A Tool for the Analysis of DMN Decision Tables

Ülari Laurson and Fabrizio Maria Maggi

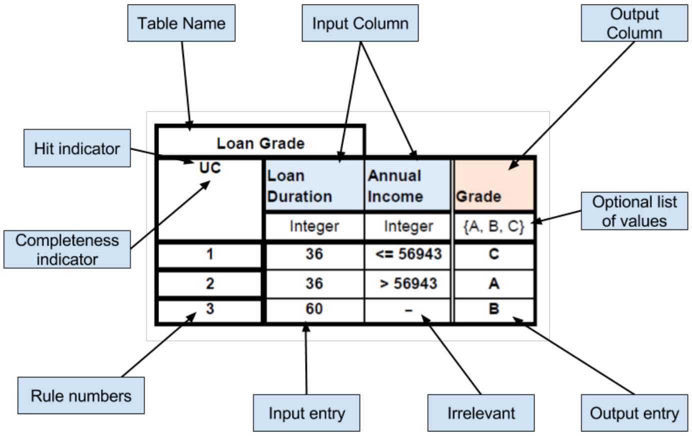
University of Tartu, Estonia
{ulaurson,f.m.maggi}@ut.ee

Abstract. The Decision Model and Notation (DMN) is a standard notation to specify decision logic in business applications. A central construct in DMN is a decision table. The rising use of DMN decision tables to capture and to automate everyday business decisions fuels the need to support analysis tasks on decision tables. This paper presents an open-source DMN editor to tackle three analysis tasks: detection of overlapping rules, detection of missing rules and simplification of decision tables via rule merging. The tool has been tested on large decision tables derived from a credit lending data-set.

Keywords. Decision Table, Decision Model and Notation, Camunda

1 Introduction

In 2014, the Object Management Group (OMG) published the Decision Model and Notation (DMN) standard [2]. The primary goal of this standard is to pro-



Loan Grade			
UC	Loan Duration	Annual Income	Grade
	Integer	Integer	{A, B, C}
1	36	<= 56943	C
2	36	> 56943	A
3	60	–	B

Fig. 1: Sample decision table with its elements

vide a common notation that is readily understandable for all business users including business managers, analysts, and developers. DMN consists of two levels: a "decision requirements level" and a "decision logic level". The decision requirements level defines input data needed to make a decision. The decision logic level describes how each decision is made. Decisions are usually expressed as decision tables. In decision tables, columns represent inputs and outputs of a decision, and rows represent rules. Columns typically are typed so that they have an associated domain. A rule is a conjunction of basic expressions (one basic expression per cell). Basic expressions are captured using a language known as S-Feel (Simplified Friendly Enough Expression Language).

In Figure 1, we see an example of a decision table. The rules in the decision table should be interpreted as follows:

1. If the customer asks for a loan duration of 36 and her annual income is less than or equal to 56943, then she gets grade "C."
2. If the customer asks for a loan duration of 36 and her annual income is greater than 56943, then she gets grade "A."
3. If the customer asks for a loan duration of 60, then she gets grade "B."

Using DMN decision tables for making critical business decisions raises the question of ensuring their correctness and simplicity to prevent costly defects. The tool presented in this paper implements scalable algorithms for two basic correctness checking tasks over DMN tables and one algorithm for simplification task [1]. The two correctness checking tasks are the detection of overlapping rules and the detection of missing rules. The detection of missing rules allows the user to check the completeness of the table, while the detection of overlapping rules allows the user to identify: (i) inconsistent rules, meaning two rules that overlap and that are associated with two different outputs; and (ii) redundant rules, meaning two rules that overlap and that have the same output.

