This additional chapter was created to supplement the criteria in the re-examination. This complementary section will aim to demonstrate the result of the new experiment.In this order the example is changed from satellite scenario to depots scenario.

5.0.1 The scenario problem.

The case depots describe all the actions that occur while working in the deposit or warehouse. We can drive, lift, drop load and unload. Drive Action means to move track from x position from y position.

- (: action Drive

  : parameters (?x - truck ?y - place ?z - place)

  : precondition (and (at ?x ?y))

  : effect (and (not (at ?x ?y)) (at ?x ?z)))

Action lift means that some hoist will pull crate in the specific surface in the special place in the depots

- (: action lift

  : parameters (?x - hoist ?y - crate ?z - surface ?p - place)

  : precondition (and (at ?x ?p) (available ?x) (at ?y ?p) (on ?y ?z)(clear ?y))

  : effect (and (not (at ?x ?y)) (at ?x ?z)))

Action load and unload are pretty much the same. There is a hoist which will pull the crate from a specific position and put it on the track in place.

- (: action Lift

  : parameters (?x - hoist ?y - crate ?z - truck ?p - place)

  : precondition (and (at ?x ?p) (at ?z ?p) (lifting ?x ?y))

  : effect (and (not (lifting ?x ?y)) (in ?y ?z(available ?x)))

- (: action Unload

  : parameters (?x - hoist ?y - crate ?z - truck ?p - place)

  : precondition (and (at ?x ?p) (at ?z ?p) (available?x)(in ?y ?z))

  : effect (and (not (in ?y ?z)) (not (available ?x))(lifting ?x ?y)))

5.1 Count and binary features

The code has two specific rules which are count and binary.

5.1.1 count rule

The main purpose of the " count rule" is to determine which of the non-bounded parameters of the specific action evaluates to an integer which will represent the cardinality of that object in the rule.Count parameter can be divided into different variables , it could be a free variable of a single wildcard. Count rules collect a higher amount of specific information which is more useful.

Generated count rules rules can look like these

- Drop (?x, ?y, ?z, ?p) :- goal:on(?fv0, ?fv1);goal:on(?fv0, ?y);ini:on(?fv1, ?fv0).
- Drop (?x, ?y, ?z, ?p) :- goal:on(?fv0, ?fv1);goal:on(?fv0, ?y);ini:on(?fv1, ?y).

5.1.2 binary

Binary rules as the name suggests will only evaluate features into "true or false". It can generate one or more count rules depending on the unique free variables.Binary rule in specific of wild card value can be static for entire pddl.

Generated binary rules can look like these

- Lift (?X, ?y, ?z, ?p) :- goal:ini:on (?fv0, ?y);goal:on(?fv0, _);goal:on(?z, ?fv0).
- Lift (?X, ?y, ?z, ?p) :- ini:at(?fv0, _);ini:clear(?fv0);ini;on(?y, ?fv0).

5.2 Models

When comparing the outcomes in order for the results to be adequate the standards must be established. In the case of this research there are many combinations of parameters according to which the planning process is conducted. The first metrics that was taken into consideration was the rules size. As described before in the section Rule 3.1.1 the rules can be created with different numbers of predicates in their bodies. However, the bigger number of predicates gets, complexity has increased significantly. Additionally, as discovered during the research the results of comparing the results from rules of size 2 or 3 do not differ significantly. In order to be able to conduct the process of investigation more efficiently the rule size was limited to 2. Another metric that has to be adjusted are different machine learning algorithms. To be able to make any conclusions it is not enough to test using only one method as the results can differ significantly while changing the algorithms used. Therefore to create adequate results, three different ML models were used: Logistic Regression, Support Vector Classification and Random Forest. They are described in the section Training of machine learning models 4.3. For each of the ML algorithms a separate model for the binary (original) solution and the improved - count solution were developed in order to compare their metrics

5.3 What do we want to achieve?

The main aim of the study will be to calculate the values obtained. For this purpose, the whole procedure will be calculated by 3 different algorithms. The digits we get will be evaluated based on precision, recall and f1score. The table in chapter 5.5.1 will present the scores on the above-mentioned criteria on a positive and negative value basis. Simplifying the whole procedure, negative actions will be given in column 0, we want the numbers in this place to be as small as possible. Next to the zero there will be 1. This column will represent the real actions. A good result will be if the numbers are as close as possible to 1.

5.3.1 Steps to perform

1. Download entire from https://github.com/Czubbi/MachineLearning-PDDL
2. Save the file to the desktop and then load it in visual studios.
3. Choose the scenario and run command in command line example like this:C:\Users\leszek47\Desktop\reexam pddl\MachineLearning-PDDL-subdominization>python ./src/subdominization-training/gen-subdom-rules.py --store_rules ./results/depots_with_runs.csv --rule_size 3 --run ./useful-actions-dataset-main/depots/runs/optimal ./useful-actions-dataset-main/depots/runs/optimal/p01-1-1-2-2-2-4/domain.pddl
4. Change post processing-py to the specific scenario(in this case depots)
5. Run the script rules-rules.bs command in terminal
6. Run rules-training command in terminal
7. Run select-features command in terminal
8. Run train-model command in terminal
9. After steps from 6 to 9 we will get a learning model.
10. Run model and we will get results.
11. After performing the result , the person is able to do it once again with a different algorithm by changing the last word in the train model command.Logr to logistic regressions , rf to random forest or svc to support vector machines.

