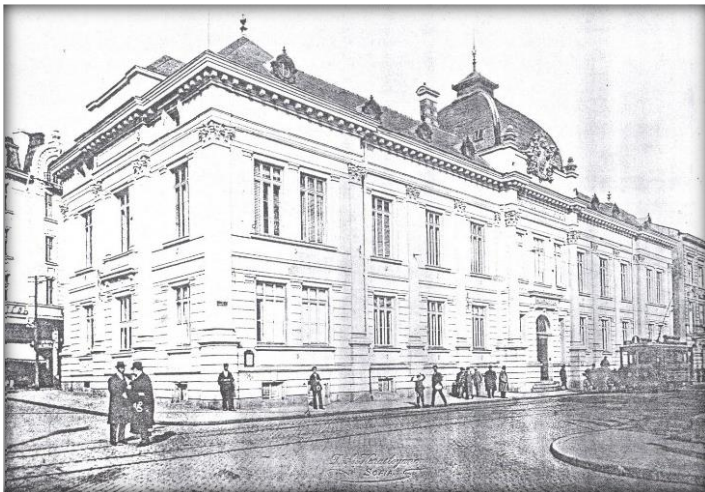


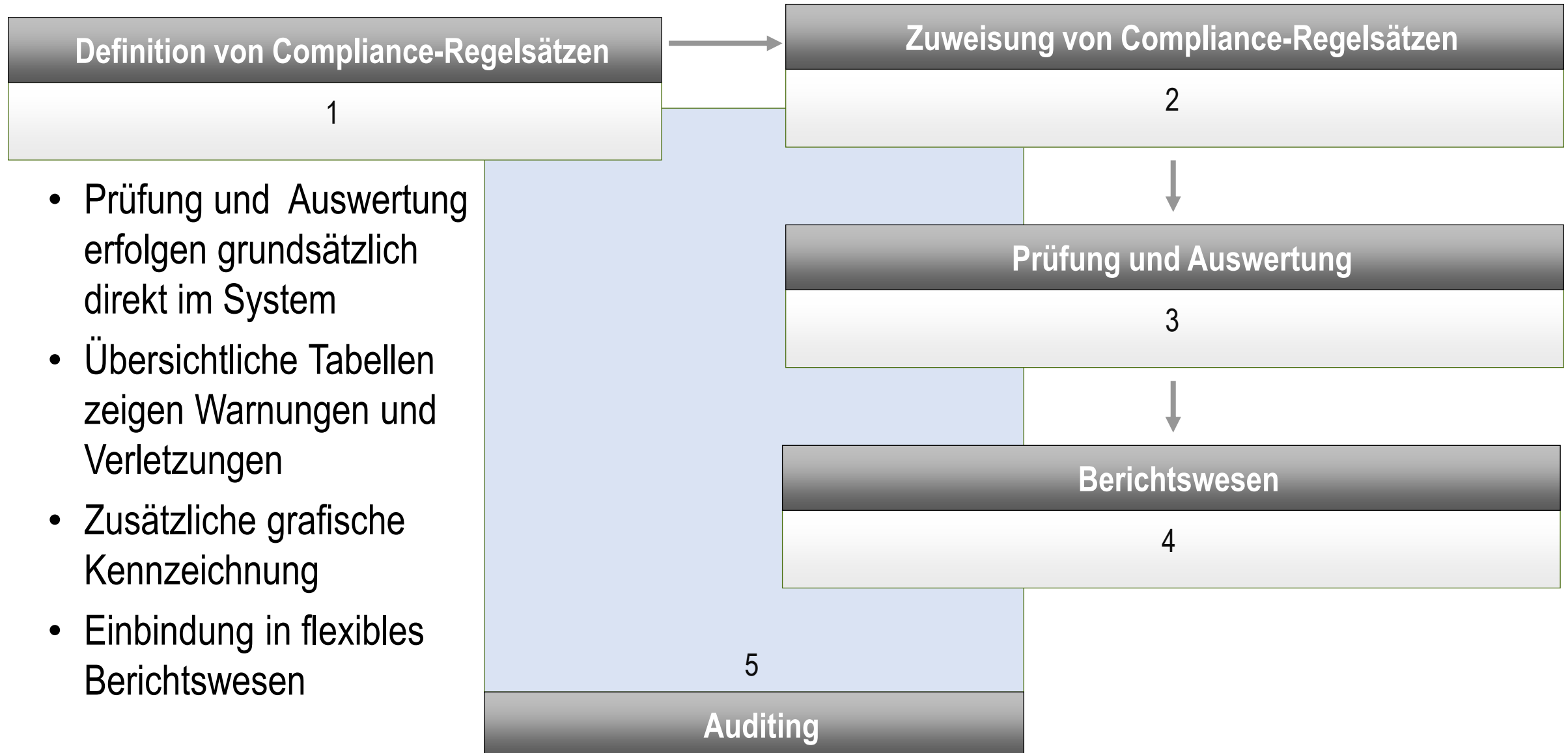


Eurorisk Systems Ltd
General Kiselov Str. 31
9002 Varna, Bulgaria
Tel. +359 52 612 367
Fax +359 52 612 371
E-mail: info@eurorisksystems.com
Web: www.eurorisksystems.com

Compliance Checks

Workflow Concept

Dr. Anatoliy Antonov
Dr. Ventsislav Nikolov
Eurorisk Systems Ltd.



Definition von Compliance-Regelsätzen

- Aufbau der Compliance-Strukturen in Baumform
- Hinterlegung der Einzelregeln direkt an den Baumknoten
- Definition von Verletzungs- und Warngrenzen
- Charakteristiken: regulatorisch, allgemein, individuell etc.
- Auditing bei Regeln

Zuweisung von Compliance-Regelsätzen

2

Regulatorische Regelsätze

- **Regulatorische Richtlinien (FINMA, SBV, InvFG etc.)**
- Definition analog zu allgemeinen Regelsätzen
- Vordefinierte Templates
- Einfache Anpassung bei Änderung der Regulatorien

Allgemeine Regelsätze

- **Richtlinien gültig für Portfoliogruppen (standardisierte Strategie , Anlegerprofil etc.)**
- Zugriff auf sämtliche Kriterien aus den Titel- und Portfolio-Stammdaten
- Vereinfachte Regeldefinition über Kriterienselektion und logische Verknüpfung

Individuelle Regelsätze

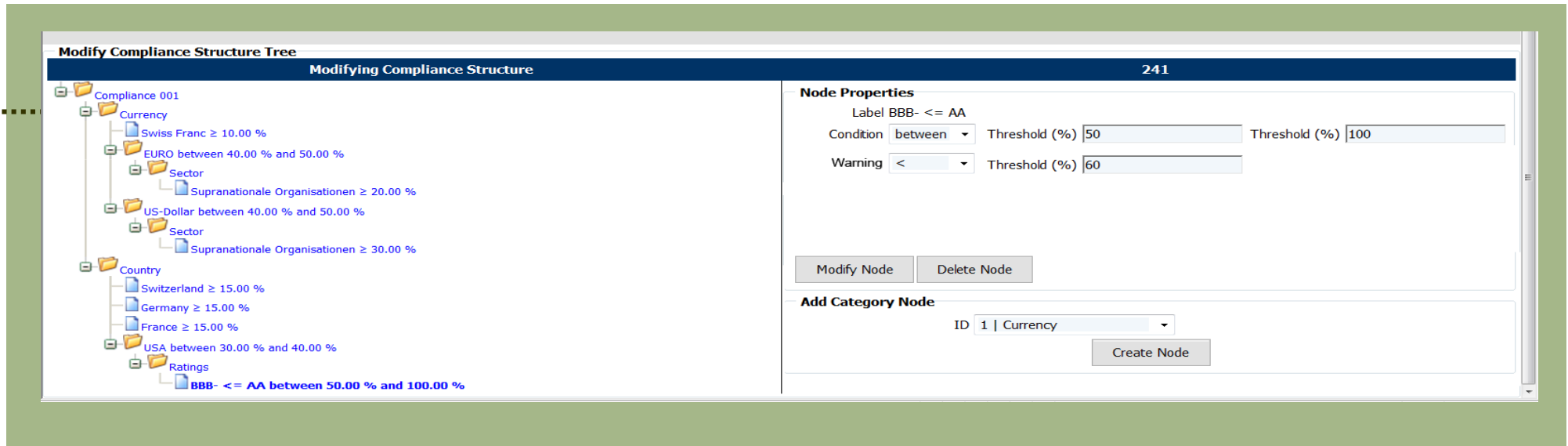
- **Richtlinien gültig für Portfoliogruppen (individualisierte Strategie , Einzelportfolio-Regeln etc.)**
- Erstellung wie allgemeine Regelsätze
- Zusätzliche Möglichkeiten durch flexiblen Formel-Editor zur Regeldefinition (inkl. Sensitivitäten)

Definition von Compliance-Regelsätzen

- Aufbau der Compliance-Strukturen in Baumform
- Hinterlegung der Einzelregeln direkt an den Baumknoten
- Definition von Verletzungs- und Warngrenzen
- Charakteristiken: regulatorisch, allgemein, individuell etc.
- Auditing bei Regeln

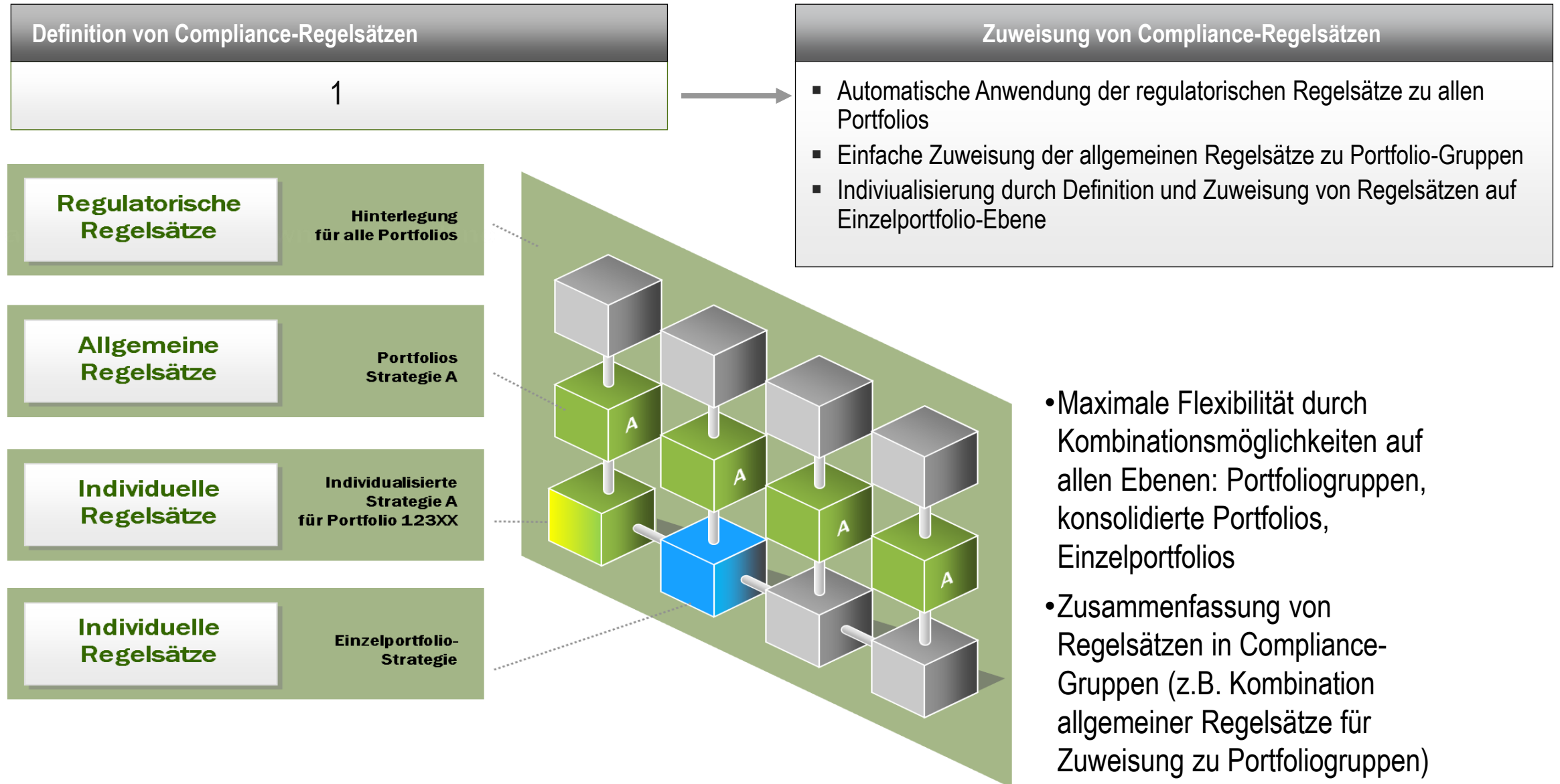
Zuweisung von Compliance-Regelsätzen

2



The screenshot displays the 'Modify Compliance Structure Tree' interface. On the left, a hierarchical tree structure is shown under the title 'Modifying Compliance Structure'. The tree includes categories like 'Compliance 001', 'Currency', 'Sector', 'Country', and 'Ratings', each with associated rules and thresholds. For example, under 'Currency', there are rules for 'Swiss Franc' and 'EURO' with specific percentage thresholds. On the right, the 'Node Properties' panel is visible, showing fields for 'Label', 'Condition', 'Warning', and 'Threshold (%)'. The 'Label' is set to 'BBB- <= AA', and the 'Condition' is set to 'between'. Below this, there are buttons for 'Modify Node' and 'Delete Node'. At the bottom, the 'Add Category Node' section shows a dropdown menu with 'ID 1 | Currency' selected and a 'Create Node' button.

2. Zuweisung von Compliance-Regelsätzen



Definition von Compliance-Regelsätzen

1

Zuweisung von Compliance-Regelsätzen

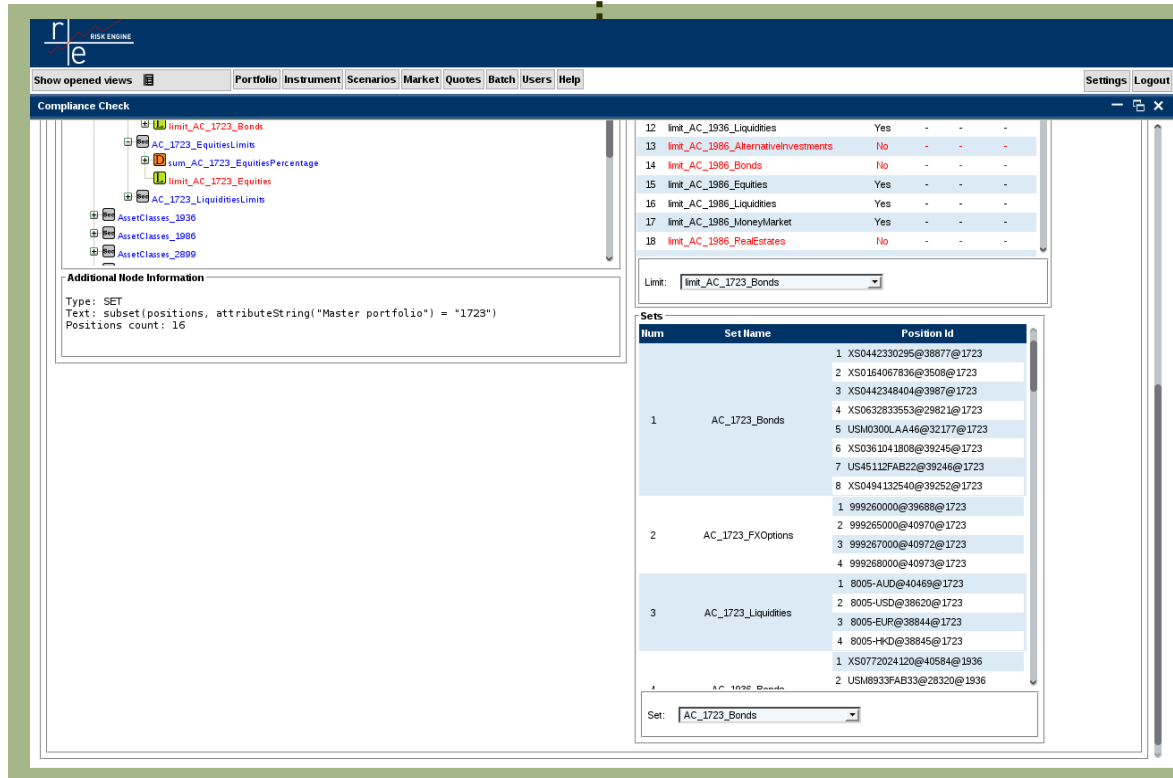
2

Prüfung und Auswertung

- Prüfung jederzeit ad-hoc möglich (on screen)
→ Monitoring Screen
- Prüfung über definierte zeitgesteuerte Abläufe
- Übersichtlicher Ausweis beim Überschreiten von Warn- und Verletzungsgrenzen

Berichtswesen

4



The screenshot shows the 'Compliance Check' window of the Risk Engine. It features a tree view on the left with nodes like 'limit_AC_1723_Bonds', 'AC_1723_EquitiesLimits', and 'AC_1723_LiquiditiesLimits'. The main area displays a table of compliance checks with columns for 'Num', 'Set Name', and 'Position Id'. The table lists various checks for 'AC_1723_Bonds', 'AC_1723_FXOptions', and 'AC_1723_Liquidities'. Below the table, there is a 'Set' dropdown menu and a 'Limit' input field. The 'Additional Node Information' section on the left provides details about the selected node, including its type and position count.

