Rheinisch-Westfälische Technische Hochschule Aachen
Chair of Process and Data Science
Prof. Dr. Wil van der Aalst

Process Conformance Checking in Python in SS 2021

Alignments on NFA(s) in Micropython

Group:3

Testing, Assessment & The Project Retrospective

18.07.2021

supervisor: Alessandro Berti

Contents

List of Figures

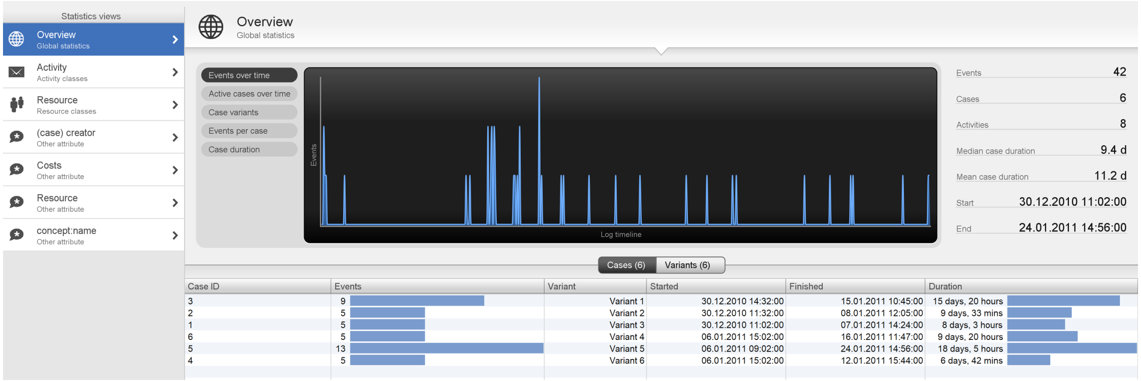
1 Introduction

The goal of this project is to implement conformance checking in python with no dependencies. In order to achieve it, in the first sprint we started with converting regular expressions into NFAs, implemented NFA model and checked its fitness. However, in the second sprint was dedicated for computation of optimal alignments, development of unit tests and optimization of the code of first sprint by catering the edge cases. The third sprint was about improving the implementation by adapting to a new and more optimal approach in terms of memory and performance. In this phase we performed the final testing and assessing of our algorithm which is the most essential step into qualifying and quantifying the performance and usefulness of an algorithm. In particular when considering the usage of algorithms in a workplace, it is important for the program to be able to adjust to fast changing situations and therefore make sure that all results are accurate and are able to be produced as efficiently as possible. This document covers a brief summary of what we had planned and how we have achieved it in the current phase. Chapter 2 discusses event logs used in the testing assessment in more details,event logs used in scalability evaluation and other comparative studies,chapter 3 contains the evaluation of our algorithm for the scalability in terms of time and memory, Chapter 4 is comprised of phase reviews of the entire team.

2 Event Logs

We divided the over all evaluation into sub-tasks in order for each team member to tackle a sub-task independently, so each team member had to find a way to get suitable event logs or even create his own event log to sufficiently test and evaluate our alignment model and that is due to the nature of our model that accepts the input in a certain format which is not available for all online event logs.

The event log used in comparison of the two approaches for conformance checking using Alignments is *running-example.xes*. Figure 1 gives an overview of the event log statistics.



Events	42
Cases	6
Activities	8
Median case duration	9.4 d
Mean case duration	11.2 d
Start	30.12.2010 11:02:00
End	24.01.2011 14:56:00

Case ID	Events	Variant	Started	Finished	Duration
3	9	Variant 1	30.12.2010 14:32:00	15.01.2011 10:45:00	15 days, 20 hours
2	5	Variant 2	30.12.2010 11:32:00	08.01.2011 12:05:00	9 days, 33 mins
1	5	Variant 3	30.12.2010 11:02:00	07.01.2011 14:24:00	8 days, 3 hours
6	5	Variant 4	06.01.2011 15:02:00	16.01.2011 11:47:00	9 days, 20 hours
5	13	Variant 5	06.01.2011 09:02:00	24.01.2011 14:56:00	18 days, 5 hours
4	5	Variant 6	06.01.2011 15:02:00	12.01.2011 15:44:00	6 days, 42 mins

Figure 1: Overview of the log

2.1 Scalability:

Scaleability measurement is a pivotal role in estimating the barriers of the implementation. A common way to measure quantify scalability is to survey how the implementation reacts to a great amount of data. In this sub-task we came up with an idea to create three different sized event logs from a regular expression. This idea came from the situation that we need a variety of event logs in different sizes in order to check the scalability of out algorithm, and also all live event logs have different format from our model's input so we had two options: Firstly, try to find three different event log with various sizes and then try to discover the regular expression that represents these event logs and then build our NFA model based on this regular expression,and then convert these event logs to a trace in a string format in order to fit in our model and then run our memory and time evaluations. The other approach which we adapt in this phase of the project is as follows: -Firstly, we decided to get a reasonable regular expression. -Then, generate all possible strings that matches this regular expression. -After that, we divided these strings into three different files with the following sizes 1K, 10K and 100K traces -And we manipulate each file to have 50% of the traces match our model and the other half don't match so we will calculate the alignment for various traces. -Then, we create our NFA model from the regular expression we agree on. -And finally, we run our algorithm on the different three files with the different traces

