ДОПОВІДІ
НАЦІОНАЛЬНОЇ
АКАДЕМІЇ НАУК
УКРАЇНИ

ІНФОРМАТИКА

doi: https://doi.org/10.15407/dopovidi2018.03.022
УДК 004.8

А.Ф. Кургаев

Институт кибернетики им. В.М. Глушкова НАН Украины, Киев
E-mail: afkurgaev@ukr.net

Новое определение языка веб-онтологий OWL2

Представлено академиком НАН Украины А.В. Палагиным

Даны в метаязыке нормальных форм знаний (НФЗ) описания манчестерского синтаксиса и синтаксиса функционального стиля языка веб-онтологий OWL 2 — центрального языка семантического стека Тима Бернерс-Ли. Наличие таких описаний гарантирует реализуемость языка OWL 2 с реализацией интерпретатора метаязыка НФЗ. Показано, что выразительные возможности метаязыка НФЗ для формального описания OWL 2 вполне сопоставимы с выразительными возможностями метаязыка Extended Backus-Naur Form.

Ключевые слова: *метаязык нормальных форм знаний, формальное описание, язык веб-онтологий OWL, Semantic Web, манчестерский синтаксис, синтаксис функционального стиля.*

Поиск и использование нужной информации становятся все более трудоемкими и неэффективными. Уже осознана неизбежность перехода от хранения и обработки данных к накоплению и обработке знаний для борьбы с информационным насыщением общества.

Одно из направлений исследований и разработок, в котором для решения указанной проблемы сосредотачиваются значительные научно-технические ресурсы, — переход от классического интернета к семантическому, сохраняя основные принципы Web [1]:

– *децентрализация*, как отсутствие единого центра управления и распределенность ресурсов сети, создаваемых самими пользователями;

– *терпимость*, как универсальность доступа, независимо от аппаратной или программной платформы, сетевой инфраструктуры, языка, культуры, географического положения, физического или умственного нарушения.

Semantic Web — это веб-сайт, включающий документы, содержащие семантическую информацию в форме, доступной как людям, так и непосредственно компьютерам, для поиска, чтения, восприятия и использования информации при решении задач с помощью автоматизированных агентов и Web-сервисов [2].

Semantic Web создается на основе ряда стандартов, развиваемых и рекомендуемых консорциумом W3C [1—3]. В настоящее время на Semantic Web работают многие научные подразделения мира, разрабатывая и совершенствуя новые протоколы, технологии, среды программирования, языки, пользовательские интерфейсы, методы поиска знаний.

Целью статьи является экспериментальное исследование выразительных возможностей метаязыка нормальных форм знаний (НФЗ) [4—6] на примере описания языка онтологий OWL 2 (Web Ontology Language) — центрального языка семантического стека Semantic Web (рис. 1).

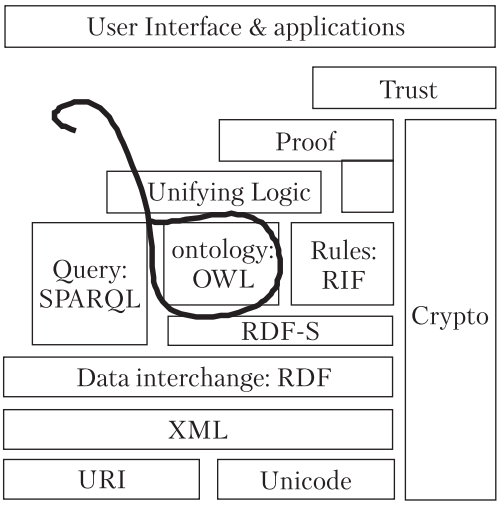

Рис. 1. Место OWL в семантическом стеке Тима Бернерс-Ли [3]

Характеристика языка OWL. Язык OWL в настоящее время является наиболее развитым языком онтологий базовой модели стека стандартов Semantic Web для описания контента, понимаемого компьютерами [3, 7, 8]. OWL — это логический язык представления онтологии в виде документов, которые могут храниться и передаваться в глобальной сети подобно любым другим данным или информации.

На рис. 2 дана структура языка OWL 2 с его основными компонентами. Эллипс в центре объединяет абстрактную структуру OWL 2 с его RDF (Resource Description Framework) графом [7, п.2.1], которые могут быть воплощены в разные языковые формы [7, п.2.2] для тиражирования и обмена онтологиями. Синтаксис Manchester [9] — это синтаксис языка OWL, упрощенного для восприятия пользователями. Синтаксис Functional-Style [10] предназначен для целей спецификации и создания основы для реализации OWL 2-инструментов. Синтаксис OWL XML (eXtensible Markup Language) — это синтаксис OWL функционального стиля, определенный XML-схемой [11]. Синтаксис RDF / XML для OWL — это просто RDF / XML, с конкретным переводом для конструкций OWL [12]; он обязательно должен поддерживаться всеми инструментами OWL 2 для хранения и обмена онтологиями, в основном, в форме документов RDF.

Для экспериментального исследования выбраны два нормативных синтаксиса OWL 2 — манчестерский и синтаксис функционального стиля с использованием официальных редакций нормативных документов [9, 10].

В приведенных ниже формальных описаниях языка OWL 2 использованы следующие метасимволы метаязыка НФЗ [4-6]:

"=" — разделитель, отделяет имя понятия (нетерминала) от его определения;

";" — конец определения понятия;

" " (пробел) — отношение конкатенации;

"/" — отношение альтернативного выбора;

"(", ")" — итерационные скобки обрамляют повторяемую (нуль или больше раз) структуру понятий;

"^" — отношение отрицания примыкающего понятия;

" ' " — текстовая кавычка;

true — тождественно истинное понятие с пустым объемом;

знаки /* и */ обрамляют комментарий.

