1. **Apply FCA to generate hierarchy and matrix of concepts**

the formal context is input to the FCA tool to generate concepts and their hierarchy. Formal context refer to the matrix/table with objects as rows and properties as columns, it contains binary values 0-1 to represent if ith object possesses jth object. If ith row and jth column is marked as 1 it means ith object possesses the jth properties otherwise not. FCA tool will generate the concepts and hierarchy for the input table. here we will input two matrices one for student-skill and other for project-skill, the tool will generate the set of concepts, each concept consists of group of objects and properties. Here we have the tables for k skills example.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | data integration | frequent patterns | supervised learning | neural chips | core memory | manufacturing systems | flexible structures | probability | m-p model | internet |
| student1 |  | X | X |  |  | X |  | X |  | X |
| student2 | X |  |  |  |  |  |  |  |  |  |
| student3 |  | X |  |  |  |  |  |  | X | X |
| student4 |  |  | X |  |  |  |  |  |  |  |
| student5 | X | X |  |  | X | X |  |  |  | X |
| student6 |  |  |  |  | X |  | X |  |  |  |
| student7 |  |  |  |  |  |  |  | X |  |  |
| student8 | X |  | X |  |  |  |  |  |  |  |
| student9 |  |  |  |  |  |  | X |  |  |  |
| student10 |  |  | X |  |  |  |  | X |  | X |

Student-skill table (m students – k skills)

Project-skill table ( n projects – k skills )

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | data integration | frequent patterns | supervised learning | neural chips | core memory | manufacturing systems | flexible structures | probability | m-p model | internet |
| project 1 |  |  | X | X |  |  | X |  |  | X |
| project 2 | X |  |  |  |  |  |  | X |  | X |
| project 3 | X |  | X |  | X |  |  |  |  |  |
| project 4 |  |  |  |  |  |  |  |  | X |  |
| project 5 |  |  |  |  |  | X |  |  |  | X |
| project6 |  | X |  |  |  |  |  |  |  |  |
| project7 | X |  |  | X |  |  |  |  | X | X |
| project8 |  |  |  |  |  |  |  | X |  |  |
| project9 |  |  |  |  |  | X |  |  |  | X |
| project10 | X |  | X |  | X |  | X | X |  | X |

Output concepts generated by the Tool

For student-skill table,

*c0 = () > ('data integration', 'frequent patterns', 'supervised learning', 'neural chips', 'core memory', 'manufacturing systems', 'flexible structures', 'probability', 'm-p model', 'internet')*

*c1 = ('student1',) > ('frequent patterns', 'supervised learning', 'manufacturing systems', 'probability', 'internet')*

*c2 = ('student3',) > ('frequent patterns', 'm-p model', 'internet')*

*c3 = ('student5',) > ('data integration', 'frequent patterns', 'core memory', 'manufacturing systems', 'internet')*

*c4 = ('student6',) > ('core memory', 'flexible structures')*

*c5 = ('student8',) > ('data integration', 'supervised learning')*

*c6 = ('student1', 'student5') > ('frequent patterns', 'manufacturing systems', 'internet')*

*c7 = ('student1', 'student10') > ('supervised learning', 'probability', 'internet')*

*c8 = ('student5', 'student6') > ('core memory',)*

*c9 = ('student6', 'student9') > ('flexible structures',)*

*c10 = ('student1', 'student3', 'student5') > ('frequent patterns', 'internet')*

*c11 = ('student1', 'student7', 'student10') > ('probability',)*

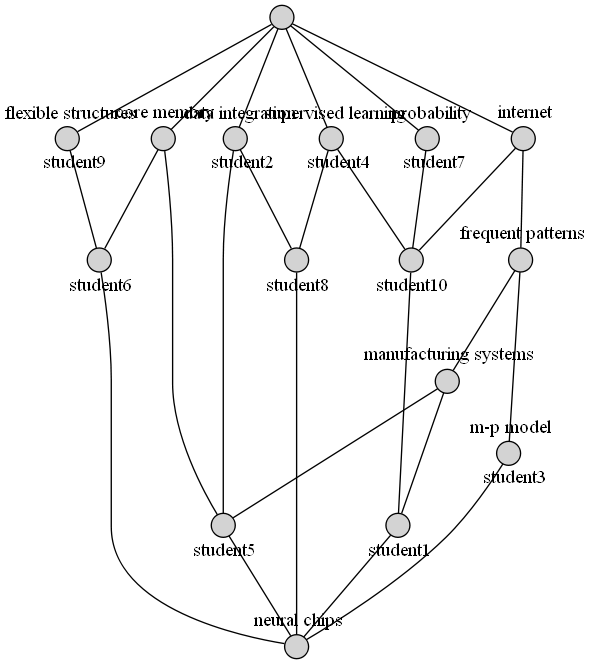
*c12 = ('student2', 'student5', 'student8') > ('data integration',)*

