Frequent Submap Discovery

Stéphane GOSSELIN, Guillaume DAMIAND,
and Christine SOLNON

LIRIS

Université de Lyon, France

June 2011

Motivations

- Combinatorial maps are nice structures for modeling images
  - Partition of images in regions
  - Model the topology of these regions
- There exist efficient algorithms for:
  - Deciding of submap isomorphism
  - Searching for a map in a database of maps
- But comparing and classifying maps are challenging issues

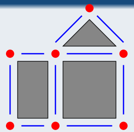
Contribution

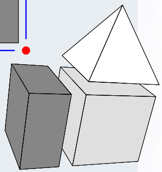
Algorithm for extracting frequent patterns from maps
$\rightsquigarrow$ May be used to compare and classify maps

Plan

Very short (and incomplete) history of combinatorial maps...

- Introduced in 2D / planar graphs
  [Edmonds60, Tutte63, Jacques70, Cori73]
- Extended in 3D
  [ArquesKoch88, Lienhardt88, Spehner91]
- Generalized in nD
  [Lienhardt89]





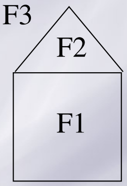
Basic idea

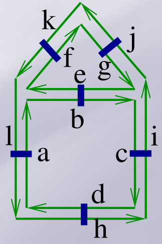
- Subdivision of an nD space into cells
  $\rightsquigarrow$ Vertices (0D), Edges (1D), Faces (2D), Volumes (3D), ...
- Model the topology of these cells
  $\rightsquigarrow$ Adjacency and incidence relationships

This talk introduces ideas and algorithms for 2D maps
$\rightsquigarrow$ Refer to the paper for an extension to nD

From 2D objects to 2D maps

- 2D object $\rightsquigarrow$ faces, edges and vertices
- Each edge is decomposed in 2 adjacent darts
- Darts are related by 2 functions: let x be a dart,
  - $\beta_1(x) \rightsquigarrow$ dart which follows x when turning around the face
  - $\beta_2(x) \rightsquigarrow$ dart adjacent to x (in the face adj. to the face of x)





darts	a	b	c	d	e	f	g	h	...
β_1	b	c	d	a	f	g	e	i	...
β_2	l	e	i	h	b	k	j	d	...

Definition of 2D maps

A 2D map is defined by $M = (D, \beta_1, \beta_2)$ such that

1. D is a finite set of darts;
2. β_1 is a permutation on D;
3. β_2 is an involution on D.

Previous work used in this talk

- Algorithm for deciding of submap Isomorphism [CVIU 2011]
  - Input: a connected pattern map M and a target map M'
  - Output: decide if there exists a copy of M in M'
  - Time complexity: $\mathcal{O}(|D| \cdot |D'|)$
- Map signatures [TCS 2011]
  - Construction of the signature of a map M in $\mathcal{O}(|D|^2)$
  - Search for a map M given a set of k signatures in $\mathcal{O}(|D|^2)$

Plan

Definition of our frequent pattern extraction problem

- Input: a database S of maps and a real $0 < \sigma \leq 1$
- Output: set of all frequent patterns
  - Pattern $\rightsquigarrow$ Set of connected faces
  - Frequent $\rightsquigarrow$ Occurs in at least $\sigma \cdot |S|$ maps of S

Bad news

- A map may contain an exponential number of patterns
  $\rightsquigarrow$ Finding all frequent patterns is NP-hard

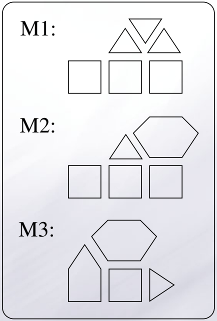
Good news

- Submap isomorphism is in P
- Anti-monotonicity: $M \notin$ Freq. $\Rightarrow$ Supermaps of $M \notin$ Freq.