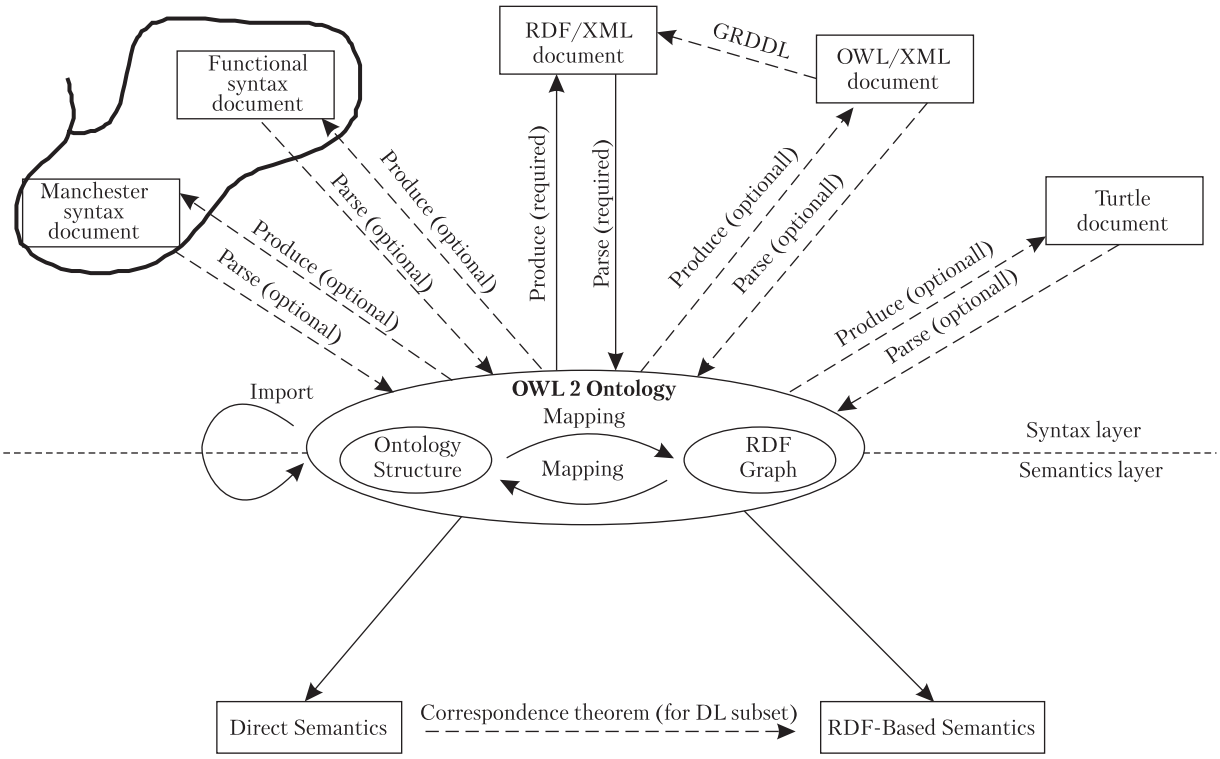

Рис. 2. Структура языка OWL 2 [7]

Описание в метаязыке НФЗ манчестерского синтаксиса. В табл. 1 дано в метаязыке НФЗ текстовое описание манчестерского синтаксиса языка OWL 2, а на рис. 3 — описание структуры верхнего уровня этого синтаксиса в форме графа.

Описание в метаязыке НФЗ синтаксиса функционального стиля языка OWL 2. В табл. 2 дано текстовое описание в метаязыке НФЗ синтаксиса функционального стиля языка OWL 2, а на рис. 4 — описание структуры верхнего уровня этого синтаксиса в форме графа.

Сопоставление предлагаемых с нормативными описаниями OWL 2. Нормативное описание манчестерского синтаксиса языка онтологий OWL 2 метаязыком Extended Backus-Naur Form (EBNF) [9] включает 87, а аналогичное описание метаязыком НФЗ (см. табл. 1) — 100 продукций.

Дополнительные правила НФЗ-описания манчестерского синтаксиса использованы для моделирования структурных скобок: (29) — floatingBody, (30) — fractional, (32) — floatingPost, (82) — classFrame_ и (89) — fact_, а необязательности: (35) — sgn, (44) — ontologyIRI_, (46) — versionIRI_, (63) — primary_, (64) — dataPrimary_, (67) — datatypeFrame_, (70) — anns_ и (89) — not.

Нормативное описание метаязыком EBNF [10] синтаксиса функционального стиля языка OWL 2 включает 120 продукций, а аналогичное НФЗ-описание (см. табл. 2) — 124 продукции. Дополнительные четыре правила НФЗ-описания нетерминалов функционального синтаксиса использованы для моделирования необязательности: (12) — ontologyIRI_, (14) — versionIRI_, (67) — ClassExpression_ и (74) — DataRange_.

Таблица 1. **Описание в метаязыке НФЗ манчестерского стиля языка OWL 2**

Интернационализированные идентификаторы ресурса (IRIs), целые числа, литералы и сущности			
1	fullIRI	=	/* IRI, как определено в [13], заключенное в пару символов < (U + 3C) и > (U + 3E) */
2	prefName	=	/* конечная последовательность символов, соответствующая продукции PNAME_NS из [14] и не соответствующая любому из терминалов ключевого слова синтаксиса */
3	abbrIRI	=	/* конечная последовательность символов, соответствующая продукции PNAME_LN из [14] */
4	simIRI	=	/* конечная последовательность символов, соответствующая продукции PN_LOCAL из [14] и не соответствующая любому из терминалов ключевого слова синтаксиса */
5	IRI	=	fullIRI / abbrIRI / simIRI;
6	nonNegInt	=	zero / posInt;
7	posInt	=	nonZero (digit);
8	digits	=	digit (digit);
9	digit	=	zero / nonZero;
10	nonZero	=	'1' / '2' / '3' / '4' / '5' / '6' / '7' / '8' / '9';
11	zero	=	'0';
12	clIRI	=	IRI;
13	Dtype	=	dtypeIRI / 'integer' / 'decimal' / 'float' / 'string';
14	dtypeIRI	=	IRI;
15	objPrIRI	=	IRI;
16	dPrIRI	=	IRI;
17	annPrIRI	=	IRI;
18	ind	=	indIRI / nodeID;
19	indIRI	=	IRI;
20	nodeID	=	/* конечная последовательность символов, соответствующая продукции BLANK_NODE_LABEL из [14]
21	lit	=	typLit / strLitNoLang / strLitWithLang / intLit / decLit / flPointLit;
22	typLit	=	lexVal '^^' Dtype;
23	strLitNoLang	=	quotStr;
24	strLitWith-Lang	=	quotStr langTag;
25	langTag	=	/* @ (U + 40) следовали за непустой последовательностью символов, соответствующих продукции langtag из [15] */
26	lexVal	=	quotStr;
27	quotStr	=	/* заключенная в пару символов (U + 22) конечная последовательность символов, в которой " (U + 22) и \ (U + 5C) встречаются только в парах вида \" (U + 5C, U + 22) и \\ (U + 5C, U + 5C) */
28	flPointLit	=	sgn flBody exp flPost;
29	flBody	=	digits fract / '.' digits ;
30	fract	=	'.' digits / true;
31	exp	=	'e' sgn digits / 'E' sgn digits / true;
32	flPost	=	'f' / 'F';
33	decLit	=	sgn digits '.' digits;
34	intLit	=	sgn digits;
35	sgn	=	'+' / '-' / true;

