Journal of Theoretical and Applied Information Technology
10th December 2015. Vol.82. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NON-DETERMINISM REDUCING METHOD FOR OWL SHOIN CONCEPT CONSISTENCY CHECKING

[1]ANDREY VIKTOROVICH GRIGORYEV, [2]ALEXANDER GRIGORYEVICH IVASHKO

[1,2]Tyumen State University
Semakova Street, 10, Tyumen, 625003, Russian Federation

ABSTRACT

Description logics the widely used tool for the knowledge base representation. The main task of this formalism is concept consistency checking problem and it is solved by tableau algorithm. However, complexity of the tableau algorithm is NEXPTIME and there are many tries to develop optimization for this algorithm. This paper tells about probability determining method of two concepts conjunction consistency, which was described in SHOIN DL. The developed method of coherence determination uses a Kruskal's method for segmentation. Based on the developed probability determination method the method of reducing choice non-determination in case of merging individuals executed in case of solving rule "n>=" was created. For testing of the developed method a module, which implements presented techniques was developed and integrated to TReasoner system. Computer experiment was made for efficiency evaluation of the developed methods and algorithms. Results shows that count of performed operations on «n >=» rule expansion reduces to 28%. Also article presents the theoretical explanation of the advantage of the presented method.

Keywords: *Description Logics; Consistency Checking; Rule Expansion; Kruskal's Algorithm.*

1. INTRODUCTION

Modern state of information technologies already allow to solve process automation problems related to issues of health and human life. Among the variety of such problems it is necessary to distinguish the developments of Austrian [1] and British [2] scientist. It allows performing decision-making support related to diagnosis of cancer (in particular cancer of the lung). These opportunities became available due to formalism development of knowledge base representation, which called as description logics [3] and the OWL, which is based by this formalism [4]. Reasoning about knowledge bases allows determining subsumption of the one concept to another. It is performed by applying specials methods: tableau method [5] hyper tableau method [6] and resolution method [7]. The tableau method is the most widely used among such methods.

Unfortunately, tableau method has an NEXPTIME complexity, thus there are many count of attempts to optimize this method by heuristics reducing answers search space of tableau algorithm: every year description logic workshop (DL) is hold and at the workshop there is OWL Reasoner competition. It performed to determine best of the tableau algorithms implementations.

2. PRELIMINARIES

The concept is a class of individuals (objects of subject area), which satisfy to certain property in description logics. For instance, "Human" concept defines a set of all objects that are humans. Concepts aren't only names of certain properties. In the table 1 are presented all constructors, which define an individual's classes of SROIQ description logics (mathematical basis of the OWL) of the web ontology language (OWL).

Knowledge base is defined by sets of equivalence axioms (C≡D) and subsumption axioms (**C⊑D**). These axioms are described in TBox axioms set containing general rules about subject area. Concrete assertions about individuals of subject area are contained in the ABox axioms set. Such assertions has the form of the class assertions axioms (i:C, where "i" is the individual name and "C" is a concept name) or object property assertion axioms ((a, b): R, where "a" and "b" are the individuals names and "R" is the object property name).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1. SROIQ Description Logics Concepts Constructors

Name	Syntax	Description
Conjunction	$C_1 \sqcap C_2$	Defines individuals having the properties C_1 and C_2
Disjunction	$C_1 \sqcup C_2$	Defines individuals having the properties C_1 or C_2
Existence quantifier	∃R.C	Defines individuals having the relation *R* with the one or more another individuals, which have the property *C*
Universal quantifier	∀R.C	Defines individuals having the relation *R* with the only those individuals, which have the property *C*
<= cardinality constructor	n R <= C	Defines individuals having the relations *R* with more or equal than *n* individuals, which have a property *C*
>= cardinality constructor	n R >= C	Defines individuals having the relations *R* with less or equal than *n* individuals, which have a property *C*

Kernel of the tableau method is the model consistency checking of the concept defined by description logics rules. Model of the concept is a couple of individuals set X, set of relations between individuals R and sets q(x) for all x□X, that contains subject area concepts containing individual x. If set q(x) contains concept ⊥ or pair of the counter liters C and ¬C, or set of relations R contains pair (x, y) in both two counter relations ri and ¬ri (ri □ R), then there is a clash in model else it is a right model.