3 Alignments on NFA(s) Evaluation

3.1 Scalability:

In software engineering evaluation and testing Scaleability measurement is a pivotal role in estimating the barriers of the implementation. A common way to measure quantify scalability is to survey how the implementation reacts to a great amount of data. In the following, we will quantify how our implementation of the alignments on NFA models handles three different types of data sizes and how our algorithm responds to different lengths of traces. This will be quantified in terms of time and memory usage. In the end, we will give a summary of the results.

3.1.1 Memory and time performance:

This section going demonstrate the performance in terms of running time and memory usage for our implementation of alignment on NFA(s). For that, we created four different event logs with different sizes. First, a small event log that contains 1000 traces in string format with different traces lengths, second medium event log which contains 10000 traces also in string format with different string lengths,third large event log with 100000 traces also, and at last an event log that contains traces with lengths from 1 to 50 in order to check how our algorithm responds to different lengths of the traces. Our method of measuring time was to measure the execution time of the block of code that deal with each file separately. To achieve the measured time we used the Timeit package and wrap the code we want to measure the running time in the timeit() function. For the memory we calculated the maximum memory usage for each function with the memory profile module in python. The results of the methodology can be seen in the following.

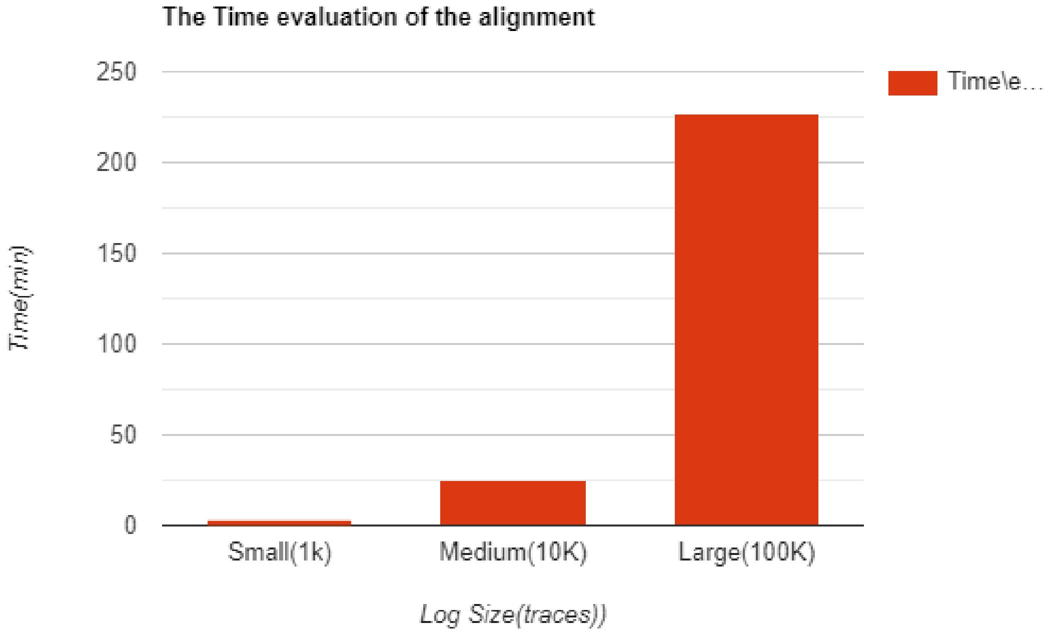


Figure 2: The runnig time for each event log [10].

In Figure 2 we can see that the execution time of the algorithm takes. Namely, 195 seconds or 3.5 minutes for the small event log, 1454 seconds or 24.5 minutes for the medium event log, and 227 minutes for the large event log. It is noticeable to see how the execution time increases greatly from the small event log to the medium event log as this log has a lot more events. Another significant increase can be seen from the medium log to the large event log. Still, it is within a reasonable time for its size. The figure 3 will illustrate the memory performance.

Interestingly, as we can see in fig 3, the memory consumption between the small and medium log changed slightly as the two files can easily handled in the memory. For the large event log, the memory consumption is a little bit big but it is not as we expect to be huge compared to the running time.

In figure 4 we measured the running time of 50 traces with lengths from 1 char to traces with 50 char to check the behaviour of our implementation towards different traces lengths and try to figure out if there is a pattern or a correlation between the trace length and the running time, and as we expected the shorter the trace the less running time it takes as the algorithm search for the shortest trace to align the input. And the longer the trace the more time it consume to calculate the alignment. So in general there is a pattern that tells us that the run time increases linearly with the increase of the length of the trace.Of course there is some difference between the traces that have nearly the same length or close to each others.