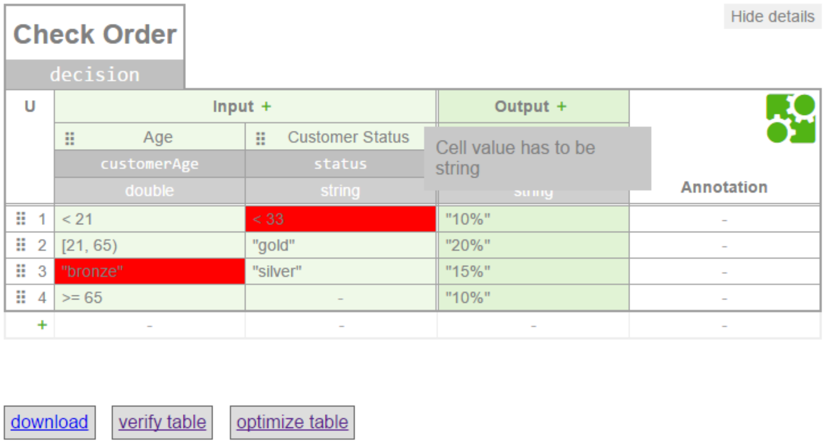
2 Tool

We implemented our tool starting from dmn-js, the open-source rendering and editing toolkit of Camunda DMN.[2] We extended dmn-js to support correctness verification and simplification. Our tool can be found at https://github.com/ulaurson/dmn-js and a deployed version for testing is available at http://dmn.cs.ut.ee/. A screencast of the tool demo can be found at https://www.dropbox.com/s/nc0vibw2jlv0p3l/DMN_Screencast.mp4?dl=0.

The tool provides a syntactic check that verifies whether each cell of a decision table has the right content. The implemented algorithm checks if the content of each cell matches its column type.[3] If not, the cell becomes red and a tooltip is added. In Figure 2, we see two syntactic errors. In the third rule first column we have a syntactic error because this cell should contain a double but it contains a string. Another syntactic error is in the first rule second column. This cell should contain a string, but it contains an integer.

[2] https://camunda.org/

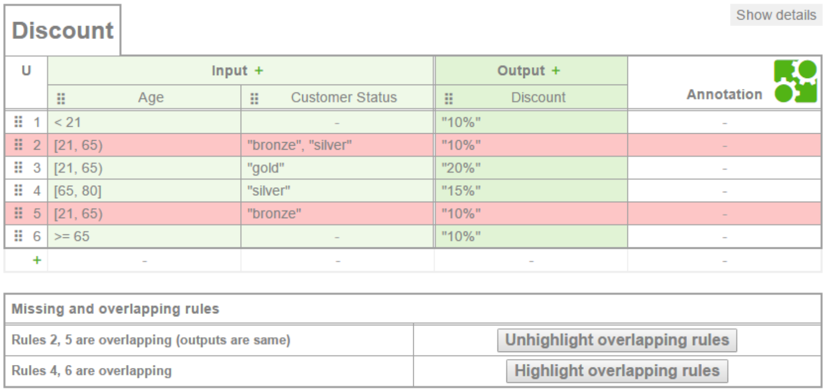
[3] Our tool supports types integer, string, Boolean, and double.



Cell value has to be string

U	Input +		Output +	Annotation
	Age customerAge double	Customer Status status string	string	
1	< 21	< 33	"10%"	-
2	[21, 65)	"gold"	"20%"	-
3	"bronze"	"silver"	"15%"	-
4	>= 65	-	"10%"	-
+	-	-	-	-