Tableau algorithm has an exponential complexity due to a non-deterministic choice on performing satisfiability checking on C⊔D or n R >= concepts:

During concept processing **C⊔D**, a status of interpretation should be saved and one of the concepts C or D should be added to the set of concepts q(x), by the table algorithm. Thus, if on any step there was a contradiction, it should be return to the next remembered status, in which other choice case is possible.

During concept processing a of n R >= individual, which belongs to this concept, shouldn't have more than n of individuals, which are in the relation of R with the considered individual and if it is some such individuals, that, probably, some couple of individuals represents one individual and for this purpose their combining is required.

To realize such choice, which will allow to find quickly the correct concept, during processing a disjunction or the correct configuration of combining of individuals, during processing n R> = rule expanding, the new method of determination of coherence of conjunction of two concepts was offered in this paper.

In [8] authors introduced a method that can determine satisfiablity of two concepts conjunction in certain cases. We propose an extension of such method and use it to determine consistency of merged individuals.

3. RELATED WORKS

Worldwide scientists hold researches of such directions. The most important developments, realized in state-of-the-art reasoners are researches of German scientists Volker Haarslev and Ralf Moller [9-10]. They were solving a non-determinism branching problem for the ALCQHR+ logic. Nasim Farsiniamarj [11], who used SHQ logic, suggested the extension of the proposed idea. The general idea based on the earlier determining of individuals disjointness in this developments. Also, many authors developed method which rewrites ontology such as there are less rules with qualified number restriction [12-15].

We introduce a new method, which determines how individuals must be united in this article.

4. SEARCH SPACE REDUCING METHOD

Assume that, knowledge base has N atomic concepts (A1, A2, …, AN). Let for every arbitrary (complex or atomic) concept D there are a corresponding vector V(v1, v2, …, v2N) and a number G. Every component of the vector V determines total count of tableau algorithm built models having a concept Ai. Total count of Ai concept negation is determined by vi+N vector component. Number G determines total count of concept D models. Consider models total count (number G) computing method for every constructor from table 1.

1. If concept ***D*** is an atomic concept (literal) or atomic concept negation then ***G*** if equal to 1;
2. If concept ***D*** is a conjunction of two other (maybe complex) concepts ($C_1 \sqcap C_2$) and total count of models for this two concepts are G_1 and G_2, then total count of concept ***D*** models is equal to $G_1 * G_2$;
3. If concept ***D*** is a disjunction of two other (maybe complex) concepts ($C_1 \sqcup C_2$) and total count

Journal of Theoretical and Applied Information Technology
10th December 2015. Vol.82. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of models for this two concepts are G_1 and G_2, then total count of concept D models is equal to $G_1 + G_2$;

4. If concept D is built by existence quantifier (∃) or universal quantifier (∀) or cardinality constructor (n R >= or n R <=), then total count of models is equal to models total count of concept under quantifier (or cardinality constructor);

To compute the value of an every component of the vector it is used a method similar to the presented above:

1. If concept D is an atomic concept (a literal) or negation of the atomic concept then corresponding component of the vector V set equal to 1;
2. If concept D is a disjunction of two other (maybe complex) concepts ($C_1 \sqcup C_2$) then total count of models for each liter became equal to $V_L = V_{1L} + V_{2L}$, where V_1 and V_2 vector for concepts C_1 and C_2 respectively and L defines a value for L liter component of the vectors;
3. If concept **D** is a conjunction of two other (maybe complex) concepts ($C_1 \sqcap C_2$) then total count of models for each liter became equal to $V_L = V_{1L} * G_2 + V_{2L} * G_1 - V_{1L} * V_{2L}$ where G_1 and G_2 are total counts of models for concepts C_1 и C_2 respectively and V_{1L}, V_{2L} are total counts of concept models for liter L.
4. If concept D is built by existence quantifier (∃) or universal quantifier (∀) or cardinality constructor (n R >= or n R <=), then total count of models with L liter is equal to total count of models with liter L concept under quantifier (or cardinality constructor).