Num	Set Name	Position Id
1	AC_1723_Bonds	1 XS0442330295@38877@1723
		2 XS0164067836@3508@1723
		3 XS0442348404@3987@1723
		4 XS0632833553@29821@1723
		5 USM0300LAA46@3217@1723
		6 XS0361041808@39245@1723
		7 US45112FAB22@39246@1723
		8 XS0494132540@39252@1723
2	AC_1723_FXOptions	1 999260000@39688@1723
		2 999265000@40970@1723
		3 999267000@40972@1723
		4 999268000@40973@1723
3	AC_1723_Liquidities	1 8005-AUD@40469@1723
		2 8005-USD@38620@1723
		3 8005-EUR@38844@1723
		4 8005-HKD@38845@1723
		1 XS0772024120@40584@1936
		2 USM8933FAB33@28320@1936

Definition von Compliance-Regelsätzen

1

Zuweisung von Compliance-Regelsätzen

2

Prüfung und Auswertung

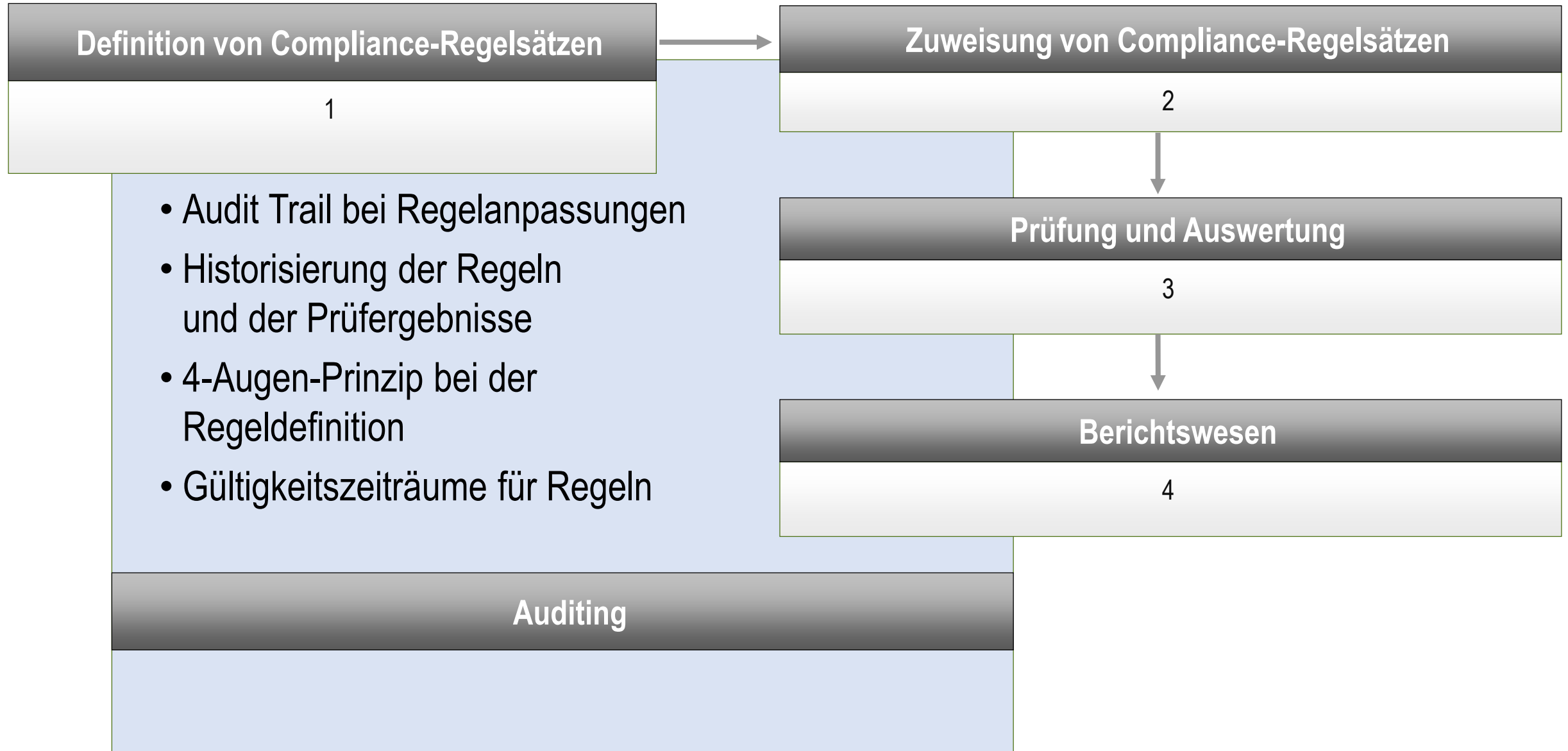
3

Berichtswesen

- Einbindung in Berichtswesen
- Regulatorische Berichte
- Individualisierte Berichte

The screenshot displays the 'Compliance Check' window in the EURO RISK COMPLIANCE CHECKER application. The interface includes a menu bar (Portfolio, Instrument, Scenarios, Market, Quotes, Batch, Users, Help) and a toolbar (Calculate, Save results). The main area is divided into two panes. The left pane shows a 'Results Tree' with a hierarchical structure of asset classes and positions. The right pane displays a table of 'Limits' with columns: Item, Limit Name, Satisfied, LHS, RHS, and Remaining. Below the table, there is a 'Sets' section showing a list of sets and their positions.

Item	Limit Name	Satisfied	LHS	RHS	Remaining
1	limit_AC_1723_Bonds	No	-	-	-
2	limit_AC_1723_Bonds_down	No	21.21	25.00	15.17%
3	limit_AC_1723_Bonds_up	Yes	21.21	31.00	31.59%
4	limit_AC_1723_Equities	No	-	10.00	100.00%
5	limit_AC_1723_FXOptions	Yes	-	-	-
6	limit_AC_1723_FXOptions_down	Yes	51.19	51.00	0.36%
7	limit_AC_1723_FXOptions_up	Yes	51.19	52.00	1.57%
8	limit_AC_1723_Liquidities	Yes	-	-	-
9	limit_AC_1723_Liquidities_down	Yes	27.61	9.00	67.40%
10	limit_AC_1723_Liquidities_up	Yes	27.61	48.00	42.48%
11	limit_AC_1936_Bonds	Yes	-	-	-
12	limit_AC_1936_Liquidities	Yes	-	-	-
13	limit_AC_1986_AlternativeInvestments	No	-	-	-
14	limit_AC_1986_Bonds	No	-	-	-
15	limit_AC_1986_Equities	Yes	-	-	-
16	limit_AC_1986_Liquidities	Yes	-	-	-
17	limit_AC_1986_MoneyMarket	Yes	-	-	-
18	limit_AC_1986_RealEstate	No	-	-	-





- Positionsfluss: Definition der Struktur des Positionsflusses
- Positionsfluss: Steuerung der Positionsflusses über Filterelementen
- Positionsfluss: Freie Strukturbildung des Positionsflusses
- Datenfluss: Laden von Daten und Berechnung von Aggregate
- Logik-Fluss: Definition und Berechnung von Regelsätzen:
 - Datenelemente
 - Arithmetische Formeln
 - Vergleichsoperatoren
 - Logisch Ausdrücke

- **Definition der Struktur des Positionsflusses**

Topologische Strukturen (verlinkte Baumstrukturen, einschl. Asset Allokationen) von Positionslisten.

- **Steuerung der Positionsflusses über Filterelementen**

Die Filterelemente filtern Eingangsmengen von Positionen über Zugriffe in der Datenbank und produzieren Ausgangsmengen von Positionen.

Gefiltert wird nach Parameter und Ergebnisse auf Positionsebenen und aus den verlinkten Gattungsdaten und Issuer-Daten.

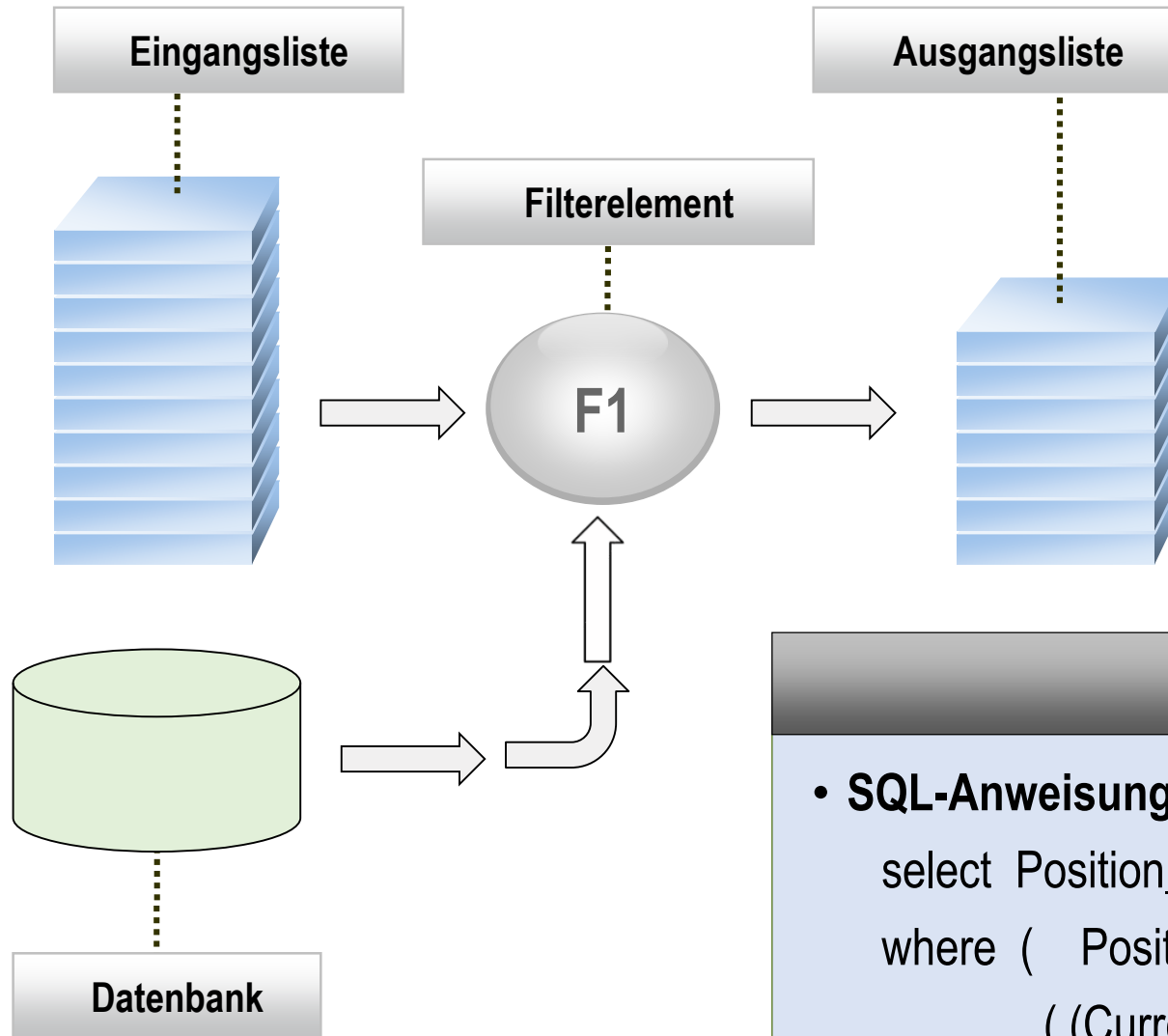
Ergebnisse auf Sub-Portfolio-Ebenen können in den Filterelementen berücksichtigt werden.

- **Freie Strukturbildung des Positionsflusses**

Die Positionsmengen können über hintereinander geschalteten oder parallel geschalteten Filterelemente bearbeitet werden.

Exklusivität kann ein- oder ausgeschaltet werden.

Mengenoperationen auf den Positionslisten, wie Union und Intersection, können ausgeführt werden.