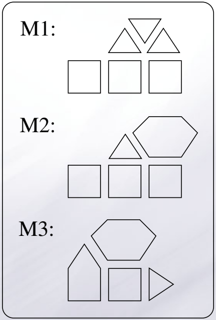
$\rightsquigarrow$ Extraction in incremental polynomial time

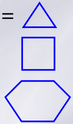
Algorithm 1 mSpan(S: a base of maps, $\sigma \in]0;1]$)

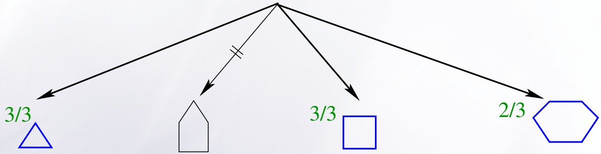
```
F_1 ← all frequent patterns composed of 1 face
F ← F_1
while F_1 ≠ ∅ do
    Choose a pattern f in F_1
    Cand ← {f}
    while Cand ≠ ∅ do
        Choose a pattern p in Cand
        F_p ← grow(p, F_1)
        Cand ← Cand ∪ F_p
        F ← F ∪ F_p
    /* All frequent patterns which contain f are in F */
    remove f from F_1
return F
```



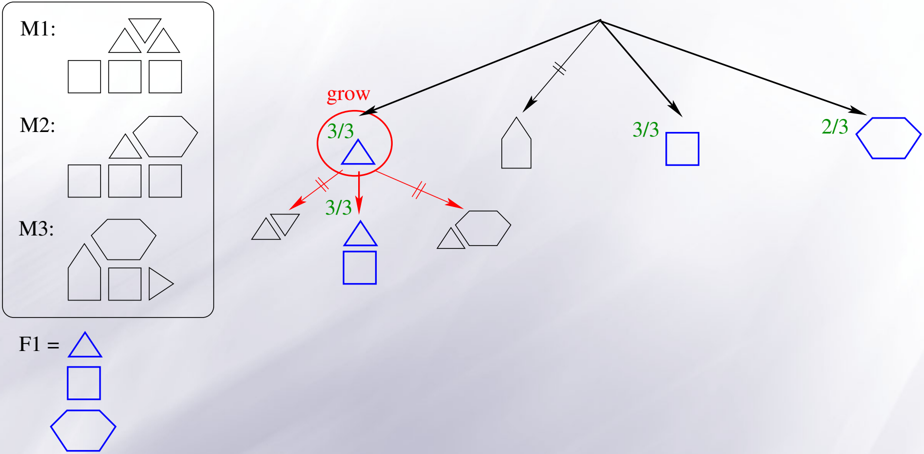
Frequency sigma = 2/3

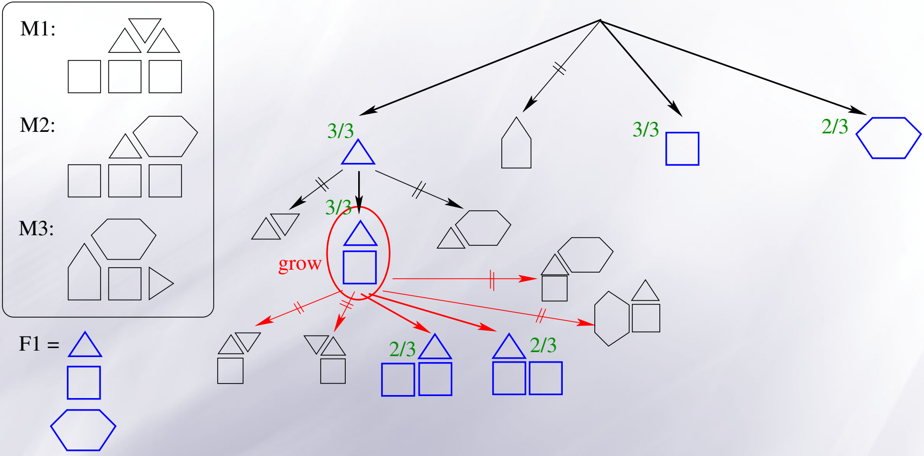


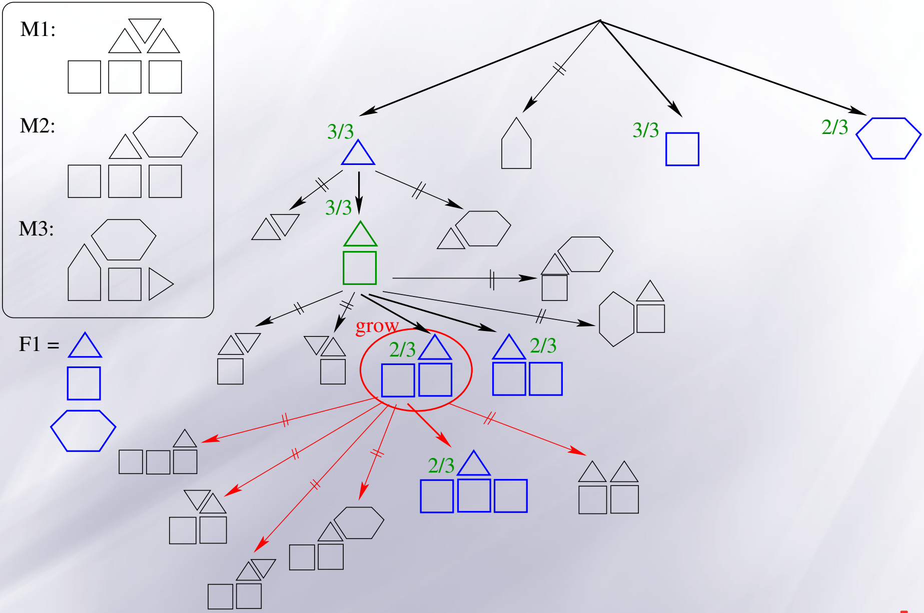
F1 = 

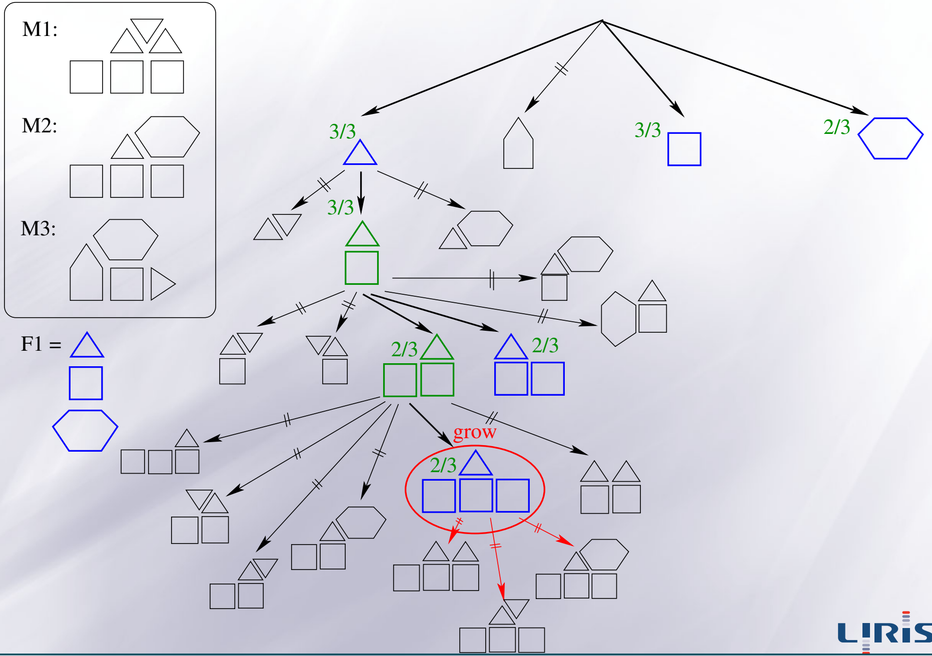


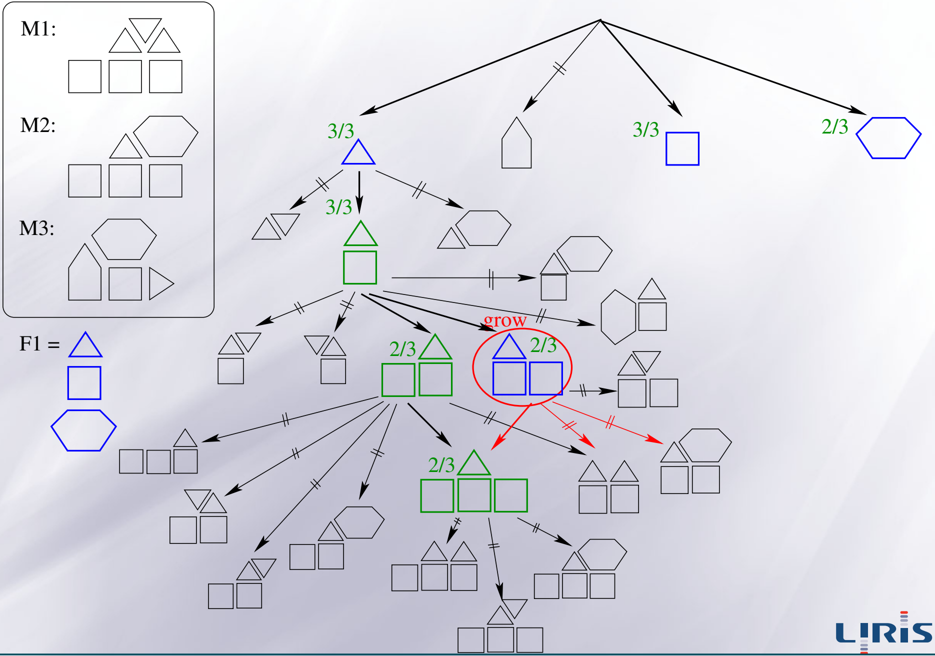
Frequency sigma = 2/3

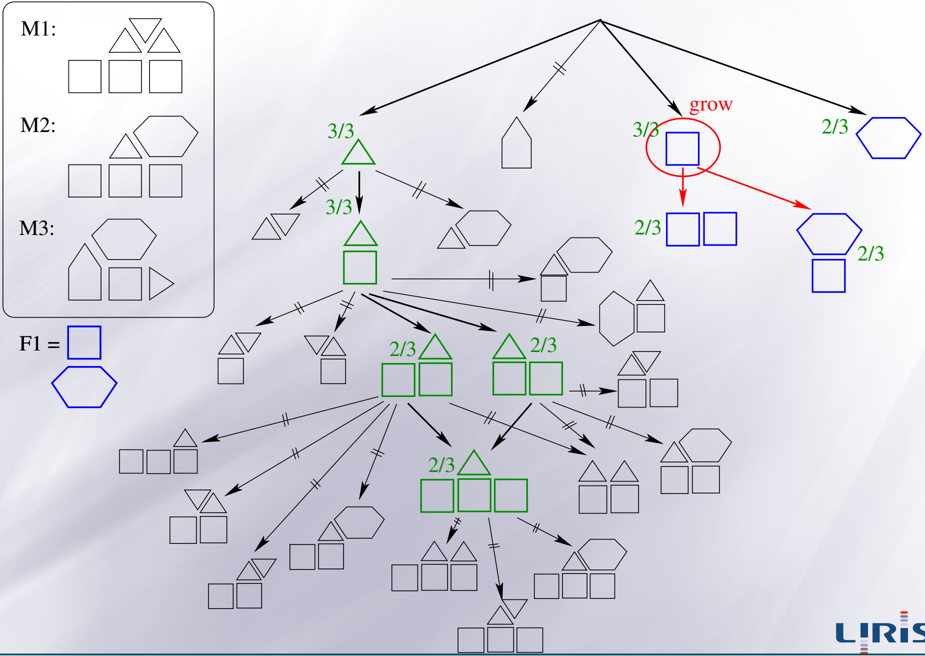