Thus, for each concept the probability of appearance in it each of names of simple concepts is defined. We will consider possibility of application of the described method for implementation of search of model of a concept.

On processing a n R >= rule it is determined all R-successors of the current individual [16-17]. If there are more than n such successors then it is needed to unite some of them while its count more than n. Subsequently united individuals are processed as the one individual. If there will a clash on processing of the tableau method then last remembered interpretation will loaded, if last non-deterministic rule will n R >= rule it will perform union another individuals. And it is took place looking over all configurations of union. Two different unions are presented on the figures 1 and 2.

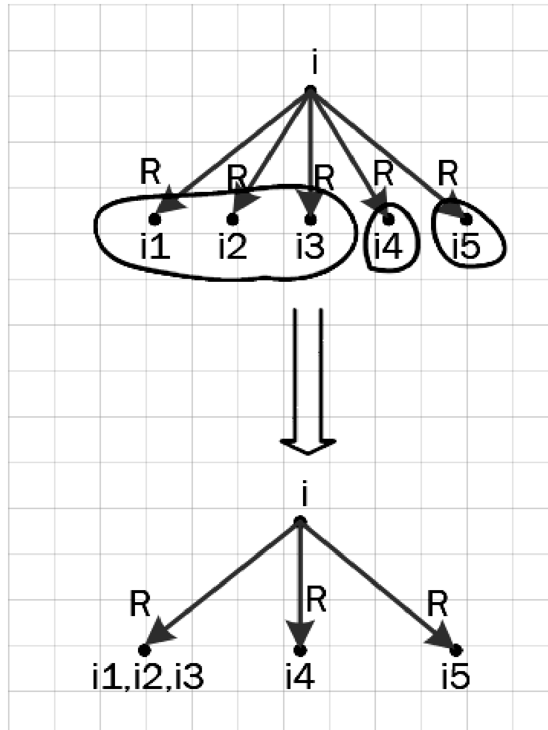

Figure 1. Union variant

For the probability determining of the two individuals consistency checking use next assumption. No complex concept should have two contrary atomic concepts A and ¬A. Therefore the total union probability of the two individuals i and j is

$$a_{ij} = \prod_{C\in Sign} \left(1 - \frac{W_C^i * W_{\neg C}^j}{G_i} * \frac{W_{\neg C}^i * W_C^j}{G_i}\right) \quad (1)$$

where W is the function defining count of models that contain corresponding concept (C and $\neg C$ in the formula) for the i_{th} individual and $Sign$ is the set containing all concepts in the ontology.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

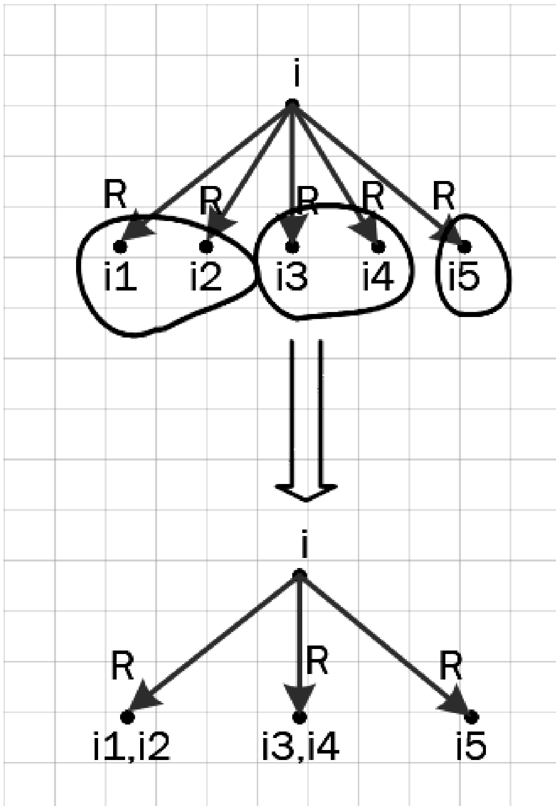

Figure 2. Alternate union variant

For non-determinism reducing it can be used a special form graph. Let vertices of the graph are individual preimages, there are edges between all pairs of vertices and weight of the edge is the concepts consistency probability of the individuals at the end of the edge. Form of this graph is presented on figure 3.