Продолжение табл. 1

36	entity	=	'Datatype' '(' Dtype ')' / 'Class' '(' clIRI ')' / 'ObjectProperty' '(' objPrIRI ')' / 'DataProperty' '(' dPrIRI ')' / 'AnnotationProperty' '(' annPrIRI ')' / 'NamedIndividual' '(' indIRI ')' ;
			Онтологии и аннотации
37	anns	=	'Annotations:' annAnnList;
38	ann	=	annPrIRI annTarg;
39	annTarg	=	nodeID / IRI / lit;
40	ontoDoc	=	(prefDecl) ontology;
41	prefDecl	=	'Prefix:' prefName fullIRI ;
42	ontology	=	'Ontology:' ontIRI_ (import) (anns) (frame);
43	ontIRI	=	IRI;
44	ontIRI_	=	ontIRI versIRI_ / true;
45	versIRI	=	IRI;
46	versIRI_	=	versIRI / true;
47	import	=	'Import:' IRI ;
48	frame	=	dtypeFr / clFr / objPrFr / dPrFr / annPrFr / indFr / misc;
			Выражения свойств и типов данных
49	objPrExpr	=	objPrIRI / invObjPr;
50	invObjPr	=	'inverse' objPrIRI;
51	dPrExpr	=	dPrIRI;
52	dRange	=	dConj 'or' dConj ('or' dConj) / dConj;
53	dConj	=	dPrim 'and' dPrim ('and' dPrim) / dPrim;
54	dPrim	=	not dAtom;
55	dAtom	=	Dtype / '{' litList '}' / dtypeRestr / '(' dRange ')';
56	dtypeRestr	=	Dtype '[' facet restrVal (',' facet restrVal) ']';
57	facet	=	'length' / 'minLength' / 'maxLength' / 'pattern' / 'langRange' / '<=' / '<' / '>=' / '>';
58	restrVal	=	lit;
			Описания
59	descr	=	conj 'or' conj ('or' conj) / conj;
60	conj	=	clIRI 'that' not restr ('and' not restr) / prim 'and' prim ('and' prim) /prim;
61	prim	=	not restr / not atom;
62	restr	=	objPrExpr 'some' prim / objPrExpr 'only' prim / objPrExpr 'value' ind / objPrExpr 'Self' / objPrExpr 'min' nonNegInt prim_ / objPrExpr 'max' nonNegInt prim_ / objPrExpr 'exactly' nonNegInt prim_ / dPrExpr 'some' dPrim / dPrExpr 'only' dPrim / dPrExpr 'value' lit / dPrExpr 'min' nonNegInt dPrim_ / dPrExpr 'max' nonNegInt dPrim_ / dPrExpr 'exactly' nonNegInt dPrim_;
63	prim_	=	prim / true;
64	dPrim_	=	dPrim / true;
65	atom	=	clIRI / '{' indList '}' / '(' descr ')';
			Фреймы и разное
66	dtypeFr	=	'Datatype:' Dtype ('Annotations:' annAnnList) dtypeFr_ ('Annotations:' annAnnList);
67	dtypeFr_	=	'EquivalentTo:' anns dRange / true;
68	litList	=	lit (',' lit);
69	annAnnList	=	anns_ ann (',' anns_ ann);
70	anns_	=	anns / true;

