Mining Decision to Discover the Relation of Rules among Decision Points in a Non-Free Choice Construct

Riyanarto Sarno, Putu Linda Indita Sari, Dwi Sunaryono, Bilqis Amaliah
Informatics Departmentand
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
riyanarto@if.its.ac.id, dwi@its-sby.edu, bilqis@its-sby.edu

Imam Mukhlash
Mathematics Department
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia

***Abstract*—Decision mining is a combination of process mining and machine learning algorithms to retrieve information on how data attributes in a business process affect routing of a case. It analyzes decision point by looking for XOR-splits in petri-net workflow model and examining rules for each choice based on available attributes using decision tree. The rules for each decision point are based on the attribute's influence to the case. Meanwhile, a non-free choice construct is a mixture of choice and synchronization, which will create limited choices in the workflow. The limitation of choice will then affect the rules found in non-free choice construct using decision mining technique. Limitation of rules makes it possible to examine the relation among rules in the workflow. The relation of these rules will emerge a certain property of a non-free choice construct. Rules for two decision points within a non-free choice construct will have similarities. Regarding to this, when the same rule is found during a decision mining process, we can determine that the decision points have a non-free choice relationship**

Keywords—decision mining, decision tree, non-free choice, process mining

I. INTRODUCTION

Decision mining is a technique to retrieve rules for choices in a business process. It combines process mining and machine learning algorithms to analyze a Process Aware Information System's event log and discover how the attributes in the process affects routing of a case [8,9]. The decision mining algorithm has been applied as a plugin in ProM framework named Decision Point Analysis. It will first discover the decision points in a process model by identifying its XOR-split. Since XOR-split is a place in Petri-net which has more than one outgoing arcs, the split will then be identified as the decision point [12]. In the meantime, another split type called the AND-split indicates all choices taken. Therefore, it cannot be identified as decision point.

A non-free choice is a situation where there is a mixture of choice and synchronization [13]. The choice and synchronization are not separated and will create implicit dependencies. Some discovery algorithms still cannot detect the non-free choice construct, while other algorithms have already been improved to detect non-free choice. Thus, it is possible that a process model that in reality has the property of non-free choice is not detected as a non-free choice construct. In this case, we can use the relationship of rules between decision points in a non-free choice workflow to identify whether it is a non-free choice or an ordinary exclusive choice workflow.

In this paper, we will use decision mining to find the relationship of rules among decision points in a non-free choice workflow. The relationship of rules in a limited choice can be used to identify a non-free choice construct if the process model does not show the non-free choice, for example, if the model is mined using alpha algorithm. Decision mining method which is conducted to find non-free choices can be applied only if the decision points used in the non-free choices are found in the process model.

II. RELATED WORK

A. Decision Mining

Decision mining is a method to learn how the value of attributes in a business process can affect routing of a case. Decision mining uses a combination of process mining to deal with Process Aware Information System's event log and machine learning to discover the rule for each decision point in process model. Decision mining method used in this paper is similar to the one presented by [8,9]. Decision mining generally consists of two phases: the first one is the discovery of decision points from process model; and, the second one is a construction of decision tree. Rules for routing in each decision point can be found using decision tree.

In a decision mining process, the decision points are discovered by identifying conditional routing in Petri-net model. In this case, all XOR-split will be identified as the decision points. Therefore, it is highly recommended to find a proper model needs before doing decision mining. Should the model not show conditional routing as proposed, the result of decision mining might be unsatisfying. After all the decision points are identified, decision tree can be built for each

decision point. Decision tree will be built using values of attributes in event log.

A decision tree will identify the most affecting variables for classification of data records. The most affecting variable will become the root of the tree, and continued by other variables as nodes until reaching leaves where the classification is concluded. Each variable on decision tree's nodes are chosen based on information gain. The variable with the largest gain will be chosen.

B. *Non-Free Choice*

Non-free choice is a mixture of synchronization and choice, where the synchronization and choice are not separated and may create implicit dependencies [13]. The implicit dependencies make the remaining choice become a pseudo choice, because the resulting choice is dependent to the previous choice. The implicit dependencies are caused by the extra place connecting two choices from different decision point. Fig. 2 illustrate P3 and P4 as the places which cause implicit dependencies to the workflow. Concerning to this, if T1 is chosen, T4 will also be executed. But, if T1 is executed, T5 cannot be chosen. The situation creates a non-free choice relationship in this workflow.

