Root Cause Analysis in Insurance Claim Processing

Working Draft

Tobias Trapp
AOK Systems GmbH
Breitlacher Str. 90
60489 Frankfurt

January 1, 2016

Abstract

Insurance companies use rule systems for automated checks of insurance claims. Many of these rule systems can be modeled and implemented using decision tables. In this paper we analyze an existing rule system and develop a model using propositional logic. It shows that root cause analysis can be performed with a variant of propositional abduction. In this approach we use rectification for explaining a rule system calculation.

1 Introduction

Business rules management systems are software systems for definition, deployment, execution and monitoring decision logic that is used by operational systems within an enterprise ([16]). Business rules are used for classifications ([10]), complex calculations and determinations and also for checks of business objects in automated processes.

Often business requirements define fine-grained checks for business objects. In statutory health insurance in Germany certain rule systems are quite comprehensive and specified in about 300 pages. If business rules are violated often officials have to control the data set according to the output of the rule system. This process can be supported in different ways: usually messages of the ruleset outcome are prioritized. In this paper we try out a different approach by searching the root cause for calculations of the rule system. Looking for root causes is well understood in abductive logic which looks for "best explanations". This method of non-monotonic reasoning has been intensively studied by Peirce (see [12]) and introduced into modern logic. The special case of propositional abduction is well understood (see [11] for current results).

In this section we start with a simple example of rule system and provide a propositional model in section 2. In section 3 we will give a formal definition of the root cause analysis problem in the paper and compare it with other defintions from literature. In section 4 we present a case study from a rules system in statutory health insurance.

Many business rules management systems like ARIS, ILOG-JRules and BRFplus (see [1]) offer decison tables which are widely used to define business rules in a precise and compact way that is well understood by business people and can be easily maintained using spreadsheet software. They are known since the 50s of the last century (see [13] and [15]). Both books contain formal definitions of decision tables which will be omitted here. In this paper we will consider multiple hit decision tables where each row is evaluated and an action is triggered if the corresponding logical expression is fulfilled.

Decision tables consist of rows and columns where each row represents a rule. Each column represents a property of an business object like an insurance claim checked with that decison table. A typical property is the number of procuments of type A1 in an insurance claim for example. Each cell of the decion table consists of a comparison of that property and a rational number. A cell can be empty which means the conditions always evaluates to true.

We present a simple example of a rule system containing two checks C_1 and C_2:

number of procurements of type A1	number of procurements of type A2	patient's age	check
> 2	> 5		C_1
> 10		> 12	C_2

C_1 evalutates to true if and only if the number of procurements of type A1 is greater than 2 and the number of procurements of type A2 is greater than 5. C_2 evalutates to true if and only if the number of procurements of type A1 is greater than 10 and patient's age is greater than 12.

In business rule systems like BRFplus the columns of a decision can be arbitrary expressions. The case study in section 4 shows that the restriction to simple comparison is reasonable. The reason why we consider BRFplus is that BRFplus is the strategic BRMS solution of SAP and the rule system in statutary health insurance which are analyzed in this paper are developed on the SAP platform.

Let us now consider the case a business entity is checked using a rule system and we get a set of checks which evaluated to true. In the following we will focus on the task of finding the root cause for the outcome of the rule system by using propositional logic.

2 Modeling rule systems using propositional logic

At first we define some common basic notations in the area of propositional logic. Let $\mathcal{L}_V$ be the language of propositional logic over an alphabet V of propositional variables with the usual syntactic operators $\vee$, $\wedge$ and $\neg$, moreover $\oplus$ means exclusive-or. A *theory* is a finite set $T \subseteq \mathcal{L}_V$ which is the conjunction of its expressions for reasons of convenience. A *literal* is an expression v (*positive literal*) or $\neg v$ (*negative literal*) for a propositional variable v. A *clause* is a disjunction of pairwise disjoint literals. We denote $\top$ with *truth value* true and

$\bot$ with false. An *assignment* of a propositional formula is a function from a set of propositional variables of that expression to $\{\top, \bot\}$. This assignment defines an *interpretation* that assigns the expression a truth value. Given a expression φ we denote with $Lits(\varphi)$ the set of literals of φ. For an expression $\varphi \to \psi$ we call φ *antecedent* and ψ *consequent.* A disjunction of with at most one negated literal is called a *Horn clause.* A Horn clause with exactly one positive literal is a *definite clause.* A disjunction of literals with at most one negated literal is called a dual-Horn clause. Sometimes we use *implication form* where a definite clause φ has the form $\psi \to v$ where $Lits(\varphi)$ are positive.