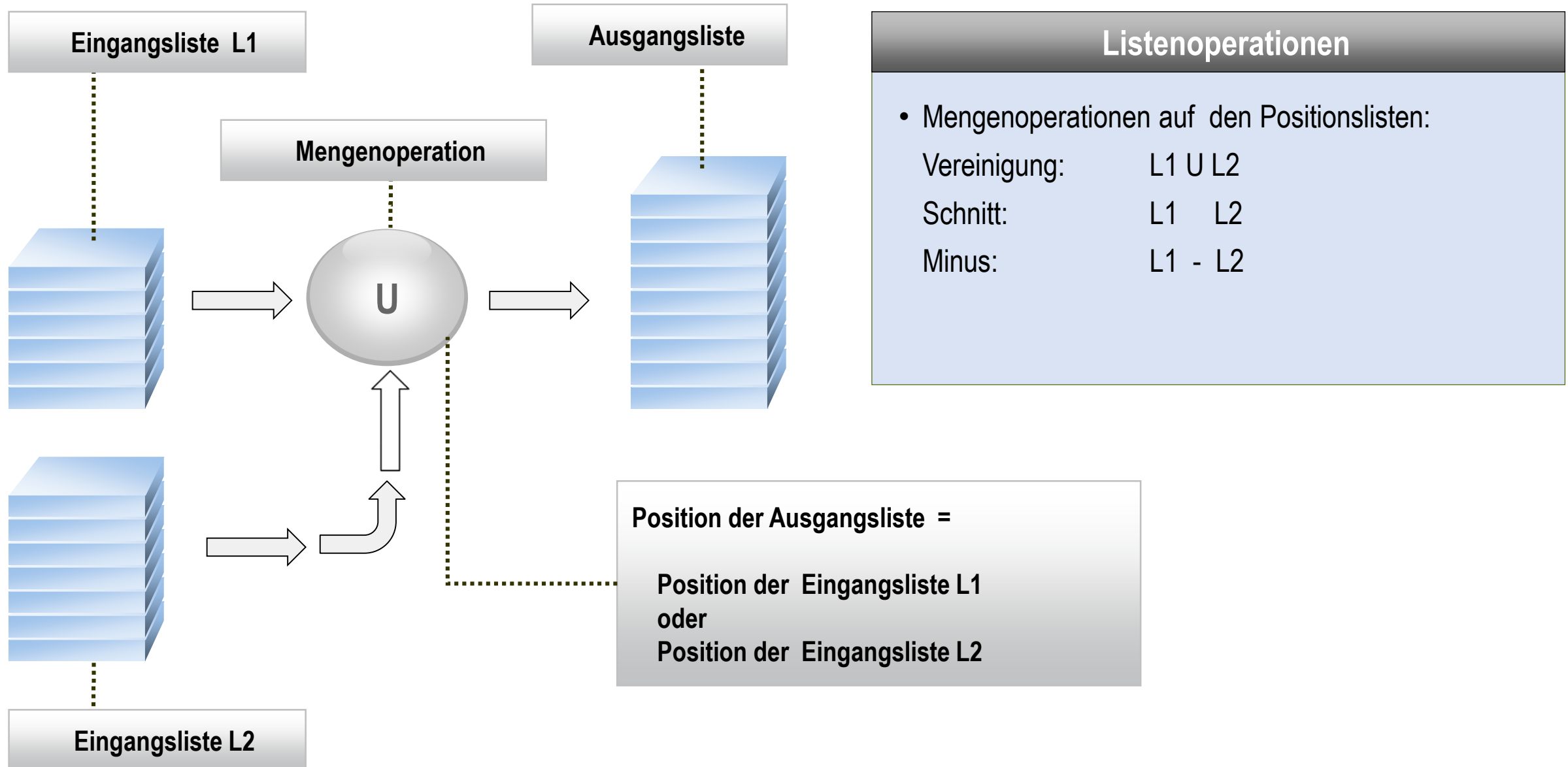
Filterelemente

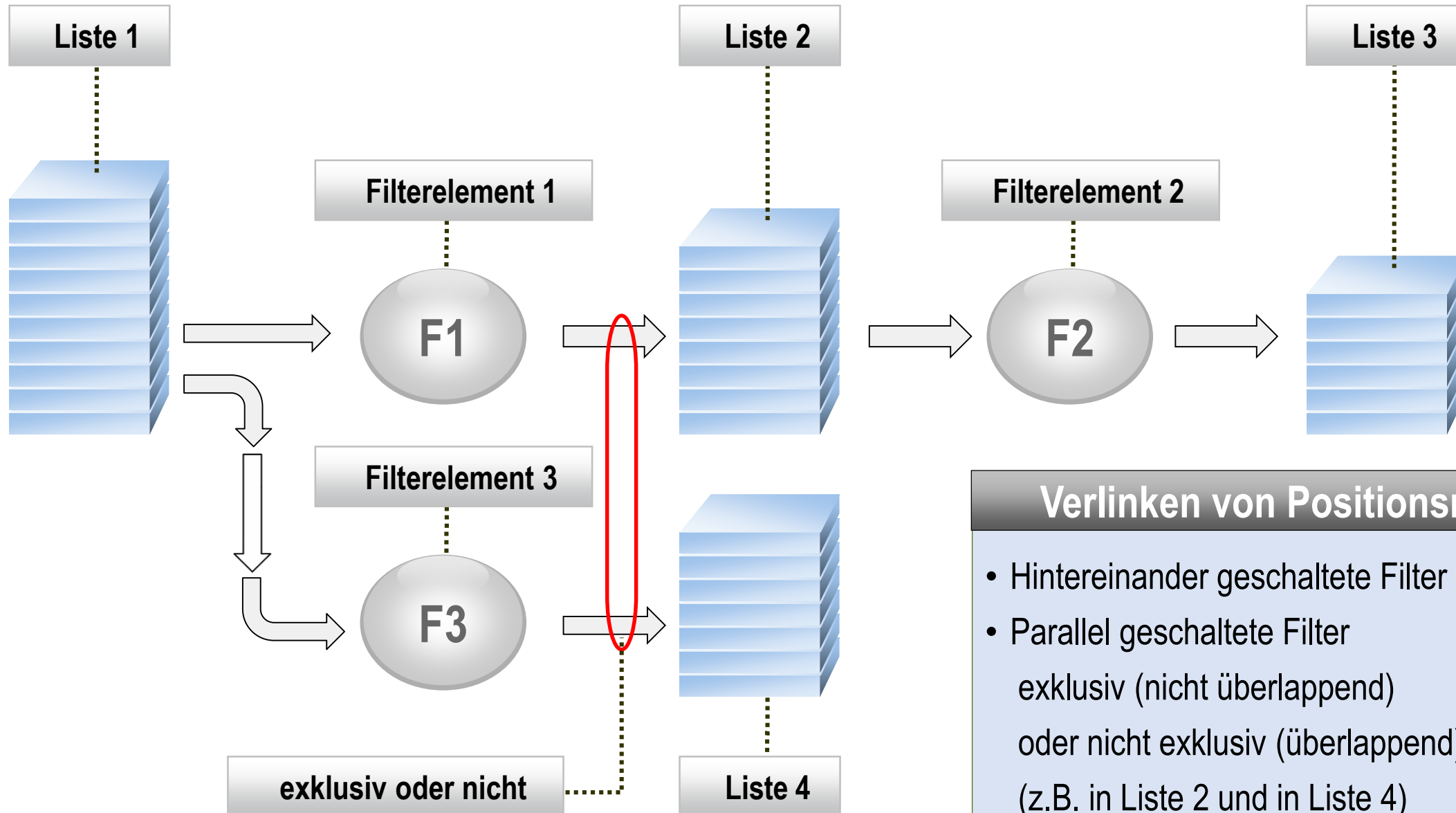
- Steuerung der Positionsflusses über Filterelemente
- Filtern Eingangslisten von Positionen über Zugriffe in der Datenbank und produzieren Ausgangslisten von Positionen

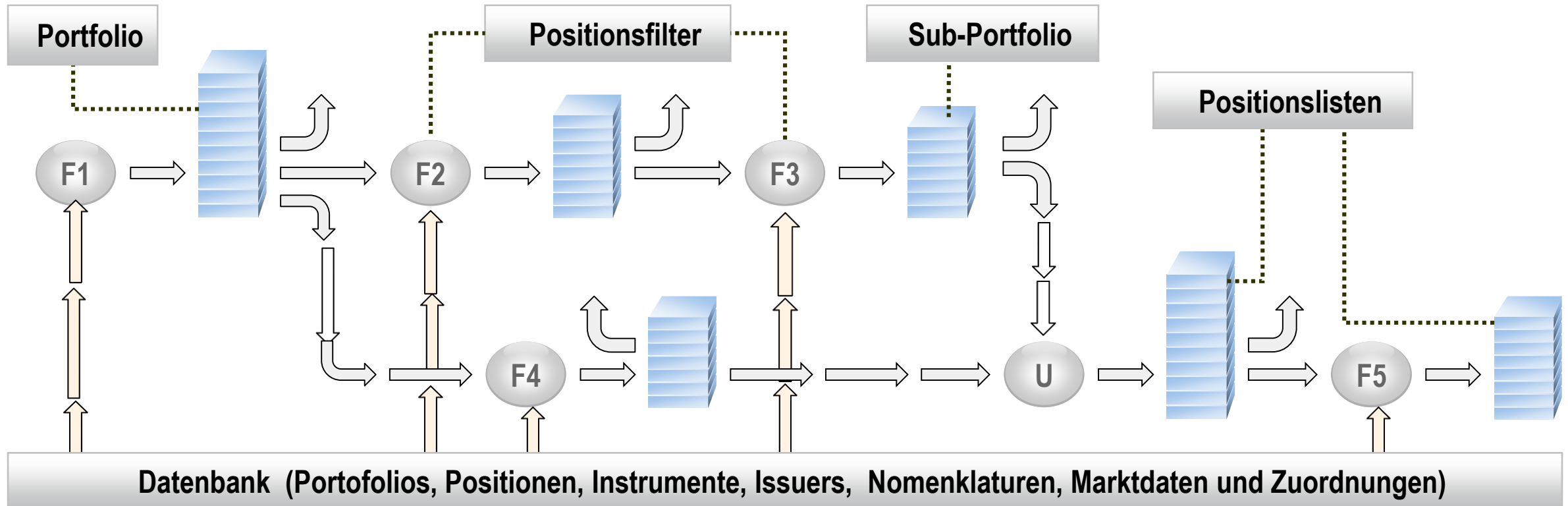
Filterbeispiel: SQL-Anweisung für F1

- **SQL-Anweisung für die Ausgangsliste:**

```
select Position_Id from Position, Instrument, Issuer  
where ( Position_Id in [Eingangsliste] and  
      ( (Currency = CHF and Duration < 3,5) or  
        Issuer.Country <> GB) )
```







Filterelementen

- Topologische Strukturen von Positionslisten und Filterelementen
- Verlinkte Baumstrukturen zur Bestimmung von Portfoliolisten
- einschließlich Asset Allokationen und Sub-Portfolio-Hierarchien

• Laden von Daten aus der Datenbank

Parameter, Ergebnisse und Aggregate der gefilterten Positionsmengen werden aus der Datenbank geladen und für die Bildung der Regelsätze herangezogen.

• Definition und Berechnung von Aggregate

Auf den gefilterten Positionsmengen werden Aggregate definiert:

Neue Daten auf Positionsebene über Formeln, z.B. Positionswert * Währungskurs.

Aggregate auf der Positionsmengen über Funktionen: max, min, sum, average, etc.

Eine Formelsprache und ein Formeleditor werden für die Definition verwendet.

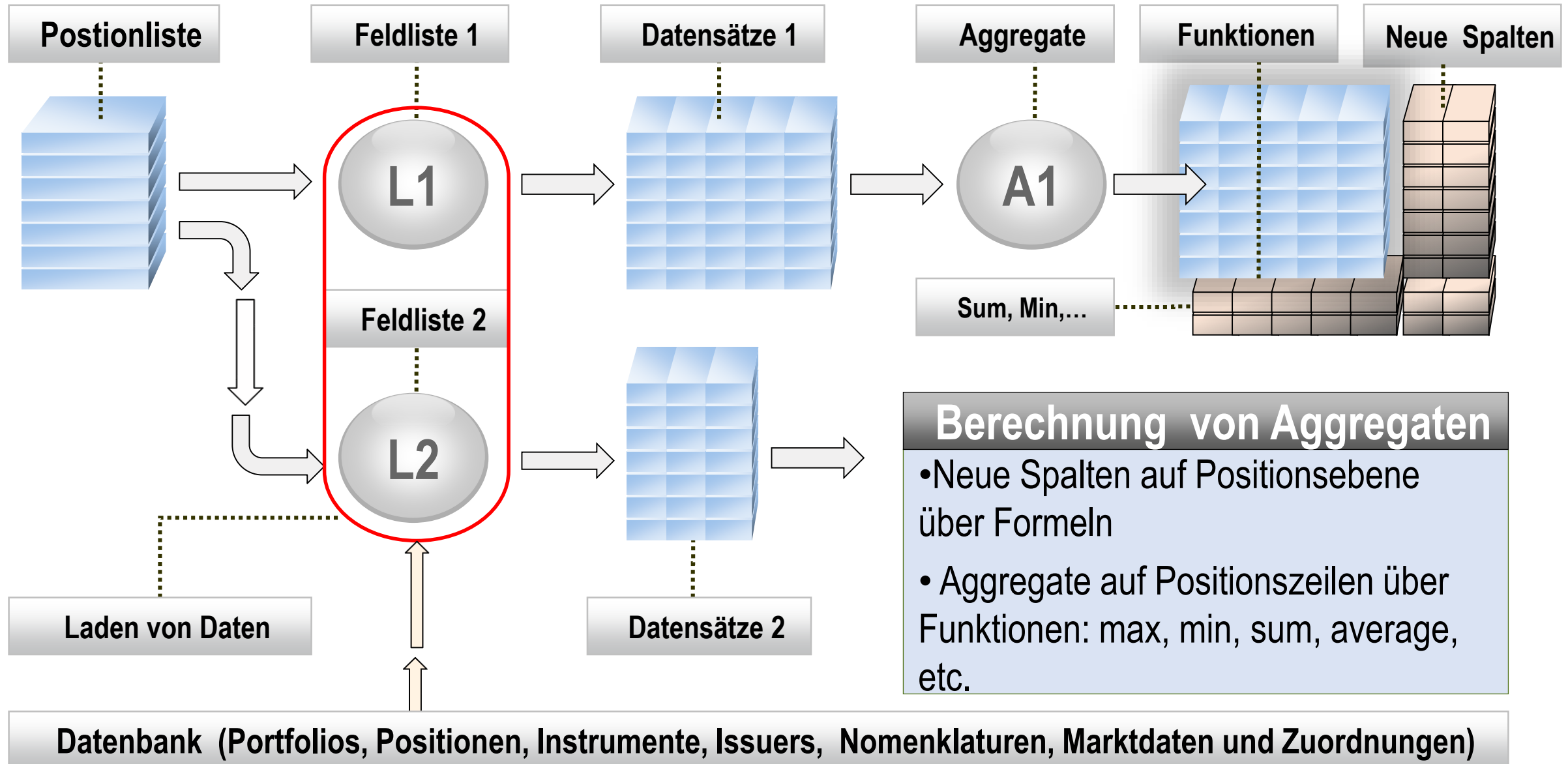
• Definition und Berechnung von Regelsätzen

Die Regelsätze sind logische Ausdrücke bestehend aus folgenden Komponenten:

- arithmetischen Formeln: z.B. (Clean Value + Accr. Interest) / Nominal Amount
- Vergleichsoperatoren: >, >=, <, <=, =, <>, between, z.b. Volume < 20%
- logische Operatoren: and ,or, not

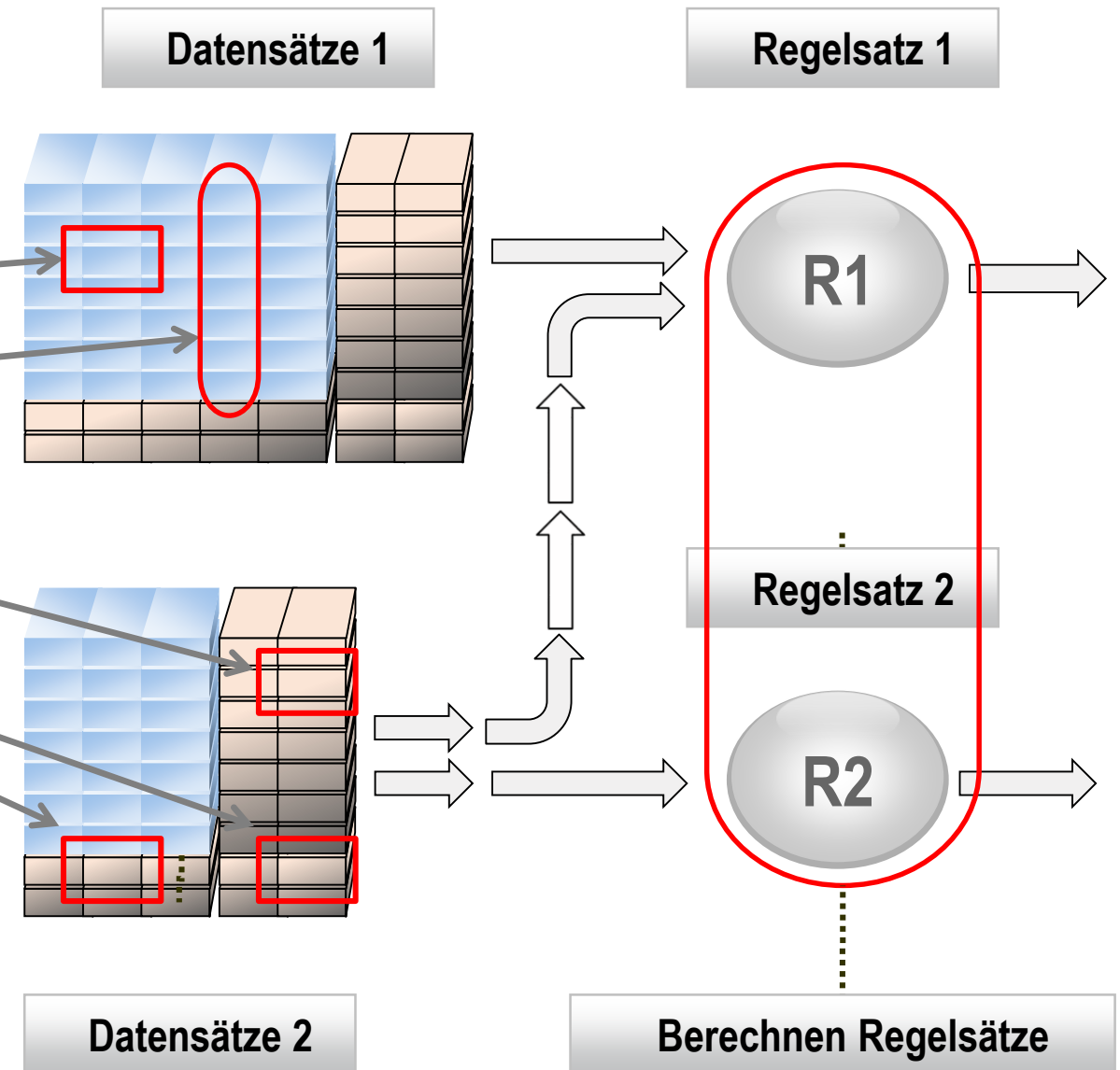
Definition und Berechnung von komplexen Regelsätzen wie z.B.:

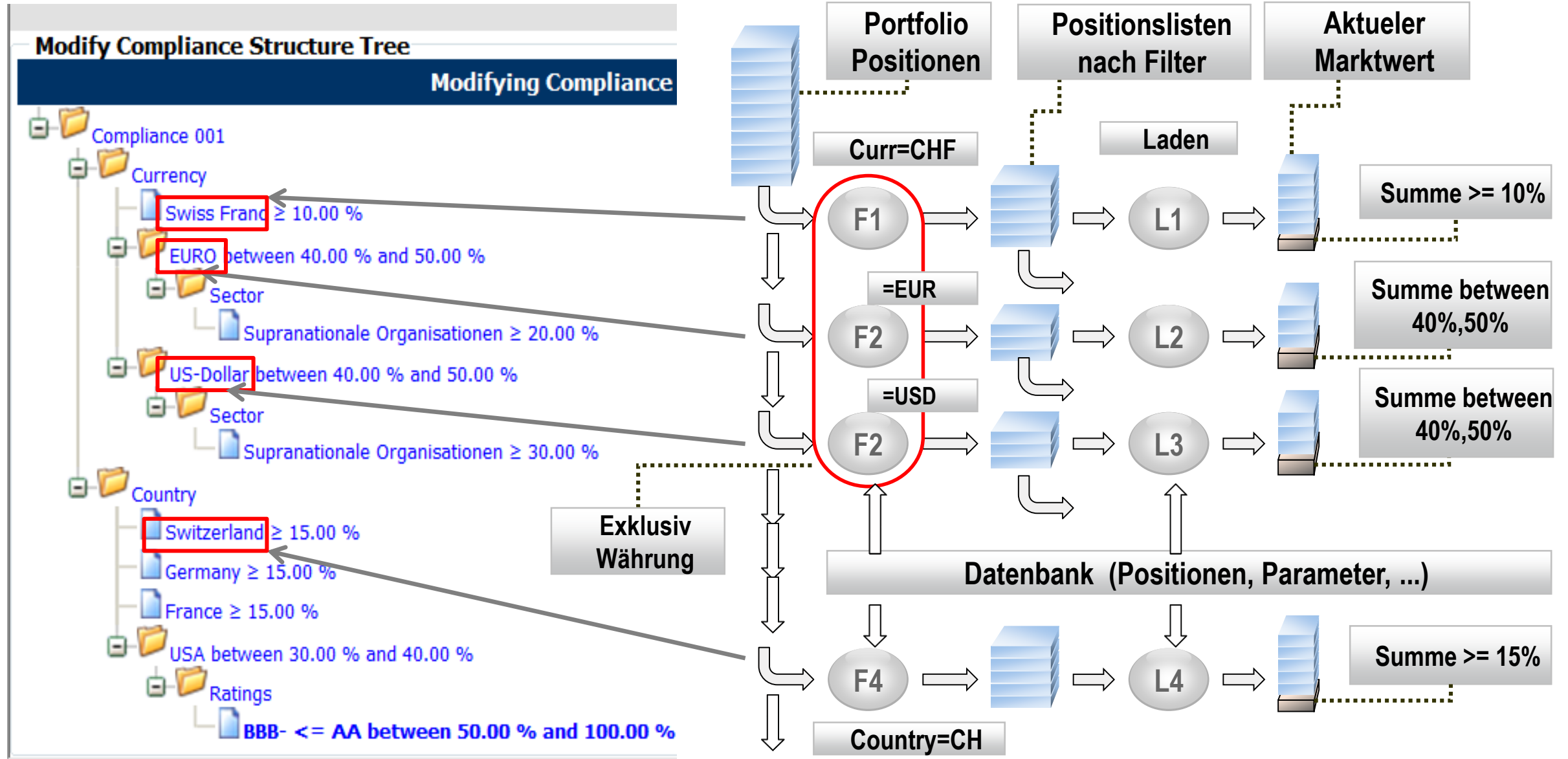
Der Wert der 10 größten Positionen < als der Wert der maximale Position*0,3.



