Set Solving - Contrast Sequential Patterns*) hereafter. In this paper we want to carry on the work on *MASS-CSP* by facing one of the major problems in pattern mining, i.e. the huge number of patterns most of which might be not useful. To this aim we explore the so-called *condensed representations* to decrease the size of output.

The paper is organized as follows. In Section 2 we briefly recall the basics of ASP and give the necessary background on CSPM. In Section 3 we define the problem of mining condensed representations for the CSPM task and report the experimental results obtained on the a couple of datasets. Section 4 concludes the paper with final remarks.

2. Background

2.1. Answer Set Programming

Answer Set Programming (ASP) [17, 18] is a declarative programming paradigm that allows for the representation and solving of complex combinatorial problem. It is based on the logical formalism of logic programs, specifically disjunctive logic programs with answer set semantics. ASP provides a powerful tool for solving problems in various domains such as planning, scheduling, reasoning about actions, and knowledge representation. In an ASP program, rules are defined using predicates and logical connectors such as conjunction, disjunctions, and negations. The program consists of a set of rules which define relationships between different elements in the problem domain. These rules are then used to generate answer sets - sets of consistent interpretations that satisfy all constraints specified in the program. One advantage of ASP over other declarative programming paradigms is its expressiveness and flexibility in representing complex problems concisely through logical constraints. Additionally, ASP programs can be easily modified or extended without changing their overall structure due to their modular nature. ASP solvers use sophisticated algorithms based on efficient search techniques to compute answer sets, the most important are Clingo [19] and DLV [20]. An example of general rule is: $a_1 \vee \ldots \vee a_n \leftarrow b_1, \ldots, b_k, not\, b_{k+1}, \ldots, not\, b_m$. The rule says that if $b_1, \ldots, b_k$ are true and there is not reason for believing that $b_{k+1}, \ldots, b_m$ are true then at least one of the $a_1, \ldots, a_n$ is believed to be true. The left hand side and the right hand side of the $\leftarrow$ are called *head* and *body* respectively. Rules without body are called *facts*. The head is unconditionally true and the arrow is usually omitted. Conversely, rules without head are called *constraints* and are used to discard stable models, thus reducing the number of answers returned by the ASP solver.

2.2. Contrast Sequential Pattern Mining

Contrast sequential Pattern Mining (CSPM) [3] is a data mining technique that aims to discover interesting patterns in sequential data by comparing and contrasting different sequences. This approach goes beyond traditional sequential item pattern mining, which focuses solely on finding frequent patterns, by also considering the differences between sequences. The main idea behind CSPM is to identify patterns that occur frequently in one group of sequences but infrequently in another group. This allows for the detection of significant differences between two sets of sequences and can provide valuable insights into the underlying relationships or trends within data. By focusing on contrasting patterns, CSPM can uncover hidden associations or trends that may not be apparent when analyzing each group separately. This can lead to new discoveries and insights into complex datasets where traditional pattern mining techniques may fall short. Below, we will show how to arrive at the final definition of CSPM starting from SPM, finally we will show an example to understand the technique.

[1]We refer to this encoding as *MASS-SP (Mining with Answer Set Solving - Sequential Patterns).*
[2]We refer to this encoding as *MASS-CP (Mining with Answer Set Solving - Contrast Patterns).*

ID	Sequence	Class
1	$\langle a\,b\,a\,c\,d\rangle$	C_1
2	$\langle a\,b\,c\rangle$	C_1
3	$\langle c\,a\,b\,c\rangle$	C_1
4	$\langle c\rangle$	C_1
5	$\langle b\,c\,a\,a\rangle$	C_2
6	$\langle c\,b\,a\rangle$	C_2
7	$\langle c\,b\,a\rangle$	C_2
8	$\langle a\,b\,a\,c\,b\,a\rangle$	C_2

Table 1
Example of a sequence dataset. Each sequence has a class label, that is used in CSPM