III. Rule Relationship in Non-Free Choice Construct

A non-free choice construct has a limited series of choices even if there are more than one decision points in the construction. Fig. 3 demonstrates that activity E has implicit dependencies to activity B, while activity F has implicit dependencies to activity C. In other words, if in decision point P2 activity B is chosen, activity E will be executed; however, activity F will not be executed. When activity C is chosen, activity F will also be executed. In this case, activity D will always be chosen. If activity A has attribute X, then the value of X can be used to construct a decision tree and to find rules for both decision points. Since there is only one variable, gaining computation to select the root is not essential. X will be the root for the decision tree.

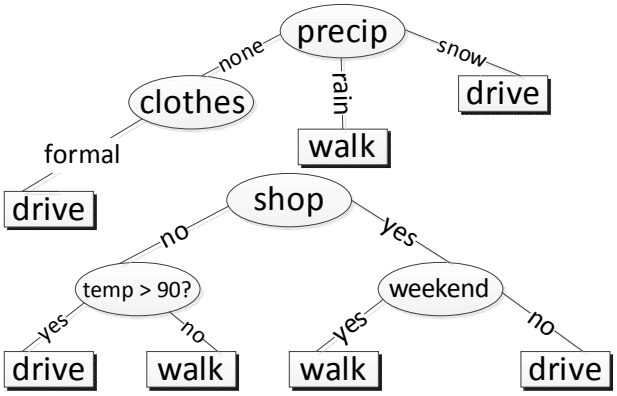

Fig. 1. Example of a decision tree

From Table I, we can see that in P2, activity B will be chosen when X>999. At the same time, activity C will be chosen when X ≥ 999. Since activity E has implicit dependencies to activity B every time B is chosen from P2, activity E in P5 must be chosen; and, activity F cannot be executed. Meanwhile, every time C is chosen in P2, activity F in P5 must also be chosen. This condition results in a situation where activity E in P5 will be chosen when X > 999, while activity F will be chosen when X ≥ 999.

The rule for choosing B in P2 is the same as rule for choosing E in P5, that is X > 999. And the rule for choosing C in P2 is the same as rule to choose F in P5, X ≥ 999. From this data, we can see the relationship between the rules for decision points in a non-free choice construct. Both decision points have exactly the same rules for each dependent branch in the workflow. However, the similarity of the rules only applies if the attribute comes from one activity. Different sources of variables can affect the gain value for node assignment of the decision tree. The rules found are still valid, yet there might be no similarities for two decision points in a non-free choice construct. If there are other activities which contribute to the variables, then only the nearest activity can be used as a source for attributes to build the decision tree.

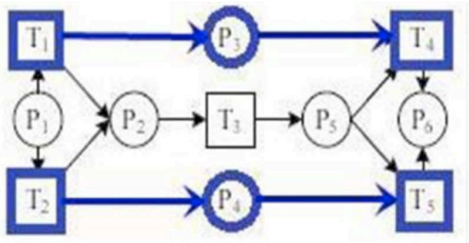

Fig. 2. Non-free choice

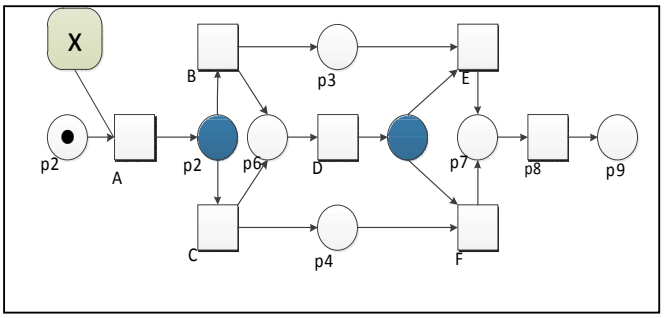

Fig. 3. Non-free choice example with activity A having attribute x

IV. IMPLEMENTATION AND EXPERIMENT

Experiment will be done using to case examples. Both cases are workflow with non-free choice construct. Experiment will be done using ProM 5.2 framework. Discovery process will be done using alpha and alpha++ algorithm. Rules identification will be done using decision miner, the Decision Point Analysis plugin.

The first example is shown in Fig.4. The workflow contains two decision points which have implicit dependencies (P3 and P4). If in P3 activity B is chosen, then P4 must choose activity C, and vice versa. However, when P3 chooses activity D, then P4 must also choose activity D. Activity A has attribute X and Y. Event log for the first example is shown in Table 2.

The second example uses credit application workflow as shown in Fig.5. Modification is done to add non-free choice property between checkSmallAmount and startApproval, as well as between checkLargeAmount and notifyRejection. From the above figure, we can conclude that if from P4 the decision is checkSmallAmount, the decision in P6 will be startApproval. At the same time, if in P4 the decision is checkLargeAmount, in P6 notifyRejection will be chosen. This modification is done only to show the non-free choice construct in a large workflow.

