

REBS – Assignment 2

Albert Friis-Hansen (ghv657), Felix Fleckenstein (vcb456)

Group 3

5. januar 2024

1 Introduction

The objective of this assignment was to implement a conformance checker for DCR Graphs and subsequently evaluate predefined patterns against the Dreyers log. The conformance checker was designed to handle various inputs, including DCR Graphs and logs in CSV format. This report outlines the implementation approach, design decisions, and results obtained.

2 Code description

Because of the issues we had making the conformance checker, we made sure we completed one step at the time when working. First off we split up the Dreyer's log into each individual ID, each representing the list of events, so that our conformance checker later on could go through the rules for each case. After this we were not sure how to define our rules, because we had a hard time parsing the XML file, and therefore we hard-coded the conformance checker, based on the rules we had from assignment 1. We knew this was not sufficient, but we thought that we might as well use them as tests later on.

Rule #	Number of satisfied rules	Number of unsatisfied rules	Total
1	594	0	594
2	305	289	594
3	369	225	594
4	594	0	594
5	591	3	594
6	592	2	594
7	576	18	594
8	592	2	594

Tabel 1: Table of the hard-coded conformance check on the Dreyer's log

Next up we wanted to make sure we had parsed and sorted all necessary data from our XML file. We wanted to make sure that our activities and the names for them aligned. We did this with hash tables, so that they will be easy to modify. Afterwards we made lists for each relationship, for each relationship specification like response, condition etc. Now all our necessary data was parsed and sorted, and we just had to figure out, how to compare the data with the rules specified in the XML file.

We could not get our own engine to work, so we ended up translating and printing the relationship arrows and activities from the XML file representing the DCR graph. This means we can input the printed arrows and activities into the engine made by our professors and from there it will make a conformance check. This means you can give any giving DCR graph in XML form, and a fitting Dreyer's log, and it will translate it, so the output can be used in the engine that we were given.

```

Responses relations:
Fill out application --> Execute abandon
Fill out application --> Lawyer review
Fill out application --> Architect Review
Lawyer review --> Approve
Lawyer review --> Reject
Architect Review --> Approve
Architect Review --> Reject
Reject --> Applicant informed
Reject --> Change phase to aborted
Approve --> Account number changed
Execute abandon --> Change phase to abandon
Receive end report --> Change phase to completed
Account number changed --> Approve changed account
Approve changed account --> Change phase to payout
Change phase to payout --> First payment
First payment --> Change phase to end report
Change phase to end report --> First payment
Change phase to end report --> Receive end report

Excludes relations:
Execute abandon --% Fill out application
Lawyer review --% Fill out application
Architect Review --% Fill out application
...
First payment -->* Undo Payment
Fill out application -->* Execute abandon
Fill out application -->* Lawyer review
Fill out application -->* Architect Review
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```

Figure 1: Output from our program that writes the DCR graph from the XML model as readable for the JS engine to run it with a Dreyer's log

```

Conformance Check Results:
Rule 1: Satisfied 594, Not Satisfied 0, Total 594
Rule 2: Satisfied 305, Not Satisfied 289, Total 594
Rule 3: Satisfied 369, Not Satisfied 225, Total 594
Rule 4: Satisfied 594, Not Satisfied 0, Total 594
Rule 5: Satisfied 591, Not Satisfied 3, Total 594
Rule 6: Satisfied 592, Not Satisfied 2, Total 594
Rule 7: Satisfied 576, Not Satisfied 18, Total 594
Rule 8: Satisfied 592, Not Satisfied 2, Total 594

```

Figure 2: Output from checking the rules against the Dreyer's log.

3 Handtracing the first ID

Running the first ID parser by hand:

1 means that we've defined rules for this activity

0 means that we haven't implemented any rules for this activity

I will assume all activities without rules are valid, since this is an open world DCR-graph.

Fill out application 1 - Valid

Approved to board 0 - Valid

Change phase to Review 0 - Valid

Architect Review 1 - Valid

Architect Review 1 - Valid

Review 0 - Valid

Review 0 - Valid

Architect Review 1 - Valid

Review 0 - Valid

Review 0 - Valid

Lawyer Review 1 - Invalid

(We can't have Lawyer review after Architect review since they exclude each other)

Register Decision 0 - Valid

Change phase to Board meeting 0 - Valid

Round ends 0 - Valid

Round approved 0 - Valid

Inform application of board review 0 - Valid

Round approved 0 - Valid

Reject 1 - Valid

Applicant informed 1 - Valid

Change phase to Abort 1 - Valid

In this case we see that both Lawyer review and Architect review is executed, which is against rule 2. Other than that, all traces are valid in this ID.