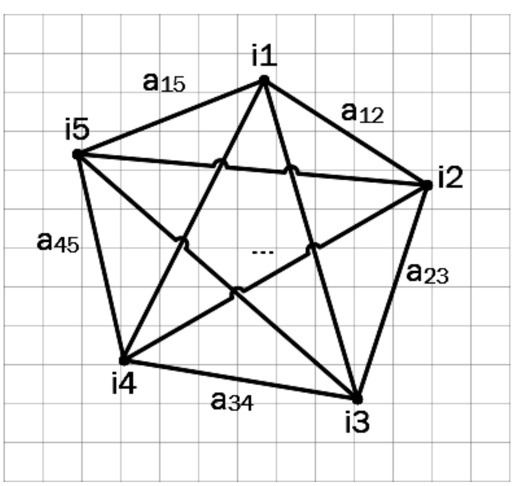

Figure 3. Individuals graph

Weight of the edges is aij and is computed by the formula (3.1).

The modification of Kruskal's algorithm [18] is used for individuals combining. This modification often used to locate segments on a picture. Let our task segment is an individual union set. If individuals has maximum combining probability then they must be united firstly and therefore edges are considered in weight decreasing order. Kruskal's algorithm will works correctly and will search maximum weight segments firstly [19]. Aslo it is needed to consider segments in order of decreasing value of

$$P_{it} = \prod_{a_{ij} \in Segm} a_{ij} \quad (2)$$

Where Segm is the set of all edges choosed by Kruskal's algorithm and i is the number of the combination and Piteration represents the combination consistency probability on the itth iteration. But the Kruskal's algorithm searches segmentations in order of decreasing sums of aij so such values is exchanged to ln(aij).

Besides, if there is a clash in some determined segment tableau method must exclude individuals set from searching space. Thus we input new set F for sets of individuals where clash was founded. Modification of Kruskal's algorithm is presented in listing 1.

Listing 1 – Kruskal's algorithm modification.

Input: vertices set $I = \{i_1, i_2, \ldots, i_m\}$, vertices necessary count n.

Step 1. Initializing m sets of S_i. Every S_i contains only one element: **i**

Step 2. Sort edges e_{ij} in order by increase of a_{ij} value.

Step 3. For each edge e_{ij}

Step 4. While a_{ij} value equals to value of the next edge (denoted a_{kh})

Step 5. If vertices k and h are belong to different sets and union of sets S_k and S_h are contained if F set, then unite sets S_k and S_h

Step 6. If count of sets less or equal to n then return set of sets S else go to step 3

Output: variant of vertices union.

If there is a clash in some phase of the tableau algorithm then it is necessary to determine those individual set which must to be added to F set. It is performed on loading of saved interpretation. Part of the algorithm realizing tableau method is presented in listing 2.

Journal of Theoretical and Applied Information Technology
10th December 2015. Vol.82. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Listing 2 – Modification of the tableau method.

Step 1. If there is a backtrack to ***n R*** >= rule then

Step 2. ***A*** = ***GetAllAncestors***(***Current individual***);

Step 3. For all segments allocated in last processing if the rule

Step 4. For all vertices ***v*** of the current segment

Step 5. For all ***a*** ∈***A***

Step 6. If ***a*** = ***v*** then

Step 7. Add considered segment to ***F*** set;

Step 8. Get out of all cycles;

Function GetAllAncestors is simple implementations of DFS algorithm. It moves up from Current individual on all relations until reach a root. Presented algorithm will look over every variants of individual union, because set F will monotonically increase.

