May 28, 2021

This document is part of the paper "$\mathcal{ELKG}_{app}$: An Alternative Approach to Represent Multi-dimensional MK in the Web of Data". It presents the matching algorithm which is used in the paper.

1 Algorithm

Algorithm Step 3.2: matching_function

Input: line send by Algorithm 4 (see the paper).
Output: Result send back to Algorithm 4 (see the paper).

```
1: procedure EXECUTE query pattern containing query
2:     for each query pattern do
3:         if sub is variable, obj is variable then.
4:             if sub variable is new then
5:                 Append the contents of sub_dict of result dictionary with that of sub_dict of
   current dictionary. And Add variable to list in vars['sub']
6:             end if
7:             if sub is variable and is previously encountered then
8:                 Find the position of previous encounter and select the associated_dictionary.
9:                 for each key in associated_dictionary do
10:                    if key exists in current dictionary's sub_dict then
11:                        Make union of current dictionary's sub_dict[key] and result dictionary's
   associated_dictionary[key].
12:                        Store it as value of associated_dictionary[key].
13:                    end if
14:                    if key not found in sub_dict of current dictionary then
15:                        Delete the key from associated_dictionary of result dictionary.
16:                    end if
17:                end for
18:            end if
19:            if obj variable is new then
20:                Append the contents of obj_dict of result dictionary with that of obj_dict of
   current dictionary. Add variable to list in vars['obj']
21:            end if
22:            if obj is variable and is previously encountered then
23:                Find the position of previous encounter and select the associated_dictionary.
24:                for each key in associated_dictionary do
25:                    if key exists in current dictionary's obj_dict then
26:                        Make union of current dictionary's obj_dict[key] and result dictionary's
   associated_dictionary[key].
27:                        Store it as value of associated_dictionary[key].
28:                    end if
29:                    if key not found in obj_dict of current dictionary then
30:                        Delete the key from associated_dictionary of result dictionary.
31:                    end if
32:                end for
33:            end if
34:        end if
35:        if sub is given, obj is variable then
36:            Keep the matching given sub in result dictionary's sub_dict.
37:            for each non_matching key in sub_dict do
38:                Remove sub from obj_dict of result dictionary.
39:                Remove obj from uid_dict of result dictionary.
40:            end for
41:            Remove the non_matching keys.
42:            if obj variable is new then
43:                Append the contents of obj_dict of result dictionary with that of obj_dict of
   current dictionary and add variable to list in vars['obj']
44:            end if
45:            if obj is variable and is previously encountered then
46:                Find the position of previous encounter and select the associated_dictionary.
47:                for each key in associated_dictionary do
48:                    if key exists in current dictionary's obj_dict then
49:                        Make union of current dictionary's obj_dict[key] and result dictionary's
   associated_dictionary[key].
50:                        Store it as value of associated_dictionary[key].
51:                    end if
52:                    if key not found in obj_dict of current dictionary then
53:                        Delete the key from associated_dictionary of result dictionary.
54:                    end if
55:                end for
56:            end if
57:        end if
```

```
58:         if sub is variable, obj is given then
59:             Keep the matching given obj in result dictionary's sub-dict.
60:             for each non-matching key in obj-dict do
61:                 Remove sub from sub-dict of result dictionary.
62:                 Remove obj from uid-dict of result dictionary.
63:             end for
64:             Remove the non-matching obj keys.
65:             if sub variable is new then
66:                 Append the contents of sub-dict of result dictionary with that of sub-dict of
    current dictionary.
67:                 Add variable to list in vars['sub']
68:             end if
69:             if sub is variable and is previously encountered then
70:                 Find the position of previous encounter.
71:                 Previously at sub, obj or uid position, select the associated_dictionary.
72:                 for each key in associated_dictionary do
73:                     if key exists in current table's sub-dict then
74:                         Make union of current dictionary's sub-dict[key] and result table's associ-
    ated_dictionary[key].
75:                         Store it as value of associated_dictionary[key].
76:                     end if
77:                     if key not found in sub-dict of current dictionary then
78:                         Delete the key from associated_dictionary of result dictionary.
79:                     end if
80:                 end for
81:             end if
82:         end if
83:         if sub is given, obj is given then
84:             Keep the matching given obj in result dictionary's sub-dict.
85:             for each non-matching key in obj-dict do
86:                 Remove sub from sub-dict of result dictionary.
87:                 Remove obj from uid-dict of result dictionary.
88:             end for
89:             Remove the non-matching keys.
90:             Keep the matching given sub in result dictionary's sub-dict.
91:             for each non-matching key in sub-dict do
92:                 Remove sub from obj-dict of result dictionary.
93:                 Remove obj from uid-dict of result dictionary.
94:             end for
95:             Remove the non-matching keys.
96:         end if
97:         Start UID1 processing.
98:         if UID is variable in user query then check:
99:             if UID variable is new then
100:                Append the contents of uid-dict of result dictionary with that of uid-dict of
    current dictionary.
101:                Add variable to list in vars['uid']
102:            end if
103:            if UID is variable and is previously encountered then
104:                Find the position of previous encounter.
105:                Previously at sub, obj or uid position, select the associated_dictionary.
106:                for each key in associated_dictionary do
107:                    if key exists in current table's uid-dict then
108:                        Make union of current dictionary's uid-dict[key] and result table's asso-
    ciated_dictionary[key].
109:                        Store it as value of associated_dictionary[key].
110:                    end if
111:                    if key not found in uid-dict of current dictionary then
112:                        Delete the key from associated_dictionary of result dictionary.
113:                    end if
114:                end for
115:            end if
116:        end if
```

```
        if UID1 is given then
            Keep the matching given UID in result dictionary's uid-dict.
            for each non-matching key in uid-dict do
                Remove sub from sub-dict of result dictionary.
                Remove obj from obj-dict of result dictionary.
            end for
            Remove the non-matching keys.
        end if
        Predicate parameters processing.
        for each parameter do
            if parameter if variable then
                Check if parameter is new OR Check if parameter matches with previous parameter variables and match
            end if
            if parameter is given then
                Match given parameter accordingly with result table.
            end if
            if parameter is not required then
                No processing required.
            end if
        end for
        Start UID2 processing.
        if UID2 is variable then
            Check if UID2 is new OR
            Check if UID2 matches with previous variables and match accordingly
        end if
        if UID2 is given then
            Match given UID2 with result table
        end if
    end for
end procedure
```