Let D be a database containing a set of sequences $S = \{s_1, s_2, \ldots, s_k\}$, where each sequence s_i consists of ordered elements or items from an alphabet Σ. A sequence is represented as $s_i = \langle i_1, i_2, \ldots, i_m\rangle$, where each item i_k belongs to Σ and appears in the sequence in order according to some timestamp or position information. A sequential pattern P is defined as an ordered list of items $\pi = \langle a_1, a_2, \ldots, a_k\rangle$ such that each $a_i \in \Sigma$ and occurs consecutively in at least one sequence in S. The *support* of a sequential pattern π is the number of sequences in which it occurs. Given a minimum support threshold *minsup*, SPM aims to find all frequent sequential patterns π, such that $supp(\pi) \geq minsup$. frequent sequential patterns are those that occur frequently enough within the dataset base on the specified support threshold. A contrast sequential pattern is defined as a sequential pattern tha occurs frequently in one sequence dataset but not in the others. It is necessary to introduce the concept of *growth rate* and *contrast rate* to find contrast sequential patterns. Given two sequences dataset, D_1 labeled with the C_1 class and D_2 labeled as C_2, first we compute the *growth rate* from D_2 to D_1 of a sequential pattern π as $GR_{C_1}(\pi) = \frac{supp(\pi, D_1)/|D_1|}{supp(\pi, D_2)/|D_2|}$. If the $supp(\pi, D_2) = 0$ and $supp(\pi, D_1) \neq 0$ then $GR_{C_1}(\pi) = \infty$. After, the growth rate from D_1 to D_2 of π is defined as $GR_{C_2}(\pi) = \frac{supp(\pi, D_2)/|D_2|}{supp(\pi, D_1)/|D_1|}$. If the $supp(\pi, D_1) = 0$ and $supp(\pi, D_2) \neq 0$ then $GR_{C_2}(\pi) = \infty$. The contrast rate of π is defined as $CR(\pi) = max\{GR_{C_1}(\pi), GR_{C_2}(\pi)\}$ and if $GR_{C_1}(\pi) = 0$ and $GR_{C_2}(\pi) = 0$ then $CR(\pi) = \infty$. π is a contrast sequential pattern if $CR(\pi) \geq mincr$, where *mincr* is the minimum contrast rate threshold.

Table 1 shows a sequences dataset D that we split in D_1 and D_2 according to the classes C_1 and C_2, respectively. We start by finding sequential patterns first and given $minsup = 2$, $\langle a\,b\,c\rangle$ is a sequential pattern because in occurs in sequences 1, 2, 3, and 8. Another example is $\langle c\,b\,a\rangle$ within sequences number 6, 7, and 8. Assuming we have found all the sequential patterns, we check whether these are contrasting for one of the two classes. Given $mincr = 2$, $\pi_1 = \langle a\,b\,c\rangle$ and the metrics $supp(\pi_1, D_1) = 3$, $supp(\pi_1, D_2) = 1$, $GR_{C_1}(\pi_1) = 3$, $GR_{C_2}(\pi_1) = 0.33$, and $CR(\pi_1) = 3$, π_1 is a contrast sequential pattern for C_1 because $CR(\pi_1) \geq mincr$. Given $\pi_2 = \langle c\,b\,a\rangle$ and its metrics $supp(\pi_2, D_1) = 0$, $supp(\pi_2, D_2) = 3$, $GR_{C_1}(\pi) = 0$, $GR_{C_2}(\pi) = \infty$, π_2 is a contrast sequential pattern for C_2. In this specific case it has $GR = \infty$ therefore it is only a pattern for the C_2 class.

3. Mining Condensed Representations of Contrast Sequential Patterns

In traditional pattern mining, algorithms often generate a large number of patterns that may contain redundant or overlapping information. This can lead to issues such as increased computational complexity, difficulty in interpretation, and inefficiency in storing and processing the discovered patterns. Condensed representation techniques address these challenges by summarizing the set of mined patterns into a more compact form without losing important insights or key relationship within data. One common method used for condensed representation concerns the concept of closed and maximal patterns [11]. A pattern s is *closed*, w.r.t. a dataset D, if no other pattern t exists such that $s \subseteq t$ and $supp(s, D) = supp(t, D)$. A pattern s in *maximal*, w.r.t. a dataset D, if there are no other patterns t such that $s \subseteq t$ and $supp(s, D) \geq minsup$. With reference to the example reported in Section 2.2, we

