EIOPA Solvency II validations syntax

Ver 1.1

The document contains non-binding information, and is subject to substantial further changes

LAST UPDATE: 01/06/2016

INDEX

I Modification history

Date	Main change description
30/09/2015	First version of the document
15/07/2016	'rNNN' & 'cNNN' syntax added to the document. Replacing syntax 'for every' with 'not(isfallback)'

II Introduction

Aim of this document is to describe syntax, wording and patterns used in definition of business rules for Solvency II taxonomy version 2.0.1. It is complementary documentation to the EIOPA_SolvencyII_Validations.xlsx spreadsheet that contains entire set of business validations applicable to 2.0.1 release.

III Syntax use cases

III.1   Generic mathematical and logical operators

Below table describes basic operators used in business rules

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
=	Equation
>=	Greater than or equal
<=	Less than or equal
<>	Not equal (other than)
Sum()	Calculates summation of components inside the parenthesis
Max()	Finds maximum value from the components inside the parenthesis
Min()	Finds minimum value from the components inside the parenthesis
Abs()	Returns absolute value from the components inside the parenthesis
And	Both components must be true
Or	At least one component must be true

III.2   Syntax specific for EIOPA Solvency II validations

Some syntax used in Solvency II validations is specific for EIOPA project. Below particular case with explanation and examples are provided.

III.2.1   Data type constrains

Data type constrain is used to identify applicable patterns for a given reportable fact. In majority of the cases it refers to one of the ISO codification standards, like ISO 4217 for currencies or ISO 8601 for dates.

Examples:

Validation ID	Validation	Explanation
BV2	{C0390} - data type constrains [ISO 8601: (yyyy-mm-dd)]	Value in column C0390 must be in line with the ISO 8601 format (yyyy-mm-dd)

BV58	{C0080} - data type constrains [ISO 4217: alphabetic code]	Value in column C0080 must be in line with the ISO 4217 format
BV7	{c0040} - data type constrains [ISO 3166-1: alpha-2 code;or 'XA';or 'EU';or "AA"]	Value in column C0040 must be in line with the ISO 3166-1 alpha-2 code format but additionally possible values are also "XA", "EU" and "AA" which do not belong to the standard
BV36	{Z0030} - data type constrains [ISO 4217: alphabetic code + Total/NA]	Value on z-axis Z0030 must be in line with ISO 4217 format and additionally it can be reported as a "Total/NA"

NOTE:

This type of validation is represented in a taxonomy as a reference to list of domain members defined in the dictionary or by data type as an XML attribute, hence technically no XBRL formulas are generated for data type constrains.

III.2.2 'Empty'

This operator is used to check whether particular reportable element was or was not reported.

Examples:

Validation ID	Validation	Explanation
EV48	{er0010,ec0020} <> empty	Cell er0010,ec0020 in template E.03.01 must not be empty (*information about template code is provided elsewhere*)
BV237	If {c0290} like '##75' or {c0290} like '##95' then {c0280} = empty	If for a given row in open table S.06.02, third and fourth digits from code reported in column c0290 is 75 or 95, then column c0280 must be empty (*information about template code is provided elsewhere*)

III.2.3 'sNNN'

This expression is used to indicate situations where component of the equation is a summation of cells for a given column, row and/or z-axis. *sNNN* is always preceded by the *Sum()* operator

Examples:

Validation ID	Validation	Explanation
BV140	{S.02.01, r0770,c0010}>= sum({S.31.01, c0140,(sNNN)})	Value in cell r0770,c0010 in table S.02.01 must be greater or equal to summation of values from all rows for column c0140 in table S.31.01. *Note: If there will be drop down list on z-axis*

Validation ID	Validation	Explanation
BV513	sum({S.29.04, r0110,c0040,(SNNN)}) = {S.29.03, r0200,c0050} + {S.29.03, r0200,c0060}	Summation of values from cell r0100,c0040 for all z-axis combinations in table S.29.04 must be equal to summation of cells r0200,c0050 and r0200,c0060 from table S.29.03

in table S.31.01, it will be multiplied by these options as well

III.2.4 Dictionary element reference

Since some of the reported facts are components of the dictionary (e.g. *s2c_SE:x10* which is an domain member from the SE domain and its label is *Undertakings pursuing both life and non-life insurance activity*), they are also used in a number of business rules. In the expressions, these cases are identified by putting relevant dictionary component within the square brackets.

