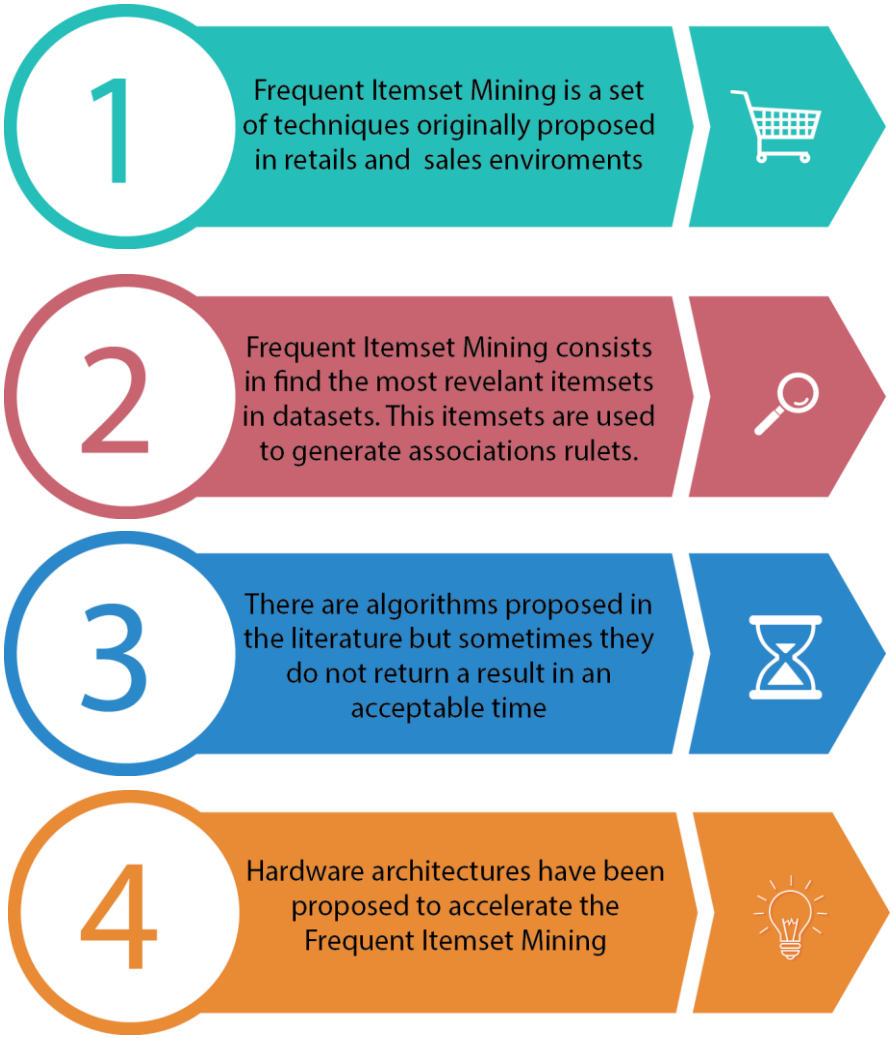
Hardware Architectures for Frequent Itemset Mining Based on Equivalence Classes Partitioning

Martin Letras
René Cumplido
Raudel Hernández-León

Computer Sciences Department

National Institute of Astrophysics, Optics, and Electronics

Motivation


1
Frequent Itemset Mining is a set of techniques originally proposed in retails and sales enviroments
2
Frequent Itemset Mining consists in find the most revelant itemsets in datasets. This itemsets are used to generate associations rulets.
3
There are algorithms proposed in the literature but sometimes they do not return a result in an acceptable time
4
Hardware architectures have been proposed to accelerate the Frequent Itemset Mining