Dataset	$\lvert\Sigma\rvert$	$\lvert$D$\rvert$	$\lvert\lvert$D$\rvert\rvert$	max$\lvert$T$\rvert$	avg$\lvert$T$\rvert$	density
iPRG	21	8628	111,743	12	11.95	0.62
iPRG_25_25	20	50	657	12	11.88	0.64
iPRG_100_100	20	200	2591	12	11.83	0.64
iPRG_500_500	21	1000	12,933	12	11.92	0.62
iPRG_1000_1000	21	2000	25,841	12	11.91	0.61
UNIX	2672	9099	165,748	1256	18.22	0.01
UNIX_25_25	70	50	365	55	7.3	0.10
UNIX_100_100	178	200	2281	175	11.41	0.06
UNIX_500_500	420	1000	13,289	187	13.29	0.03
UNIX_755_755	540	1510	20,234	214	13.4	0.02

Table 2
Features of iPRG and UNIX User sub-datasets: The number of distinct symbols, the number of sequences, the total number of symbols in the dataset, the maximum sequence length, the average sequence length, and the density (calculated by $\frac{||D||}{|\Sigma||D|}$)

know that $\langle a\, b\, c\rangle$ and $\langle c\, b\, a\rangle$ are sequential patterns. Following the definition of closed and maximal patterns, $\langle a\, b\, c\rangle$ and $\langle c\, b\, a\rangle$ are not only maximal but also closed and because $CR(\langle a\, b\, c\rangle) \geq mincr$ and $CR(\langle c\, b\, a\rangle) \geq mincr$ also contrast patterns.

In the next section we examine the computational behavior of the condensed representations for the CSPM task comparing with the results obtained in [16] in Figures 1, 2 and, 3. In pattern mining, it is usual to evaluate the effectiveness (number of extracted patterns), runtime and memory consumption of an algorithm. Moreover in ASP-based DPM approaches it is important to know the solver and grounder time. To this end, we conducted experiments on two datasets (Table 2) creating several subsets of increasing size. In iPRG, each transaction is a sequence of peptides that is known to cleave in presence of a Trypsin enzyme,[3] while in UNIX User, each transaction is a sequence of shell commands executed by a user during one session.[4]. We have chosen these datasets because (i) they are suitable for the task considered in this paper (classified sequences), (ii) they have been already used in the DPM literature, in particular in [11, 4] although for a different task, and (iii) they are publicly available. Notably, transactions in both datasets are labelled with one of two classes, *pos* and *neg*.

Due to lack of space we do not report the ASP encodings and all the experiments carried out, which however can be found in the Github repository devoted to *MASS-CSP*.[5]

3.1. Evaluation

In the following we report and discuss the results obtained from scalability tests on iPRG and UNIX User. We have used the version 5.4.0 of Clingo, with default solving parameters. The timeout (T.O) has been set to 1 hour. The ASP programs were run on a laptop computer with Windows 10 (with Ubuntu 20.04.4 subsystem), AMD Ryzen 5 3500U @ 2.10 GHz, 8GB RAM without using the multi-threading mode of clingo. Multi-threading reduces the mean runtime but introduces variance due to the random allocation of tasks. Such variance is inconvenient for interpreting results with repeated executions.

Tables 3 and 4 summarize the experiments conducted on iPRG and UNIX. It is clear that with the extraction of maximal patterns the output is further reduced compared to closed ones. Obviously a pattern can be both closed and maximal as shown for all sub-datasets when *minsup* grows up to 20%. As the input to the program increases, the total execution time and occupied memory increases accordingly as well as grounding time *(time - solv. t.)* but not much. Only when the size of the dataset reaches the order of thousands of rows (iPRG_1000_1000) or when *minsup* is 10% (iPRG_500_500 maximal) the process ends because of the timeout. The reason is that a high value of *minsup* is able to reduce the search space and runtime. Figures 1, 2 (iPRG), and 3 (UNIX) compare basic, closed and maximal representations.