There is graph diagram representing segments of the individuals created on step 5 of the Kruskal's modification on figure 4.

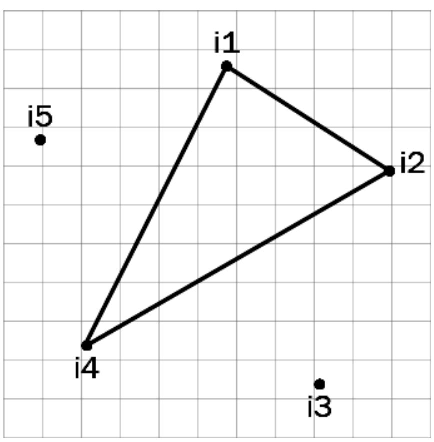

Figure 4. Variant of segmentation

Such segmentation will occurs if weights of the edges a12, a14, a24 are less than weights of the other edges.

Let us show comparison of the default approach for individuals combining and presented approach. In the default approach every combination can be considered only one time and if there is only one correct combination then the probability of the correct combination appearance is $p = \frac{1}{N}$, where N is the total count of combinations (equals to the combinations with repetitions $\overline{C_V^D}$, where V is the total count of individuals and D is the needed count of individuals). It is easy to see that expected value of the iterations count is equal to $\frac{N}{2}$.

Now we try to show that presented method gives smaller expected value than the default approach. Assume that the each combination is differ from another combination and at the each iteration only one additional concept appears in interpretation (and this concept is differ from any another concept). It is easy to see that this scenario if the worst case. Then on the last iteration, total probability of combination consistency is equal to

$$P_{last} = \left(1-\frac{1}{G_{i_1} * G_{i_2}}\right) * \left(1-\frac{1}{G_{i_3} * G_{i_4}}\right) * \ldots * \left(1-\frac{1}{G_{L-1} * G_{L-2}}\right) \quad (3)$$

Assume that all G_i are equals to G, then

$$P_{last} = \left(1-\frac{1}{G^2}\right)^{N-1} * \frac{1}{G^2} \quad (4)$$

Where D is the total count of combinations. To compare P_{last} probability formed by presented approach and $\frac{1}{N}$ it must be calculated

$$\lim_{N\to\infty}\left(\left(1-\frac{1}{G}\right)^{N-1} * \frac{1}{G} * N\right) \quad (5)$$

And it is cleary that it equals to 0 and therefore P_{last} is less than $\frac{1}{N}$, so it can be concluded that expected value of the iterations count of presented method is less than expected value of the iterations count of default approach.

5. COMPUTER EXPERIMENT

The TReasoner system [20] was used for testing of the developed method. TReasoner was developed by author and has open source code (available at URL: https://code.google.com/p/treasoner/) and distributed under GNU General Public License v. 2.0. It took part at the ORE 2013 reasoners competition and wins first prize on OWL RL ontologies classification. The QualOpt class implementing presented method was developed and it was integrated to TReasoner for testing. Ontologies from ORE 2013 OWL DL Classification competition having number restrictions (ALC N logics at least) were used for the experiment. All knowledge bases from test dataset are available at http://mowl-

Journal of Theoretical and Applied Information Technology
10th December 2015. Vol.82. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

power.cs.man.ac.uk/ore2013/ore2013-ontologies-offline.zip. So anyone can reproduce results of the computational experiment.