Examples:

Validation ID	Validation	Explanation
BV339	If{S.01.02, r0190,c0010} = [s2c_AP:x9] then {S.01.01, r0370,c0010}=[s2c_CN:x1]	If value reported in cell r0190,c0010 in table S.01.02 is *s2c_AP:x9* (*Use of transitional measure on the risk-free interest rate*)*,* then value of cell r0370,c0010 in table S.01.01 must be *s2c_CN:x1* (*Reported*)
BV603	If {z0020}=[s2c_PU:x60] then {z0030}<>empty	If a value reported on z-axis z0020 is *s2c_PU:x60* (The items reported refer to a RFF/MP), then z-axis z0030 must not be empty

III.2.5 'Like'/'not like'

This operator provides mechanism to distinguish pattern or a given sign from the reported element. It is used primarily to filter out particular rows from open tables.

Examples:

Validation ID	Validation	Explanation
EV19	If {c0290} like '##8#' then {ec0141} <> empty	If for a given row in open table SE.06.02, third digit from four digit code reported in column c0290 is 8, then column ec0141 must not be empty (*information about template code is provided elsewhere*)
BV4	If {c0290} like '##3#' or {c0290} like '##4#' or {c0290} like '##71' or {c0290} like '##72' or {c0290} like '##73' or {c0290} like '##75' or {c0290} like '##09' or	If for a given row in open table S.06.02, third digit from code reported in column c0290 is either 3 or 4 or 9 or third and fourth digits are 71 or 72 or 73 or 75 or 09 then column c0390 must be empty (*information about template code is provided elsewhere*)

	{c0290} like '##9#' then {c0390} = empty	
BV368	{S.02.01, r0150,c0010}=sum({S.06.02, c0170,(sNNN)}); Filter - {S.06.02, c0290} like '##2#' and {S.06.02, c0090}=[s2c_LB:x91]	Value in cell r0150,c0010 in table S.02.01 must be equal to summation of values from column c0170 in table S.06.02 for a specific rows defined by the filter. Filter takes out only rows, where third digit in column c0290 is 2 and where value reported in column c0090 is "s2c_LB:x91" (*Neither unit-linked nor index-linked*)
TV11	si1495 like "LEI/[A-Z0-9]{20}" or "None"	Value reported for metric si1495 must be in line with LEI which is 20 character alphanumeric code, preceded by "LEI/". The only other accepted value is "None"
TV4	dim:GO like "LEI/[A-Z0-9]{20}" or "None"	Value reported for dimension GO must be in line with LEI which is 20 character alphanumeric code, preceded by "LEI/". The only other accepted value is "None"

III.2.6   Conditional validations

These validations are represented by *If x then y* notation. Often, logical test (x) and the result if true (y) are complex expressions and are using other operators described in this document.

Examples:

Validation ID	Validation	Explanation
BV565_1	If {r0020,c0010}=[s2c_AP:x34] then {r0110,c0060}=MAX(0,({r0110,c0020}-{r0110,c0030})-({r0110,c0040}-{r0110,c0050}))	If value reported in cell r0020,c0010 is *s2c_AP:x34 (Simplifications not used)* then value in cell r0110,c0060 must be equal to the maximum value of either 0 or result of subtracting r0110,c0030; r0110,c0040 and r0110,c0050 from r0110,c0020

III.2.7   'not(isfallback)'

This expression is used to assure that the report is consistent and that reported information is provided in correct locations.

Examples:

Validation ID	Validation	Explanation
BV445	If not(isfallback({S.15.02, c0060})) then	If there is a row reported in table S.15.02, the row with the same key

	not(isfallback({S.15.01, c0090}))	must be reported in table S.15.01. Column codes are provided as examples of datapoints that must be provided in each table.

III.2.8 'Has to be reported'

This expression checks if any cell from the specified "has to be reported" table exists in the report.

Examples:

Validation ID	Validation	Explanation
BV337	If {S.01.02, r0150,c0010}=[s2c_PU:x4] or {S.01.02, r0170,c0010}=[s2c_PU:x51] then {SR.01.01} has to be reported	If value in cell r0150,c0010 is *s2c_PU:x4* or value in cell r0170,c0010 is *s2c_PU:x51* in table S.01.02, then table SR.01.01 must contain at least one reported fact

III.2.9 'rNNN' & 'cNNN'

This expression is used to indicate situation that cross template validation should be executed for a range of rows ('rNNN') or columns ('cNNN') from one of tables that it refers.

Examples:

Validation ID	Validation	Explanation
BV239_2	If {SR.01.01, r0790,c0010}=[s2c_CN:x1] then {SR.12.01, r0120,cNNN}<=0	If value in cell r0790,c0010 in table SR.01.01 is *s2c_CN:x1*, then value reported in SR.12.01 table in row r0120 must be smaller or equal to zero for all the columns specified in the columns range (*information about columns range is provided elsewhere*)