[3]https://dtai.cs.kuleuven.be/CP4IM/cpsm/datasets.html
[4]https://archive.ics.uci.edu/ml/datasets/UNIX+User+Data
[5]https://github.com/mpia3/Contrast-Sequential-Pattern-Mining

It seems that maximal representation requires more time compared to the closed one while closed representation requires more memory compared to the maximal one. The advantage of using condensed representations instead of the basic frequent formulation is when we want to reduce the output size. It is strange that for UNIX_100_100 (Figure 3) the number of condensed patterns is higher than the basic ones and it will be necessary to delve deeper into this phenomenon to understand the underlying reasons. As the reader can see from the Figures mentioned before, condensed representations require more execution time and memory due to the increase in new atoms to be derived to represent these new patterns which increment the program size. Also in this case, as in various other contexts, it is necessary to define a trade-off between what types of information are wanted and at what cost.

Closed

(a) iPRG_25_25, mincr = 3

minsup	#pat	time	solv. t.	memory
10%	164	4.72	4.52	31.84
20%	23	2.59	2.41	31.82
30%	2	1.46	1.3	31.38
40%	0	0.93	0.77	30.67
50%	0	0.36	0.19	29.83

(a) iPRG_100_100, mincr = 3

minsup	#pat	time	solv. t.	memory
10%	536	75.48	74.46	96.69
20%	14	22.26	21.36	84.27
30%	0	8.19	7.33	82.53
40%	0	4.72	3.88	80.4
50%	0	5.51	4.68	77.77

(a) iPRG_500_500, mincr = 3

minsup	#pat	time	solv. t.	memory
10%	71	993.41	982.49	604.75
20%	12	554.70	543.17	600.45
30%	0	246.49	235.8	600.45
40%	0	1242.32	1231.62	579.17
50%	0	1745.27	1735.21	585.21

(a) iPRG_1000_1000, mincr = 3

minsup	#pat	time	solv. t.	memory
10%	14	T.O	3560.14	1883.13
20%	9	T.O	3559.32	1875.15
30%	0	736.96	694.01	1861.71
40%	0	T.O	3559.38	1851.53
50%	0	3132.80	3093.54	3270.31

Maximal

(b) iPRG_25_25, mincr = 3

mincr	#pat	time	solv. t.	memory
10%	115	13.01	12.73	34.35
20%	23	9.09	8.77	39.13
30%	2	5.58	5.26	38.5
40%	0	3.01	2.7	34.7
50%	0	1.29	0.94	30.86

(b) iPRG_100_100, mincr = 3

mincr	#pat	time	solv. t.	memory
10%	460	116.96	115.91	88.4
20%	14	30.37	29.22	82.51
30%	0	15.54	14.65	84.4
40%	0	10.92	9.98	83.43
50%	0	7.14	6.24	82.55

(b) iPRG_500_500, mincr = 3

mincr	#pat	time	solv. t.	memory
10%	28	T.O	3589.91	555.71
20%	12	2167.71	2156.21	549.48
30%	0	187.37	176.54	546.68
40%	0	1523.46	1513.25	553.42
50%	0	766.99	757.04	600.09

(b) iPRG_1000_1000, mincr = 3

mincr	#pat	time	solv. t.	memory
10%	11	T.O	3560.77	2787.54
20%	9	T.O	3557.44	1779.18
30%	0	T.O	3574.37	1773.29
40%	0	2389.90	2342.93	1740.44
50%	0	T.O	3558.29	1729.63

Table 3
Number of closed and maximal patterns, runtime (seconds), solver time (seconds) and memory consumption (MB) on all iPRG sub-datasets by varying *minsup* and leaving fixed *mincr*. T.O means timeout

4. Conclusions

Contrast Sequential Pattern Mining offers a powerful tool for exploring sequential data and discovering meaningful patterns by highlighting differences between groups of sequences. It has applications across various domains such as market analysis, healthcare research, fraud detection, and more where understanding contrasts in sequential data is crucial for decision-making and problem-solving. Combinatorial explosion is typically involved in pattern mining. Condensed representation for patterns are the solutions proposed in the literature to address this issue. We have reported the results of the evaluation of closed/maximal contrast sequential patterns comparing with the basic CSPM representation in ASP. We have used two datasets from two different domains for our evaluation. The experiments illustrate