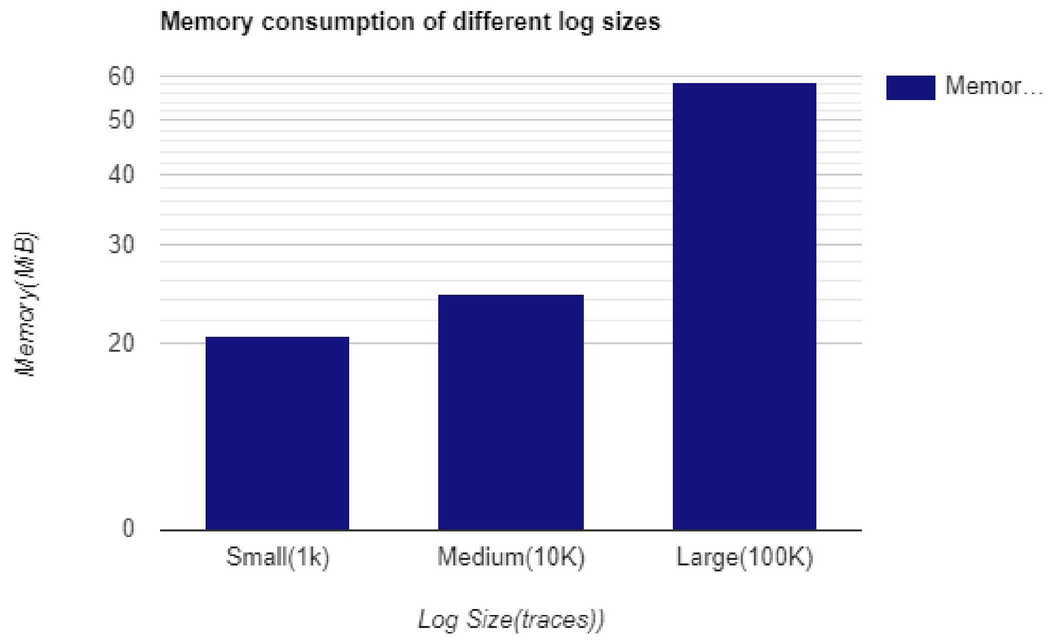


Figure 3: The memory usage for each event log size [10].

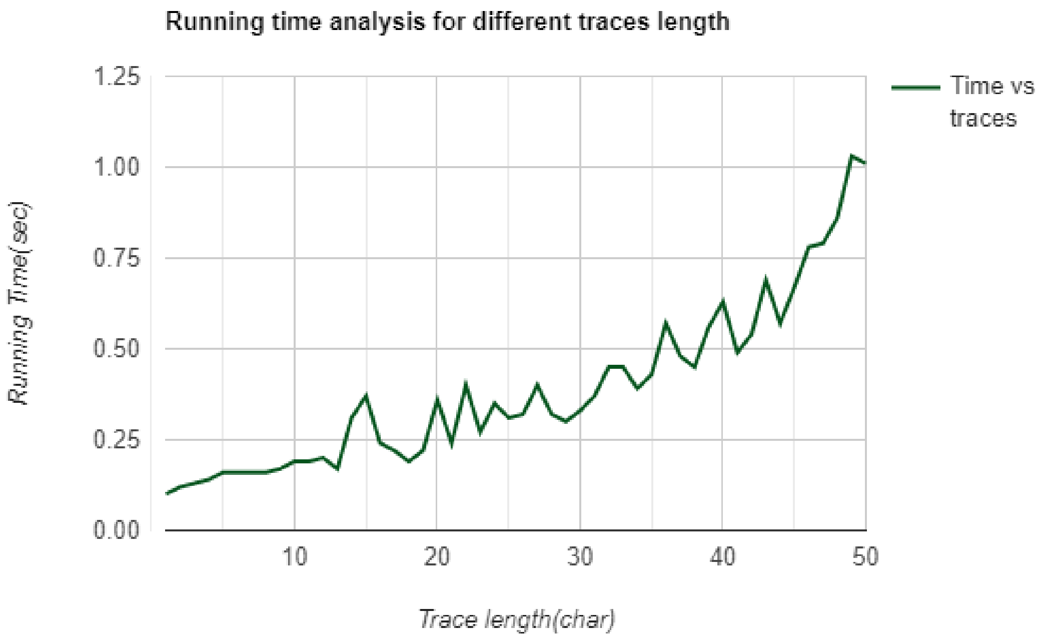


Figure 4: The running time of different traces from length 1 to 50 [10].

4 Comparison of Conformance checking via Alignments: PM4Py vs Our approach

In conformance checking, a comparison between an event log and a process model is performed in order to find deviations and understand the problems related to the process so that it can be improved by redesigning.