Frequent Itemset Mining

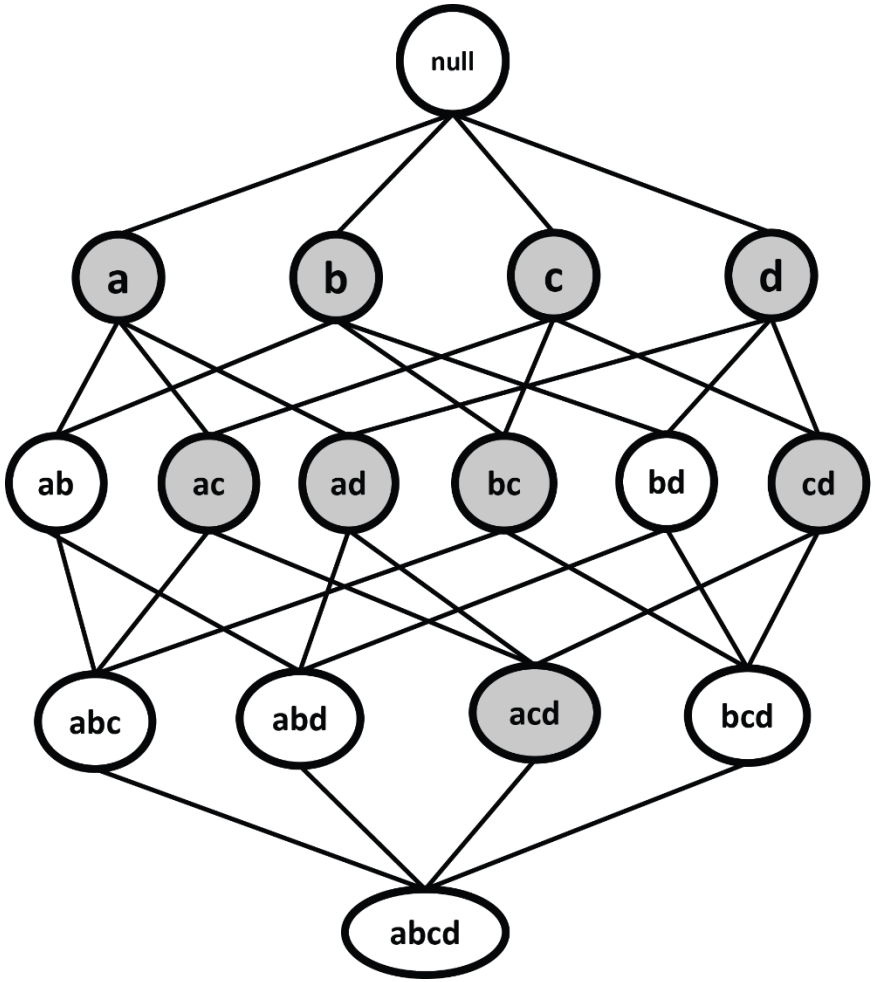
Id	Items
1	Milk, Bread
2	Butter, Bread
3	Butter
4	Milk, Butter, Bread
5	Bread
6	Milk, Butter, Bread

Transactions dataset.

Itemset	Cover	Support	Frequency
{}	1,2,3,4,5,6	6	100%
{Milk}	1,4,6	3	50%
{Butter}	2,3,4,6	4	66%
{Bread}	1,2,4,5,6	5	83%
{Milk, Butter}	4,6	2	33%
{Milk, Bread}	1,4,6	3	50%
{Butter, Bread}	2,4,6	3	50%
{Milk, Butter, Bread}	4,6	2	33%

Support calculations.

Search Space



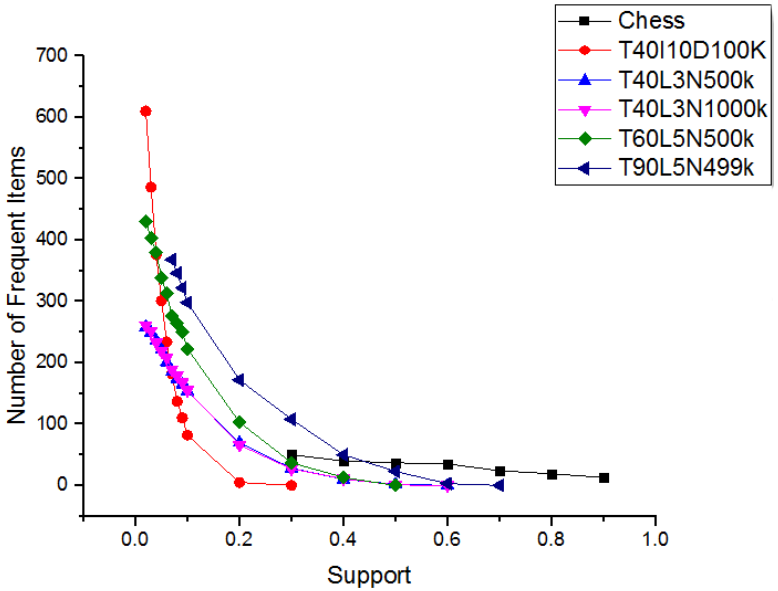
ID	Items
1	a, d
2	b, c, d
3	a, c
4	a, c, d
5	a, b
6	a, c, d
7	b, c
8	a, c, d
9	b, c
10	a, d

For example in this dataset there are only **four** items. 16 itemsets can be generated, this shows the exponential nature of this problem.