In the following we show a very simple example for a propositional rule set. This approach is quite common since logical expressions without equality can often transferred into propositional logic which makes computations easier. We will omit the procedure and jump right into an example. Let v_1, v_2, v_3, v_4 and v_5 be the following propositional variables with following meaning:

$v_1 :=$ "number of procurements of type A1 is greater than 2"

$v_2 :=$ "number of procurements of type A2 is greater than 5"

$v_3 :=$ "number of procurements of type A1 is greater than 10"

$v_4 :=$ "patient's age is above 12"

$v_5 :=$ "treatment type B is contained in an insurance claim"

A rule system consists of propositional expressions that describe checks of anomalies like the following which define rules for other propositional variables m_1, m_2, and m_3 called *manifestations*:

$$v_1 \wedge v_2 \to m_1 \qquad (C_1)$$

$$v_2 \wedge v_4 \to m_2 \qquad (C_2)$$

$$v_3 \oplus v_5 \to m_3 \qquad (C_3)$$

The theory $\{C_1, C_2\}$ is exactly a propositional version of the decision table in section 1. The expression C_3 does not come from propositional models of decision tables and is a generalization we also take into account.

In abductive logic we are interested in finding root causes. Therefore we introduce hypotheses and search for a subset of hypotheses which is consistent with the knowledge base so that hypotheses together with the knowledge base entail the manifestation. When working with decision tables the definition of hypotheses is quite canonic if we look at a specific instance of a rule system. We continue the example from above. The rule system checks an insurance claim with the following properties:

$h_1 :=$ "number of procurements of type A1 is 15"

$h_2 :=$ "number of procurements of type A2 is 10"

$h_3 :=$ "patient's age is 18"

$h_4 :=$ "treatment type C is contained in an insurance claim"

For those hypotheses the propositional variables v_1, v_2, v_3, and v_4, are $\top$ and v_5 is $\bot$. If we apply the checks C_1, C_2 and C_3 to them and obtain manifestantions m_1, m_2 and m_3. Now we give a formal definition for rule systems coming from decision tables.

Definition 1 (Conjunctive Rules). *Let V and M be sets of propositional variables and T a theory with $Lits(T) = V$ then (V, M, T) is called conjunctive rule instance if the following conditions hold:*

- *$H \cup M = V$ and $H \cap M = \emptyset$,*
- *T consists of definite Horn clauses $(\bigwedge h_i) \rightarrow m$ with $h_i \in H$ and $v \in V$. and definite dual Horn clauses $h_1 \rightarrow h_2$ with $h_1, h_2 \in H$.*

When the T contains also expressions $(\bigoplus h_i) \rightarrow m$ for $h_i \in V$ and $m \in M$, we speak of an instance with exclusive disjunction-rules.

3 Propositional abduction and rectification

According to [5] we define the propositional abduction and its solution as follows:

Definition 2 (PAP). *A propositional abduction problem PAP consists of a tuple (V, H, M, T) where V is a finite set of propositional variables, $H \subseteq V$ is a set of hypotheses, $M \subseteq V$ is a set of manifestations, and $T \subset \mathcal{L}_V$ is a consistent theory.*

Definition 3 (Solution of a PAP). *Let $\mathcal{P} = (V, H, M, T)$ be a PAP. $S \subseteq H$ is a solution or explanation to $\mathcal{P}$ if and only if $T \cup S$ is consistent and $T \cup S \models M$.*

We denote with $Sol(\mathcal{P})$ the set of solutions of a PAP. $Sol_{\preceq}(\mathcal{P})$ is defined as the set of minimal solutions due to subset inclusion and $Sol_{\leq}(\mathcal{P})$ is the set of minimum solutions due to cardinality.

