

Chaque prompt est précédé par:

In the field of Formal Concept Analysis (FCA), explain the theoretical framework necessary to implement a concept derivation algorithm. The objective is to find all of the correct concepts. Your explanation should include:

1. What constitutes a formal context within the FCA framework, and how can it be represented in code?
2. What is a formal concept in FCA, and how is it defined mathematically?
3. What is the role of extents and intents in defining formal concepts, and how do they emerge through the Galois connection?
4. What is the process by which formal concepts are derived?
5. How should formal concepts be represented programmatically?
6. What are algorithms such as NextClosure or other enumeration strategies commonly used to compute all formal concepts from a given context?

Zero Shot

Now using what you've just explained, write Java code that reads a semicolon-separated CSV representing a formal context and computes all formal concepts (intent and extent).

One Shot

The CSV file looks like this:

```
;a1;a2  
o1;0;1  
o2;1;1
```

This means that the first row contains attribute names (columns), the first column contains object names (rows), and a cell with 1 means the object has that attribute.

For this input, the formal concepts are:

```
Extent: [o1, o2]  
Intent: [a2]  
Extent: [o2]  
Intent: [a1, a2]
```

Now using the same logic, write Java code that reads the following semicolon-separated CSV file and computes all formal concepts (intent and extent). Print each concept like:

```
Extent: [object list]  
Intent: [attribute list]
```

New input:

```
;a1;a2;a3;a4;a5  
o1;1;0;0;1;0  
o2;0;0;0;1;0  
o3;1;0;0;0;1  
o4;1;1;1;0;1  
o5;0;0;1;0;1
```

Few Shots

Example 1

Input:

.a1,a2
o1:0,1
o2:1,1

Explanation:

The first row contains attribute names, and the first column contains object names. A 1 means the object has that attribute.

Output:

Extent: [o1, o2]
Intent: [a2]
Extent: [o2]
Intent: [a1, a2]

Example 2

Input:

.a1,a2,a3
o1:1,1,0
o2:1,1,0
o3:0,1,1

Output:

Extent: [o1, o2, o3]
Intent: []
Extent: [o1, o2]
Intent: [a1]
Extent: [o3]
Intent: [a2, a3]
Extent: []
Intent: [a1, a2, a3]

Now write Java code that reads the following semicolon-separated CSV file and computes all formal concepts (intent and extent). Print each concept like:

Extent: [object list]
Intent: [attribute list]

New Input:

.a1;a2;a3;a4;a5
o1:1;0;0;1;0
o2:0;0;0;1;0
o3:1;0;0;0;1
o4:1;1;1;0;1
o5:0;0;1;0;1

Role Prompting

You are a computational theory researcher specializing in Formal Concept Analysis (FCA) and Java-based algorithm prototyping. As part of your role, your task is to write well-structured and documented Java code that reads a semicolon-separated CSV representing a formal context and computes all formal concepts (intent and extent). Use your expertise to do this:

Chain of Thought:

Prompt 1: Begin by explaining how to parse the CSV file into a formal context (objects, attributes, and their relations).

Prompt 2: Describe the data structures you would use to represent the context and to track extents and intents.

Prompt 3: Explain how you would systematically compute formal concepts using closure operators.

Prompt 4: Describe how you would avoid duplicates and infinite loops during the computation.

Prompt 5: Now using what you've just explained, write Java code that reads a semicolon-separated CSV representing a formal context and computes all formal concepts (intent and extent).

Stepback Prompting

Prompt 1: I want to compute all formal concepts (intent and extent) from a semicolon-separated CSV representing a formal context using Java.

The CSV file looks like this:

;a1;a2

o1;0;1

o2;1;1

What kind of computational task is this?

Prompt 2: What kind of input is involved?

Prompt 3: What is the goal of this kind of task?

Prompt 4: What general strategy or reasoning process applies to solving it?

Prompt 5: What common challenges should be considered when implementing this in Java (e.g., infinite loops, duplicate concepts)?

Prompt 6: Now, using what you've just explained, write Java code that reads a semicolon-separated CSV representing a formal context and computes all formal concepts (intent and extent). Now, using what you've just explained, write Java code that reads a semicolon-separated CSV representing a formal context and computes all formal concepts (intent and extent).

Self Consistency

Solve the following problem multiple times independently. For each attempt, you may use your own reasoning and implementation strategy.

Task:

Write Java code that reads a semicolon-separated CSV file representing a formal context and computes all formal concepts (intent and extent).

The CSV will look like this:

```
;a1;a2
```

```
o1;0;1
```

```
o2;1;1
```

This means that the first row contains attribute names (columns), the first column contains object names (rows), and a cell with 1 means the object has that attribute.

Each concept should be printed in this format:

Extent: [object list]

Intent: [attribute list]

Generate at least three independent implementations. Then compare them and determine which solution is the most correct and consistent with the task requirements.

Tree of Thought:

Prompt 1: You are tasked with solving the following problem: Write Java code that reads a semicolon-separated CSV file representing a formal context and computes all formal concepts (intent and extent). Print each concept like:

Extent: [object list]

Intent: [attribute list]

Prompt 2: List 3 different ways to solve this problem in Java. Each way should be a different strategy (e.g., different data structures, parsing methods, or concept generation algorithms).

Prompt 3: For each of the 3 solutions above, describe the main steps involved in implementing it. Include pros and cons if possible.

Prompt 4: Evaluate the 3 approaches. Which seems the most efficient, readable, and correct based on the problem requirements? Choose 1 or 2 to expand.

Prompt 5: Using the chosen approach, write the complete Java code that solves the problem, and make sure it matches the expected output format.