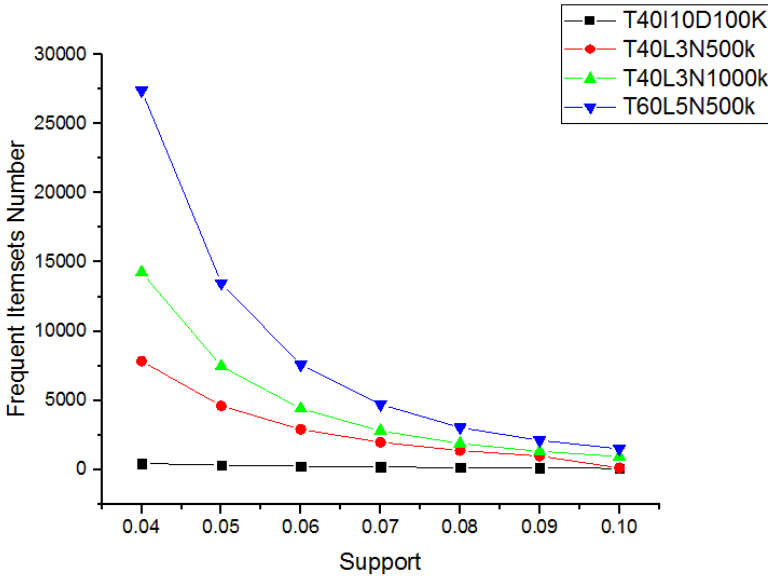
Search Space

Dataset	Binary Size(MB)	Average Length per Transaction	Number of Transactions	Number of items
Chess	0.013	20	3196	75
T40I3N500k	11.9	40	500k	299
T40I3N1000k	24.1	40	1000k	300
T60I5N500k	18.9	60	500k	500

Datasets employed in to test FIM algorithm. The number of frequents itemsets generated grow exponentially.