There are more than one sources of attributes in the workflow, but since we want to identify the non-free choice relationship in a non-free choice condition, then the sources of attributes will be limited to checkLoanAmount where the attributes will the amount of loan. Choices in event log for the second example is shown in Table 3.

TABLE I. EXAMPLE OF CHOICES FOR WORKFLOW IN FIG.3

Case ID	Value of X	P2	P5
Case1	1000	B	E
Case2	999	C	F
Case3	500	C	F
Case4	1200	B	E
Case5	800	C	F
Case6	1500	B	E

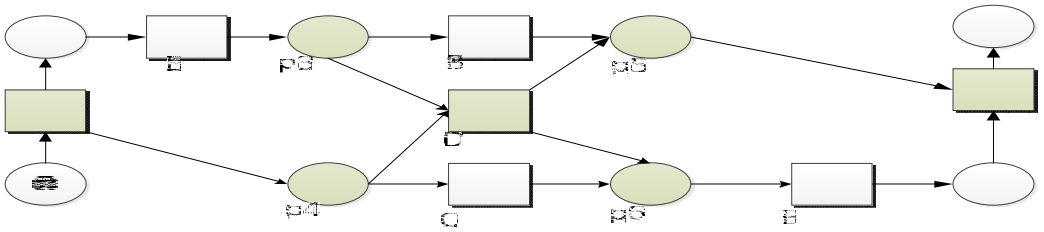

Fig. 4. Non-free choice example with activity A having attribute x

A. Experimental Result

The first example is discovered by using alpha++ algorithm. Alpha algorithm cannot find one if the decision points, so decision mining cannot be conducted by using the alpha model. The result from alpha++ model is analyzed using Decision Point Analysis. The result can be seen in Fig.6. The rules from both decision points are exactly the same.

The second example uses both alpha and alpha++ algorithms in model discovery process. Alpha cannot identify a non-free choice construct in the model, while alpha++ can. The process models from both discovery algorithms are presented in Fig.7. However, after conducting analysis using decision mining, both models yield the same result. The rules for decision points contained by non-free choice are identical. The rules are shown in Fig.8.

TABLE II. EVEN LOG FOR FIRST EXAMPLE

Case ID	Value of X	Value of Y	Log
Case1	500	Premium	A E B C F G
Case2	1000	Gold	A E C B F G
Case3	999	Silver	A C E B F G
Case4	800	Premium	A E C F B G
Case5	1001	Premium	A E D F G
Case6	1500	Gold	A E D F G
Case7	1100	Premium	A E D F G

TABLE III. CHOICES IN SECOND EXAMPLE OF NON-FREE CHOICE

Case ID	Amount	P4	P6
Case1	500	checkSmallAmount	startApproval
Case2	490	checkSmallAmount	startApproval
Case3	550	checkLargeAmount	notifyRejection
Case4	510	checkLargeAmount	notifyRejection

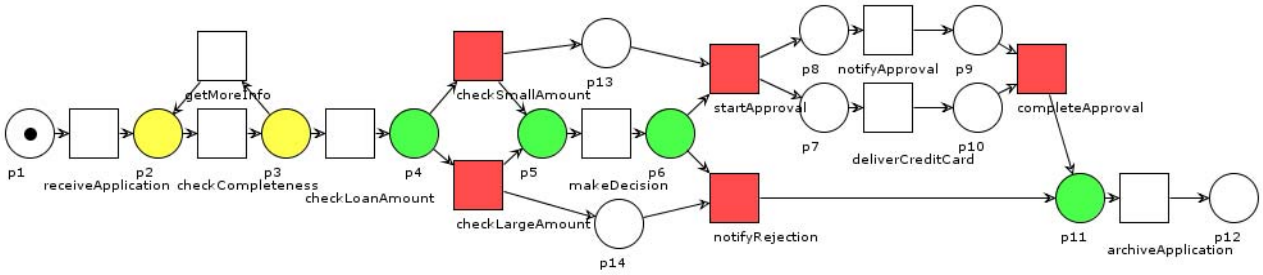

Fig. 5. Second example of a non-free choice workflow, modified credit application workflow