The approach we had to use for this purpose was **Alignments on NFA(s)**. NFA(s) are non-deterministic finite automaton that was first introduced by Michael O. Rabin and Dana Scott in 1959 [2]. Formally they are described by a tuple of 5 parts (Q, Σ, Δ, q0, F).
With Q being the finite set of states.
Σ being the finite set of input symbols.
$\Delta : Q\Sigma \rightarrow P(Q)$ being the transition function where it is possible to have non deterministic behaviour such that an input can have multiple places as the target from a given start place.
q0 being the initial start place. F is a set of places that are allowed final states.

In the approach used by us, we have converted regular expressions to NFA and performed verification of a process execution against that regular expression by replaying traces and calculating their alignments. Whereas in the already existing approach used in PM4Py library, Petri net is created from a given log using Inductive Miner and then alignments are computed using that model.

This section compares the two approaches in terms of Time and Memory being used by each of them.

4.1 Memory and time performance:

The log used for comparing both the approaches is *running-example.xes*, details of which are already given above in the Event logs section of this document.

In the already existing PM4Py approach, the mentioned log is extracted and the Petri net as shown in figure 5 is generated from it using inductive miner. Alignments are then computed using the traces in the log.

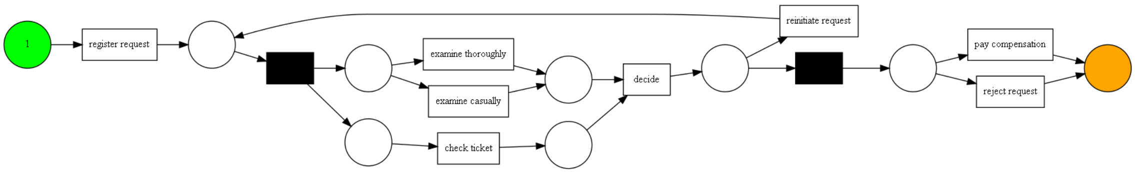


Figure 5: Generated Petri Net

For being able to compare the two approaches we had to ensure that the input used in both was the same. For this purpose we created the process tree as shown in figure 6 of the similar log in ProM and manually translated it into a regular expression that we then used in our approach for creating NFA model and consequently computing alignments by replaying traces of log. Since the log was in XES format, we had to convert it to string format in order to fit in our model and then run our memory and time evaluations. The complete event names were also mapped on to alphabets using lookup table that was given as input to the evaluation algorithm. The file showing this mapping is also uploaded on Git named *events_mapping.csv*.

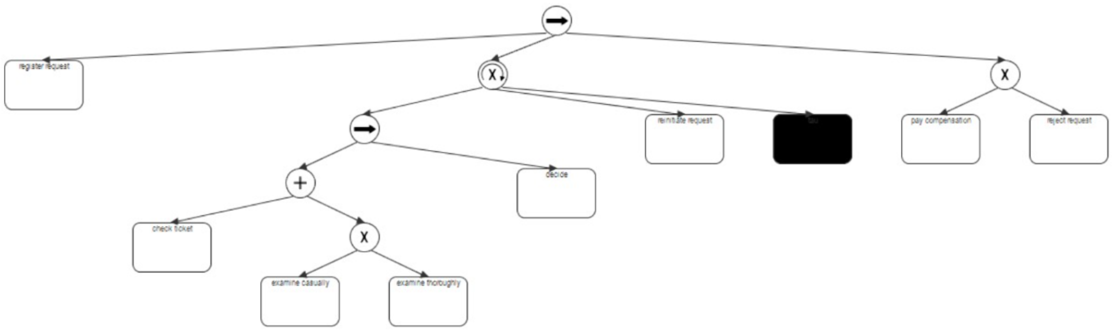


Figure 6: Process Tree of the log

To achieve the measured time we have used the Timeit package and have wrapped the code we want to measure the running time in the timeit() function. For the memory we have calculated the average memory usage for each function with the memory usage module in python.

The results we received can be seen in figure 7 and figure 8 below.