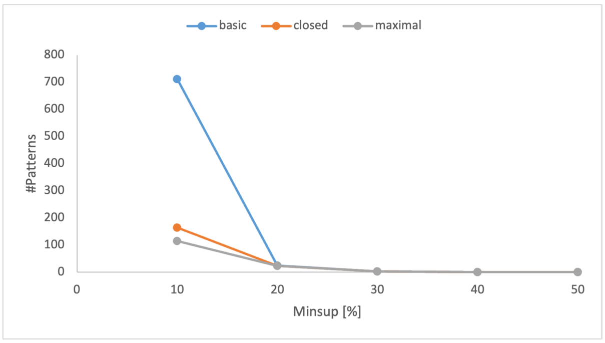

Figure 1: Comparison on number of basic, closed and maximal contrast sequential patterns extracted on iPRG_25_25

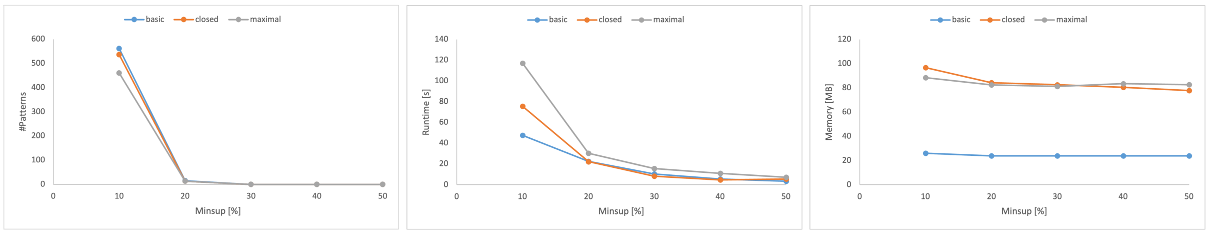

Figure 2: Number of basic, closed and maximal contrast sequential patterns extracted , runtime, and memory consumption on iPRG_100_100

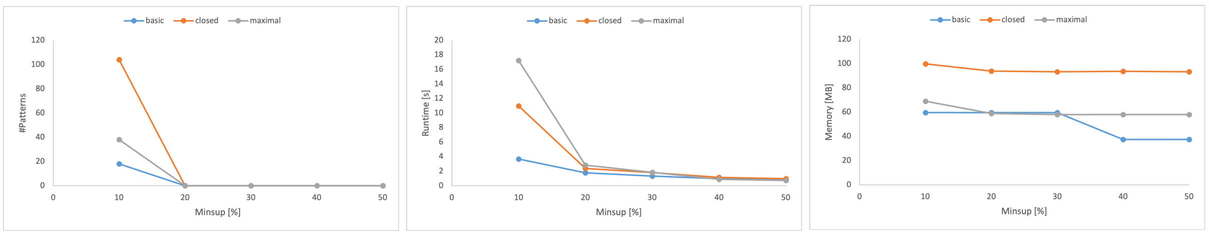

Figure 3: Runtime and memory consumption comparison on basic, closed, and maximal contrast sequential patterns for UNIX_100_100

what are the advantages and weaknesses of condensed representations and in particular pros and cons of closed and maximal patterns. On one hand, they reduce the number of patterns, on the other hand, they tend to consume more computational resources. Overall, condensed representations for pattern mining offers a valuable tool for extracting actionable insights from data by simplifying complex pattern structures into concise yet informative summaries that facilitate better decision-making processes across various domains such as market analysis, bioinformatics research, customer behavior prediction among others where understanding underlying trends is crucial for making informed decisions.

Acknowledgments

This work was partially supported by the project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

References

[1] C. H. Mooney, J. F. Roddick, Sequential pattern mining–approaches and algorithms, ACM Computing Surveys (CSUR) 45 (2013) 1–39.
[2] G. Dong, J. Bailey, Contrast data mining: concepts, algorithms, and applications, CRC Press, 2012.
[3] Y. Chen, W. Gan, Y. Wu, P. S. Yu, Contrast pattern mining: A survey, 2022. arXiv:2209.13556.