In the following we mention some complexity results of propositional abduction. See [9] for example for an introduction into complexity theory. In [5] it is proven, that deciding whether a propositional abduction problem (V, H, M, T) has a solution is Σ_2^P-complete even if $H \cup M = V$ and T is in clausal form. Please note that according [5] neither $H \cap M = \emptyset$ nor $H \cup M = V$ is a restriction from the point of view of computational complexity. Computing a $\leq$-solution for a propositional abduction problem is NP-hard if the knowledge base is definite Horn. If the knowledge base is definite Horn and hypotheses are positive literals and the manifistation is give by a positive term we have polynomial complexity. In [11] the results are generalized where the manifestation is a general propositional expression instead of a conjunction of variables. The results are proven for other sets of propositional formulae related to boolean functions in Post's lattice (see [3] and [4]).

Applying the PAP to a set of conjunctive rules is straight forward:

1. An insurance claim that is checked by a rule systems has certain properties. Those properties correspond to a assignment of propositional variables H.
2. The theory T represents the rules which evaluate to $\top$. The set M is defined as the union of the consequents of those rules.
3. The set of propositional variables V is defined as $H \cup M$.

Please remark that $H \cup T \models M$ hold. Since consist of Horn clauses PAP with conjunctive rules can solved efficiently.

Proposition 1. *Let $\mathcal{P} = (V, H, M, T)$ be PAP with conjunctive rules for a decision table instances. Then P is solvable in polynomial time.*

Proof. The set of clauses T consists of Horn clauses and in [5] it is proven in polynomial time. $\square$

It is easy to see that the same is true if the PAP for a decision table instance contains also exclusive disjunction-rules.

It turns out that this abduction problem is not suitable in the case studied here. The reason is due to conjunctive rules: every property, that is part of a rule leading to a manifestation, is part of the solution of the abduction problem. As a consequence the solution is in general much too big. Therefore another approach is chosen here: instead of looking of minimum model we ask, which kind of hypotheses have to be negated so that no rule is valid. This corresponds to the following intuition: if we explain the results of a rule system that looks for conspicous features of an insurance claim, we have to find out, which properties have to be changed or rectified so that the insurance claim passes all checks.

Definition 4 (RECTIFICATION)**.** *Given an instance $\mathcal{P} = (V, H, M, T)$ of PAP, a rectification is a minimum set $R \subseteq H$ with the following property: the assignment, which sets all $r \in R$ to $\bot$ and all $h \in H \backslash R$ to $\top$, is an interpretation that sets all terms of T to $\bot$ and so all manifestations.*

The solution of the rectification problem is the smallest set of hypotheses whose removal from the set of hypotheses is not a model of the manifestations. This is motivated due to the fact that business rules systems are applied to an insurance claim and we want to find out which properties have to be removed, so that the rule systems does not show conspicuous features. These conspicuous features represent checks occurring as manifestations that we want to explain. Instead of explaining these with the traditional methods of abduction, we look how the insurance claim must be altered so that manifestations vanish, so that the propositional expressions representing the checks turn from true to false which means that there are no more conspicuous features.

When we continue the example from section 2, we get the those following result:

- The solution of the minimum PAP defined by the checks C_1 and C_2 is $\{h_1, h_2, h_3\}$.
- The set $\{h_1, h_3\}$ is the solution of the rectification problem since together with the knowledge base it entails the manifestations $\{m_1, m_2\}$. Moveover this is a rectification with minimum cardinality.

Minimum solutions of the rectification problem are of special interest since they indicate a small set of hypotheses that should be changed to achieve a valid instance.

Proposition 2. *Finding a minimum solution of the rectification problem is NP-hard even for conjunctive rules.*

Proof. Let $\mathcal{P} = (V, H, M, T)$ be an PAP for a rule system instance. Finding a rectification is in NP since for given $R \subseteq H$ we can check in polynomial time that the expressions of T are set to $\perp$ by the assignment in the definition in the rectification.