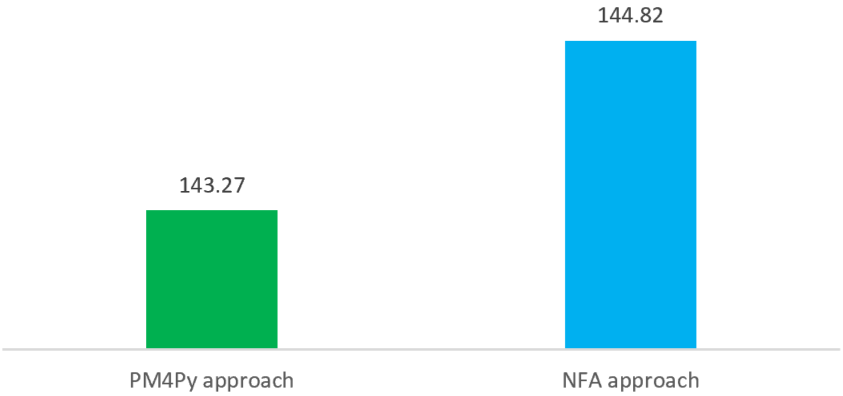


Figure 7: Memory Comparison

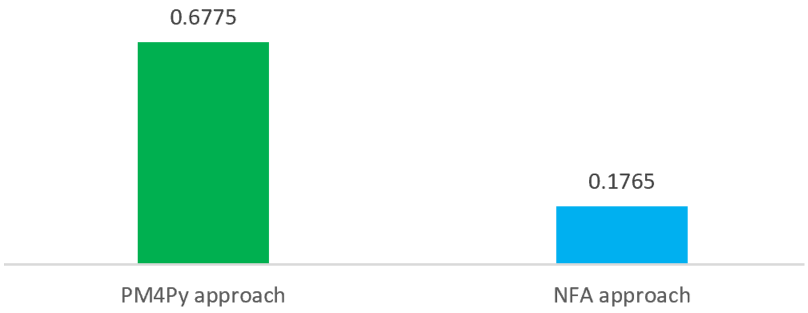


Figure 8: Time Comparison

5 Competing Approaches

5.1 Conformance Checking of Processes Based on Monitoring Real Behavior

Discrepancies, among a log and a process model, of two kinds are detected by techniques used for conformance checking. First one being the behavior seen in the log but is not allowed by the model referred as unfitting behavior and the second one being the behavior which not seen in the log which is allowed by the model referred as additional behavior. Token-based replay [6] is a simple approach used to measure and detect unfitting behavior. Each trace is replayed against the model, depicted as a Petri net. The order followed by the transitions is directed by the trace given. For firing, enabling a transition is necessary, meaning that at least one token I required in its every incoming place. When firing of a transition does not happen because of not being enabled, the determination of which token should be added is done by this technique. The tokens that are left in the Petri-net's non-sink place after the token is replayed are called remaining tokens. To quantify the difference between the log and the model, we the use the number of tokens that are added and remaining called replay errors.

5.2 Conformance Checking using Cost-Based Fitness Analysis

Trace Alignment [7] is another technique used for conformance checking which deals with the limitations of replay based conformance checking. For each row in the log (trace) an alignment is computed using the model which is as much similar to the trace as possible. The places which differ between the log and the model are highlighted and a cost is assigned to them. The process starts with converting the trace into a Petri net. A product of both the Petri Nets, Petri Net of the trace and Petri Net of the model is computed. The product is basically computed by pairing transitions with the same label. This product consists of all the paths that could be taken i.e. all possible alignments. In order to find the optimal alignment a search algorithm A* is used which returns the path with the minimal cost. All synchronous moves between log and model are assigned a cost of 0 and all others with a cost of 1. If you return only one optimal alignment then it will miss out on some of the behavioral differences.

5.3 Interpretable Conformance Checking of Business Processes

Behavioral alignment [8] is another technique that deals with both (unfitting and additional behavior) collectively. This technique converts the model and the log into event structure. Similar to Trace alignment technique a product with minimal error is computed. From this product, a set of associations are derived which define all the behavioral relations between the events that are present in the model but are not found in the log and similarly events which were present in the log but not identified in the model. This technique is also very exhaustive as it tries to capture all the set of differences between the log and the model.

5.4 Scalable Conformance Checking of Business Processes

. The approach presented in this paper [9] is very much similar to what we are doing in our approach. It uses automata as automata based implementations are dramatically faster and have more consistent speed and are memory-efficient representation for event logs and process models. The event log is converted into a Deterministic Acyclic Finite Automaton and a Finite State Machine is used to represent the model. This techniques computes the set of optimal alignments and the set of differences (basically combines the work done in [7] [8]) and tries to improve the efficiency of state of the art conformance checking techniques by using automata and memory-efficient techniques.

5.5 Conformance Checking using Alignments based on NFA

. The approach we have used in our project makes use of automata in order for it to be fast and memory efficient. Inductive miner is a technique used for process discovery. It outputs a Process Tree which can be easily converted into a Regular Expression. We in our approach use a regular expression as input to build a NFA representing our process model.In order to find an optimal alignment we use Dijkstra algorithm to search in the search space for the path with the minimum cost. A search space covers all possible combinations of positions in the trace together with states of the given process model. Therefore shortest path in this search space is equivalent to an optimal alignment with the given cost function. We used the conventional cost function where synchronous move cost is 0 and asynchronous move cost is 1.

6 Project Retrospective

In this section, all the main project successes and problems will be analysed. This includes every aspect of the project including which major requirements were met and which were not, as well as agile project development techniques that were used throughout the project.

6.1 Project Successes: What Worked Well?

6.1.1 Trello Dashboard

Trello was used effectively throughout the project in managing and assigning different tasks to each member of the group. At the start of the project, all backlog items were discussed and reviewed, Later on changes were made to the cards as per requirement of each sprint. Each member was responsible to update and move their Trello card to the correct label.Currently all the cards have been moved to Delivered tab as we were able to complete all the requirements.

6.1.2 Work Distribution

At the start of each sprint, we used to have a meeting to divide the tasks. Everyone was assigned the task according to their preferences. We luckily had a team where every person had some strong suite which really helped us as we had people with experience in the tasks they were performing.We had a Whatsapp group where if any team member faced difficulties in doing their task would highlight the issue and everyone was there to help that person out.