Definition der Regelsätze

- Die Regelsätze = logische Ausdrücke bestehend aus:
 - Daten der Positionsdatensätze:
 - Einzelne Positionsfelder
 - Alle Positionsfelder einer Spalte
 - Felder berechneter Spalten
 - Felder aggregierter Zeilen
 - Arithmetische Formeln und Funktionen: +, -, *, /, ln, exp, (), ...
 - Vergleichsoperatoren: >, >=, <, <=, =, <>, between, ...
 - Logische Operatoren: and, or, not
- Die Regelsätze berechnen das logische Ergebnis (Prüfung erfolgreich oder nicht) und Warnmeldungen





Limit System Implementation in Risk Engine

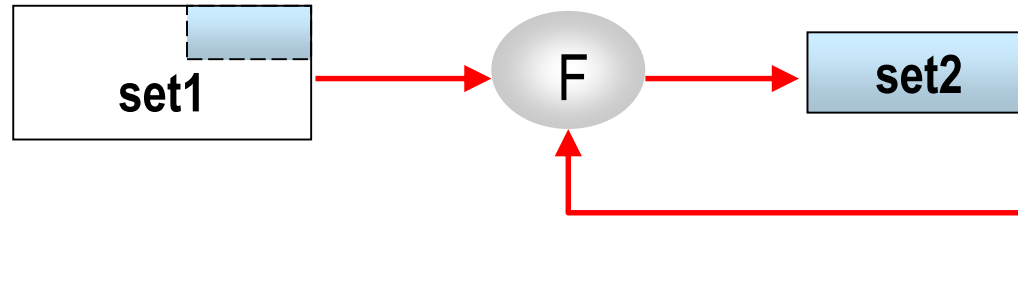
- *Simple language* for definition of sets, values and limits
- *Syntactic support* for automatic generation of language elements
- Inference by Drools – Java based *rule engine* (similar to CLIPS), that implements RETE algorithm
- New positions could be added and removed and the system *automatically changes its logical inferences*
- Allows development of an “*Alert system*” that always resides in the memory and every new position leads to necessary local changes and logical inferences

Assigned set:



```
set set2 = set1;
```

Subset:

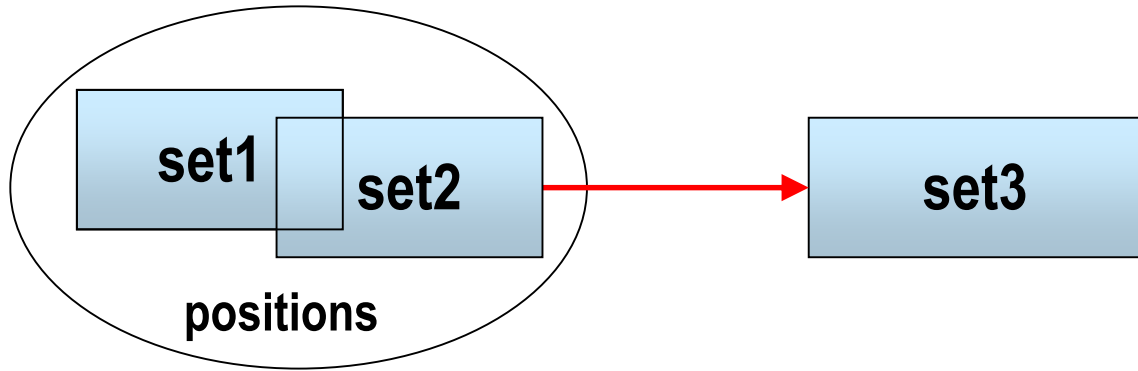


```
set set2 = subset(set1, ((assetCategory = 'Bonds') and (riskCurrency = 'CHF')) or ((assetCategory = 'Equities') and (Country = 'CH')));
```

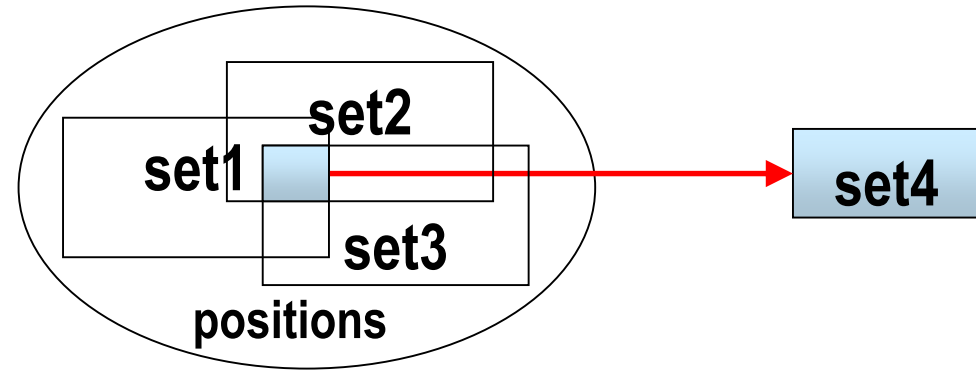
```
set set4 = min(set2, (value + 2.0), 0, 10); // first 10 minimal elements
```

```
set set5 = max(set2, PV, 2, 15); // 15 maximal elements starting from 3-th
```

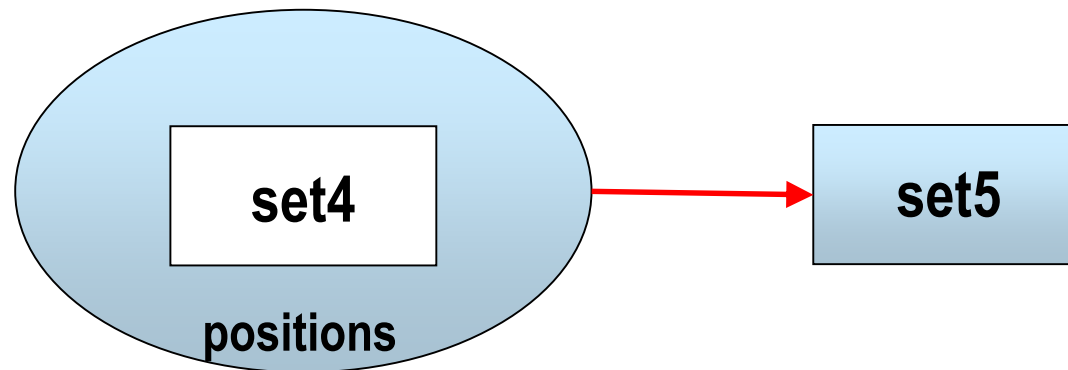
Logical sets operations (conjunction, disjunction, negation):



set set3 = set1 or set2;



set4 = set1 and set2 and set3;



set5 = not (set4);

Definition of named variables for use in other expressions:

```
double_result <result_name> = <expression>;
```

The following aggregates could be used:

- ☐ Size – number of positions in a set
- ☐ Avg – average value of an expression with position fields in a set
- ☐ Sum – sum value of an expression with position fields in a set
- ☐ Value – value of an expression with a position fields

Examples for named variables:

```
double_result set4Size = size(set4);  
double_result result2 = result1 + 130.6;  
double_result avgSet1Expr = avg(set1, result1 + 0.2 * volume);  
double_result val7Set2 = value(set2, PV, 7);  
double_result result3 = sum(set3, 1.0 + 0.2 * volume);  
double_result result4 = result2;
```


Assigned limits

```
limit <limit_name> = <existing_limit_name>;
```

Expression limits using compare operators

```
limit <limit_name> = <expression_1> <op> <expression_2>;           // single comparing
limit <limit_name> = forall(<set_name>, <expression_1> <op> <expression_2>;    // for all within set
```

Examples for limits and aggregates:

```
limit limit1 = result3 < 99.8;
limit limit2 = avg(set3, value) >= 30%;
limit limit3 = sum(set3, result5 + value) < sum(set2, value) * 10%;
limit limit4 = forall(set1, PV > 10.0%);
```

Logical limit operations (and, or, not):

```
limit <limit_name> = <limit_name_1> <op> <limit_name_2> ...;
limit <limit_name> = forall(<set_name>, (<expr_1> <op> <expr_2>) <op> (<expr_3> <op> <expr_4>) ...);
```

Examples:

```
limit limit5 = limit1 or limit2;           // 'or' of existing limits
limit limit6 = (avgSet1Expr > 10%) and (avgSet1Expr < 15%);    // 'and' of inline limits
limit limit7 = forall(set3, (PV < result5) and (PV > 30.0%));    // for all positions in set3
```

Classifier definition

```
classifier <classifier_name> = classify(<set>, <column_name> , <column_content>, <limit>);
```

Determines whether or not the <column_content> must be written in the column named <column_name> for the <set> if the <limit> is satisfied. The column name is string and the column content depends on the type of that column.

Example

```
classifier classifier1 = classify(set0, "PV", result.someValue + 15.0, limit1); // someValue is name of a double_result variable  
classifier anyclassifier2=classify(set0, "country", "DENMARK", limit2);  
// if limit2 is satisfied the column "country"  
classifier classifier2 = classify(set0, "date", to_date("10.01.2010", "dd.MM.yyyy"), limit3);
```

The column type can be double, string or date. In case of double type the column content can be expression (+, -, *, /) of numbers and/or double_result names.

In order to structure the Compliance check language elements sections can be used. They do not perform any calculations.

```
classifier <classifier_name> = classify(<set>, <column_name> , <column_content>, <limit>);
```

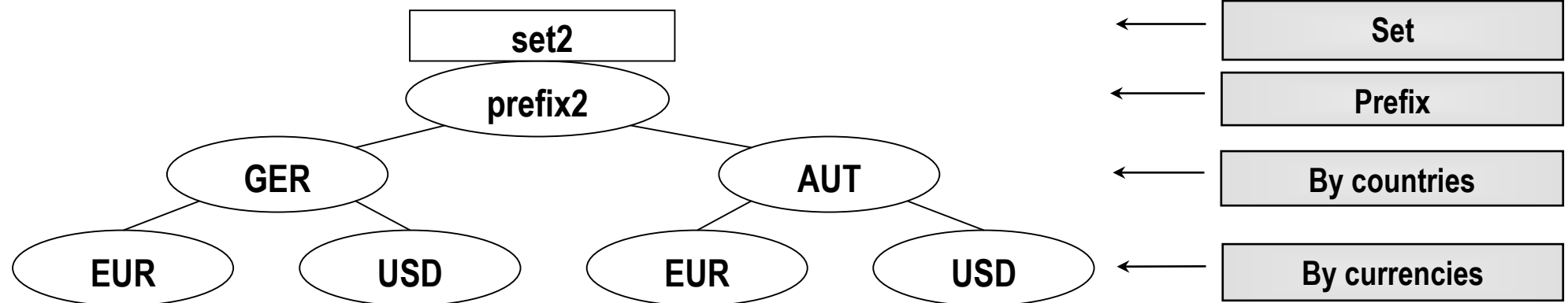
Determines whether or not the <column_content> must be written in the column named <column_name> for the <set> if the <limit> is satisfied. The column name is string and the column content depends of the type of that column.

Example

```
classifier classifier1 = classify(set0, "PV", result.someValue + 15.0, limit1); // someValue is name of a double_result variable  
classifier anyclassifier2=classify(set0, "country", "DENMARK", limit2);  
// if limit2 is satisfied the column "country"  
classifier classifier2 = classify(set0, "date", to_date("10.01.2010", "dd.MM.yyyy"), limit3);
```

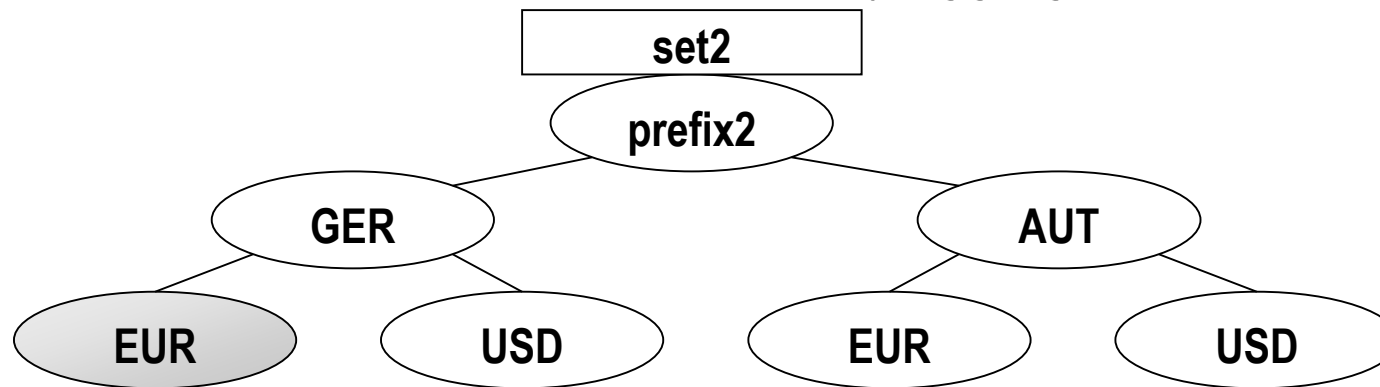
The column type can be double, string or date. In case of double type the column content can be expression (+, -, *, /) of numbers and/or double_result names.