Closed | **Maximal**

(a) UNIX_25_25, mincr = 3

minsup	#pat	time	solv. t.	memory
10%	13	0.23	0.09	28.77
20%	1	0.17	0.05	23.57
30%	0	0.15	0.03	20.25
40%	0	0.14	0.02	21.3
50%	0	0.17	0.04	25.45

(b) UNIX_25_25, mincr = 3

mincr	#pat	time	solv. t.	memory
10%	8	0.22	0.11	24.39
20%	1	0.14	0.04	19.63
30%	0	0.12	0.02	21.42
40%	0	0.11	0.02	24.00
50%	0	0.13	0.04	18.8

(a) UNIX_100_100, mincr = 3

minsup	#pat	time	solv. t.	memory
10%	104	10.99	9.82	99.74
20%	0	2.40	1.55	93.93
30%	0	1.82	1.02	93.39
40%	0	1.16	0.34	93.71
50%	0	0.98	0.21	93.32

(b) UNIX_100_100, mincr = 3

mincr	#pat	time	solv. t.	memory
10%	38	17.24	16.55	68.95
20%	0	2.83	2.29	58.94
30%	0	1.84	1.23	58.00
40%	0	0.87	0.31	57.99
50%	0	0.74	0.19	57.99

(a) UNIX_500_500, mincr = 3

minsup	#pat	time	solv. t.	memory
10%	13	175.92	162.35	902.93
20%	1	68.30	53.48	890.00
30%	0	54.61	41.47	888.28
40%	0	31.15	17.86	890.66
50%	0	21.76	8.84	890.66

(b) UNIX_500_500, mincr = 3

mincr	#pat	time	solv. t.	memory
10%	12	135.03	126.04	487.08
20%	1	65.08	56.13	494.34
30%	0	46.67	37.83	483.34
40%	0	26.66	17.75	483.21
50%	0	17.70	8.80	483.21

(a) UNIX_755_755, mincr = 3

minsup	#pat	time	solv. t.	memory
10%	4	T.O	3569.58	1855.13
20%	0	206.60	176.4	1834.99
30%	0	124.90	95.39	1812.50
40%	0	80.04	50.43	1811.34
50%	0	51.06	21.00	1811.33

(b) UNIX_755_755, mincr = 3

mincr	#pat	time	solv. t.	memory
10%	3	T.O	3579.70	936.05
20%	0	190.55	169.98	936.61
30%	0	109.95	89.79	930.67
40%	0	71.47	50.74	930.55
50%	0	41.23	21.01	930.55

Table 4
Number of closed and maximal patterns, runtime (seconds), solver time (seconds) and memory consumption (MB) on all UNIX sub-datasets. T.O means timeout

[4] B. Negrevergne, T. Guns, Constraint-based sequence mining using constraint programming, in: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, Springer, 2015, pp. 288–305.

[5] M. Gebser, T. Guyet, R. Quiniou, J. Romero, T. Schaub, Knowledge-based sequence mining with ASP, in: IJCAI 2016-25th International joint conference on artificial intelligence, AAAI, 2016, p. 8.

[6] S. Jabbour, L. Sais, Y. Salhi, Decomposition based SAT encodings for itemset mining problems, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2015, pp. 662–674.

[7] T. Guns, A. Dries, S. Nijssen, G. Tack, L. De Raedt, Miningzinc: A declarative framework for constraint-based mining, Artificial Intelligence 244 (2017) 6–29.

[8] T. Guyet, Y. Moinard, R. Quiniou, Using answer set programming for pattern mining, arXiv preprint arXiv:1409.7777 (2014).

[9] A. Samet, T. Guyet, B. Negrevergne, Mining rare sequential patterns with ASP, in: ILP 2017-27th International Conference on Inductive Logic Programming, 2017.