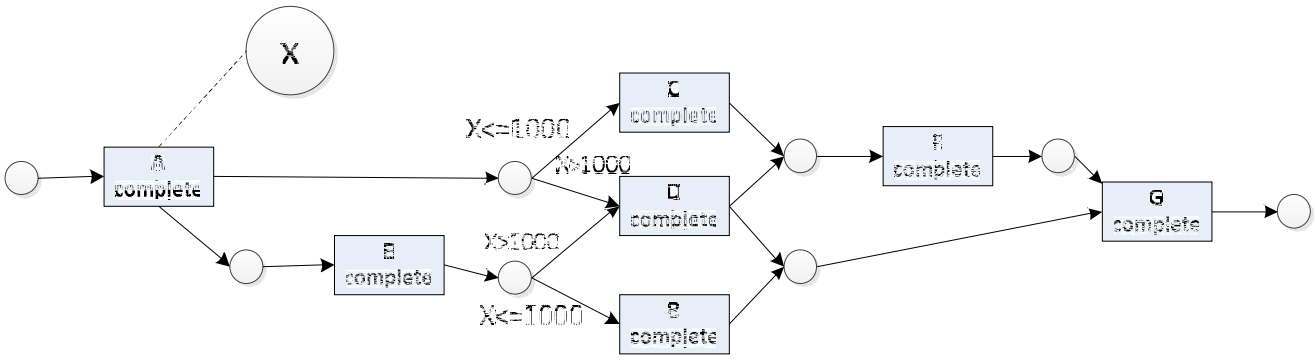

Fig. 6. Upper and lower decision points in the first example results in the same rules