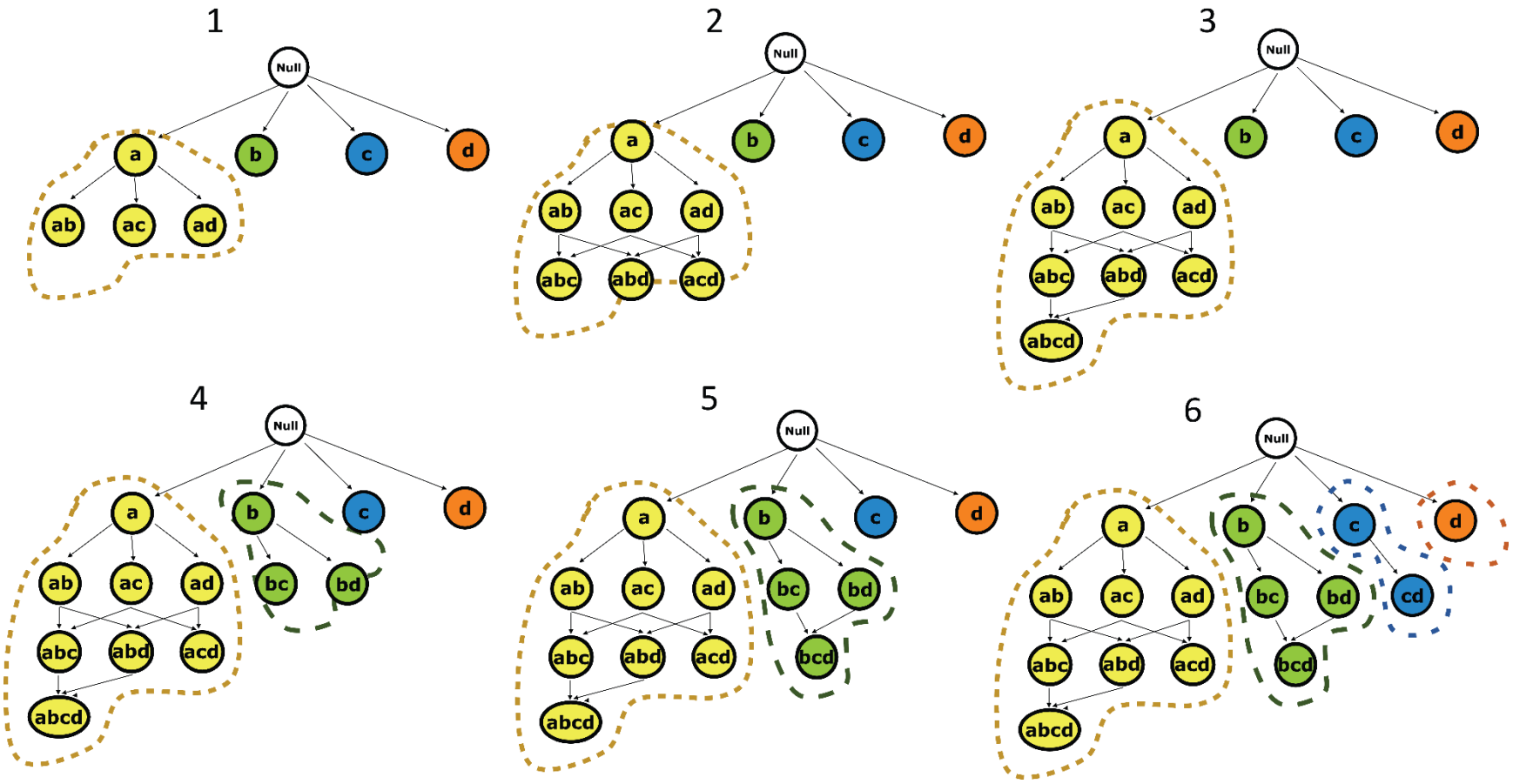


Frequent Items generated



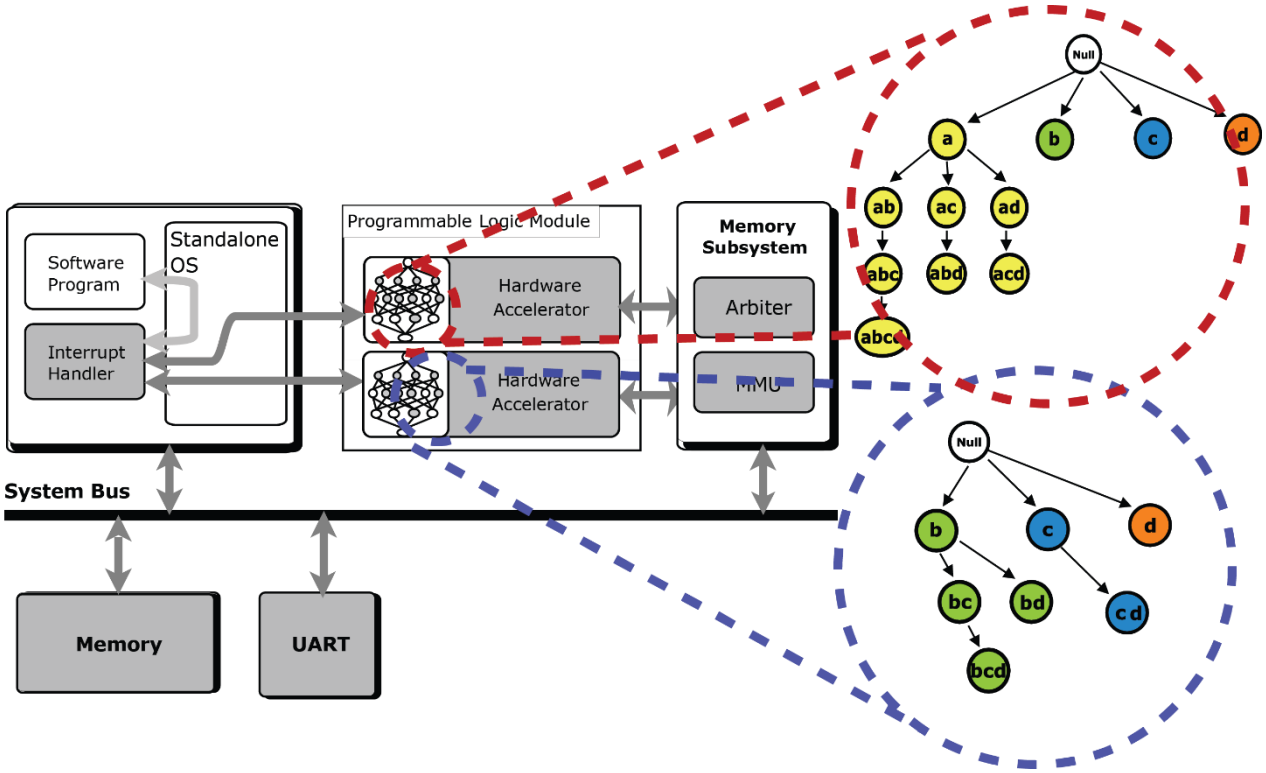
Frequent Itemsets generated

Search Strategy Proposed



Search strategy based on equivalence classes partition. Each equivalence class is processed independently each other.

Hardware Architecture



Two compact hardware architectures have been proposed to approach this problem. Figure shows the dual core architecture that divides the entire search space between two processor elements.

Thanks!, see you in poster session.