6.1.3 GitHub Repository

In order to have a clear organization structure when multiple developers work on the same project, a clear system has to be used in order for code to always be in working condition and conflicts to be avoided. Throughout the project, the 'git flow' structure of maintaining a repository was used as the basis. At the start, there was a small learning curve for all the developers, however once all developers got used to the format, all code committed to the repository was well maintained and structured in a way where there was always working code in the main and develop branches.

6.1.4 Overleaf

In order to work on the submission document simultaneously we made use of overleaf. For every sprint a file was created on the overleaf which was accessible to all the team members. Everyone was responsible for writing their own part and with the help of overleaf every one was able to work on the document at the same time. Before the submission of every document we conducted a meeting to review everyone's part. Reviewing it as a team helped us identify any underlying problems. Every one was allowed to give suggestions for improvement. The final version was always the one where every one was satisfied with the structure of the document.

6.1.5 Documentation

From the start our main emphasis was to not only focus on the implementation of the approach but also on documenting everything in such a way that people prefer to use our library. We firstly focused on our code that each part of the code is well explained in the comment section so that any person who uses this code can easily understand it just by looking at the comments. For documenting the code, we have used Pdoc. It is a software package for generating API documentation for Python language. Documentation is generated from the source code docstrings and HTML documentation is generated for chosen Python modules. The html files are uploaded in the Git repository and can be accessed by going to html - nfa.html,conformance.html. Project manual (README) is also added to guide the users regarding the details of our project like its purpose, why is it useful, what to refer to in case of any ambiguities.Its a brief overview of the overall idea and the details of its execution to make it easy for the end users to not only get the idea of our work but also enable them to use it in future,wherever required.

6.2 Project Challenges: What did not work well?

6.2.1 Suitable Meeting Time

When we started the project, due to a very diverse team, we faced issues in planning discussion sessions because of different schedules each individual had. But once we had a clear vision of the project requirements and deadlines, we worked this out and everybody was very generous in accommodating other members of the group. This got better with each sprint to the level that communication became the strongest part of our team.

6.2.2 Multiple Perspectives

Since everyone was over enthusiastic in coming up with the best possible solutions, at times we had different approaches for the implementation of the same module. Initially we decided of making two different teams, both implementing the entire solution end to end. This made us work on two approaches. Eventually we had to let go of one approach so all the work done by the other team went in vain and we also wasted a lot of time in two possible implementations. We learnt our lesson from this and from that point on wards we always did the theoretical analysis first, decided on a single approach and then started working on it with clear distribution of tasks where replication of tasks was strictly avoided.

6.3 Extended Phase Review & What We Learned

6.3.1 Asad Tariq

This phase started with the usual distribution and organization of tasks. We started off with a meeting to discuss the work load distribution and also how to proceed towards the finishing phase. We tried to maintain the level of communication we had in all the previous sprints and it lead us to achieve our goals successfully. Every member was always there to help each other out and depicted a great example of team work.

Talking about the lab as a whole, I believe this was a great learning experience for me in every perspective.Even with such a diversified team, we managed to put up such great work, mainly because we tried to utilize the distinct strengths of every member for our lab. The situation always remained agreeable for us and we got comfortable with everyone's working style.

From the beginning my learning has increased with every sprint, starting with the presentation of our project as a business model. firstly, I tried to get my hands on the theoretical aspects of the project to strengthen my understanding and concepts. Then I worked on the implementation as well by adding the input checks. We made sure that the input is entered in the connvention that is allowed for computing the NFA. In our case, the input should have been a list of alpha-numeric values for a trace and alpha-numeric values along with some special characters which are predefned for a regular expression. We added the input checks in a way so that the user gets notifed with an alarm message containing the respective error.

With this project, I was able to figure out my strengths and weaknesses. I tried to acquire as much knowledge as I can in both theory and practication implementation.

6.3.2 Mahmoud Emara

In this phase, we organized our task distributions as we always have been. We meet at the beginning and discuss what we want to do for the remaining time. My job for this phase was to evaluate the scalability of our algorithm. Something that seems to be not that easy as it sounds as one has to think what things could be scalable in alignment on NFA(s) and the absence of sufficient event logs . Still I managed to complete my part. Providing different-sized event logs and analyzing the scalability behaviour of our algorithm of the alignment on NFAs was very beneficial, as we were able to dig deeper into the pros and cons of our implementation. Comparing the running time and memory consumption of our implementation provided different sized event logs provided us with a deeper understanding of the behavior of the implementation. I feel that this last sprint was very fruitful as the assessment of the project is like harvesting our efforts and putting our work into action. The planning of this sprint was very smooth and there was a high collaboration between all team members which made this sprint enjoyable. This project was challenging yet fun. First and foremost, I centered my attention on the theoretical aspect of the project. This step was a double-edged sword, on one hand, I have understood the underlined theory, on the other hand, I did not spend a lot of time in the implementation since my tasks were only based on theory. In the first sprint I Co-operate with other team members in the implementation of the NFA model class which represents the core data structure of our algorithm. In the second sprint, my task was to implement the unit tests for the most important functions of NFA class and the initial version of the alignment algorithm. In the last sprint I was able to implement unit tests for all functions of the NFA class and the optimized version of our algorithm. Over all taking part in this project helped me a lot to figure out my strengths and weaknesses. I have learned a lot in this project moreover, I have acquired a piece of knowledge in both practical including theoretical aspects.I am proud of our report and the results we delivered at the end. Everyone gave their best despite the current situation and I can humbly say that I learned a lot in this lab. I can proudly say that we produced something of value that could be used in future projects.