Окончание табл. 1

71	descrAnnList	=	anns_ descr (',' anns_ descr);
72	objPrChar-AnnList	=	anns_ objPrChar (',' anns_ objPrChar);
73	objPrEx-prAnnList	=	anns_ objPrExpr (',' anns_ objPrExpr);
74	dRangAnnList	=	anns_ dRange (',' anns_ dRange);
75	IRIAnnList	=	anns_ IRI (',' anns_ IRI);
76	dPrExprAnn-List	=	anns_ dPrExpr (',' anns_ dPrExpr);
77	annPrIRIAnn-List	=	anns_ annPrIRI (',' anns_ annPrIRI);
78	factAnnList	=	anns_ fact (',' anns_ fact);
79	indAnnList	=	anns_ ind (',' anns_ ind);
80	clFr	=	'Class:' clIRI ('Annotations:' annAnnList / 'SubClassOf:' descrAnnList / 'EquivalentTo:' descrAnnList / 'DisjointWith:' descrAnnList / 'DisjointUnionOf:' anns descr2List) / 'HasKey:' anns clFr_ (clFr_);
81	clFr_	=	objPrExpr / dPrExpr;
82	objPrFr	=	'ObjectProperty:' objPrIRI ('Annotations:' annAnnList / 'Domain:' descrAnnList / 'Range:' descrAnnList / 'Characteristics:' objPrCharAnnList / 'SubPropertyOf:' objPrExprAnnList / 'EquivalentTo:' objPrExprAnnList / 'DisjointWith:' objPrExprAnnList / 'InverseOf:' objPrExprAnnList / 'SubPropertyChain:' anns objPrExpr 'o' objPrExpr ('o' objPrExpr));
83	objPrChar	=	'Functional' / 'InverseFunctional' / 'Reflexive' / 'Irreflexive' / 'Symmetric' / 'Asymmetric' / 'Transitive';
84	dPrFr	=	'DataProperty:' dPrIRI ('Annotations:' annAnnList / 'Domain:' descrAnnList / 'Range:' dRangeAnnList / 'Characteristics:' anns 'Functional' / 'SubPropertyOf:' dPrExprAnnList / 'EquivalentTo:' dPrExprAnnList / 'DisjointWith:' dPrExprAnn-List);
85	annPrFr	=	'AnnotationProperty:' annPrIRI ('Annotations:' annAnnList) / 'Domain:' IRIAnn-List / 'Range:' IRIAnnList / 'SubPrOf:' annPrIRIAnnList;
86	indFr	=	'Individual:' ind ('Annotations:' annAnnList / 'Types:' descrAnnList / 'Facts:' factAnnList / 'SameAs:' indAnnList / 'DifferentFrom:' indAnnList);
87	fact	=	not fact_;
88	fact_	=	objPrFact / dPrFact;
89	not	=	'not' / true;
90	objPrFact	=	objPrIRI ind;
91	dPrFact	=	dPrIRI lit;
92	misc	=	'EquivalentClasses:' anns descr2List / 'DisjointClasses:' anns descr2List / 'EquivalentProperties:' anns objPr2List / 'DisjointProperties:' anns objPr2List / 'EquivalentProperties:' anns dPr2List / 'DisjointProperties:' anns dPr2List / 'SameIndividual:' anns ind2List / 'DifferentIndividuals:' anns ind2List;
93	descr2List	=	descr ',' descrList;
94	descrList	=	descr (',' descr);
95	objPr2List	=	objPrIRI ',' objPrList;
96	objPrList	=	objPrIRI (',' objPrIRI);
97	dPr2List	=	dPrIRI ',' dPrList;
98	dPrList	=	dPrIRI (',' dPrIRI);
99	ind2List	=	ind ',' indList;
100	indList	=	ind (',' ind);

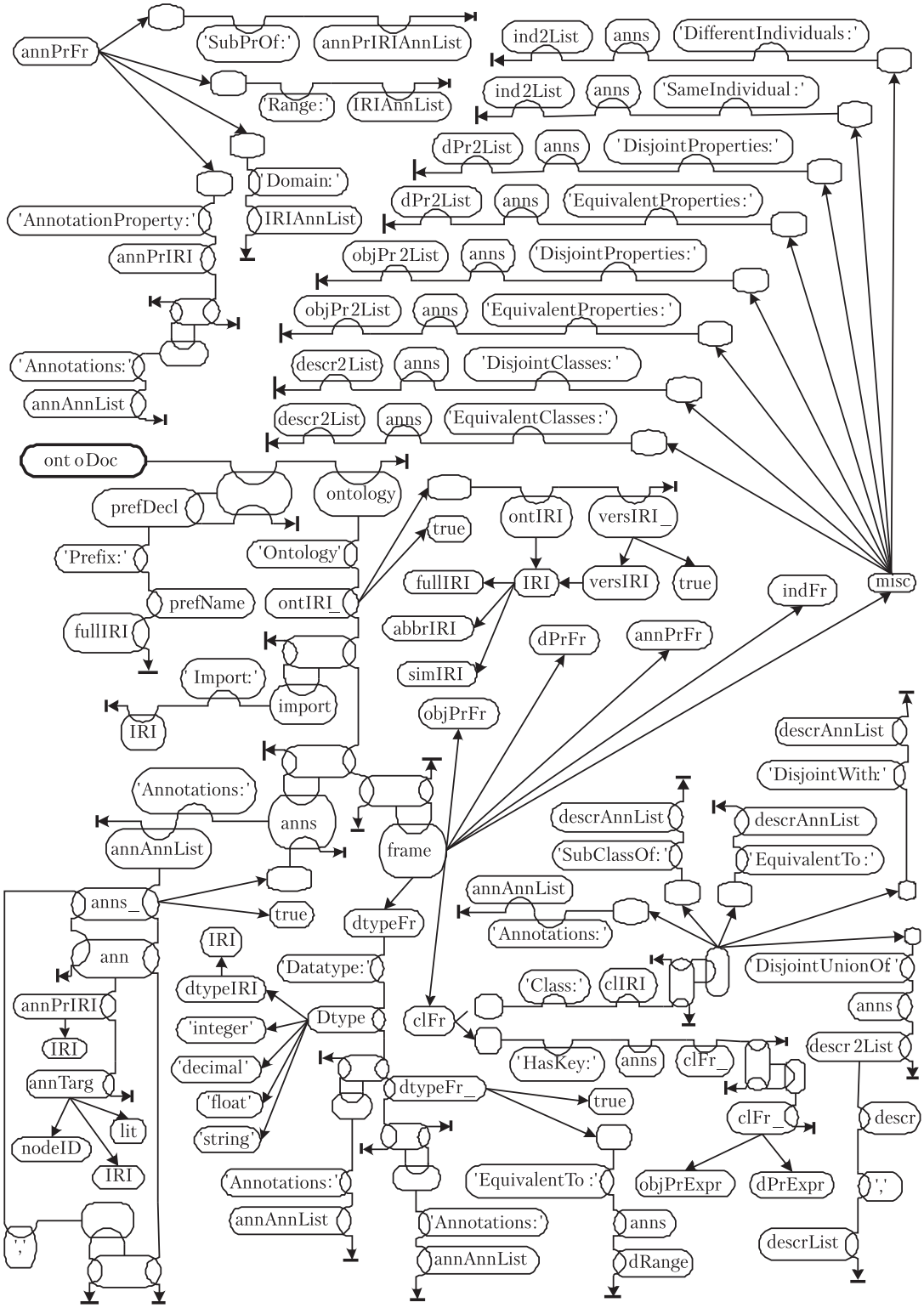

Рис. 3. Графическое описание в метаязыке НФЗ верхнего уровня структуры манчестерского синтаксиса языка OWL2