Table 2. Test data information

Name	Axioms Count	Concepts Count	Roles Count	Individuals Count
49f5cae6-e9be-4614-ad3d-2e696f7e3d6c_MSC671.owl	181	40	6	5
c692093c-46b4-4728-b062-42a1cdb13f0c_pizza.owl	693	97	8	5
ca9585af-418b-4bdd-9432-2a3aaf49678b_Pizza.owl	695	97	8	5
b4885109-03d6-4e14-a89d-bc05a64b31df_pizza.owl	703	99	14	5
f43b18b9-d218-493b-bc2d-27d49eb7c753_pizza.owl	703	99	14	5
00a1118a-5420-46f0-b4b2-a2585165b28a_ePizza.owl	705	99	14	5
26ffd334-3398-4054-993b-f8e22411435c_pizza.owl	705	99	14	5
95a2df41-085b-41ef-998e-9dc32bd75a30_pizza.owl	705	99	14	5
6126d1bf-ffc8-4a6f-ad9d-b8173f4d99db_ePizza.owl	706	100	14	5
00793.owl	712	99	8	5
5d572996-f4b6-40f3-8003-1afa4f477964_pizza.owl	712	99	8	5
6610e240-3629-4127-b44c-86ae60ee00f9_pizza.owl	712	99	8	5
66d0a1c6-dc39-437e-90d6-63c91fe8beb6_pizza.owl	712	99	8	5
7dacb73c-7366-4099-aba0-16dac9dda108_pizza.owl	712	102	14	5
f9ac3350-6de9-4481-bff6-b19458f3f81e_pizza.owl	712	99	8	5
4243a7a0-1e9f-492f-a26b-e3232ff60f29_pizza.owl	796	99	14	5
853e0872-cc76-472f-acdd-d8407d489447_e_fred.owl	889	138	18	206
52cf3ab5-1662-4296-835e-b22ac92339e7_.2_DUL.owl	1240	156	359	0
c4f69db5-3f46-479a-a92e-678f107cdb30_wineCR.rdf	1546	213	31	367

Information about test data is presented in a table 2. First column of the table contains name of the ontology as it given at competition. Columns 2, 3, 4 and 5 contain information about axioms count, concepts count, roles count and individuals count respectively. Data in the table is sorted by increasing order of axioms count. Some rows contains equal count of axioms count, concepts count and roles count. This is not means that correspond ontologies are identical. Ontologies form such rows describes same subject area but with different axioms with different form.

Classification operation was held for each ontology for testing presented method. Computer experiment used ASUS Notebook VX7SX Series Intel Core i7-2630QM CPU@2.00 GHz 2.00 GHz; 6.00 GB RAM under Windows 7 OS control. A reasoning operations performing in my system takes much time due to the fact that system was developed on Java and interpreted by JRE, some code wasn't written optimal (I use standard Java collections), also I didn't pay attention to details which can increase runtime of the system. Therefore I do not compare my system with state-of-the-art reasoners and restrict experiment with comparing count of performing operations on n >= rule expansions with and without developed optimization. This can show that using such technique may be more effective compared to methods without presented optimization.

Results of the experiment are showed in the table 3. First column contains name of the ontology, second column contains information about count of n >= rule expansions. This column show results of the system with implemented optimization. Column 3 contains same information as column 2 but there system was used without developed optimization.

Results show that presented technique of individuals union reduce count of n >= rule expansions for classification of some ontologies and therefore count of restores in tableau method. Figure 5 shows count of rule n >= rule expansions. "With optimization" line corresponds to results of the system with optimization and line "Without optimization" corresponds to results of the system without implemented optimization.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3. Experiment results

Name	Count Opt	Count No
49f5cae6-e9be-4614-ad3d-2e696f7e3d6c_MSC671.owl	121	114
c692093c-46b4-4728-b062-42a1cdb13f0c_pizza.owl	29468	32533
ca9585af-418b-4bdd-9432-2a3aaf49678b_Pizza.owl	20308	22025
b4885109-03d6-4e14-a89d-bc05a64b31df_pizza.owl	22703	30125
f43b18b9-d218-493b-bc2d-27d49eb7c753_pizza.owl	34776	39244
00a1118a-5420-46f0-b4b2-a2585165b28a_ePizza.owl	22047	23825
26ffd334-3398-4054-993b-f8e22411435c_pizza.owl	22045	23825
95a2df41-085b-41ef-998e-9dc32bd75a30_pizza.owl	22056	23825
6126d1bf-ffc8-4a6f-ad9d-b8173f4d99db_ePizza.owl	22458	24256
00793.owl	22099	23173
5d572996-f4b6-40f3-8003-1afa4f477964_pizza.owl	22123	23173
6610e240-3629-4127-b44c-86ae60ee00f9_pizza.owl	22109	23173
66d0a1c6-dc39-437e-90d6-63c91fe8beb6_pizza.owl	22093	23173
7dacb73c-7366-4099-aba0-16dac9dda108_pizza.owl	24201	26020
f9ac3350-6de9-4481-bff6-b19458f3f81e_pizza.owl	22118	23173
4243a7a0-1e9f-492f-a26b-e3232ff60f29_pizza.owl	14058	14059
853e0872-cc76-472f-acdd-d8407d489447_e_fred.owl	108691	216027
52cf3ab5-1662-4296-835e-b22ac92339e7_.2_DUL.owl	6183	6123
c4f69db5-3f46-479a-a92e-678f107cdb30_wineCR.rdf	236361	534887