A dynamic set is defined by prefix and enumeration position fields:

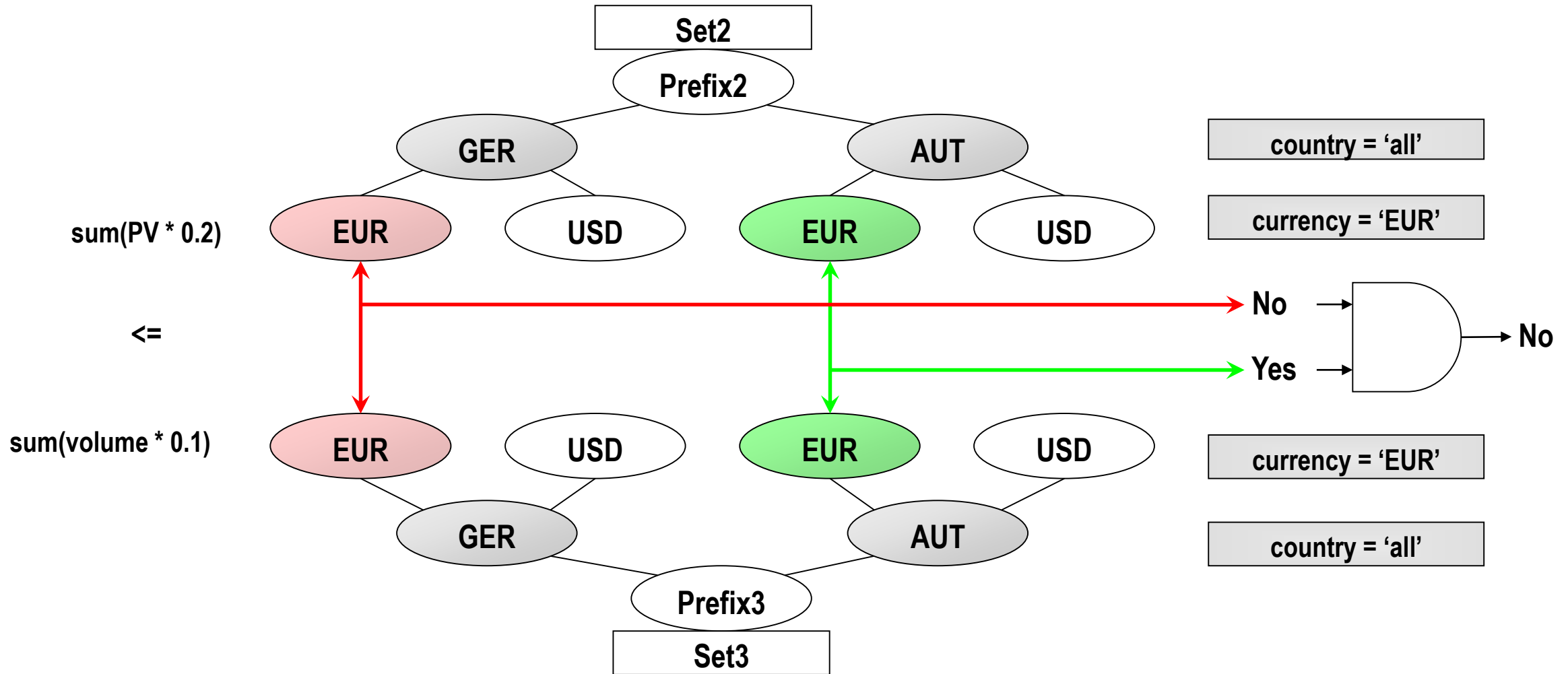


```
dynamicSet(set2, country, currency, 'prefix2'); // Dynamic set definition
```

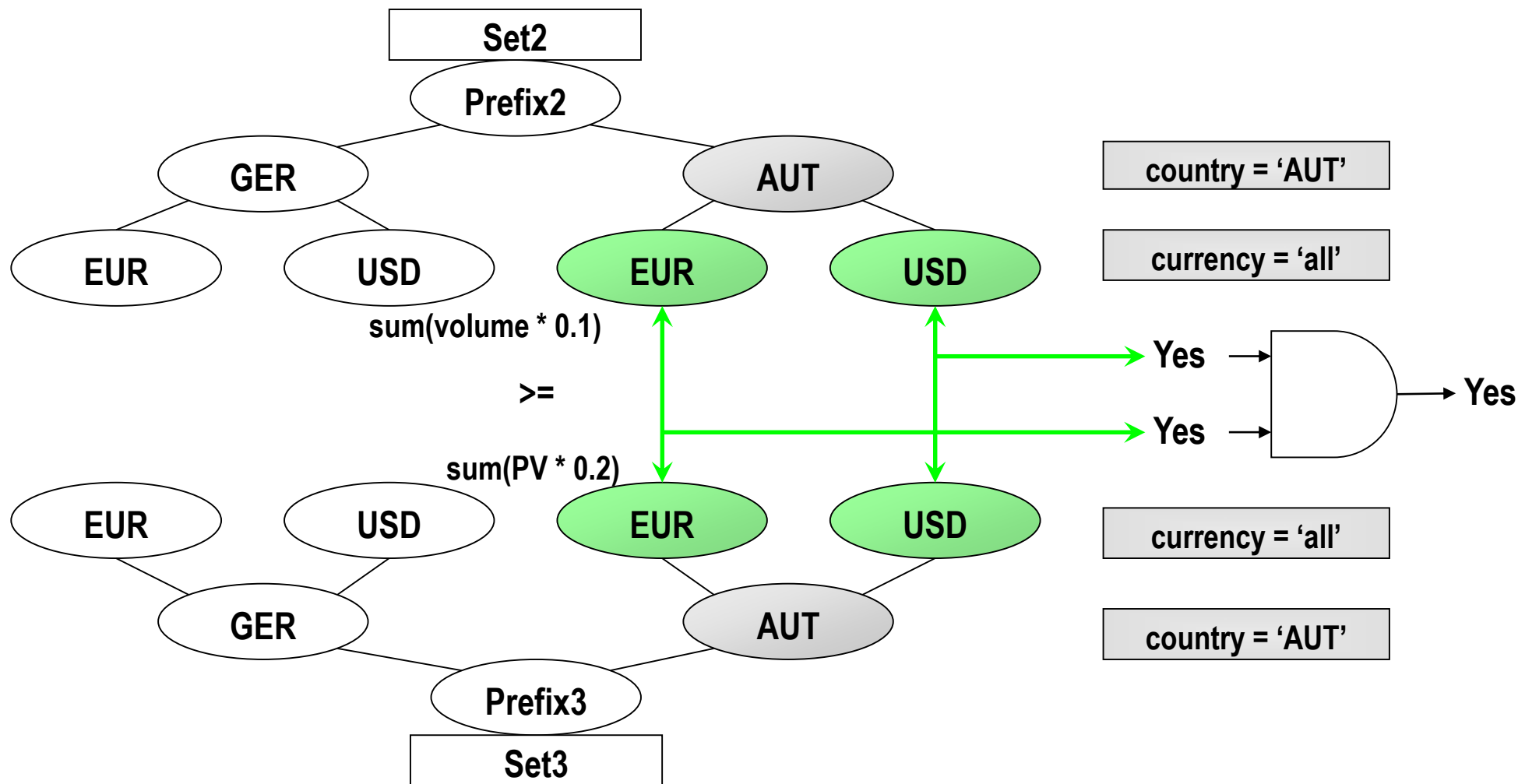
A supposed subset name could be used in an ordinary aggregate:



```
double_result result1 = sum(prefix2_GER_EUR, volume); // Usage of the sub-set GER&EUR
```

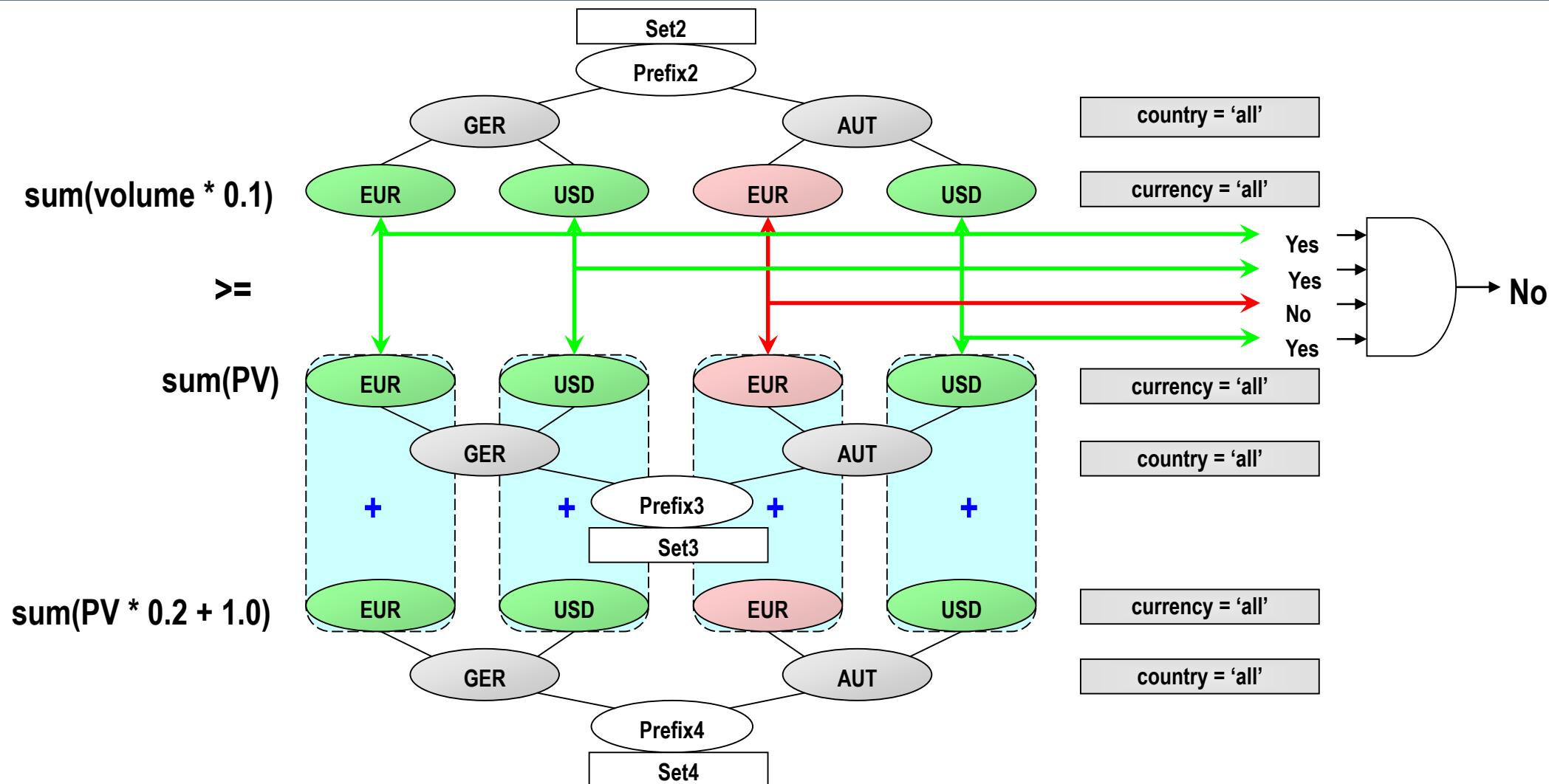


```
limit limit9 = Forall( 'prefix2', 'prefix3', distinct_by(country='all', currency='EUR'), Sum('prefix3', volume * 0.1) >= Sum('prefix2', PV * 0.2) );
```

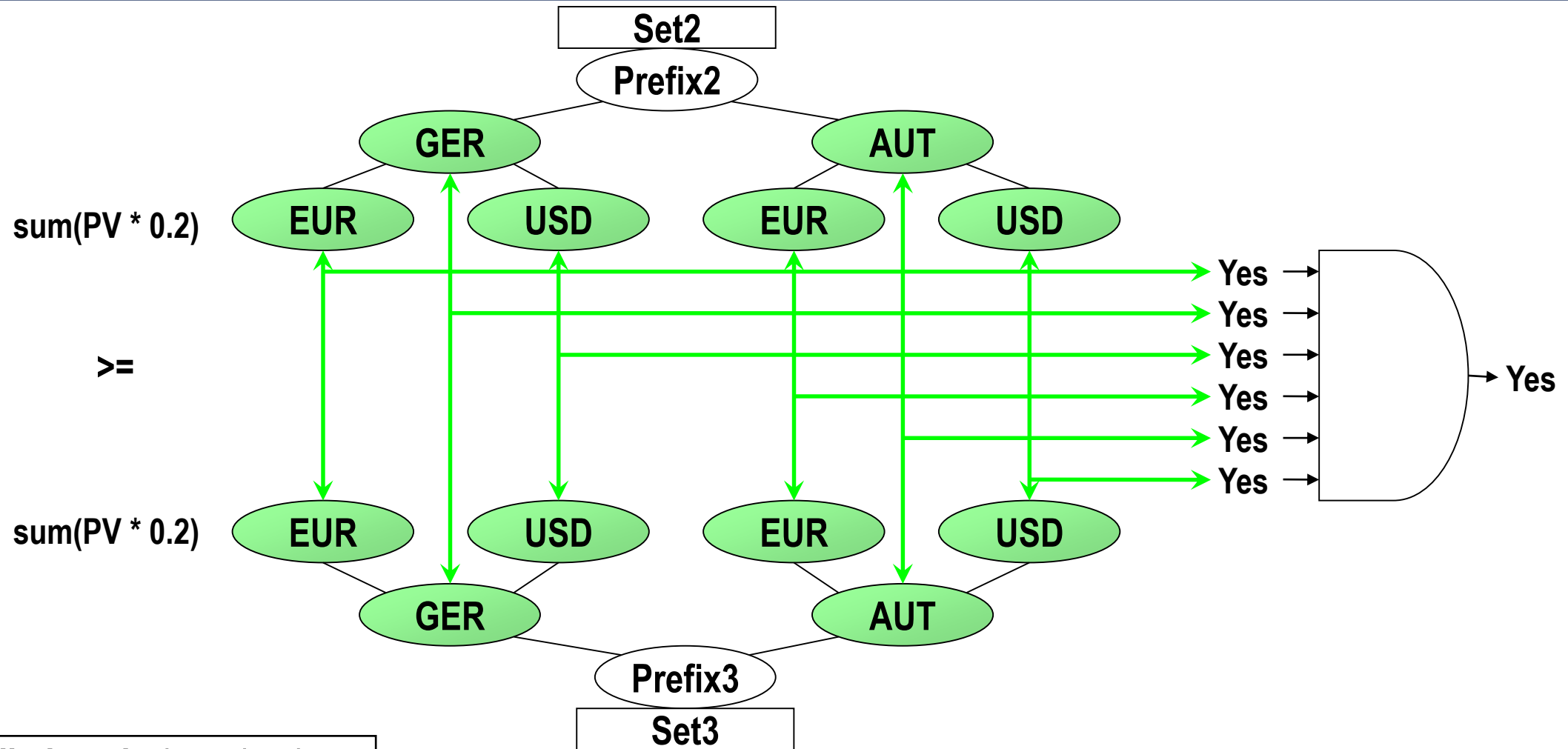


```
limit limit10 = Forall ( 'prefix3', 'prefix2', distinct_by(country='AUSTRIA', currency = 'all'), Sum('prefix2', volume *0.1) >= Sum('prefix3',PV*0.2) );
```

Limit on dynamic sets – aggregations on leaves

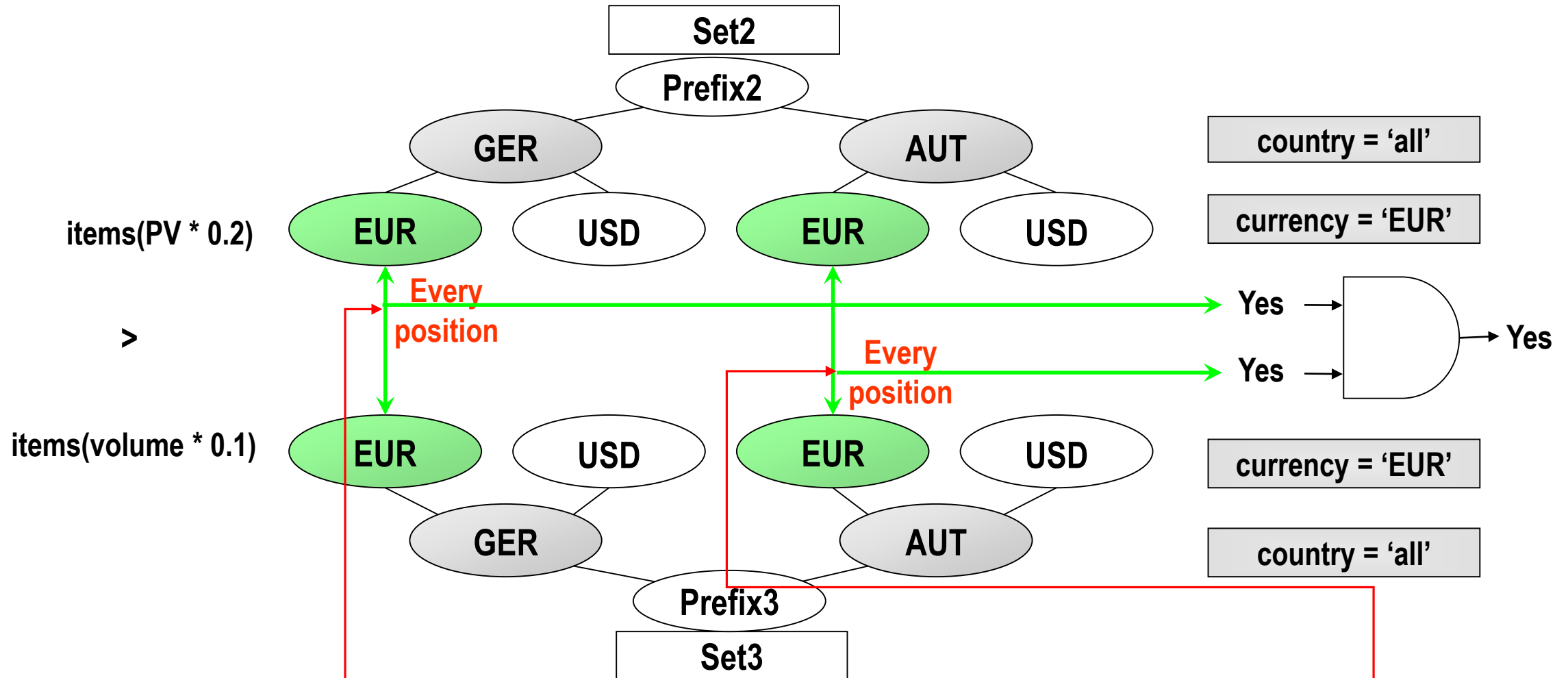


```
limit limit11 = Forall( 'prefix2', 'prefix3', 'prefix4', distinct_by(country='all', currency='all'),
    Sum('prefix2', volume * 0.1) >= Sum('prefix3', PV) + Sum('prefix4', PV * 0.2 + 1.0) );
```