*c13 = ('student1', 'student3', 'student5', 'student10') > ('internet',)*

*c14 = ('student1', 'student4', 'student8', 'student10') > ('supervised learning',)*

*c15 = ('student1', 'student2', 'student3', 'student4', 'student5', 'student6', 'student7', 'student8', 'student9', 'student10') > ()*

Concept-lattice for above concepts



For project-skill table,

*c0 = () > ('data integration', 'frequent patterns', 'supervised learning', 'neural chips', 'core memory', 'manufacturing systems', 'flexible structures', 'probability', 'm-p model', 'internet')*

*c1 = ('project 1',) > ('supervised learning', 'neural chips', 'flexible structures', 'internet')*

*c2 = ('project6',) > ('frequent patterns',)*

*c3 = ('project7',) > ('data integration', 'neural chips', 'm-p model', 'internet')*

*c4 = ('project10',) > ('data integration', 'supervised learning', 'core memory', 'flexible structures', 'probability', 'internet')*

*c5 = ('project 1', 'project7') > ('neural chips', 'internet')*

*c6 = ('project 1', 'project10') > ('supervised learning', 'flexible structures', 'internet')*

*c7 = ('project 2', 'project10') > ('data integration', 'probability', 'internet')*

*c8 = ('project 3', 'project10') > ('data integration', 'supervised learning', 'core memory')*

*c9 = ('project 4', 'project7') > ('m-p model',)*

*c10 = ('project 5', 'project9') > ('manufacturing systems', 'internet')*

*c11 = ('project 1', 'project 3', 'project10') > ('supervised learning',)*

*c12 = ('project 2', 'project7', 'project10') > ('data integration', 'internet')*

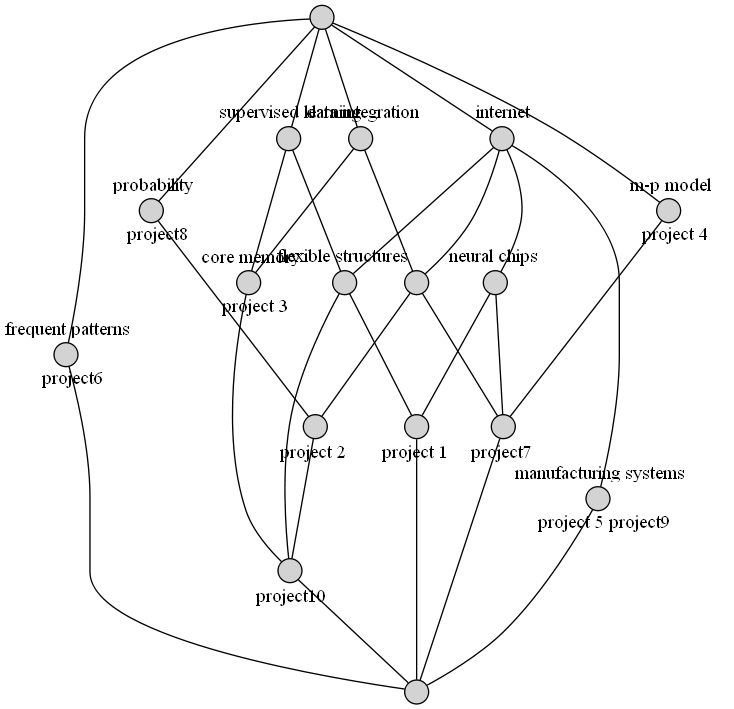
*c13 = ('project 2', 'project8', 'project10') > ('probability',)*