Fig. 2: dmn-js decision table with syntactic errors



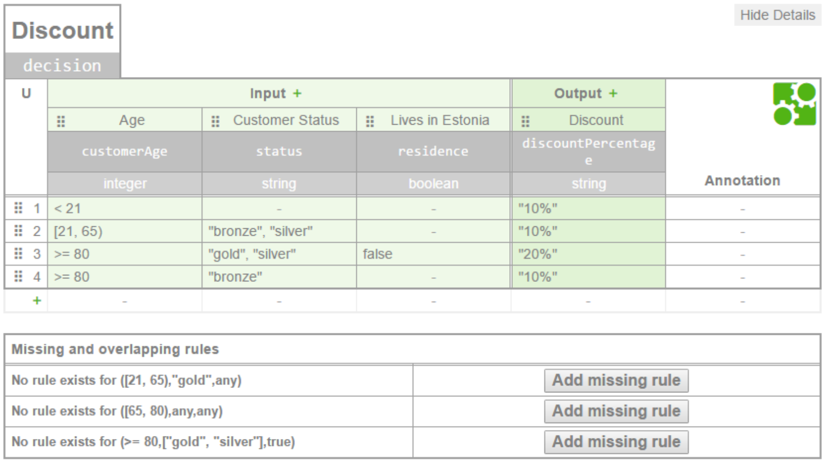
U	Input +		Output +	Annotation
	Age	Customer Status	Discount	
1	< 21	-	"10%"	-
2	[21, 65)	"bronze", "silver"	"10%"	-
3	[21, 65)	"gold"	"20%"	-
4	[65, 80]	"silver"	"15%"	-
5	[21, 65)	"bronze"	"10%"	-
6	>= 65	-	"10%"	-
+	-	-	-	-

Missing and overlapping rules	
Rules 2, 5 are overlapping (outputs are same)	Unhighlight overlapping rules
Rules 4, 6 are overlapping	Highlight overlapping rules

Fig. 3: dmn-js decision table with overlapping rules

When clicking verify table (see Figure 2), the tool finds all the overlaps in the decision table. The tool can find two types of overlaps: overlaps where inputs and outputs are the same and overlaps where inputs are the same and at least one output is different. In Figure 3, we see that the tool has found two overlaps. Rules 2 and 5, and rules 4 and 6 have overlaps. The overlapping rules are summarized in the "Missing and overlapping rules" table. This table contains maximal sets of overlapping rules with a non-empty intersection. If "(outputs are the same)" is explicitly mentioned in a set of overlapping rules, then these rules have the same inputs and outputs, otherwise they have different outputs. The overlapping rules are highlighted in light red.

When clicking verify table, the tool also finds the missing rules in the decision table. In Figure 4, the decision table has three missing rules. The first value inside



U	Input +			Output +	
	Age	Customer Status	Lives in Estonia	Discount	
	customerAge	status	residence	discountPercentage	
	integer	string	boolean	string	Annotation
1	< 21	-	-	"10%"	-
2	[21, 65)	"bronze", "silver"	-	"10%"	-
3	>= 80	"gold", "silver"	false	"20%"	-
4	>= 80	"bronze"	-	"10%"	-
+	-	-	-	-	-

Missing and overlapping rules	
No rule exists for ([21, 65),"gold",any)	Add missing rule
No rule exists for ([65, 80),any,any)	Add missing rule
No rule exists for (>= 80,["gold", "silver"],true)	Add missing rule

Fig. 4: dmn-js decision table with missing rules

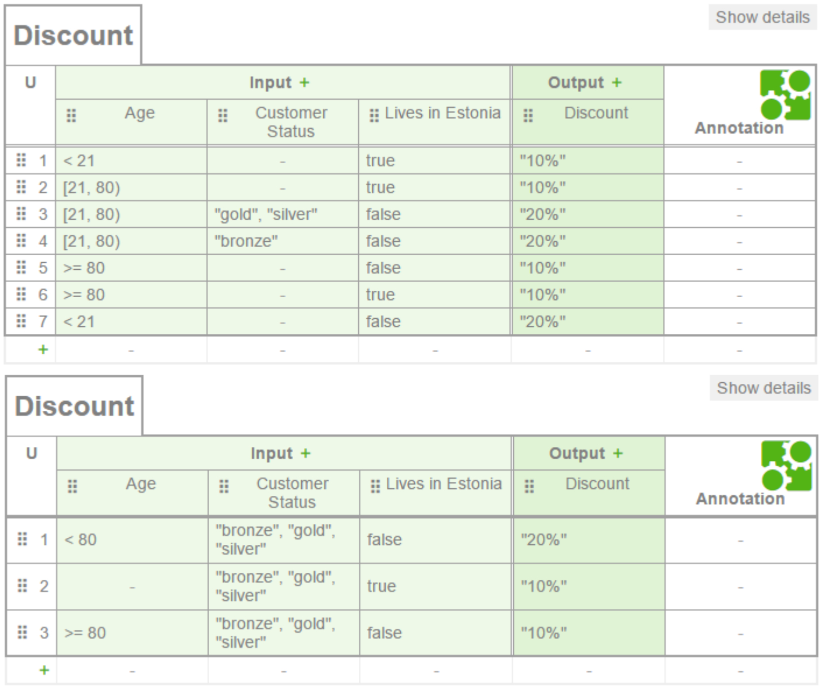
the brackets shows what is missing in the first input column, the second value shows, what is missing in the second input column and so on. In Figure 4, we can see that there are three types of missing rules. In the first case, a categorical value is missing. Corresponding to interval [21, 65) there is no rule with "Customer Status" = "gold". In the second case, an interval is missing. In the decision table, there is no rule where "Age" = [65, 80). The third case has a missing Boolean value. With "Age" >= 80 and "Customer Status" = "gold" or "silver" there is no rule with "Lives in Estonia" = true. The missing rules are summarized in the "Missing and overlapping rules" table. The first column of this table contains information about the missing rules. The second column contains a button for adding the missing rules into the table.