6.3.3 Syed Faizan Hassan

The assessment phase was tackled the same way how we tackled all the other phases. We started of with a meeting to discuss and distribute the tasks to what everyone thought was the most logical distribution. We conducted a meeting with the professor to discuss how to proceed with the assessment phase as the approach we worked on has never been implemented before. Since I worked as a Theory Expert throughout the whole project I was assigned the task to look for other similar approaches and how they work. I studied different research papers thoroughly to develop an understanding of what algorithms and logic have been used by them and how our approach differs from them. Just like the other sprints our communication was the key to complete everything on time. We were always there to help each other out in case any one got stuck.I am very happy with the way we worked in this sprint and I am looking forward to the Oral Exam.

Overall if we look at how things started from scratch, I believe we have come very far. I got a lot to learn throughout the lab. Initially I was a bit worried about working with such a diverse team but as soon as I got to know everyone I became very comfortable. Our communication remained the key throughout the project for completing everything the way we intended it to.

I learned how a project is presented as a Business case to the stake holders which is something I never did before.

In the requirement engineering phase I learned how requirements are gathered from stake holders and are written down in different levels of hierarchy.

The coding part started from sprint 1. We initially divided the team into two groups and decided that both groups will try to build an NFA from a regular expression. The teams came up with two different implementations and then we decided to move forward with one approach which looked more reasonable. The most important thing I learned from this sprint was that there could be multiple solutions to one problem and you should have proper logical reasoning to justify the approach you use.

In sprint 2 our goal was to implement the alignment computation for traces. In this sprint I did the theoretical analysis of how the computation can be performed. Since we all were busy during this sprint due to exams we decided to leave the part of optimization for sprint 3 and just focus on completing the implementation of the approach. However we got some marks deducted due to this reason which was a set back for us. We then contacted the professor and told him about how we planned to do the optimization in the third sprint. I learned from this incident that strong communication only between the team is not sufficient and such things should be communicated to the stake holders as well so that everyone is on the same page.

We were told that if we optimize our code the deducted marks from sprint 2 would be fixed. We were very enthusiastic about this sprint as we wanted to fix the misunderstanding that led to a less score in sprint 2. We again first did the theoretical analysis and were successful in optimizing our code. Our efforts didn't go to waste and not only we scored full marks in sprint 3 but also an improvement in the sprint 2 marks which lifted the mood of the whole team again.

Lastly with the last assessment phase, I understood that how it is so important to not only compare the quality of your product with other competing products but also the scalability and benefit it offers in itself. For others to use your library it should be very well documented that what benefits it bring along with itself and why someone should opt for your work.

Overall, I learnt many new things throughout the entire life-cycle of this project. I feel like I have improved my communication skills, I have learned how to work in a diverse team and especially presenting of business case arguments which will be very useful in all the future projects.

6.3.4 Lukas Liß

In this last phase we analysed the work we had done in all the previous sprints. This allowed us to reflect not only on the quality of the implementation but also on the way we worked together as a team. We started this last sprint as always with a meeting. Throughout the past sprints we have learned how important this continuous communication is in order keep everyone up to date. I think that we have all improved our communication skills in the course of this project as we are way more precise now compared to our first meetings.

Overall we were able to achieve the goals we defined in the requirement analysis and I am happy with the resulting product we created. Moreover, I am most proud that we did not only manage to deliver a minimal viable product but we also had time left in the last sprint to drastically improve our implementation. In my opinion, this is the result of great team work, because all of this was only possible because every team member delivered their part reliably. I think their was not that much that did not worked according to plan. Maybe in the first sprint when we started coding we needed to spend more time than we planned on organizing the work distribution. But we let this not stop us from reaching our goal and overcame this. And this really helped us to improve the effectiveness of our communication in the sprints that followed the first one.

When it comes to the thinks I learned, I would like to describe the learning's in more details for each of the sprints.

In the project setup I got to know multiple tools that can be used to work together in larger development teams. Although I already worked in development teams before I found it interesting to take a broader look over all the possible tools that are available.

The requirement engineering phase was one of the most interesting to me. I think this phase is to often skipped in projects. This phase trained me in seeing thinks from the eyes of different stake holders. Organizing the requirements into a clear hierarchy is something that helped us throughout the following implementation sprints.