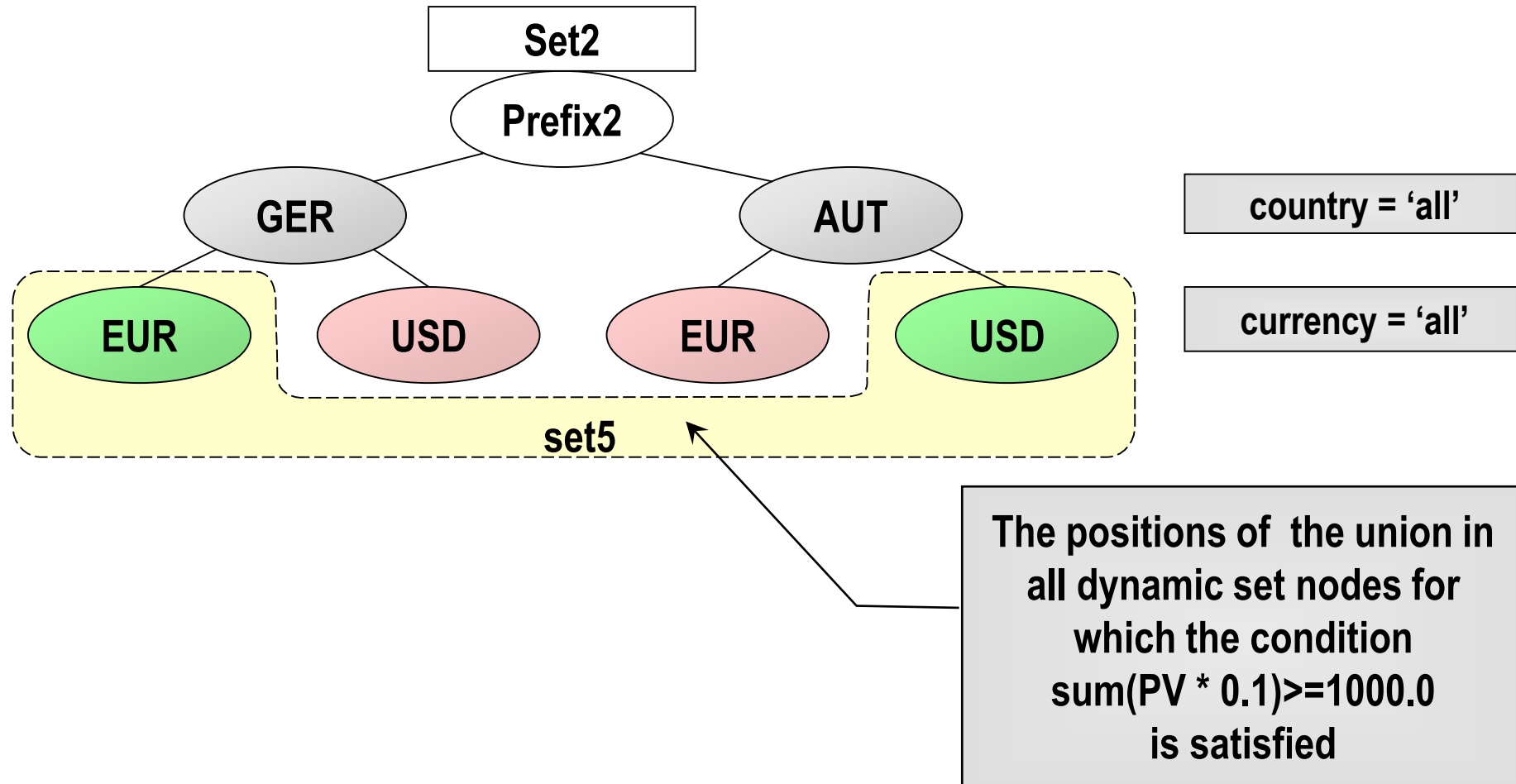


distinct_by is omitted

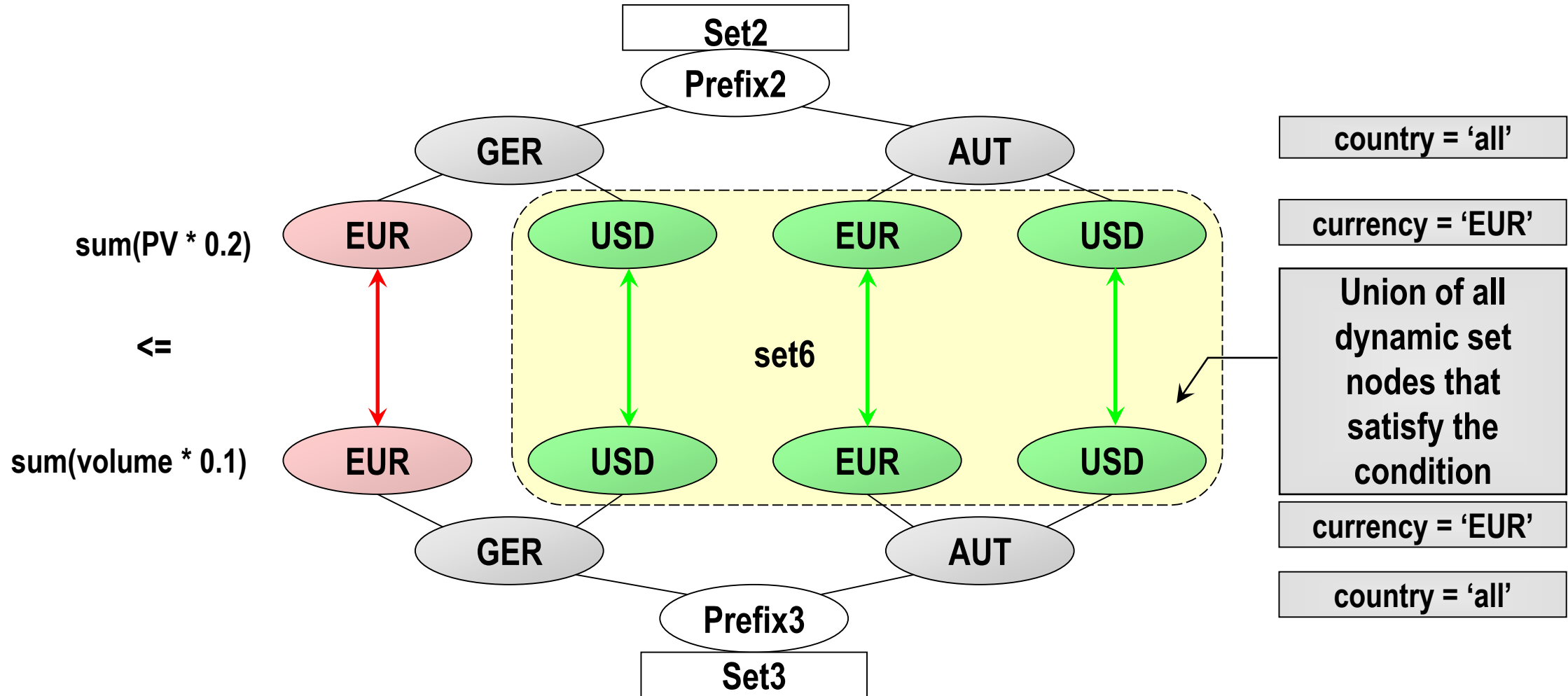
```
limit limit10 = Forall('prefix2', 'prefix3', Sum('prefix2', volume * 0.1) >= Sum('prefix3', PV * 0.2))
```

```
limit limit9 =Forall( 'prefix2', 'prefix3', distinct_by(country='all', currency='EUR'), Items('prefix3', volume * 0.1) > Items('prefix2', PV * 0.2));
```



```
set set5 = dynamicUnion('prefix2', distinct_by(country='all', currency='all'), Sum('prefix2', volume * 0.1) >= 1000.0);
```



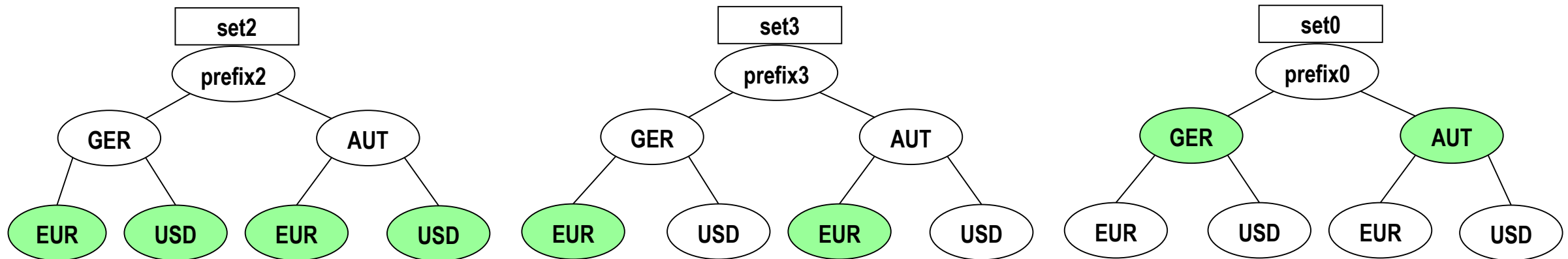
```
set set6 = dynamicUnion('prefix2', 'prefix3', distinct_by(country='all', currency='all'), Sum('prefix3', volume * 0.1) >= Sum('prefix2', PV * 0.2));
```

```
dynamicSetSize(<prefix>, distinct_by(<propertiesList>);
```

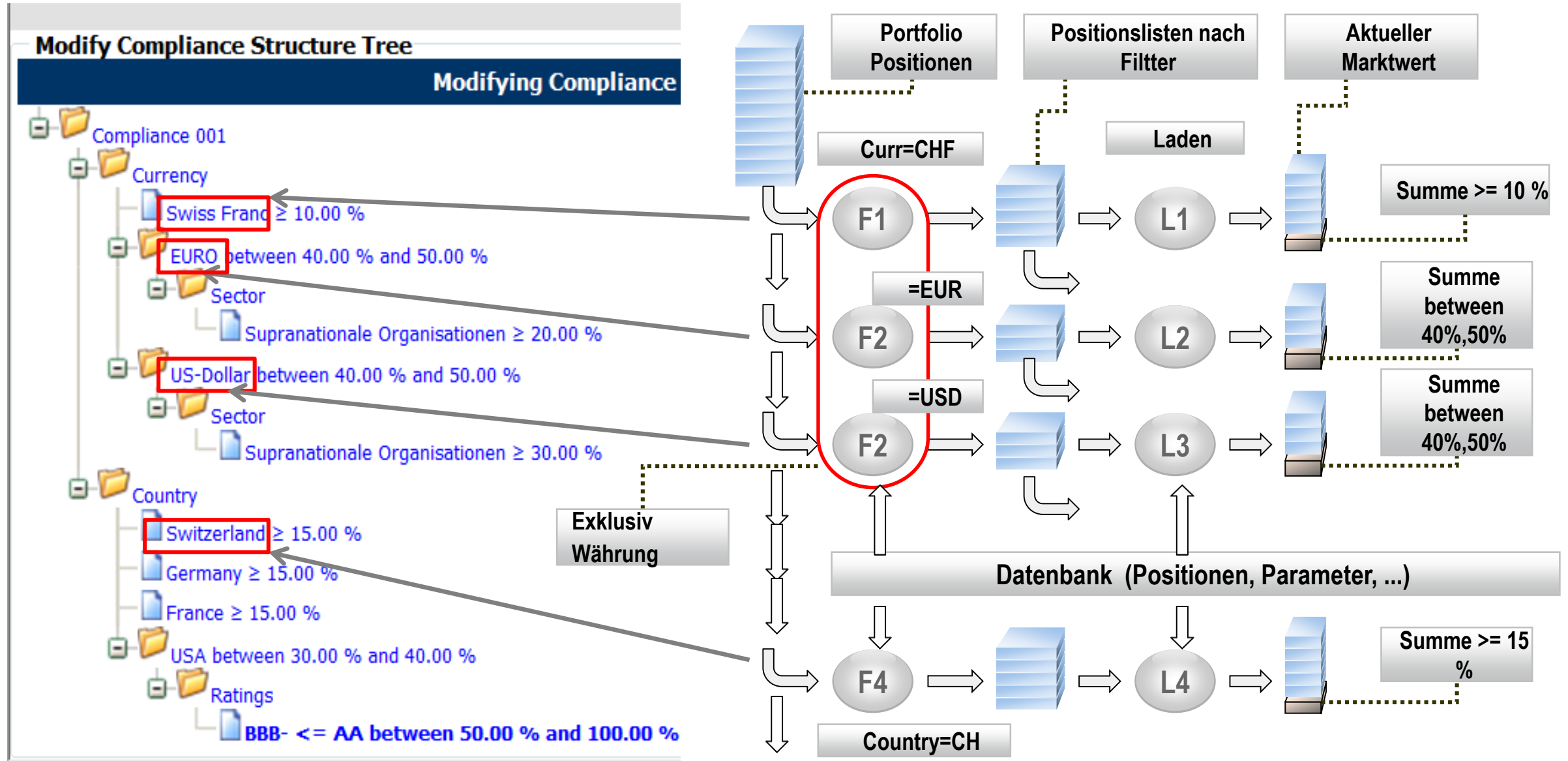
Determines the number of sets of a dynamic set given by prefix. The sets are determined by the property list.

Examples for named variables:

```
double_result allCountriesAllCurrencies = dynamicSetSize('prefix2', distinct_by(country='all', currency='all'));  
double_result allCountriesEur = dynamicSetSize('prefix3', distinct_by(country='all', currency='EUR'));  
double_result allCountries = dynamicSetSize('prefix0', distinct_by(country='all'));  
double_result allCurrencies = dynamicSetSize('prefix0', distinct_by(currency='all')); // 2: EUR and USD
```