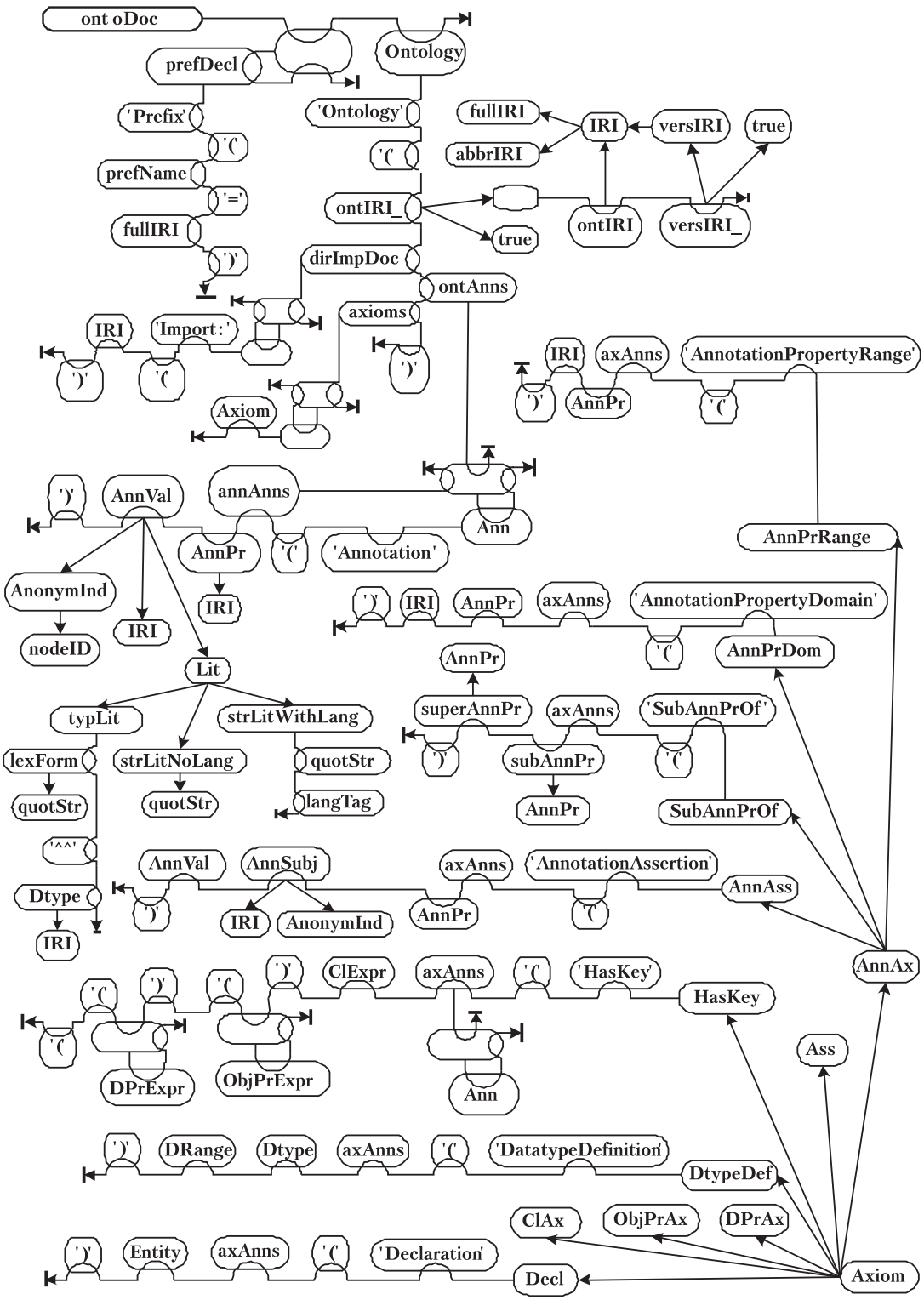

Рис. 4. Графическое описание в метаязыке НФЗ верхнего уровня структуры синтаксиса функционального стиля языка OWL2

Таблица 2. **Описание в метаязыке НФЗ синтаксиса функционального стиля языка OWL2**

		Целые числа, символы, строки, теги языка и идентификаторы узлов	
1	nonNegInt	=	/* непустая конечная последовательность цифр между 0 и 9 */
2	quotStr	=	/* заключенная в пару символов (U+22) конечная последовательность символов, в которой символы "(U+22) и \(U+5C) встречаются лишь в парах \"(U+5C, U+22) и \\(U+5C, U+5C) */
3	langTag	=	*/ @ (U+40) следовали за непустой последовательностью символов, соответствующих продукции langtag из [15] */
4	nodeID	=	/* конечная последовательность символов, соответствующая продукции BLANK_NODE_LABEL из [14] */
		Интернационализированные идентификаторы ресурсов (IRI)	
5	fullIRI	=	/* IRI, как определено в [13], заключенное в пару символов < (U+3C) > (U+3E) */
6	prefName	=	/* конечная последовательность символов, соответствующая продукции PNAME_NS из [14] */
7	abbrIRI	=	/* конечная последовательность символов, соответствующая продукции PNAME_LN из [14] */
8	IRI	=	fullIRI / abbrIRI;
		Онтологии	
9	ontoDoc	=	(prefDecl) Ontology;
10	prefDecl	=	'Prefix' '(' prefName '=' fullIRI ')' ;
11	Ontology	=	'Ontology' '(' ontIRI_ dirImpDoc ontAnns axioms ')';
12	ontIRI_	=	ontIRI versIRI_ / true;
13	ontIRI	=	IRI;
14	versIRI_	=	versIRI / true;
15	versIRI	=	IRI;
16	dirImpDoc	=	('Import' '(' IRI ')');
17	ontAnns	=	(Ann);
18	axioms	=	(Axiom);
19	AnnVal	=	AnonymInd / IRI / Lit;
20	axAnns	=	(Ann);
		Аннотации	
21	Ann	=	'Annotation' '(' annAnns AnnPr AnnVal ')';
22	annAnns	=	(Ann);
23	AnnAx	=	AnnAss / SubAnnPrOf / AnnPrDom / AnnPrRange;
24	AnnAss	=	'AnnotationAssertion' '(' axAnns AnnPr AnnSubj AnnVal ')';
25	AnnSubj	=	IRI / AnonymInd;
26	SubAnnPrOf	=	'SubAnnPrOf' '(' axAnns subAnnPr superAnnPr ')';
27	subAnnPr	=	AnnPr;
28	superAnnPr	=	AnnPr;
29	AnnPrDom	=	'AnnotationPropertyDomain' '(' axAnns AnnPr IRI ')';
30	AnnPrRange	=	'AnnotationPropertyRange' '(' axAnns AnnPr IRI ')';
		Сущности, литералы и анонимные индивидуумы	
31	Class	=	IRI;
32	Dtype	=	IRI;
33	ObjPr	=	IRI;
34	DPr	=	IRI;