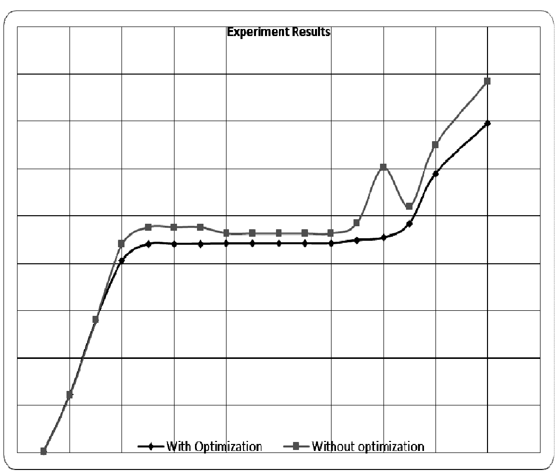

Figure 5. Results Comparing

Experiment results show that developed method allows reducing the performed operations count of n >= rule expansions. Total count of operations on selected ontologies reduces on 25%. Ontologies 853e0872-cc76-472f-acdd-d8407d489447_e_fred.owl and c4f69db5-3f46-479a-a92e-678f107cdb30_wineCR.rdf are not showed at the diagram on the figure because count of operations reduces extremely and it hides other results.

6. CONCLUSION

Article represents method for probability determining of concepts conjunction consistency and Kruskal's algorithm modification based on developed method. It is applied for non-determinism reducing on individuals to merging selection in n R >= rule. Experiment results shows that developed algorithm allows reducing time for concept consistency checking. In future research, we will apply developed techniques to cover SROIQ logic. In addition, we will research field of ontologies application and in future, we will present knowledge base approach for program static analysis.

7. ACKNOWLEDGEMENTS

This work is developed on Information system department of the Tyumen State University. In addition, I want to say thanks to my colleagues Alexander Kropotin and Mikhail Samoilov.

REFERENCES:

[1] Samwald M. "Genomic CDS: an example of a complex ontology for pharmacogenetics and clinical decision support", *in proceeding of the 2nd OWL Reasoner Evaluation Workshop (ORE 2013)*, Ulm, 22 July 2013. P. 128-133. URL: http://ceur-ws.org/Vol-1015/

[2] Sesen B., Jimenez-Ruiz E., Banares-Alcantara R., Brady M. "Evaluating OWL 2 Reasoners in the context of Clinical Decision Support in Lung Cancer Treatment Selection", *in proceeding of the 2nd OWL Reasoner Evaluation Workshop (ORE 2013)*, Ulm, 22 July 2013. P. 128-133. URL: http://ceur-ws.org/Vol-1015/

[3] Schmidt-Schauss M., Smolka G. "Attributive concept descriptions with complements". *Artificial Intelligence*. 1991. Vol. 48(1). P. 1-26.