Subsets and limits definitions - example



```

/*****
*           Subsets definitions
*****/

set F1 = subset(positions, currency = 'CHF');
set F2 = subset(positions, currency = 'EUR') and not (F1);           // Exclusive about F1
set F3 = subset(positions, currency = 'USD') and not (F1) and not (F2); // Exclusive about F1 & F2

/*****
*           Limits definitions
*****/

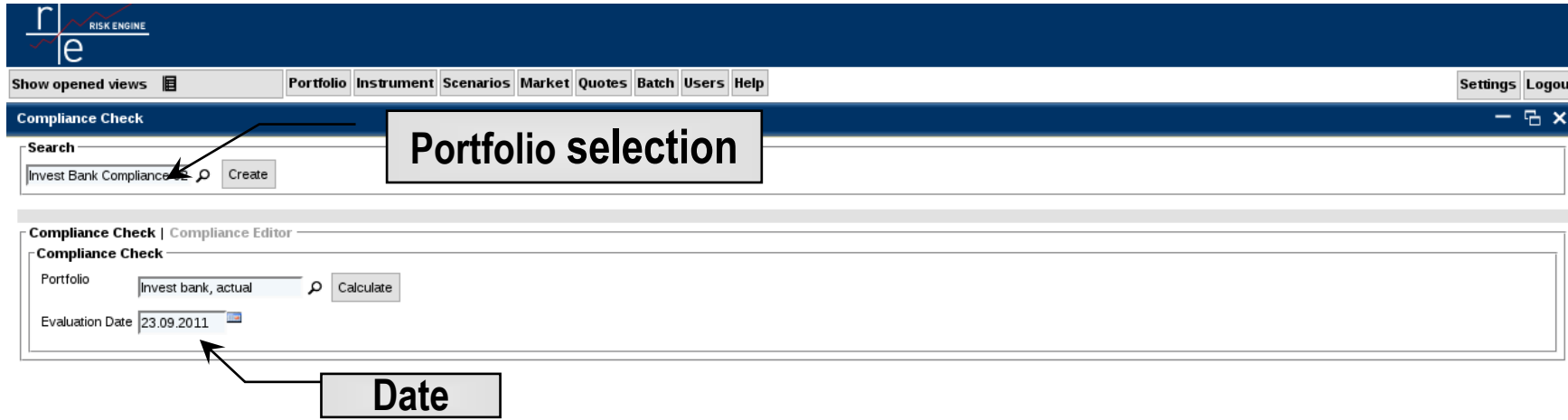
double_result  positionValues = sum(positions, value);           // sum of all position values
limit L1 = sum(F1, value) > 10% * positionValues;                // Limit L1

double_result  sumValueF2 = sum(F2, value);                      // sum of position values in F2
limit L2 = (sumValueF2 > 40% * positionValues) and               // Limit L2
           (sumValueF2 < 50% * positionValues);

double_result  sumValueF3 = sum(F3, value);                      // sum of position values in F3
limit L3 = (sumValueF3 > 40% * positionValues) and               // Limit L3
           (sumValueF3 < 50% * positionValues);

double_result  sumValueF4 =
               sum(subset(positions, country = 'CH'), value);    // sum of position values in F4
limit L4 = sumValueF4 > (15% * positionValues);                 // Limit L4

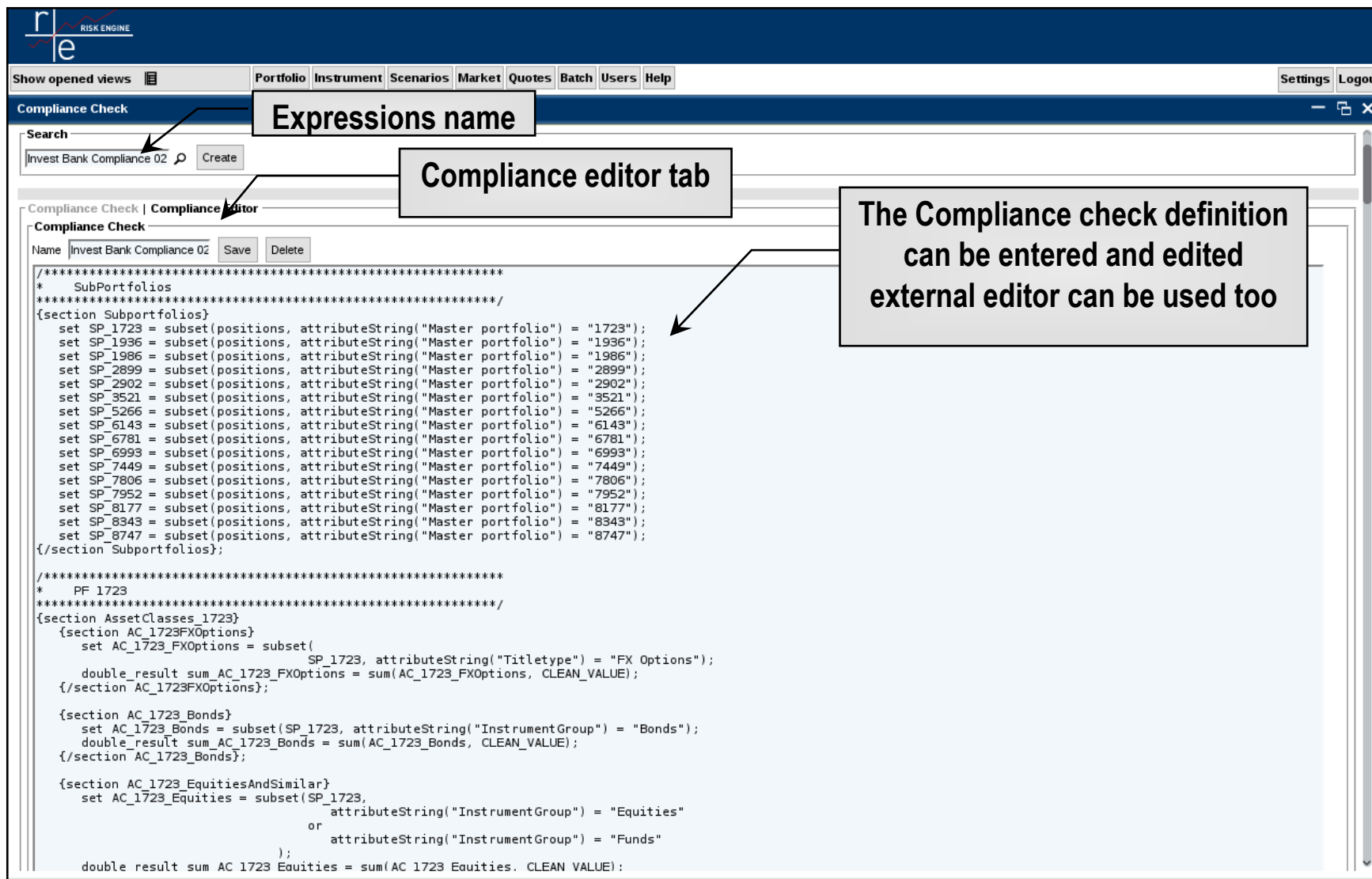
```



The screenshot displays the 'Compliance Check' window within the Risk Engine application. The interface includes a top navigation bar with the 'r|e RISK ENGINE' logo and a menu with options: 'Show opened views', 'Portfolio', 'Instrument', 'Scenarios', 'Market', 'Quotes', 'Batch', 'Users', 'Help', 'Settings', and 'Logout'. The 'Compliance Check' window has a title bar with standard window controls. Inside, there is a 'Search' section with a text input field containing 'Invest Bank Compliance' and a 'Create' button. Below this is a 'Compliance Check | Compliance Editor' section. It contains a 'Portfolio' field with the value 'Invest bank, actual' and a 'Calculate' button. The 'Evaluation Date' field is set to '23.09.2011'. Two annotations are present: a box labeled 'Portfolio selection' with an arrow pointing to the 'Invest Bank Compliance' search field, and a box labeled 'Date' with an arrow pointing to the 'Evaluation Date' field.

Portfolio selection

Date



The screenshot shows the 'Compliance Check' window in the Risk Engine application. The interface includes a menu bar with 'Portfolio', 'Instrument', 'Scenarios', 'Market', 'Quotes', 'Batch', 'Users', and 'Help'. A 'Search' bar at the top contains the text 'Invest Bank Compliance 02'. Below this, the 'Compliance editor tab' is active, displaying a list of compliance checks. The selected check, 'Invest Bank Compliance 02', is shown in a detailed view. This view includes a 'Name' field with the same text and 'Save' and 'Delete' buttons. The main area contains a code editor with a complex expression defining subportfolios and asset classes. Annotations with arrows point to the 'Expressions name' field, the 'Compliance editor tab', and a text box stating 'The Compliance check definition can be entered and edited external editor can be used too'.

Expressions name

Compliance editor tab

The Compliance check definition can be entered and edited external editor can be used too

```
*****
* SubPortfolios
*****
{section Subportfolios}
  set SP_1723 = subset(positions, attributeString("Master portfolio") = "1723");
  set SP_1936 = subset(positions, attributeString("Master portfolio") = "1936");
  set SP_1986 = subset(positions, attributeString("Master portfolio") = "1986");
  set SP_2899 = subset(positions, attributeString("Master portfolio") = "2899");
  set SP_2902 = subset(positions, attributeString("Master portfolio") = "2902");
  set SP_3521 = subset(positions, attributeString("Master portfolio") = "3521");
  set SP_5266 = subset(positions, attributeString("Master portfolio") = "5266");
  set SP_6143 = subset(positions, attributeString("Master portfolio") = "6143");
  set SP_6781 = subset(positions, attributeString("Master portfolio") = "6781");
  set SP_6993 = subset(positions, attributeString("Master portfolio") = "6993");
  set SP_7449 = subset(positions, attributeString("Master portfolio") = "7449");
  set SP_7806 = subset(positions, attributeString("Master portfolio") = "7806");
  set SP_7952 = subset(positions, attributeString("Master portfolio") = "7952");
  set SP_8177 = subset(positions, attributeString("Master portfolio") = "8177");
  set SP_8343 = subset(positions, attributeString("Master portfolio") = "8343");
  set SP_8747 = subset(positions, attributeString("Master portfolio") = "8747");
{/section Subportfolios};

*****
* PF 1723
*****
{section AssetClasses_1723}
  {section AC_1723FXOptions}
    set AC_1723_FXOptions = subset(
      SP_1723, attributeString("Titletype") = "FX Options");
    double_result sum_AC_1723_FXOptions = sum(AC_1723_FXOptions, CLEAN_VALUE);
  {/section AC_1723FXOptions};

  {section AC_1723_Bonds}
    set AC_1723_Bonds = subset(SP_1723, attributeString("InstrumentGroup") = "Bonds");
    double_result sum_AC_1723_Bonds = sum(AC_1723_Bonds, CLEAN_VALUE);
  {/section AC_1723_Bonds};

  {section AC_1723_EquitiesAndSimilar}
    set AC_1723_Equities = subset(SP_1723,
      attributeString("InstrumentGroup") = "Equities"
      or
      attributeString("InstrumentGroup") = "Funds"
    );
    double result sum AC 1723 Equities = sum(AC 1723 Equities, CLEAN VALUE);
  {/section AC_1723_EquitiesAndSimilar};
}
```


Calculation

Limit results as a table

Nodes can be opened and closed

The structure of the compliance system is shown as a tree

Selected node

The definition of a tree element from compliance description is shown in the window

Compared values of the limit

The percentages show the distance to the limit in %, i.e. reserve before limit violation

Simple limits with limit value

Complex limits

Satisfied or unsatisfied limits

Limits Table:

Num	Limit Name	Satisfied	LHS	RHS	Remaining
1	limit_AC_1723_Bonds	No	-	-	-
2	limit_AC_1723_Bonds_down	No	21.21	25.00	15.17%
3	limit_AC_1723_Bonds_up	Yes	21.21	31.00	31.59%
4	limit_AC_1723_Equities	No	-	10.00	100.00%
5	limit_AC_1723_FXOptions	Yes	-	-	-
6	limit_AC_1723_FXOptions_down	Yes	51.19	51.00	0.36%
7	limit_AC_1723_FXOptions_up	Yes	51.19	52.00	1.87%
8	limit_AC_1723_Liquidities	Yes	-	-	-
9	limit_AC_1723_Liquidities_down	Yes	27.61	9.00	67.40%
10	limit_AC_1723_Liquidities_up	Yes	27.61	48.00	42.48%
11	limit_AC_1936_Bonds	Yes	-	-	-
12	limit_AC_1936_Liquidities	Yes	-	-	-
13	limit_AC_1986_AlternativeInvestments	No	-	-	-
14	limit_AC_1986_Bonds	No	-	-	-
15	limit_AC_1986_Equities	Yes	-	-	-
16	limit_AC_1986_Liquidities	Yes	-	-	-
17	limit_AC_1986_MoneyMarket	Yes	-	-	-
18	limit_AC_1986_RealEstates	No	-	-	-

Sets Table:

Num	Set Name	Position Id
1	AC_1723_Bonds	XS0442330295@38877@1723
		XS0164067836@3508@1723
		XS0442348404@3987@1723
		XS0632833553@29821@1723
		XS0442330295@38877@1723

Additional Node Information:

Type: SET
Text: subset(positions, attributeString("Master portfolio") = "1723")
Positions count: 16

The screenshot displays the 'Compliance Check' window in the Risk Engine. The interface includes a menu bar (Show opened views, Portfolio, Instrument, Scenarios, Market, Quotes, Batch, Users, Help) and a toolbar (Settings, Logout). The main area is divided into three sections:

- Tree View:** A hierarchical tree on the left showing nodes like 'limit_AC_1723_Bonds', 'AC_1723_EquitiesLimits', 'sum_AC_1723_EquitiesPercentage', 'limit_AC_1723_Equities', 'AC_1723_LiquiditiesLimits', 'AssetClasses_1936', 'AssetClasses_1986', and 'AssetClasses_2899'.
- Additional Node Information:** A panel below the tree showing details for the selected node:


```
Type: SET
Text: subset(positions, attributeString("Master portfolio") = "1723")
Positions count: 16
```
- Sets Table:** A table on the right showing the positions for the selected set. The table has columns 'Hum', 'Set Name', and 'Position Id'.

Hum	Set Name	Position Id
1	AC_1723_Bonds	1 XS0442330295@38877@1723
2		XS0164067836@3508@1723
3		XS0442348404@3987@1723
4		XS0632833553@29821@1723
5		USM0300LAA46@32177@1723
6		XS0361041808@39245@1723
7		US45112FAB22@39246@1723
8		XS0494132540@39252@1723
1	AC_1723_FXOptions	1 999260000@39688@1723
2		999265000@40970@1723
3		999267000@40972@1723
4		999268000@40973@1723
3	AC_1723_Liquidities	1 8005-AUD@40469@1723
2		8005-USD@38620@1723
3		8005-EUR@38844@1723
4		8005-HKD@38845@1723
4	AC_1936_Bonds	1 XS0772024120@40584@1936
2		USM8933FAB33@28320@1936

Annotations in the image highlight the workflow:

- An arrow points from the text box "Selecting an element from the combo boxes opens the tree node and shows additional information for the element" to the 'Additional Node Information' panel.
- Another arrow points from the same text box to the 'Sets' table.
- A third arrow points from the text box "The positions belonging to the asset classes of the sub-portfolios are shown in the second table" to the 'Sets' table.

rle RISK ENGINE

Show opened views Portfolio Instrument Scenarios Market Quotes Batch Users Help Settings Logout

Compliance Check

Compliance Check | Compliance Editor

Portfolio: Invest bank, actual Calculate Save results

Evaluation Date: 23.09.2011

Results Tree

- root
 - Subportfolios
 - AssetClasses_1723
 - AC_1723FXOptions
 - AC_1723_FXOptions
 - SP_1723
 - positions
 - sum_AC_1723_FXOptions (Selected node of type Double result)
 - AC_1723_Bonds
 - AC_1723_Bonds
 - sum_AC_1723_Bonds
 - AC_1723_EquitiesAndSimilar
 - AC_1723_Equities
 - sum_AC_1723_Equities
 - AC_1723_Liquidities
 - AC_1723_Liquidities
 - sum_AC_1723_Liquidities
 - AC_1723_Limits
 - AssetClasses_1936
 - AssetClasses_1986

Additional Node Information

Type: DOUBLE_RESULT
Text: double_result sum_AC_1723_FXOptions = sum(AC_1723_FXOptions, CLEAN_VALUE);
Value: 4809996.68

Limits

Num	Limit Name	Satisfied	LHS	RHS	Remaining
1	limit_AC_1723_Bonds	No	-	-	-
2	limit_AC_1723_Bonds_down	No	21.21	25.00	15.17%
3	limit_AC_1723_Bonds_up	Yes	21.21	31.00	31.59%
4	limit_AC_1723_Equities	No	-	10.00	100.00%
5	limit_AC_1723_FXOptions	Yes	-	-	-
6	limit_AC_1723_FXOptions_down	Yes	51.19	51.00	0.36%
7	limit_AC_1723_FXOptions_up	Yes	51.19	52.00	1.57%
8	limit_AC_1723_Liquidities	Yes	-	-	-
9	limit_AC_1723_Liquidities_down	Yes	27.61	9.00	67.40%
10	limit_AC_1723_Liquidities_up	Yes	27.61	48.00	42.48%
11	limit_AC_1936_Bonds	Yes	-	-	-
12	limit_AC_1936_Liquidities	Yes	-	-	-
13	limit_AC_1986_AlternativeInvestments	No	-	-	-
14	limit_AC_1986_Bonds	No	-	-	-
15	limit_AC_1986_Liquidities	Yes	-	-	-
16	limit_AC_1986_Liquidities	Yes	-	-	-
17	limit_AC_1986_MoneyMarket	Yes	-	-	-
18	limit_AC_1986_RealEstate	No	-	-	-

Sets

Num	Set Name	Position Id
1	AC_1723_Bonds	XS0442330295@38877@1723
2		XS0164067836@3508@1723
3		XS0442348404@3987@1723
4		XS0632833553@29821@1723
5		XS0442330295@38877@1723

Base limit system implementation in Risk Engine

Compliance Check | Compliance Editor

Portfolio:

Evaluation Date:

Results Tree

- root
 - Subportfolios
 - AssetClasses_1723
 - AC_1723FXOptions
 - AC_1723_FXOptions
 - SP_1723
 - positions
 - sum_AC_1723_FXOptions

Selected node of type Set

Additional Node Information

Type: SET
Text: subset(positions, attributeString("Master portfolio") = "1723")
Positions count: 16

Limits

Num	Limit Name	Satisfied	LHS	RHS	Remaining
1	limit_AC_1723_Bonds	No	-	-	-
2	limit_AC_1723_Bonds_down	No	21.21	25.00	15.17%
3	limit_AC_1723_Bonds_up	Yes	21.21	31.00	31.59%
4	limit_AC_1723_Equities	No	-	10.00	100.00%
5	limit_AC_1723_FXOptions	Yes	-	-	-
6	limit_AC_1723_FXOptions_down	Yes	51.19	51.00	0.36%
7	limit_AC_1723_FXOptions_up	Yes	51.19	52.00	1.57%
8	limit_AC_1723_Liquidities	Yes	-	-	-
9	limit_AC_1723_Liquidities_down	Yes	27.61	9.00	67.40%
10	limit_AC_1723_Liquidities_up	Yes	27.61	48.00	42.48%
11	limit_AC_1936_Bonds	Yes	-	-	-
12	limit_AC_1936_Liquidities	Yes	-	-	-
13	limit_AC_1986_AlternativeInvestments	No	-	-	-
14	limit_AC_1986_Bonds	No	-	-	-
15	limit_AC_1986_Equities	Yes	-	-	-
16	limit_AC_1986_Liquidities	Yes	-	-	-
17	limit_AC_1986_MoneyMarket	Yes	-	-	-
18	limit_AC_1986_RealEstates	No	-	-	-

Limit:

Sets

Num	Set Name	Position Id
		1 XS0442330295@38877@1723
		2 XS0164067836@3508@1723
		3 XS0442348404@3987@1723
1	AC_1723_Bonds	4 XS0632833553@29821@1723
		5 XS0442330295@38877@1723

Base limit system implementation in Risk Engine

rle RISK ENGINE

Show opened views Portfolio Instrument Scenarios Market Quotes Batch Users Help Settings Logout