5.4 Comparison plane

To visualize the progress of the research a comparison of the previous base system and the new solution must be made. For this, models corresponding to both systems will be created and used. This method evaluates the results using the same parameters from both systems. In total there will be 6 models, 3 for each solution. In order to keep consistency in nomenclature the models created using the base system will be referred as "binary models" and those computed using the enhanced solution will be referred as "count models"

The models are created for each individual action. Therefore, after computation in fact, in the case of the depot domain, 5 different models are obtained each for classification of one action schema. The evaluation will concern a specific action instead of the overall combined score. For the results to be the most accurate the binary and count models created with the same parameters will be directly compared. Such an approach ensures the most adequate juxtaposition and results in the most descriptive data.

5.5 The results of different models

This section will present the results of comparing the binary and count models. In order to clearly present the findings it is divided based on the machine learning algorithm used.

5.5.1 Logistic Regression

The first compared models were created using Logistic regression The upper part of the table illustrates metrics for load, unload and drop action for Logistic regression of count and binary models. While for the count model for LOAD action the recall of good operators is very high - 94% of all positive samples has been classified as positive (opposed to 1% for binary), it can be seen that the guesses the model did for good operators were less precise - drop from 100% for binary to 81% from count. This shows a deterioration in comparison to the binary model. While for the count model for UNLOAD action the recall of good operators is very high - 99% of all positive samples has been classified as positive (opposed to 87% for binary), it can be seen that the guesses the model did for good operators were less precise - drop from 66% for binary to 33% for count. This shows a deterioration in comparison to the binary model

5.5.2 Support Vector Classification

We have the same recall and precision for good samples for LOAD action for binary and count (100% recall, 63% precision). For the bad samples we have better results of recall for binary (0% opposed to 85%), which shows a deterioration of count model in comparison to the binary model. While for the count model for UNLOAD action the recall of good operators is significantly lower - 18% of all positive samples has been classified as positive (opposed to 43% for binary), it can be seen that the guesses the model did for good operators were more precise - increase from 79% for binary to 94% for count. This shows an improvement in comparison to the binary model.

		Precision		Recal		F1 score	
		Logistic regression					
		0	1	0	1	0	1
	L-Count	0,98	0,81	0,94	0,94	0,96	0,87
U = unload	L-Binary	0,79	1,00	1,00	0,01	0,88	0,02
L = load	U-Count	0,99	0,33	0,51	0,99	0,67	0,49
D = drop	U-Binary	0,97	0,66	0,89	0,87	0,93	0,75
	D-Count	0,97	0,00	0,97	0,00	0,97	0,00
	D-Binary	0,97	0,00	1,00	0,00	0,98	0,00
		Support vector machine					
	L-Count	1,00	0,63	0,85	1,00	0,92	0,77
	L-Binary	1,00	0,63	0,00	1,00	0,00	0,77
	U-Count	0,83	0,94	1,00	0,18	0,91	0,30
	U-Binary	0,88	0,79	0,97	0,43	0,92	0,55
	D-Count	0,97	0,00	0,97	0,00	0,97	0,00
	D-Binary	0,97	0,00	1	0,00	0,98	0,00
		Random forest tree					
	L-Count	0,93	0,80	0,93	0,80	0,93	0,80
	L-Binary	0,95	0,98	0,95	0,98	0,95	0,98
	U-Count	0,88	0,80	0,97	0,46	0,93	0,59
	U-Binary	0,89	0,87	0,98	0,51	0,93	0,64
	D-Count	0,97	0,00	0,97	0,00	0,97	0,00
	D-Binary	0,97	0,00	1	0,00	0,98	0,00

Picture 5.5.1 Tabel with results

5.5.3 Random Forest

For the count model for LOAD action the recall of good operators is significantly lower - 80% of all positive samples have been classified as positive (opposed to 98% for binary). Also it can be seen that the guesses the model did for good operators were less precise - decrease from 98% for binary to 80% for count. This shows a deterioration in comparison to the binary model. For the count model for UNLOAD action the recall of good operators is a bit lower - 46% of all positive samples have been classified as positive (opposed to 51% for binary). Also it can be seen that the guesses the model did for good operators were less precise - decrease from 87% for binary to 80% for count. This shows a deterioration in comparison to the binary model.

5.5.4 Explanation of Drop

We can see that both of the models (Binary and Count) are creating a naive model predicting only the majority class (NEGATIVE) despite having the weighted labels.

5.6 Reflections

Considering the results of comparing the models several conclusions can be made. In most of the cases the introduced features did not change the results and based on the compared metrics They perform similarly. Depending on the action schema the scores varied and in some cases the improvement could be noticed but in other the situation was opposite. However those results are neglectable as the differences are insignificant and no conclusions can be made based on them. Therefore, because of used metrics and lack of further experiments based on other methods the research can be summarized as inconclusive. Despite this, it elaborates on important aspects and the concepts could be used for further development with high chances of accomplishing more satisfying results.