Продолжение табл. 2

35	AnnPr	=	IRI;
36	Ind	=	NamInd / AnonymInd;
37	NamInd	=	IRI;
38	AnonymInd	=	nodeID;
39	Lit	=	typLit / strLitNoLang / strLitWithLang;
40	typLit	=	lexForm '^^' Dtype;
41	lexForm	=	quotStr;
42	strLitNoLang	=	quotStr;
43	strLitWithLang	=	quotStr langTag;
44	Decl	=	'Declaration' '(' axAnns Entity ')';
45	Entity	=	'Class' '(' Class ')' / 'Datatype' '(' Dtype ')' / 'ObjectProperty' '(' ObjPr ')' / 'DataProperty' '(' DPr ')' / 'AnnotationProperty' '(' AnnPr ')' / 'NamedIndividual' '(' NamInd ')';
			Выражения свойств
46	ObjPrExpr	=	ObjPr / InvObjPr;
47	InvObjPr	=	'ObjectInverseOf' '(' ObjPr ')';
48	DPrExpr	=	DPr;
			Диапазоны данных
49	DRange	=	Dtype / DInterOf / DUnOf / DComplOf / DOneOf / DtypeRestr;
50	DInterOf	=	'DataIntersectionOf' '(' DRange DRange (DRange) ')';
51	DUnOf	=	'DataUnionOf' '(' DRange DRange (DRange) ')';
52	DComplOf	=	'DataComplementOf' '(' DRange ')';
53	DOneOf	=	'DataOneOf' '(' Lit (Lit) ')';
54	DtypeRestr	=	'DatatypeRestriction' '(' Dtype constrFacet restrVal (constrFacet restrVal) ')';
55	constrFacet	=	IRI;
56	restrVal	=	Lit;
			Выражения класса
57	ClExpr	=	Class / ObjInterOf / ObjUnOf / ObjComplOf / ObjOneOf / ObjSomeValFrom / ObjAllValFrom / ObjHasVal / ObjHasSelf/ObjMinCard / ObjMaxCard / ObjExactCard / DSomeValFrom / DallValFrom / DhasVal / DMinCard / DMaxCard / DExactCard;
58	ObjInterOf	=	'ObjectIntersectionOf' '(' ClExpr ClExpr (ClExpr) ')';
59	ObjUnOf	=	'ObjectUnionOf' '(' ClExpr ClExpr (ClExpr) ')';
60	ObjComplOf	=	'ObjectComplementOf' '(' ClExpr ')';
61	ObjOneOf	=	'ObjectOneOf' '(' Ind (Ind) ')';
62	ObjSomeVal-From	=	'ObjectSomeValuesFrom' '(' ObjPrExpr ClExpr ')';
63	ObjAllValFrom	=	'ObjectAllValuesFrom' '(' ObjPrExpr ClExpr ')';
64	ObjHasVal	=	'ObjectHasValue' '(' ObjPrExpr Ind ')';
65	ObjHasSelf	=	'ObjectHasSelf' '(' ObjPrExpr ')';
66	ObjMinCard	=	'ObjectMinCardinality' '(' nonNegInt ObjPrExpr ClExpr_ ')';
67	ClExpr_	=	ClassExpression / true;
68	ObjMaxCard	=	'ObjectMaxCardinality' '(' nonNegInt ObjPrExpr ClExpr_ ')';
69	ObjExactCard	=	'ObjectExactCardinality' '(' nonNegInt ObjPrExpr ClExpr_ ')';
70	DSomeValFrom	=	'DataSomeValuesFrom' (DPrExpr (DPrExpr) DRange ')';
71	DAllValFrom	=	'DataAllValuesFrom' '(' DPrExpr (DPrExpr) DRange ')';

