Bordat Algorithm in Pseudo Code

Esteban Marquer

2020 - 2021

```
Data:
I: incidence relation
G: objects
C_k: some concept
x: some object
Result: f_{C_k}(x) (intention of x restricted to (G\X_k))

(X_k, Y_k) ← C_k; // unpack the concept C_k

// Restrict I to (G\X_k) × Y_k
J ← I;
Remove from J all the objects (rows) of X_k;
Remove from J all the attributes (columns) not in Y_k;

Let ·'J be the derivation operator over the incidence relation J;
We can now compute x'J the intent of x in J;
return x'J
```

Algorithm 1: $f_{C_k}(x)$: intention of x restricted to $(G\backslash X_k) \times Y_k$

```
Data: S: some set, implied with no duplicates
Result: maximal elements (sets) in the set of sets

max ← S forall x in S do
    forall y in max do
        if x ⊂ y then
            Remove x from max;
        end
    end
end
return max
```

Algorithm 2: $Max(S)$: all maximal elements (sets) in the set of sets S; if S has duplicate elements (*i.e.* S is not a set), remove them before applying the algorithm

Data:
I: incidence relation
M: attributes
G: objects
$\bot = (M', M)$
$\top = (\emptyset', \emptyset)$
Result: the set $\mathcal{C}$ of concepts

$C_0 \leftarrow \bot$;
$\mathcal{C} \leftarrow [\bot]$;
$n \leftarrow 0$;
while $C_n \in \mathcal{C}$ **do** `// process all the created concepts one by one`

 $(X_n, Y_n) \leftarrow C_n$; `// unpack the previous concept` C

 `// Prepare` f_{C_n}
 Store somewhere J the restricted incidence, as it will be used many times: J is I without the objects of X_n, and only the attributes in Y_n;

 `// Compute` Y_{n+1}
 $S \leftarrow \{\}$; `// set of restricted intents of objects not in` X_n
 forall x **in** $G \backslash X_n$ **do**
  Add $f_{C_n}(x)$ to S;
 end
 `// Max(S) is "all the elements (sets) in` S `which are not subsets of other elements of` S`"`
 $candidates \leftarrow Max(S)$;

 forall $Y_{candidate} \in candidates$ **do**
  if $Y_{candidate}$ **in** $\mathcal{C}$ **then**
   C_n is subsumed by the concept in which $Y_{candidate}$ appears;
   skip this candidate;
  else
   `// Compute` $X_{candidate} = X_n \cup \{x \in (G \backslash X_n)$ `such that` $f_{C_n}(x) = Y_{candidate}\}$
   $X_{candidate} \leftarrow X_n$;
   forall x ***in*** $G \backslash X_n$ **do**
    if $f_{C_n}(x) = Y_{candidate}$ **then**
     Add x to $X_{candidate}$;
    end
   end

   $C_{candidate} \leftarrow (X_{candidate}, Y_{candidate})$;
   Add $C_{candidate}$ to $\mathcal{C}$;
   C_n is subsumed by $C_{candidate}$;
  end
 end
 $n \leftarrow n + 1$;
end
return $\mathcal{C}$ and subsumption relation

Algorithm 3: Bordat: find all the concepts and build the lattice