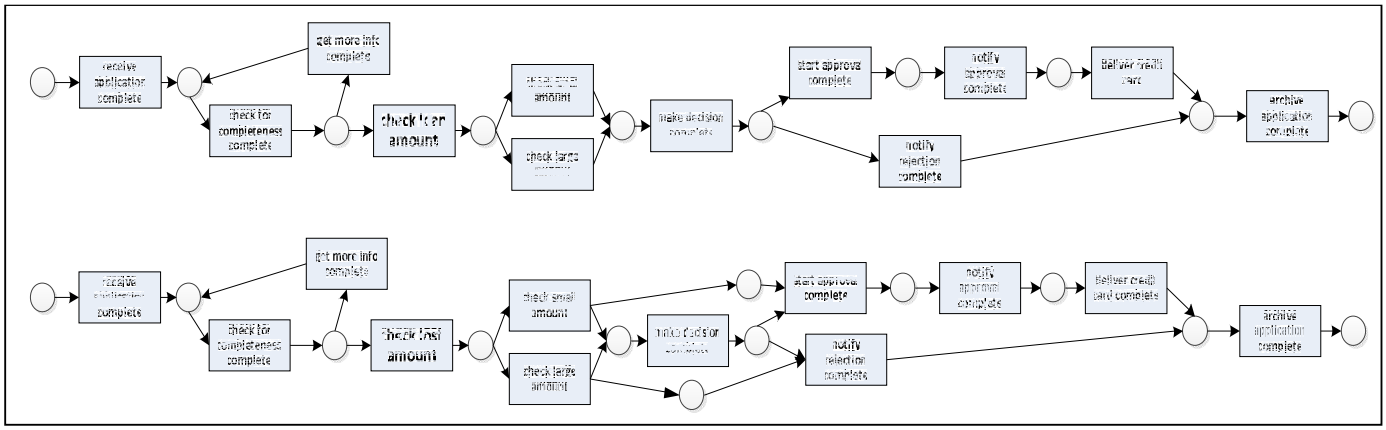

Fig. 7. Result from alpha(top) and alpha++(bottom) discovery of second example

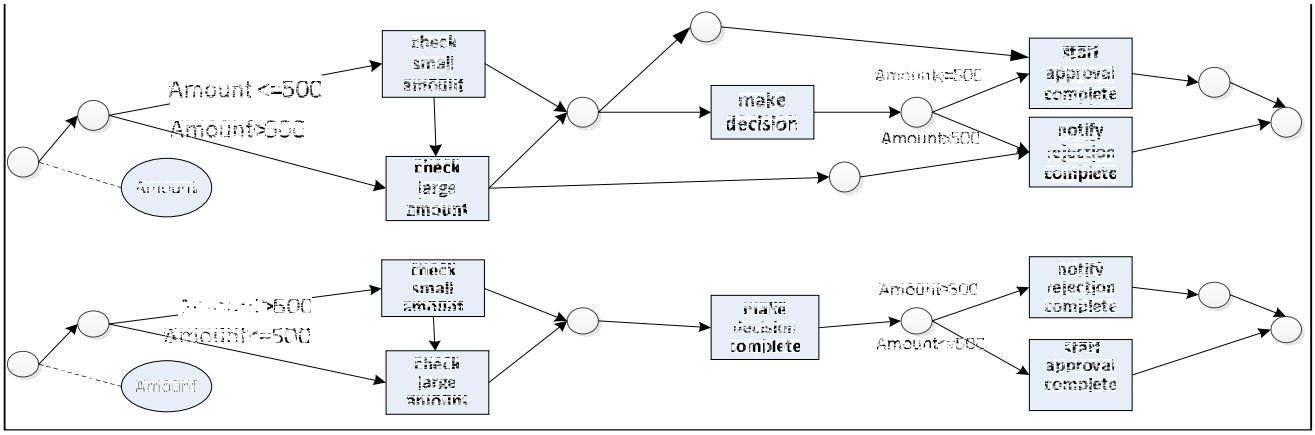

Fig. 8 CheckLargeAmount has the same rule as a notify rejection (it has implicit dependency relationship), while checkSmallAmount has the same rule as StartApproval.

V. LIMITATION

The similarities of rule relationship among the decision points in the non-free choice construct have some limitations in its application. It can be applied only to non-free choice workflows which have decision points in it. If the decision points cannot be found in the model, analysis using decision mining cannot be performed. This analysis also applies only if the attributes comes only from a single source or from sources that come from the same distance as all the decision points being analyzed. If the sources are different and the attributes have the same level of importance for the decisions, different nodes can be assigned to each decision tree which will result in different rules.

VI. CONCLUSION

Decision mining can be used to analyze rules that affect routing of a case. However, the rules found from a decision mining algorithm may also be employed to identify the form of choices using certain relationships of rules found from the workflow.

The experiment shows that rules from decision points that contained in a non-free choice construct have similarities, since implicit dependency makes limitation of choices in the cases. When the same rules between two decision points are found during a decision mining process, it is possible that the two decision points have non-free choice relationships even though the model does not show the non-free choice construct.

Acknowledgment

We would like to thank the Higher Education Directorate of the Education Ministry of Indonesia for supporting the research.

References

[1] Apte, C. and Weiss, S., Data mining with decision trees and decision rules, *Future Generation of Computer Systems,* vol. 13, pp. 197-210. 1997.

[2] Chandra, B. and Varghese, P.P., Moving towards efficient decision tree construction, *Information Sciences 179(8)*, pp.1059-1069. 2009.

[3] De Leoni, M. and van der Aalst, W.M.P., Data Aware Process Mining: Discovering Decision in Processes Using Alignments, *Proceedings the 28th Annual ACM Symposium on Applied Computing*, pp.1454-1461. 2013.

[4] Feng, Q., Miao, D., Cheng. Y., Hierarchical decision rules mining, *Expert Systems with Applications*, vol.37, pp.2081-2091. 2010.

[5] Grigorova, K., Process Modeling Using Petri-nets, *Proceedings International Conference in Computer Systems and Technologies*, 2003.

[6] Mehrez, A., Muzumdar, M., Acar, W., Weinroth, G., A Petri-net model view of decision making: an operational management analysis, Omega 23(1), pp. 63-78, 1995.

[7] Otero, F.E.B., Freitas, A.A., Johnson, C.G., Inducing decision trees with an ant colony optimization algorithm, *Applied Soft Computing*, vol. 12, pp.3615-3626. 2012.

[8] Rozinat, A. and van der Aalst, W.M.P., Decision Mining in Business Processes, BPM Center Report BPM-06-10, BPMcenter.org, 2006.

[9] Rozinat, A. and van der Aalst, W.M.P., Decision Mining in ProM, Lecture Notes in Computer Science, vol.4102, pp.420-425. 2006.

[10] Sarno, R., Sanjoyo, B.A., Mukhlash, I., Astuti, H.M., Petri-net model of ERP business process variation for small and medium enterprises, *Journal of Theoretical and Applied Information Technology*, vol.54(1), pp.31-38. 2013.

[11] Seng, J.L. and Chen, T.C., An analytic approach to select data mining for business decision, *Expert System with Applications,* vol.37, pp.8042-8057. 2010.

[12] Subramaniam, S., Kalogeraki, V., Gunopulos, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., Improving process models by discovering decision points, Information Systems 32(7), pp. 1037-1055. 2007.

[13] Wen, L., van der Aalst, W.M.P., Wang, J., and Sun, J., Mining process model with non-free-choice constructs, *BPM Center Report BPM-06-23*, BPMcenter.org, 2006.

[14] Wen, L., Wang, J., and Sun, J., Detecting implicit dependencies between tasks from event logs, Proceedings of the 8th Asia-Pacific Web conference on Frontiers of WWW Research and Development, pp.591-603.

This page is left blank on purpose