Продолжение табл. 2

72	DHasVal	=	'DataHasValue' '(' DPrExpr Lit ')';
73	DMinCard	=	'DataMinCardinality' '(' nonNegInt DPrExpr DRange_ ')';
74	DRange_	=	DRange / true;
75	DMaxCard	=	'DataMaxCardinality' '(' nonNegInt DPrExpr DRange_ ')';
76	DExactCard	=	'DataExactCardinality' '(' nonNegInt DPrExpr DRange_ ')';
			Аксиомы
77	Axiom	=	Decl / ClAx / ObjPrAx / DPrAx / DtypeDef / HasKey / Ass / AnnAx;
78	ClAx	=	SubClOf / EquCl / DisjCl / DisjUnion;
79	SubClOf	=	'SubClassOf' '(' axAnns subClExpr supClExpr ')';
80	subClExpr	=	ClExpr;
81	supClExpr	=	ClExpr;
82	EquCl	=	'EquivalentClasses' '(' axAnns ClExpr ClExpr (ClExpr) ')';
83	DisjCl	=	'DisjointClasses' '(' axAnns ClExpr ClExpr (ClExpr) ')';
84	DisjUnion	=	'DisjointUnion' '(' axAnns Class disjClExpr ')';
85	disjClExpr	=	ClExpr ClExpr (ClExpr);
86	ObjPrAx	=	SubObjPrOf / EquObjPr / DisjObjPr / InvObjPr / ObjPrDom / ObjPrRange / FunctObjPr / InvFunctObjPr / ReflObjPr / IrreflObjPr / SymmObjPr / AsymmObjPr / TransObjPr;
87	SubObjPrOf	=	'SubObjectPropertyOf' '(' axAnns subObjPrExpr supObjPrExpr ')';
88	subObjPrExpr	=	ObjPrExpr / prExprChain;
89	prExprChain	=	'ObjectPropertyChain' '(' ObjPrExpr ObjPrExpr (ObjPrExpr) ')';
90	supObjPrExpr	=	ObjPrExpr;
91	EquObjPr	=	'EquivalentObjectProperties' '(' axAnns ObjPrExpr ObjPrExpr (ObjPrExpr) ')' ;
92	DisjObjPr	=	'DisjointObjectProperties' '(' axAnns ObjPrExpr ObjPrExpr (ObjPrExpr) ')';
93	ObjPrDom	=	'ObjectPropertyDomain' '(' axAnns ObjPrExpr ClExpr ')';
94	ObjPrRange	=	'ObjectPropertyRange' '(' axAnns ObjPrExpr ClExpr ')';
95	InvObjPr	=	'InverseObjectProperties' '(' axAnns ObjPrExpr ObjPrExpr ')';
96	FunctObjPr	=	'FunctionalObjectProperty' '(' axAnns ObjPrExpr ')';
97	InvFunctObjPr	=	'InverseFunctionalObjectProperty' '(' axAnns ObjPrExpr ')';
98	ReflObjPr	=	'ReflexiveObjectProperty' '(' axAnns ObjPrExpr ')';
99	IrreflObjPr	=	'IrreflexiveObjectProperty' '(' axAnns ObjPrExpr ')';
100	SymmObjPr	=	'SymmetricObjectProperty' '(' axAnns ObjPrExpr ')';
101	AsymmObjPr	=	'AsymmetricObjectProperty' '(' axAnns ObjPrExpr ')';
102	TransObjPr	=	'TransitiveObjectProperty' '(' axAnns ObjPrExpr ')';
103	DPrAx	=	SubDPrOf / EquDPr / DisjDPr / DPrDom / DPrRange / FunctDPr;
104	SubDPrOf	=	'SubDataPropertyOf' '(' axAnns subDPrExpr supDPrExpr ')';
105	subDPrExpr	=	DPrExpr;
106	supDPrExpr	=	DPrExpr;
107	EquDPr	=	'EquivalentDataProperties' '(' axAnns DPrExpr DPrExpr (DPrExpr) ')';
108	DisjDPr	=	'DisjointDataProperties' '(' axAnns DPrExpr DPrExpr (DPrExpr) ')';
109	DPrDom	=	'DataPropertyDomain' '(' axAnns DPrExpr ClExpr ')';
110	DPrRange	=	'DataPropertyRange' '(' axAnns DPrExpr DRange ')';
111	FunctDPr	=	'FunctionalDataProperty' '(' axAnns DPrExpr ')';
112	DtypeDef	=	'DatatypeDefinition' '(' axAnns Dtype DRange ');

Окончание табл. 2

113	HasKey	=	'HasKey' '(' axAnns ClExpr '(' (ObjPrExpr) ')' '(' (DPrExpr) ')' ')';
114	Ass	=	SameInd / DiffInd / ClAss / ObjPrAss / NegObjPrAss / DPrAss / NegDPrAss;
115	sourceInd	=	Ind;
116	targetInd	=	Ind;
117	targetVal	=	Lit;
118	SameInd	=	'SameIndividual' '(' axAnns Ind Ind (Ind) ')';
119	DiffInd	=	'DifferentIndividuals' '(' axAnns Ind Ind (Ind) ')';
120	ClAss	=	'ClassAssertion' '(' axAnns ClExpr Ind ');
121	ObjPrAss	=	'ObjectPropertyAssertion' '(' axAnns ObjPrExpr sourceInd targetInd ')';
122	NegObjPrAss	=	'NegativeObjectPropertyAssertion' '(' axAnns ObjPrExpr sourceInd targetInd ')';
123	DPrAss	=	'DataPropertyAssertion' '(' axAnns DPrExpr sourceInd targetVal ')';
124	NegDPrAss	=	'NegativeDataPropertyAssertion' '(' axAnns DPrExpr sourceInd targetVal ')';

Главным результатом сопоставления пар рассмотренных описаний является следующее утверждение.

Утверждение 1. Выразительных возможностей метаязыка НФЗ достаточно для формального описания языка онтологий OWL 2.

Таким образом, в статье исследованы выразительные возможности метаязыка НФЗ по отношению к разным версиям синтаксиса языка онтологий OWL 2 — центрального языка семантического стека Тима Бернерс-Ли. Даны формальные текстовые и графические описания этих языков, чье наличие гарантирует реализуемость версий языка онтологий OWL 2 с реализацией интерпретатора метаязыка НФЗ. Показано, что выразительные возможности метаязыка НФЗ для формального описания языка онтологий OWL 2 близки выразительным возможностям метаязыка EBNF.

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

1. Berners-Lee T. Web Architecture from 50,000 feet. URL: https://www.w3.org/DesignIssues/Architecture.html
2. Web Architecture: Describing and Exchanging Data. W3C Note 7 June 1999. URL: https://www.w3.org/1999/04/WebData
3. Berners-Lee T., Hall W., Hendler J.A., O'Hara K., Shadbolt N., Weitzner D.J. A Framework for Web Science. *Foundations and Trends in Web Science*. 2006. 1, № 1. P. 1—130. 2006. doi: https://doi.org/10.1561/1800000001
4. Кургаев А.Ф., Григорьев С.Н. Нормальные формы знаний. *Допов. Нац. акад. наук Укр*. 2015. № 11. С. 36—43. doi: https://doi.org/10.15407/dopovidi2015.11.036
5. Кургаев А.Ф., Григорьев С.Н. Метаязык нормальных форм знаний. *Кибернетика и системный анализ*. 2016. **52**, № 6. С. 11—20. doi: https://doi.org/10.1007/s10559-016-9885-3
6. Кургаев А.Ф., Григорьев С.Н. Определение формальных языков в метаязыке нормальных форм знаний. *Пробл. програмування*. 2017. № 4. С. 37—50.
7. OWL 2 Web Ontology Language: Document Overview (2nd ed.). W3C Recommendation 11 December 2012. URL: http://www.w3.org/TR/2012/REC-owl2-overview-20121211/
8. OWL 2 Web Ontology Language: Primer (2nd ed.). W3C Recommendation 11 December 2012. URL: http://www.w3.org/TR/owl-primer
9. OWL 2 Web Ontology Language: Manchester Syntax (2nd ed.). W3C Working Group Note 11 December 2012. URL: http://www.w3.org/TR/2012/NOTE-owl2-manchester-syntax-20121211/

10. OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (2nd ed.). W3C Recommendation 11 December 2012. URL: http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/
11. OWL 2 Web Ontology Language: XML Serialization (2nd ed.). W3C Recommendation 11 December 2012. URL: http://www.w3.org/TR/2012/REC-owl2-xml-serialization-20121211/
12. OWL 2 Web Ontology Language: Mapping to RDF Graphs (2nd ed.). W3C Recommendation 11 December 2012. URL: http://www.w3.org/TR/2012/REC-owl2-mapping-to-rdf-20121211/
13. Duerst M., Suignard M. RFC 3987: Internationalized Resource Identifiers (IRIs). IETF, January 2005. URL: http://www.ietf.org/rfc/rfc3987.txt
14. SPARQL 1.1: Query Language. W3C Recommendation 21 March 2013. URL: http://www.w3.org/TR/2013/REC-sparql11-query-20130321/
15. Tags for Identifying Languages. BCP: 47. URL: http://www.rfc-editor.org/rfc/bcp/bcp47.txt

Поступило в редакцию 06.11.2017

REFERENCES

1. Berners-Lee, T. Web Architecture from 50,000 feet. Retrieved from https://www.w3.org/DesignIssues/Architecture.html.
2. Web Architecture: Describing and Exchanging Data. W3C Note 7 June 1999. Retrieved from https://www.w3.org/1999/04/WebData
3. Berners-Lee, T., Hall, W., Hendler, J. A., O'Hara, K., Shadbolt, N. & Weitzner, D. J. (2006). A Framework for Web Science. Foundations and Trends in Web Science. 1, No. 1, pp. 1-130. doi: https://doi.org/10.1561/1800000001
4. Kurgaev, A. & Grygoryev, S. (2015). The normal forms of knowledge. Dopov. Nac. akad. nauk Ukr., No. 11, pp. 36-43 (in Russian). doi: https://doi.org/10.15407/dopovidi2015.11.036
5. Kurgaev, A. & Grygoryev, S. (2016). Metalanguage of Normal Forms of Knowledge. Cybernetics and Systems Analysis, 52, No. 6, pp. 839-848. doi: https://doi.org/10.1007/s10559-016-9885-3
6. Kurgaev, A. & Grygoryev, S. (2017). The definition of formal languages in the meta language of normal forms of knowledge. Programming problems, No. 4, pp. 37-50 (in Russian).
7. OWL 2 Web Ontology Language: Document Overview (2nd ed.). W3C Recommendation 11 December 2012. Retrieved from http://www.w3.org/TR/2012/REC-owl2-overview-20121211/
8. OWL 2 Web Ontology Language: Primer (2nd ed.). W3C Recommendation 11 December 2012. Retrieved from http://www.w3.org/TR/owl-primer
9. OWL 2 Web Ontology Language: Manchester Syntax (2nd ed.). W3C Working Group Note 11 December 2012. Retrieved from http://www.w3.org/TR/2012/NOTE-owl2-manchester-syntax-20121211/
10. OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (2nd ed.). W3C Recommendation 11 December 2012. Retrieved from http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/
11. OWL 2 Web Ontology Language: XML Serialization (2nd ed.). W3C Recommendation 11 December 2012. Retrieved from http://www.w3.org/TR/2012/REC-owl2-xml-serialization-20121211/
12. OWL 2 Web Ontology Language: Mapping to RDF Graphs (2nd ed.). W3C Recommendation 11 December 2012. Retrieved from http://www.w3.org/TR/2012/REC-owl2-mapping-to-rdf-20121211/
13. Duerst M., Suignard M. RFC 3987: Internationalized Resource Identifiers (IRIs). IETF, January 2005. Retrieved from http://www.ietf.org/rfc/rfc3987.txt
14. SPARQL 1.1: Query Language. W3C Recommendation 21 March 2013. Retrieved from http://www.w3.org/TR/2013/REC-sparql11-query-20130321/
15. Tags for Identifying Languages. BCP: 47. Retrieved from http://www.rfc-editor.org/rfc/bcp/bcp47.txt

Received 06.11.2017

О.П. Кургаєв

Інститут кібернетики ім. В.М. Глушкова НАН України, Київ
E-mail: afkurgaev@ukr.net

НОВЕ ВИЗНАЧЕННЯ МОВИ ВЕБ-ОНТОЛОГІЙ OWL2

Дано у метамові нормальних форм знань (НФЗ) описи манчестерського синтаксису й синтаксису функціонального стилю мови веб-онтологій OWL 2 — центральної мови семантичного стека Тіма Бернерс-Лі. Наявність таких описів гарантує реалізуємість мови OWL 2 з реалізацією інтерпретатора метамови НФЗ. Показано, що виразні можливості метамови НФЗ для формального опису OWL 2 цілком порівняні з виразними можливостями метамови Extended Backus-Naur Form.

Ключові слова: *метамова нормальних форм знань, формальний опис, мова веб-онтологій OWL, Semantic Web, манчестерський синтаксис, синтаксис функціонального стилю.*

A.F. Kurgaev

V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kiev
E-mail: afkurgaev@ukr.net

NEW DEFINITION OF THE WEB ONTOLOGY LANGUAGE OWL2

In the metalanguage of normal forms of knowledge (NFK), we give descriptions of the Manchester syntax and syntax of the functional style of the web ontology language OWL 2 — the central language of the semantic stack by Tim Berners-Lee. The availability of such descriptions guarantees that OWL 2 can be implemented as long as the interpreter of the NFK meta-language is implemented first. It is shown that the expressive capabilities of the NFK meta-language for the description of OWL 2 are quite comparable with the expressive capabilities of the meta-language of Extended Backus-Naur Form.

Keywords*: metalanguage of normal forms of knowledge, formal description, OWL web ontology language, semantic Web, Manchester syntax, functional style syntax.*