We show NP-hardness by performing a reduction from the hitting set problem (see [9]). Therefore we look at the special case where $\|H\| = \|M\|$ and T consists of clauses C_i with $C_i \rightarrow m$ for $m \in M$, where all C_i consists of logical conjunctions. Finding a rectification with size $k \leq \|H\|$ is equivalent to the hitting set problem. This can be seen easily since we have to choose a set with cardinality k of positive literals of the manifestations, that turn every antecedent of a formula in T to false when assigning all variables in R to false. □

Abduction using set covering has been described in literature many times (see [14] for example). The application here is different due to the different definition of root cause analysis.

4 Case study

In statutory health insurance in Germany orthodontical treatments are checked in automated processes using rule systems. Per month more than 20.000 data sets have to processed and many of them are conspicuous and rule sets generate between 20 and 80 actions where each corresponds to a certain business rule that is violated. Every rule within the ruleset can be modeled using propositional expressions. 95% of all rules can be expressed as decision table that corresponds to a boolean formula consisting only of and-clauses. If we allow additional exclusive disjunction clauses, 98% of rules can expressed in the framework presented in this paper. The remaining rules are extraordinary complex, and others need special workflows to check them.

The rule system was implemented with a decision table with 89 rows and 124 columns. Each row represents a rule and each column a property of an insurance claim. In the following we look at the bipartite incidence graph of columns and rows of the decision table. This graph consists of 23 connected components and is sparse since it has 213 vertices and 233 edges. This means that the rule set is decomposed into different parts and abduction as well as restriction problems can be calculated for the parts.

The average degree of the clause-literal incidence graph is 2.188. The degree distribution is seen in figure 1. Those properties of the graph have been calculated using Gephi software 0.8.2 beta (see [2]). The main component consists of 137 nodes, 172 edges and has treewidth 3 which is also the treewidth of the graph. This property have been calculated using the libTW software package (see [8]).

To test whether rectification is reasonable for finding explanations of the rule system outcome we created a number of random instances and applied root cause analysis. The results are shown in the following table:

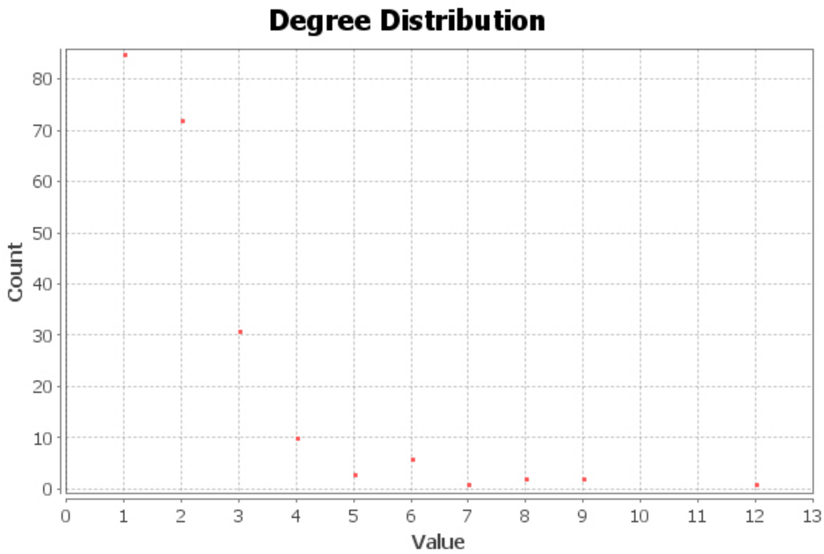

Figure 1: Distribution of vertex degrees

checks	manifest-ations	hypo-theses	size of abductive explanation	size of rectification
17	30	17	14	5
8	14	14	11	11
11	16	8	7	5
6	10	7	7	3
3	5	7	6	3

In 4 of 5 cases root cause analysis reduced the number hypotheses compared to an abductive explanation. Even in the case that where the number of hypotheses could not be reduced, shows that there are instances that cannot be corrected with a few changes. They contain many errors and should be investigated in depth.

It turned out that a greedy approach solved the approach optimally in all cases.

5 Conclusion and open questions

Abductive inference has many applications and system diagnosis is one of them. It seems that up to now there has been no reseach of the outcome of business rule systems with abductive logic although it is an obvious use case. In this paper a rule system implemented using decison tables has been studied. Due to its simple structure their outcome can be analyzed effectively using a variant of propositional abduction. First experiments showed that this approach is promising. This approach should be validated using real world data and business process experts should review the results. Moreover penalization and priorization like in [5] in algorithm for rectification should be investigated.