*c14 = ('project 2', 'project 3', 'project7', 'project10') > ('data integration',)*

*c15 = ('project 1', 'project 2', 'project 5', 'project7', 'project9', 'project10') > ('internet',)*

*c16 = ('project 1', 'project 2', 'project 3', 'project 4', 'project 5', 'project6', 'project7', 'project8', 'project9', 'project10') > ()*

Concept lattice for above concepts



after refining the concepts that are not significant for our purpose like the concepts that doesn’t have any properties or objects like c0 in both outputs, c15 in student-skill concept and c16 in project-skill concept, the concept will be converted to matrix representation where each row represents concepts and first k columns represents skills and other columns represents student/project.

Matrix rep. for student-skill concepts

Student\_skill\_concept[p][k+m]

C1: [0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

C2: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]

C3: [1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]

C4: [1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

C5: [0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0]

C6: [0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1]

C7: [1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1]

C8: [1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1]

C9: [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0]

C10: [0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0]

C11: [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1]

C12: [1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1]

C13: [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1]

C14: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1]

C15: [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1]

Similar matrix rep. for project-skill concepts

Project\_skill\_concept[q][k+n]

[0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]

[1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]

[1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]

[0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0]

[0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1]

[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0]

[0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1]

[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1]

[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1]

1. **Generate affinity between each row of generated concept matrices**

Now, we will generate the affinity between each row of student\_skill\_concept matrix with each row of project\_skill\_concept matrix according to the skill that are present in both concepts.

Let we are calculating affinity between ith student\_skill\_concept and jth project\_skill\_concept then,

1. If some skill Sk is present in both concept. This shows that Sk skill is required by all project in jth project\_skill\_concept and is present in all students of the ith student\_skill\_concept.
2. If some skill Sk is not present in either of concept. This shows that Sk is nor required in the projects nor possesses by the students.
3. If some skill Sk is present in ith student\_skill\_concept but not in jth project\_skill\_concept.This is a case where the skill is present in all students of ith concept but it is not required by any of the project in jth concept.
4. If some skill Sk is present in jth project\_skill\_concept but not in ith student\_skill\_concept. This is the case where the skills is not present in students of ith concept but it is required by all the projects in jth concepts.

Case a gives the most favourable match since it satisfies a core skill requirement of one or more tasks. The second case is also favourable since it avoids an extra skill which would remain underutilized. The last two cases are unfavourable matches. The mutual affinities between projects and students are calculated according to the above guidelines and stored in the aff[p][q]. below is the pseudo-code for generating the affinity matrix. Constants a, b, c and d are user input positive constants empirically determined to suit the nature of the contexts.

**For each** concept i in Student\_skill\_Concept **do** {

**For each** concept j in Project\_skill\_Concept **do** {

Affi[i][j]:= 0

**For each** skill Sk in Skill{1..k} **do** {

**If** (student\_skill\_concept[i][k] = 0

**AND** project\_skill\_concept[j][k] = 1) **then**

Aff[i][j] := Aff[i][j] + a\*student\_skill\_concept[i][k]

**Else if** (student\_skill\_concept[i][k] = 0

**AND** project\_skill\_concept[j][k] = 0) **then**

Aff[i][j] := Aff[i][j] + b

**Else if** (student\_skill\_concept[i][k] = 0

**AND** project\_skill\_concept[j][k] = 0) **then**

Aff[i][j] := Aff[i][j] - c\*student\_skill\_concept[i][k]

**Else**

Aff[i][j] := Aff[i][j] - d }

}

}

}

this will give us the affinity matrix Aff[p][q] where (I,j) position shows how much ith students prefer the jth projects or how much jth project prefers the ith students. Example 1st row in Aff is:

| 0| 2| -4| 0| 0| 2| 2| -2| -2| 4| 2| 0| 2| -2| 2|

And 1st column in Aff is:

| 0| 0| -4| 2| 2| 0| 4| 0| 4| 2| 0| 0| 4| 4|

Whole Aff matrix is:

0| 2| -4| 0| 0| 2| 2| -2| -2| 4| 2| 0| 2| -2| 2|

0| 6| 4| -4| 4| 2| 2| -2| 6| 4| 2| 4| 2| 2| 6|

-4| 2| 0| 0| 0| -2| 2| 2| -2| 4| -2| 4| -2| 2| 2|

2| 4| -2| 2| 2| 4| 0| 4| 4| 2| 4| 2| 4| 4| 4|

2| 4| 2| 2| 2| 4| 4| 8| 4| 2| 8| 6| 4| 8| 4|

0| 6| 0| -4| 4| 2| 2| -2| 2| 8| 2| 4| 2| 2| 6|

4| 2| 0| 4| 4| 6| 6| 2| 2| 4| 6| 4| 6| 2| 6|

0| 6| 0| 0| 4| 2| 2| 6| 6| 4| 6| 4| 6| 6| 6|

4| 6| 0| 0| 4| 6| 2| 2| 6| 4| 6| 4| 6| 6| 6|

2| 8| 2| -2| 6| 4| 4| 0| 4| 6| 4| 6| 4| 4| 8|

0| 6| 0| 0| 4| 2| 6| 2| 6| 4| 6| 4| 10| 6| 6|

0| 6| 4| 0| 4| 2| 6| 6| 6| 4| 6| 8| 6| 10| 6|

4| 6| 4| 0| 8| 6| 6| 2| 6| 8| 6| 8| 6| 6| 10|

4| 6| 0| 0| 4| 6| 2| 6| 6| 4| 10| 4| 6| 6| 6|

Now we can generate the preferences for each student and projects as the values in the ith row suggests how much ith student concept prefers the jth project and values in the jth row suggests how much jth project concept prefers the ith student concept. According to the 1st row in Aff matrix the preference order for 1st student concept is:

[9, 1, 5, 6, 10, 12, 14, 0, 3, 4, 11, 7, 8, 13, 2]

Where each value j refer to the jth project concept.

Preference order for every student concept are:

0 [9, 1, 5, 6, 10, 12, 14, 0, 3, 4, 11, 7, 8, 13, 2]

1 [1, 8, 14, 2, 4, 9, 11, 5, 6, 10, 12, 13, 0, 7, 3]

2 [9, 11, 1, 6, 7, 13, 14, 2, 3, 4, 5, 8, 10, 12, 0]

3 [1, 5, 7, 8, 10, 12, 13, 14, 0, 3, 4, 9, 11, 6, 2]

4 [7, 10, 13, 11, 1, 5, 6, 8, 12, 14, 0, 2, 3, 4, 9]

5 [9, 1, 14, 4, 11, 5, 6, 8, 10, 12, 13, 0, 2, 7, 3]

6 [5, 6, 10, 12, 14, 0, 3, 4, 9, 11, 1, 7, 8, 13, 2]

7 [1, 7, 8, 10, 12, 13, 14, 4, 9, 11, 5, 6, 0, 2, 3]

8 [1, 5, 8, 10, 12, 13, 14, 0, 4, 9, 11, 6, 7, 2, 3]

9 [1, 14, 4, 9, 11, 5, 6, 8, 10, 12, 13, 0, 2, 7, 3]

10 [12, 1, 6, 8, 10, 13, 14, 4, 9, 11, 5, 7, 0, 2, 3]

11 [13, 11, 1, 6, 7, 8, 10, 12, 14, 2, 4, 9, 5, 0, 3]

12 [14, 4, 9, 11, 1, 5, 6, 8, 10, 12, 13, 0, 2, 7, 3]

13 [10, 1, 5, 7, 8, 12, 13, 14, 0, 4, 9, 11, 6, 2, 3]

Similarly acc. To 1st column in Aff matrix preference order for 1st project concept is:

[6, 8, 12, 13, 3, 4, 9, 0, 1, 5, 7, 10, 11, 2]

Where each value I refer to the ith student concept.