When clicking simplify table (see Figure 2), the tool simplifies the decision table via rule merging. In Figure 5, the first decision table is a regular decision table that has no overlapping rules and no missing rules. The second decision table represents the simplified version of the first table. Rules 1, 2, and 6 can be merged because these rules only differ in one input (the outputs are the same). Rules 3, 4, and 7 can also be merged. Rules 1 and 3 in the second table cannot be merged because their outputs are different.

3 Maturity and Inherence

Based on the tool implementation, we have conducted empirical evaluations to compare the proposed algorithms with respect to existing approaches (in particular the one implemented in Signavio[4]) in terms of scalability, conciseness of the feedback provided to users (in case of overlapping and missing rules) and compactness of the simplified tables. The tests have been conducted on large decision tables derived from a credit lending data-set.

[4] http://www.signavio.com



U	Input +			Output +	Annotation
	Age	Customer Status	Lives in Estonia	Discount	
1	< 21	-	true	"10%"	-
2	[21, 80)	-	true	"10%"	-
3	[21, 80)	"gold", "silver"	false	"20%"	-
4	[21, 80)	"bronze"	false	"20%"	-
5	>= 80	-	false	"10%"	-
6	>= 80	-	true	"10%"	-
7	< 21	-	false	"20%"	-
+	-	-	-	-	-

U	Input +			Output +	Annotation
	Age	Customer Status	Lives in Estonia	Discount	
1	< 80	"bronze", "gold", "silver"	false	"20%"	-
2	-	"bronze", "gold", "silver"	true	"10%"	-
3	>= 80	"bronze", "gold", "silver"	false	"10%"	-
+	-	-	-	-	-

Fig. 5: Decision table before and after simplification

Execution times for missing rules detection are under 2 seconds. The detection of overlapping rules leads to higher execution times, due to the need to detect sets of overlapping rules and ensure maximality. In Signavio, if multiple rules have a joint intersection (e.g., rules {r1, r2, r3}) the output contains an overlap entry for the triplet {r1, r2, r3} but also for the pairs {r1, r2}, {r2, r3} and {r1, r3} (i.e., subsets of the overlapping set). Furthermore, in some cases, the overlap of pair {r1, r2} may be reported multiple times (and same for {r2, r3} and {r1, r3}). Meanwhile, our approach produces only maximal sets of overlapping rules with a non-empty intersection. Therefore, the number of sets of overlapping rules and the number of missing rules identified by our approach is drastically lower than the number of overlapping and missing rules identified by Signavio. For more information about our experimentation of the tool, the reader is referred to [1].

References

1. Calvanese, D., Dumas, M., Laurson, Ü., Maggi, F.M., Montali, M., Teinemaa, I.: Semantics and analysis of DMN decision tables. In Proceedings of the 14th International Conference on Business Process Management (BPM) 2016
2. Object Management Group: Decision Model and Notation (DMN) 1.0 (2015), http://www.omg.org/spec/DMN/1.0/