[10] T. Guyet, A. Happe, Y. Dauxais, Declarative sequential pattern mining of care pathways, in: Artificial Intelligence in Medicine: 16th Conference on Artificial Intelligence in Medicine, AIME 2017, Vienna, Austria, June 21-24, 2017, Proceedings 16, Springer, 2017, pp. 261–266.

[11] T. Guyet, Y. Moinard, R. Quiniou, T. Schaub, Efficiency analysis of ASP encodings for sequential pattern mining tasks, in: Advances in Knowledge Discovery and Management, Springer, 2018, pp. 41–81.

[12] S. Paramonov, D. Stepanova, P. Miettinen, Hybrid ASP-based approach to pattern mining, Theory Pract. Log. Program. 19 (2019) 505–535. URL: https://doi.org/10.1017/S1471068418000467. doi:10 .

1017/S1471068418000467.

[13] F. A. Lisi, G. Sterlicchio, Declarative pattern mining in digital forensics: Preliminary results, in: R. Calegari, G. Ciatto, A. Omicini (Eds.), Proceedings of the 37th Italian Conference on Computational Logic, Bologna, Italy, June 29 - July 1, 2022, volume 3204 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 232–246. URL: http://ceur-ws.org/Vol-3204/paper_23.pdf.

[14] F. A. Lisi, G. Sterlicchio, Mining sequences in phone recordings with Answer Set Programming, in: P. Bruno, F. Calimeri, F. Cauteruccio, M. Maratea, G. Terracina, M. Vallati (Eds.), Joint Proceedings of the 1st International Workshop on HYbrid Models for Coupling Deductive and Inductive ReAsoning (HYDRA 2022) and the 29th RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion (RCRA 2022) co-located with the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2022), Genova Nervi, Italy, September 5, 2022, volume 3281 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 34–50. URL: http://ceur-ws.org/Vol-3281/paper4.pdf.

[15] F. A. Lisi, G. Sterlicchio, A declarative approach to contrast pattern mining, in: A. Dovier, A. Montanari, A. Orlandini (Eds.), AIxIA 2022 - Advances in Artificial Intelligence - XXIst International Conference of the Italian Association for Artificial Intelligence, AIxIA 2022, Udine, Italy, November 28 - December 2, 2022, Proceedings, volume 13796 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 17–30. URL: https://doi.org/10.1007/978-3-031-27181-6_2. doi:10.1007/978-3-031-27181-6_2.

[16] F. A. Lisi, G. Sterlicchio, Mining contrast sequential patterns with ASP, in: R. Basili, D. Lembo, C. Limongelli, A. Orlandini (Eds.), AIxIA 2023 - Advances in Artificial Intelligence - XXIInd International Conference of the Italian Association for Artificial Intelligence, AIxIA 2023, Rome, Italy, November 6-9, 2023, Proceedings, volume 14318 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 44–57. URL: https://doi.org/10.1007/978-3-031-47546-7_4. doi:10.1007/978-3-031-47546-7_4.

[17] V. Lifschitz, Answer sets and the language of answer set programming, AI Magazine 37 (2016) 7–12.

[18] G. Brewka, T. Eiter, M. Truszczynski, Answer set programming at a glance, Communications of the ACM 54 (2011) 92–103. URL: http://doi.acm.org/10.1145/2043174.2043195. doi:10.1145/2043174.2043195.

[19] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, Clingo= ASP+ control: Preliminary report, arXiv preprint arXiv:1405.3694 (2014).

[20] N. Leone, C. Allocca, M. Alviano, F. Calimeri, C. Civili, R. Costabile, A. Fiorentino, D. Fuscà, S. Germano, G. Laboccetta, B. Cuteri, M. Manna, S. Perri, K. Reale, F. Ricca, P. Veltri, J. Zangari, Enhancing DLV for large-scale reasoning, in: M. Balduccini, Y. Lierler, S. Woltran (Eds.), Logic Programming and Nonmonotonic Reasoning - 15th International Conference, LPNMR 2019, Philadelphia, PA, USA, June 3-7, 2019, Proceedings, volume 11481 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 312–325. URL: https://doi.org/10.1007/978-3-030-20528-7_23. doi:10.1007/978-3-030-20528-7_23.