Preference order for every project concept are:

0 [6, 8, 12, 13, 3, 4, 9, 0, 1, 5, 7, 10, 11, 2]

1 [9, 1, 5, 7, 8, 10, 11, 12, 13, 3, 4, 0, 2, 6]

2 [1, 11, 12, 4, 9, 2, 5, 6, 7, 8, 10, 13, 3, 0]

3 [6, 3, 4, 0, 2, 7, 8, 10, 11, 12, 13, 9, 1, 5]

4 [12, 9, 1, 5, 6, 7, 8, 10, 11, 13, 3, 4, 0, 2]

5 [6, 8, 12, 13, 3, 4, 9, 0, 1, 5, 7, 10, 11, 2]

6 [6, 10, 11, 12, 4, 9, 0, 1, 2, 5, 7, 8, 13, 3]

7 [4, 7, 11, 13, 3, 2, 6, 8, 10, 12, 9, 0, 1, 5]

8 [1, 7, 8, 10, 11, 12, 13, 3, 4, 9, 5, 6, 0, 2]

9 [5, 12, 9, 0, 1, 2, 6, 7, 8, 10, 11, 13, 3, 4]

10 [13, 4, 6, 7, 8, 10, 11, 12, 3, 9, 0, 1, 5, 2]

11 [11, 12, 4, 9, 1, 2, 5, 6, 7, 8, 10, 13, 3, 0]

12 [10, 6, 7, 8, 11, 12, 13, 3, 4, 9, 0, 1, 5, 2]

13 [11, 4, 7, 8, 10, 12, 13, 3, 9, 1, 2, 5, 6, 0]

14 [12, 9, 1, 5, 6, 7, 8, 10, 11, 13, 3, 4, 0, 2]

1. **Apply extended stable marriage algorithm**

Now, we have preference order we can apply project oriented extended stable marriage algorithm. The algorithm will give us the best matching pair of concepts. The project oriented extended stable marriage algorithm is given below:

**While** P in project\_concepts and P’s list contain S not allotted to P **do** {

s := first S not allotted to H

**if** s is Assigned to some other project\_concept P’ **then**

break assignment (s, P’)

assign s to P

**for each** successor P’ of P in s’s list **do** {

remove P’ and s from each others list

}

}

For our example this will give us list pairs that is :

[(1, 9), (5, 6), (7, 4), (9, 5), (10, 13), (12, 10), (13, 11), (14, 12), (1, 1), (9, 0), (1, 7), (9, 2), (1, 8), (1, 3)]

Where for every pair (j,i) j refer to jth project concept and I refer to ith student concept

Now, every student in ith student concept can contribute to tasks of every jth project in project concept that require skills set present in that concept.

List of students that can contribute to each projects are:

project6 ( 5 ) > {'student5', 'student9', 'student3', 'student6', 'student1'}

project 1 ( 6 ) > {'student8', 'student4', 'student5', 'student10', 'student3', 'student1'}

project10 ( 8 ) > {'student8', 'student4', 'student5', 'student10', 'student7', 'student3', 'student2', 'student1'}

project 3 ( 6 ) > {'student8', 'student4', 'student5', 'student10', 'student2', 'student1'}

project 5 ( 4 ) > {'student10', 'student3', 'student5', 'student1'}

project9 ( 4 ) > {'student10', 'student3', 'student5', 'student1'}

project 2 ( 7 ) > {'student8', 'student5', 'student10', 'student7', 'student3', 'student2', 'student1'}

project8 ( 3 ) > {'student10', 'student7', 'student1'}

project7 ( 6 ) > {'student8', 'student5', 'student10', 'student3', 'student2', 'student1'}

List of projects in which each students can contribute to are:

student1 ( 9 ) > {'project 5', 'project10', 'project 2', 'project6', 'project 3', 'project 1', 'project7', 'project9', 'project8'}

student3 ( 7 ) > {'project 5', 'project10', 'project 2', 'project6', 'project 1', 'project7', 'project9'}

student5 ( 8 ) > {'project 5', 'project10', 'project 2', 'project6', 'project 3', 'project 1', 'project7', 'project9'}

student10 ( 8 ) > {'project 5', 'project10', 'project 2', 'project 3', 'project 1', 'project7', 'project9', 'project8'}

student8 ( 5 ) > {'project10', 'project 2', 'project 3', 'project 1', 'project7'}

student4 ( 3 ) > {'project 1', 'project 3', 'project10'}

student7 ( 3 ) > {'project 2', 'project8', 'project10'}

student2 ( 4 ) > {'project 2', 'project 3', 'project7', 'project10'}

student6 ( 1 ) > {'project6'}

student9 ( 1 ) > {'project6'}

More than one student with same skill set also ensure that other students can do the work if some student is unavailable for some time for some reasons.