Compliance Check

Results Tree

- AC_1723_Liquidities
 - AC_1723_Liquidities
 - SP_1723
 - positions
 - sum_AC_1723_Liquidities
- AC_1723_Limits
 - sum1723
 - AC_1723_FXOptionsLimits
 - AC_1723_BondsLimits
 - sum_AC_1723_BondsPercentage
 - limit_AC_1723_Bonds_down
 - limit_AC_1723_Bonds_up
 - limit_AC_1723_Bonds
 - AC_1723_EquitiesLimits
 - sum_AC_1723_EquitiesPercentage
 - limit_AC_1723_Equities
 - AC_1723_LiquiditiesLimits
- AssetClasses_1936
- AssetClasses_1986
- AssetClasses_2899

Additional Node Information

Type: LIMIT_CONDITION
Text: limit limit_AC_1723_Bonds_up = sum_AC_1723_BondsPercentage < 31;
Satisfied: true
LHS: 21.21
RHS: 31.00
Remaining: 31.59%

Edit limit node

Limit LHS: sum_AC_1723_BondsPercentage
Condition: <
Limit RHS: 31
Change

Limits

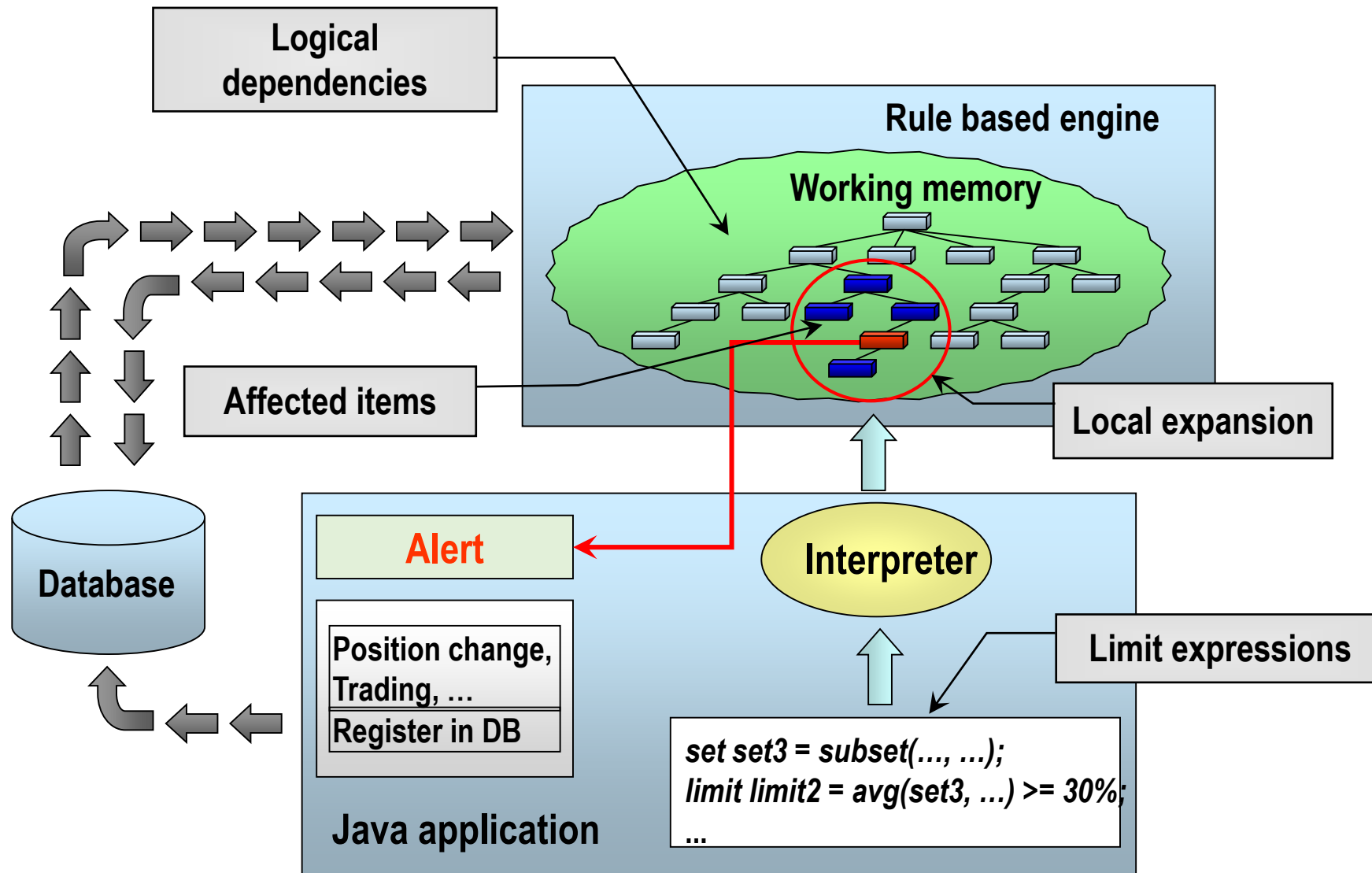
Num	Limit Name	Satisfied	LHS	RHS	Remaining
1	limit_AC_1723_Bonds	No	-	-	-
2	limit_AC_1723_Bonds_down	No	21.21	25.00	15.17%
3	limit_AC_1723_Bonds_up	Yes	21.21	31.00	31.59%
4	limit_AC_1723_Equities	No	-	10.00	100.00%
5	limit_AC_1723_FXOptions	Yes	-	-	-
6	limit_AC_1723_FXOptions_down	Yes	51.19	51.00	0.36%
7	limit_AC_1723_FXOptions_up	Yes	51.19	52.00	1.57%
8	limit_AC_1723_Liquidities	Yes	-	-	-
9	limit_AC_1723_Liquidities_down	Yes	27.61	9.00	67.40%
10	limit_AC_1723_Liquidities_up	Yes	27.61	48.00	42.48%
11	limit_AC_1936_Bonds	Yes	-	-	-
12	limit_AC_1936_Liquidities	Yes	-	-	-
13	limit_AC_1986_AlternativeInvestments	No	-	-	-
14	limit_AC_1986_Bonds	No	-	-	-
15	limit_AC_1986_Equities	Yes	-	-	-
16	limit_AC_1986_Liquidities	Yes	-	-	-
17	limit_AC_1986_MoneyMarket	Yes	-	-	-
18	limit_AC_1986_RealEstates	No	-	-	-

Limit: limit_AC_1723_Bonds

Sets

Set Name	Position Id
AC_1723_Bonds	1 XS0442330295@38877@1723
	2 XS0164067836@3508@1723
	3 XS0442348404@3987@1723
	4 XS0632833553@29821@1723
	5 USM0300LAA46@32177@1723
	6 XS0361041808@39245@1723
	7 US45112FAB22@39246@1723
	8 XS0494132540@39252@1723
	1 999260000@39688@1723

Limit nodes can be directly edited in the GUI



1. Loading of positions from the database
2. Interpretation of the limit expressions from the application
3. Building of a tree of logically dependent items in the working memory according to the expressions
4. The tree structure is residing in the working memory in an awaiting regime
5. When any change occurs (position field change, trading, etc.) the change is registered in the database
6. Changing item causes local expansion in the built tree, the other tree nodes stay unchanged
7. If a limit is violated (or became near to the violation) as a result of the change an alert is output

```
<expression> ::= <a_set> | <a_limit> | <a_result> | <a_dynamic_set> |  
                <a_classifier> | <a_section>  
<a_set> ::= 'set' <name> '=' <set> ';'   
<a_limit> ::= 'limit' <name> '=' <limit> ';'   
<a_result> ::= 'double_result' <name> '=' <doubleResultExpr> ';'   
<a_dynamic_set> ::= dynamicSet '(' <set> ',' <stringPositionProperty> (','  
                <stringPositionProperty>)* ',' <prefix> ')' ';'   
<a_classifier> ::= 'classifier' <name> '=' 'classify' '(' <set> ',' <columnName> ','  
                <columnContent> ',' <limit> ')' ';'   
<columnName> ::= <string>  
<columnContent> ::= <stringPositionValue> | <doublePositionValueExpr> | <datePositionValue>  
<a_section> ::= <start_section> <expression> <end_section>  
<start_section> ::= '{' 'section' <name> '}'  
<end_section> ::= '{' '/section' <name> '}' ';'   
<set> ::= <setExpr> (('and' | 'or') setExpr)*
```



```
<setExpr> ::=
    <name>
    | ('not')? '(' <set> ')'
    | 'subset' '(' <set> ',' <positionFilter> ')'
    | 'max' '(' <set> ',' <doublePositionValueExpr> ',' <start> ',' <size> ')'
    | 'min' '(' <set> ',' <doublePositionValueExpr> ',' <start> ',' <size> ')'
    | 'dynamicUnion' '(' <string> (',' <string>)* (',' 'distinct_by' '(' <propertiesList> ''))?
        ',' <doubleDynamicResultExpr> <doubleLimitOp>
        <doubleDynamicResultExpr> ')'
<doubleLimitOp> ::= '>' | '>=' | '<' | '<=' | '=' | '<>'
<start> ::= <uint>
<size> ::= <uint>
<positionFilter> ::= <positionFilterExpr> (('and' | 'or') <positionFilterExpr>)*
<positionFilterExpr> ::= ('not')? '(' <positionFilter> ')' | <stringPositionFilter> |
    <doublePositionFilter> | <datePositionFilter>
<stringPositionFilter> ::= <stringPositionValue> ('=' | '<>' | (('not')? 'like')) <stringPositionValue>
```

```

<stringPositionFilter> ::= <stringPositionValue> ('=' | '<' | (('not')? 'like')) <stringPositionValue>
<stringPositionValue> ::= <string> | <stringPositionProperty> | 'attributeString'
                        '(' <stringPositionAttribute> ')'
<stringPositionProperty> ::= <string_type_position_property>
<stringPositionAttribute> ::= <string_type_position_attribute>
<doublePositionFilter> ::= <doublePositionValueExpr> <doubleLimitOp>
                        <doublePositionValueExpr>
<doublePositionValueExpr> ::= <doublePositionValueExprL2> (('+' | '-')
                        <doublePositionValueExprL2>)*
<doublePositionValueExprL2> ::= <doublePositionValueExprL3> (('*' | '/')
                        <doublePositionValueExprL3>)*
<doublePositionValueExprL3> ::= ('+' | '-')? (('(' <doublePositionValueExpr> ')') |
                        <doublePositionValueExprL4>)
<doublePositionValueExprL4> ::= <number> | 'result.' <name> | <doublePositionProperty> |
                        'attributeDouble' '(' <doublePositionAttribute> ')'

```

```
<doublePositionProperty> ::= <double_type_position_property>
<doublePositionAttribute> ::= <double_type_position_attribute>
<datePositionFilter> ::= <datePositionValue> <doubleLimitOp> <datePositionValue>
<datePositionValue> ::= <date> | <datePositionProperty> | 'attributeDate' '(' <datePositionAttribute> ')'
<datePositionProperty> ::= <date_type_position_property>
<datePositionAttribute> ::= <date_type_position_attribute>
<doubleResultExpr> ::= <doubleResultExprL2> (('+' | '-') <doubleResultExprL2>)*
<doubleResultExprL2> ::= <doubleResultExprL3> (('*' | '/') <doubleResultExprL3>)*
<doubleResultExprL3> ::= ('+' | '-')? (('(' <doubleResultExpr> ')') | <doubleResultExprL4> )
<doubleResultExprL4> ::=
    number
  | name
  | 'avg' '(' <set> ',' <doublePositionValueExpr> ')'
  | 'sum' '(' <set> ',' <doublePositionValueExpr> ')'
  | 'value' '(' <set> ',' <doublePositionValueExpr> ',' <uint> ')'
  | 'size' '(' <set> ')'
  | 'dynamicSetSize' '(' <prefix> ',' 'distinct_by' '(' <propertiesList> ')<prefix> ::= <string>
```

<limit> ::= <limitExpr> (('and' | 'or') <limitExpr>)*
<limitExpr> ::= ('not')? '(' <limit> ')' | <doubleLimit> | 'forall' '(' <set> ',' <positionFilter> ')' |
<dynamicSetLimit> | <name>
<doubleLimit> ::= <doubleResultExpr> <doubleLimitOp> <doubleResultExpr>
<dynamicSetLimit> ::= 'Forall' '(' <string> (',' <string>)*(',' 'distinct_by' '(' <propertiesList> ''))?
',' <doubleDynamicResultExpr> <doubleLimitOp>
<doubleDynamicResultExpr> ')'
<propertiesList> ::= <stringPositionProperty> '=' (<string> | 'all') (',' <stringPositionProperty> '='
(<string> | 'all'))*
<datePositionProperty> ::= <date_type_position_property>
<datePositionAttribute> ::= <date_type_position_attribute>
<doubleResultExpr> ::= <doubleResultExprL2> (('+' | '-') <doubleResultExprL2>)*
<doubleResultExprL2> ::= <doubleResultExprL3> (('*' | '/') <doubleResultExprL3>)*
<doubleResultExprL3> ::= ('+' | '-')? (('(' <doubleResultExpr> ')) | <doubleResultExprL4>)

```
<doubleResultExprL4> ::=
    number
  | name
  | 'avg' '(' <set> ',' <doublePositionValueExpr> ')'
  | 'sum' '(' <set> ',' <doublePositionValueExpr> ')'
  | 'value' '(' <set> ',' <doublePositionValueExpr> ',' <uint> ')'
  | 'size' '(' <set> ')'
  | 'dynamicSetSize' '(' <prefix> ',' 'distinct_by' '(' <propertiesList> ')'

<prefix> ::= <string>
<limit> ::= <limitExpr> (('and' | 'or') <limitExpr>)*
<limitExpr> ::= ('not')? '(' <limit> ')' | <doubleLimit> | 'forall' '(' <set> ',' <positionFilter> ')' |
               <dynamicSetLimit> | <name>
<doubleLimit> ::= <doubleResultExpr> <doubleLimitOp> <doubleResultExpr>
<dynamicSetLimit> ::= 'Forall' '(' <string> (',' <string>)* (',' 'distinct_by' '(' <propertiesList> ''))?
                    ',' <doubleDynamicResultExpr> <doubleLimitOp> <doubleDynamicResultExpr> ')'
<propertiesList> ::=
    <stringPositionProperty> '=' (<string> | 'all') (',' <stringPositionProperty> '=' (<string> | 'all'))*
```

```
<doubleDynamicResultExpr> ::=
    <doubleDynamicResultExprL2> (('+' | '-') <doubleDynamicResultExprL2>)*
<doubleDynamicResultExprL2> ::=
    <doubleDynamicResultExprL3> (('*' | '/') <doubleDynamicResultExprL3>)*
<doubleDynamicResultExprL3> ::=
    ('+' | '-')? (('(' <doubleDynamicResultExpr> ')') | <doubleDynamicResultExprL4>)
<doubleDynamicResultExprL4> ::=
    <number>
    | <name>
    | 'avg' '(' <set> ',' <doublePositionValueExpr> ')'
    | 'sum' '(' <set> ',' <doublePositionValueExpr> ')' | 'value' '(' <set> ','
<doublePositionValueExpr> ',' <uint> ,)'
    | 'size' '(' <set> ,)'
    | 'Sum' '(' <prefix> ',' <doublePositionValueExpr> ')'
    | 'Avg' '(' <prefix> ',' <doublePositionValueExpr> ')'
    | 'Size' '(' <prefix> ')'
    | 'Items' '(' <prefix> ',' <doublePositionValueExpr> ')'
```

```
<prefix> ::= <string>
<name> ::= <letter> (<digit> | <letter>)*
<number> ::= (<udouble> (<numberType>)? ) | <uint>
<numberType> ::= 'Mio' | '%' | 'bp'
<uint> ::= '0' | (( '1'..'9' ) <digit>*)
<udouble> ::= <uint> '.' (<digit>+ | (<digit>+ ('e' | 'E') ('+' | '-') <digit>))
<digit> ::= '0'..'9'
<letter> ::= ('a'..'z') | ('A'..'Z') | ('_')
<date> ::= 'to_date' '(' <dateString> ',' <dateFormat> ')' | <function>
<dateString> ::= <string>
<dateFormat> ::= <string>
<string> ::= '"' ( ( '\\\\' | '\\"' ) | ~('\\\\' | '"') ) * '"'
<function> ::= 'function' '(' <string> (',' <argumentsList>)? ')'
<argumentsList> ::= (<string> | <function>) (',' (<string> | <function>))*
```

Functions:

add_year – adds specified number of years to a given date

Format:

```
function("add_year", <date>, <years_to_be_added>)
```

```
<date> ::= <string> | <function>
```

```
<years_to_be_added> ::= <string>
```

Examples:

```
function("add_year", "10.01.2009", "1"))
```

```
function("add_year", function("eval_date"), "1"))
```

The date is in the format "dd.MM.yyyy"

add_month – adds specified number of months to a given date

Format:

```
function("add_month", <date>, <months_to_be_added>)  
date>::=<string> | <function>  
<months_to_be_added> ::= <string>
```

Examples:

```
function("add_month", "10.01.2009", "1")  
function("add_month", function("eval_date"), "1")
```

The date is in the format "dd.MM.yyyy"

eval_date – returns the date for which the limits check is performed

Format:

```
function("eval_date")
```