Rectification is a promising approach for systems analysis when the set of hypotheses is quite big, is already a model of the manifestations and are hypotheses are needed for an explanation. This is the case when explaining the results of a rule systems created with decision tables. Instead of looking for small models we are looking for models as explanations, that don't have certain, unwanted properties.

Although the rectification problem defined in this paper is NP-hard, in the case study a greedy algorithm that chooses the hypotheses with most implications always found an optimum solution. There is the open question whether this approach holds due to the structure of the conjunctive rules obtained from a decision table. In general the computational complexity of abduction is very high, but it has been remarked and reported in literature that in real-world problems abduction is often tractable. For solving general rectification problems one has to find efficient algorithms. Kernelization ([6]) is promising since the problem is related to set covering and the propositional model of the rule systems allows preprocessing.

The case study showed that the treewidth of the graphs associated to the rule systems is low, although bipartite graphs have no limited treewidth in general. The low treewidth is interesting since algorithms based on tree decompositions ([7]) may lead to promising results in algorithm design.

References

[1] Thomas Albrecht, Matthias Kucharska, Christian Lechner, Daniel Ridder, Wolfgang Schaper, Tobias Trapp and Carsten Ziegler, Business Rule Management mit ABAP, Rheinwerk Verlag, 2015

[2] Mathieu Bastian, Sebastien Heymann, Sebastien and Jacomy Mathieu, Gephi, An Open Source Software for Exploring and Manipulating Networks", AAAI Publications, Third International AAAI Conference on Weblogs and Social Media, 2009

[3] E. Böhler, N. Creignou, S. Reith, and H. Vollmer, Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory. ACM SIGACT-Newsletter, 34(4), pp 38-52, 2003

[4] E. Böhler, N. Creignou, S. Reith, and H. Vollmer, Playing with Boolean blocks, part II: Constraint satisfaction problems. ACM SIGACT-Newsletter, 35(1):22-35, 2004

[5] Thomas Eiter and Georg Gottlob, The complexity of logic-based abduction, Journal of the ACM (JACM), Volume 42 Issue 1, pp 3-42, Jan. 1995

[6] Jörg Flum and Martin Grohe, Parameterized complexity theory, Texts in Theoretical Computer Science. An EATCS Series, Springer, New York, 2006.

[7] Georg Gottlob, Reinhard Pichler and Fang Wei, Bounded treewidth as a key to tractability of knowledge representation and reasoning, Artificial Intelligence, Volume 174, Issue 1, January 2010, pp 105-132

[8] Thomas van Dijk and Jan-Pieter van den Heuvel, Computing treewidth with LibTW, 2006

[9] Richard M. Karp, Reducibility Among Combinatorial Problems. In: Raymond E. Miller, James W. Thatcher (Edts.): Complexity of Computer Computations, Plenum Press, New York NY, 1972

[10] Jann Müller and Tobias Trapp, Intelligent Decision Technologies: Proceedings of the 7th KES International Conference on Intelligent Decision Technologies (KES-IDT 2015), Edts. Rui Neves-Silva, Lakhmi C. Jain, Robert J. Howlett, Springer, 2015

[11] Gustav Nordh and Bruno Zanuttini, What makes propositional abduction tractable, Artificial Intelligence, Volume 172, Issue 10, June 2008, pp 1245-1284

[12] C. S. Peirce, Abduction and induction, in J. Buchler, editor, Philosophical Wirtings of Peirce, chapter 11, Dover, New York, 1955

[13] Solomon L. Pollack, Decision Tables: Theory and Practice, Wily 1971

[14] James A. Reggia, Dana S. Nau and Pearl Y. Wang, Diagnostic expert systems based on a set covering model, Int. J. Man-Machine Studies 19, 1983, pp 437-460

[15] Horst Strunz, Entscheidungstabellentechnik, Hanser 1977

[16] James Taylor, Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics, Boston MA: Pearson Education, 2012