The first implementation phase showed us that multiple solutions can lead to the same goal and I think it was worth the time to explore the pros and cons of multiple possible solutions.

In the second sprint I improved my programming skills most dominantly with regard to write understandable code. That is something that is more important when working as a team and it helped us to improve the teamwork.

The third and last implementation sprint was very interesting for me. I learned how to refactor an existing codebase. Moreover I had not been using libraries like for example timeit before. I defenitly improved my awareness for performance and memory usage in python due to this sprint.

The final assessment sprint helped me to reflect on the complete project and allowed me to clearly point out the learnings that I had along the way. I found this to be helpfull and I plan to add a phase like that to all my upcoming projects in the future.

All in all I am very happy with the way that this project went. I liked our team and the way we worked together. Although each one of us had different roles, we all worked together and each one was willing to help another when needed. It was fun working with this team on the project and I think we all learned a lot.

6.3.5 Mina Khalid

This sprint marks the conclusion of our project. Like all previous sprints, we had multiple sessions in which the tasks were distributed and everyone discussed the ambiguities so that we were all on a common page before beginning with the execution of assigned tasks. My task in this sprint was to compare our approach with classical PM4Py approach that uses Petri Nets and to evaluate both these approaches on the basis of time and memory used by each. I learnt about the best practices when it comes to testing and how to correctly test specific performance metrics in python.

The first phase of the project was a new experience for me as I had never planned a project of this scale from beginning to end before. The Gantt Chart turned out to be a very useful tool as it kept everyone on their toes by constantly reminding us of what the next steps were and how much time was left.

The requirement gathering phase taught me the different levels of hierarchy of requirements. Until then, I had only ever focused on the either the very technical or the user requirements.

The coding part was started from sprint 1 where we all discussed all the possible approaches to go about our project and initially we came up with the idea of two teams implementing the same task which turned out not to be a very wise decision as it took twice the time and eventually we had to go forward with only one approach which was better of the two. Due to this, we had to go through quite a lot in terms of workload of that sprint. This taught us that instead of reinventing the wheel, the entire team should invest a lot on the planning part but the energies should be invested in achieving the common goal.

Sprint 2 was about alignment computation. Our focus in this sprint was to complete the project in terms of what was required irrespective of performance metrics. This resulted us in losing points which again was a lesson for us that quality is equally important as quanity of work.

Therefore in Sprint 3, we worked on optimization of our approach. Mr. Berti played a vital role in clarifying our ambiguities by addressing our concerns and guiding us as always. We were able to make up to the mistakes done by us in previous sprints so this sprint was more or less a hot fix and we gave final touches to the implementation part of our project.

Overall I have learned a variety of new things while implementing this project. It gave me a deeper understanding of the foundations of conformance checking. I have learned a lot in both practical as well as theoretical aspects of this domain.

Last but not the least, a big shout out to the entire team for keeping their spirits high no matter how much pressure we had to face and Mr. Berti for being more than our supervisor, always leading from the front and helping us out throughout. The entire experience was amazing and I can not emphasize enough on the way everyone has supported each other making every situation easier and fun throughout.

References

[1] Wil van der Aalst. Data Science in Action. Springer Berlin Heidelberg, 2016.

[2] Rabin, M. O. and Scott, D. Finite Automata and Their Decision Problems.IBM Journal of Research and Development, 1959, doi: 10.1147/rd.32.0114

[3] Alfred Vaino Aho; Monica S. Lam; Ravi Sethi; Jeffrey D. Ullman (2007). ”3.7.4 Construction of an NFA from a Regular Expression” (print). Compilers : Principles, Techniques, Tools isbn: 9780321486813

[4] Burge, W.H. (1975). Recursive Programming Techniques. isbn: 0-201-14450-6

[5] R. Rastogi, P. Mondal and K. Agarwal, ”An exhaustive review for infix to postfix conversion with applications and benefits,” 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015, pp. 95-100.

[6] A. Rozinat and W.M.P. van der Aalst. Conformance checking of processes based on monitoring real behavior. Inf. Syst., 33(1):64–95, 2008.

[7] A. Adriansyah, B.F. van Dongen, and W.M.P. van der Aalst. Conformance checking using cost-based fitness analysis. In Proc. of EDOC, pages 55–64. IEEE, 2011.

[8] L. Garc´ıa-Ba˜nuelos, N. van Beest, M. Dumas, M. La Rosa, and W. Mertens. Complete and interpretable conformance checking of business processes. IEEE TSE, 43, 2017. In press.

[9] Reißner D., Conforti R., Dumas M., La Rosa M., Armas-Cervantes A. (2017) Scalable Conformance Checking of Business Processes. In: Panetto H. et al. (eds) On the Move to Meaningful Internet Systems. OTM 2017 Conferences. OTM 2017. Lecture Notes in Computer Science, vol 10573. Springer, Cham. https://doi.org/10.1007/978-3-319-69462-7$_3$8

[10] https://en.wikipedia.org/wiki/Thompson