SWRL2SPIN support for SWRL built-ins

SWRL		**Conversion category**	**SPIN / SPARQL**		
Built-in Function	**Category**		**function**	**op**	**expression**
greaterThan(?x,?y)	Compare	binary filter	sp:gt	>	FILTER (?x > ?y)
greaterThanOrEqual(?x,?y)	Compare	binary filter	sp:ge	>=	FILTER (?x >= ?y)
lessThan(?x,?y)	Compare	binary filter	sp:lt	<	FILTER (?x < ?y)
lessThanOrEqual(?x,?y)	Compare	binary filter	sp:le	<=	FILTER (?x <= ?y)
equal(?x,?y)	Compare	binary filter	sp:eq	=	FILTER (?x = ?y)
notEqual(?x,?y)	Compare	binary filter	sp:ne	!=	FILTER (?x != ?y)
add(?y,?x_1,?x_2,…,?x_n)	Math	associative infix assign	sp:add	+	BIND ((((?x_1 + ?x_2) +…) + ?x_n) AS ?y)
multiply(?y,?x_1,?x_2,…,?x_n)	Math	associative infix assign	sp:mul	*	BIND ((((?x_1 * ?x_2) * …) * ?x_n) AS ?y)
subtract(?z,?x,?y)	Math	binary infix assign	sp:sub	-	BIND ((?x - ?y) AS ?z)
divide(?z,?x,?y)	Math	binary infix assign	sp:divide	/	BIND ((?x / ?y) AS ?z)
unaryPlus(?y,?x)	Math	unary assign	sp:unaryPlus	+	BIND ((+?x) AS ?y)
unaryMinus(?y,?x)	Math	unary assign	sp:unaryMinus	-	BIND ((-?x) AS ?y)
abs(?y,?x)	Math	assign function	sp:abs		BIND (abs(?x) AS ?y)
ceiling(?y,?x)	Math	assign function	sp:ceil		BIND (ceil(?x) AS ?y)
floor(?y,?x)	Math	assign function	sp:floor		BIND (floor(?x) AS ?y)
round(?y,?x)	Math	assign function	sp:round		BIND (round(?x) AS ?y)
mod(?z,?x,?y)	Math	assign function	spif:mod		BIND (spif:mod(?x, ?y) AS ?z)
stringConcat(?y,?x_1,?x_2,…,?x_n)	Strings	assign function	sp:concat		BIND (CONCAT(?x_1,…, ?x_n) AS ?y)
stringLength(?y,?x)	Strings	assign function	sp:strlen		BIND (STRLEN(?x) AS ?y)
upperCase(?y,?x)	Strings	assign function	sp:ucase		BIND (UCASE(?x) AS ?y)
lowerCase(?y,?x)	Strings	assign function	sp:lcase		BIND (LCASE(?x) AS ?y)
substringBefore(?y,?x_1,?x_2)	Strings	assign function	sp:strbefore		BIND (STRBEFORE(?x_1, ?x_2) AS ?y)
substringAfter(?y,?x_1,?x_2)	Strings	assign function	sp:strafter		BIND (STRAFTER(?x_1, ?x_2) AS ?y)
substring(?y,?x,?s,?l)	Strings	assign function	sp:substr		BIND (SUBSTR(?x, ?s, ?l) AS ?y)
replace(?y,?str,?s_1,?s_2)	Strings	assign function	sp:replace		BIND (REPLACE(?str, ?s_1, ?s_2) AS ?y)
endsWith(?x,?y)	Strings	filter function	sp:strends		FILTER STRENDS(?x, ?y)
startsWith(?x,?y)	Strings	filter function	sp:strstarts		FILTER STRSTARTS(?x, ?y)
contains(?x,?y)	Strings	filter function	sp:contains		FILTER CONTAINS(?x, ?y)
matches(?x,?y)	Strings	filter function	sp:regex		FILTER REGEX(?x, ?y)
tokenize(?x,?y,?z)	Strings	magic property	spif:split		?x spif:split (?y ?z) .
integerDivide(?z,?x,?y)	Math	complex assign			BIND (spif:cast(?x / ?y, xsd:integer) AS ?z)
pow(?pow,?x,?n)	Math	complex assign			BIND (spif:cast((((?x_1 * ?x_2) * …) * ?x_n), xsd:integer) AS ?pow) .
normalizeSpace(?y,?x)	Strings	complex assign			BIND (REPLACE(REPLACE(REPLACE(?x, "\\s+", " "), "^\\s+", ""), "\\s+$", "") AS ?y)
date(?y,?year,?month,?day)	Date, Time, Duration	complex assign			BIND (spif:cast(CONCAT(spif:cast(?year, xsd:string), "-", spif:cast(?month, xsd:string), "-", spif:cast(?day, xsd:string)), xsd:date) AS ?y)
containsIgnoreCase(?s_1,?s_2)	Strings	complex filter			FILTER CONTAINS(LCASE(?s_1),LCASE(?s_2))
stringEqualIgnoreCase(?s_1,?s_2)	Strings	complex filter			FILTER (LCASE(?s_1) = LCASE(?s_2))

SWRL		Conversion category	SPIN / SPARQL		
Built-in Function	**Category**		**function**	**op**	**expression**
empty(?list)	Lists	complex filter			FILTER (?list = rdf:nil)
first(?e,?list)	Lists	complex expr			?list rdf:first ?e .
rest(?e,?list)	Lists	complex expr			?list rdf:rest ?e .
member(?e,?list)	Lists	complex expr			?list (rdf:rest)*/rdf:first ?e .

length(?length,?list)	Lists	complex expr			{
					SELECT ?x ?list (COUNT(?e) AS ?length)
					WHERE {
					?list (rdf:rest)*/rdf:first ?e .
					}
					GROUP BY ?x ?list
					} . *# ?x is an instance with a property whose value is the list ?list*