[4] Motik B., Patel-Schneider P.F., Grau B.C. OWL 2 Web Ontology Language Direct Semantics (Second Edition). *W3C Official Site*. 2012. URL: http http://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211

[5] Baader F., Calvanese D., McGuinness D. L., Nardi D., Patel-Schneider P. F. The description logic handbook: theory, implementation, and applications. Cambridge University Press New York, NY, USA 2003, P. 622.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[6] Motik B., Shearer R., Horrocks I. "Hypertableau Reasoning for Description Logics", *Journal of Artificial Intelligence Research*. 2009. Vol. 36(1). P. 165-228.

[7] Rudolph S. "Foundations of Description Logics". *Reasoning Web. Semantic Technologies for the Web of Data - 7th International Summer School*, Galway, Ireland, August 23-27, 2011. P. 76-136. doi:10.1007/978-3-642-23032-5

[8] Ivashko A.G., Grigoryev A.V., Grigoryev M.V. "Tableau algorithm modification based on complex concepts disjointness checking". *Bulletin of the Tyumen State University*. - 2012, №4. - P. 143-150.

[9] Haarslev V., Moller R. "Optimizing Reasoning in Description Logics with Qualified Number Restrictions", *in proceedings of the International Workshop on Description Logics (DL'2001)*, Stanford 1-3 August, 2001. P. 142-151. URL: http://ceur-ws.org/Vol-49/HaarslevMoeller-142start.ps

[10] Haarslev V., Timmann M., Moller R. "Combining Tableaux and Algebraic Methods for Reasoning with Qualified Number Restrictions". *Proceedings of the International Workshop on Description Logics (DL'2001)*, Stanford 1-3 August, 2001. P. 151-160. URL: http://ceur-ws.org/Vol-49/HaarslevTimmannMoeller-152start.ps

[11] Farsiniamarj N., Haarslev V. Practical reasoning with qualified number restrictions: A hybrid Abox calculus for the description logic SHQ // Journal AI Communications - Practical Aspects of Automated Reasoning. 2010. Vol. 23(2-3). P. 205-240.

[12] Vescovi M., Sebastiani R., Haarslev V.: Automated Reasoning on TBoxes with Qualified Number Restrictions via SMT. Tech. Rep. DISI-11-001, Universita di Trento (April 2011)

[13] Mosurovic M., Krdzavac N. Some Optimizations R+IQ Tableau Algorithm // Proceedings of the 2nd International conference "Optimization and applications", Petrovac, Montenegro, 25 September - 2 October, 2011. P. 47. URL: http://www.ccas.ru/optima2011/abstracts_e.html

[14] Giordano L., Gliozzi V., Olivetti N., Pozzato G.L. *Proceedings of the International Workshop on Description Logics (DL'2014)*, Vienna 17-20 July, 2014. P. 543-555. URL: http://ceur-ws.org/Vol-1193/paper_20.pdf

[15] Thinh Dong, Chan Le Duc, Philippe Bonnot, Myriam Lamolle. "Tableau-based revision in SHIQ". *Proceedings of the International Workshop on Description Logics (DL'2015)*, Athens 7-10 June, 2015. P. 101-105. URL: http://ceur-ws.org/Vol-1350/paper-18.pdf

[16] Hollunder B., Baader F., Qualifying Number Restrictions in Concept Languages. In Proc. of KR, Boston, USA (1991).

[17] Horrocks I., Kutz O., Sattler U. "The Irresistible SRIQ", *in proceedings of the OWLED'05 Workshop on OWL: Experiences and Directions*. Galway, Ireland. November 11-12, 2005. P. 87-99. URL: http://ceur-ws.org/Vol-188/

[18] Cormen T. Leiserson C., Rivest R., Stein C. Introduction to Algorithms (3rd ed.) Cambridge: MIT Press. 2003.

[19] Kalachik R.A. "Automatic segmentation algorithm with segments aggregation in image searching systems" *Bulletin of RSRTU*. Vol. 23. Ryazan: RSRTU publishing. 2008, P.135 – 139.

[20] Grigoryev A.V., Ivashko A.G. "TReasoner: System Description", *in proceedings of the 2nd OWL Reasoner Evaluation Workshop (ORE 2013)*. - 2013, P. 26-31.