UNIVERSITÄT KOBLENZ · LANDAU

IWVI

BUSINESS PROCESS MANAGEMENT – EXERCISE

GENERIC MODEL QUERY LANGUAGE (GMQL)

GMQL – BASIC IDEA

- Recognize any model as two basic sets
  - Set V of model vertices
  - Set E of model edges
- Provide set-altering functions and operators that perform operations on these basic sets
- Nest functions and operators to assemble a query

PARAMETERS

- X = any set of elements (e.g., Z, V, E)
  - Y_V = any set of vertices (e.g., V)
  - Y_E = any set of edges (e.g., E)
- t = a particular element type (e.g., event, f_e)
  - t_v = a particular vertex type (e.g., event, function)
  - t_E = a particular edge type (e.g., f_e, x_e)
- u = a particular value or data type (e.g., “*invoice*”)
- n_x = a natural number (e.g., 1, 2)

GMQL FUNCTIONS

- Element characteristics
- Elements and their relationships
- Adjacent elements
- Element paths/loops

GMQL FUNCTIONS - 1

ELEMENT CHARACTERISTICS

- ElementsOfType(X,t)
- ElementsWithAttributeOfValue(X,t_v,u)
- ElementsWithAttributeOfDataType(X,u)

→ return a simple set

Example: $\{v_1, v_2, v_3\}$

GMQL FUNCTIONS – 2

ELEMENTS AND THEIR RELATIONSHIPS

- ElementsWithRelations(X, Y_E)
  - ElementsWithSuccRelations(X, Y_E)
  - ElementsWithPredRelations(X, Y_E)

- ElementsWithRelationsOfType(X, Y_E, t_E)
  - ElementsWithSuccRelationsOfType(X, Y_E, t_E)
  - ElementsWithPredRelationsOfType(X, Y_E, t_E)

→ return a set of sets

Example: {{v_1,e_1},{v_2,e_2,e_3}}

GMQL FUNCTIONS – 2

ELEMENTS AND THEIR RELATIONSHIPS

- ElementsWithNumberOfRelations(X,n_x)
  - ElementsWithNumberOfSuccRelations(X,n_x)
  - ElementsWithNumberOfPredRelations(X,n_x)

- ElementsWithNumberOfRelationsOfType(X,t_E,n_x)
  - ElementsWithNumberOfSuccRelationsOfType(X,t_E,n_x)
  - ElementsWithNumberOfPredRelationsOfType(X,t_E,n_x)

→ return a set of sets

Example: {{v_1,e_1},{v_2,e_2,e_3}}

GMQL FUNCTIONS – 3

ADJACENT ELEMENTS

- ElementsDirectlyRelated(X_1,X_2) — undirected edges
- AdjacentSuccessors(X_1,X_2) — directed edges

→ return a set of sets

Example: {{v_1,v_2,e_1},{v_2,v_3,e_2}}

GMQL FUNCTIONS - 4

ELEMENT PATHS

- Paths(X_1,X_n)
  - PathsContainingElements(X_1,X_n,X_c)
  - PathsNotContainingElements(X_1,X_n,X_c)

undirected edges

- DirectedPaths(X_1,X_n)
  - DirectedPathsContainingElements(X_1,X_n,X_c)
  - DirectedPathsNotContainingElements(X_1,X_n,X_c)

directed edges

→ return a set of sets

GMQL FUNCTIONS – 4

ELEMENT LOOPS

- Loops(X)
  - LoopsContainingElements(X_1,X_c)
  - LoopsNotContainingElements(X_1,X_c)

undirected edges

- DirectedLoops(X)
  - DirectedLoopsContainingElements(X_1,X_c)
  - DirectedLoopsNotContainingElements(X_1,X_c)

directed edges

→ return a set of sets

GMQL OPERATORS

- Most operators take exactly two parameters
  - two simple sets or two sets of sets
    - (UNION, INTERSECTION, COMPLEMENT)
  - two sets of sets or one set of sets and one simple set
    - (INNERINTERSECTION, INNERCOMPLEMENT)
  - two sets of sets
    - (JOIN)

- Operators that turn sets of sets into simple sets only take one parameter
  - one set of sets
    - (SELFUNION, SELFINTERSECTION)

COMMON ERRORS – OPERATOR NESTING

- If you want to apply operators to more than two sets, you have to nest them

- Example: the set of all connectors

```
UNION(
        UNION(
                EOT(V, XOR),
                EOT(V, OR)),
        EOT(V, AND))
```

DON'T DO THIS:

```
UNION(
        EOT(V, XOR),
        EOT(V, OR),
        EOT(V, AND))
```

VARIABLES

- In order to keep your GMQL functions clear, you can make use of variables
- Idea: Define sets or sets of sets that you can then use in your actual functions

Example: All connectors with exactly one predecessor

C = UNION(UNION(EOT(V, XOR),EOT(V, OR)),EOT(V, AND))

EWNOPR(C,1)

SPLIT-JOIN-CONNECTORS

Connectors that have more than one predecessor and more than one successor

- COMPLEMENT
  - UNION of all connectors (nested, see previous slides)
  - UNION of all connectors (nested)
    - with 0 predecessors
    - with 0 successors
    - with exactly 1 predecessor
    - with exactly 1 successor

COMMON ERRORS – OPERATORS

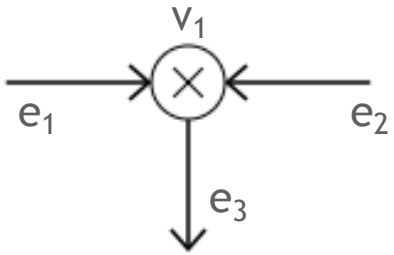
- Think about what your sets/subsets look like, before applying operators

Example: the set of all connectors with exactly two incoming and one outgoing edge

C2 = EWNOPR(C,2) → {{v_1,e_1,e_2}}

C1 = EWNOSR(C,1) → {{v_1,e_3}}

INTERSECTION(C1,C2) = {}



C has to be defined beforehand!

GMQL VS. CTL

- Examples of GMQL functions that can be expressed by equivalent CTL formulas

GMQL	CTL
Adjacent Successors(X_1,X_2)	M, x_1 | = EX x_2
Paths(EWAOV(V,label,"o_boss"), EWAOV(V,label,"n_grant_credit"))	M, x_0 | = EF(n_grant_credit ∧ o_boss)

- CTL is not able to count elements (e.g., EWNOR) or detect loops

For further reference: Riehle, D. (2018). Checking Business Process Models for Compliance–Comparing Graph Matching and Temporal Logic. In *International Conference on Business Process Management* (pp. 403-415). Springer, Cham.

UNIVERSITÄT KOBLENZ · LANDAU

IWVI

BUSINESS PROCESS MANAGEMENT – EXERCISE

INSTITUTE FOR IS RESEARCH

www.uni-koblenz.de