M1:
M2:
M3:
F1 =
grow
3/3
3/3
3/3
2/3

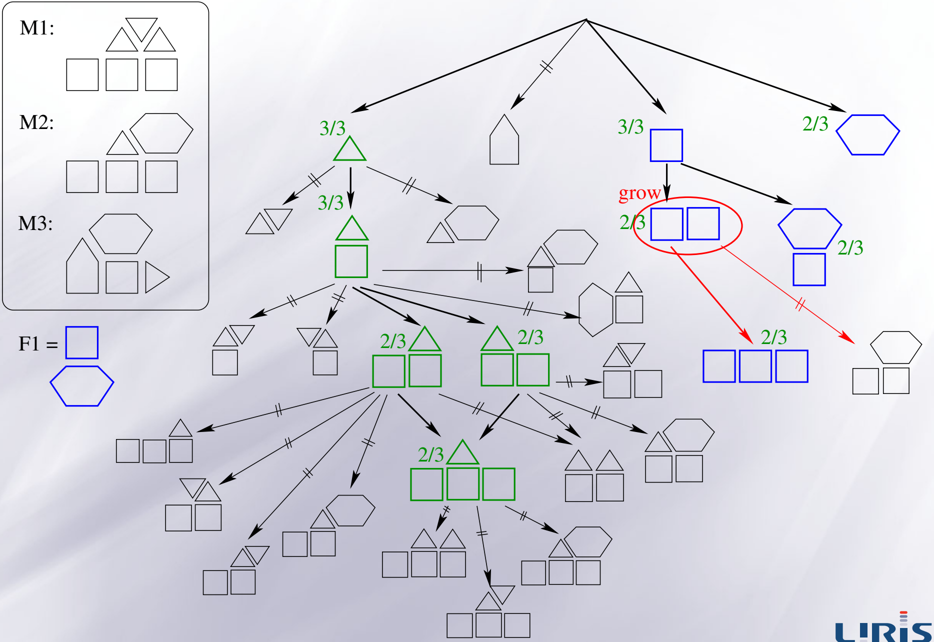

M1:
M2:
M3:
F1 =
3/3
3/3
2/3
3/3
grow
2/3
2/3


M1:
M2:
M3:
F1 =
3/3
3/3
2/3
3/3
grow
2/3
2/3
2/3

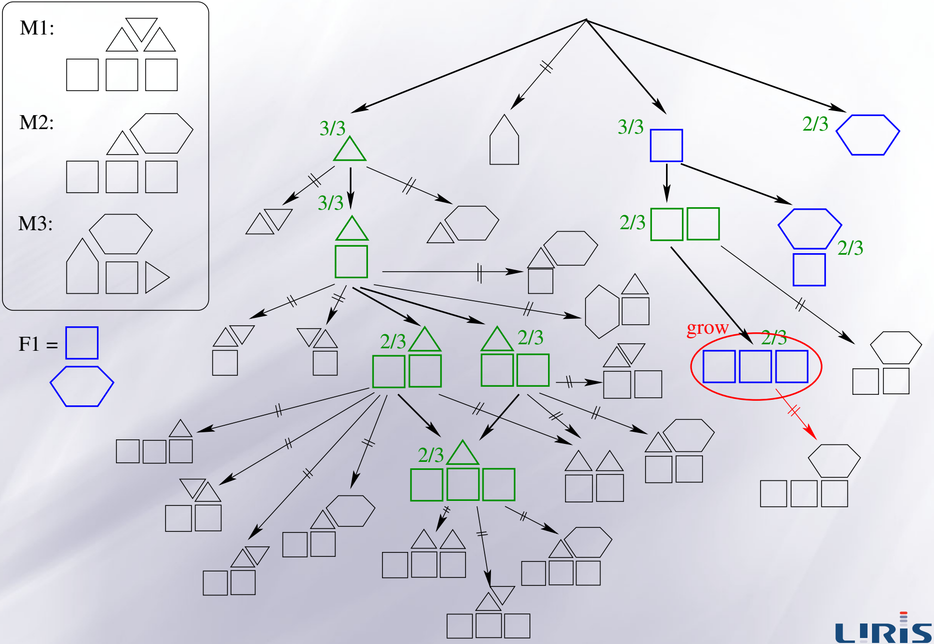

M1:
M2:
M3:
F1 =
3/3
3/3
2/3
3/3
2/3
2/3
grow
2/3

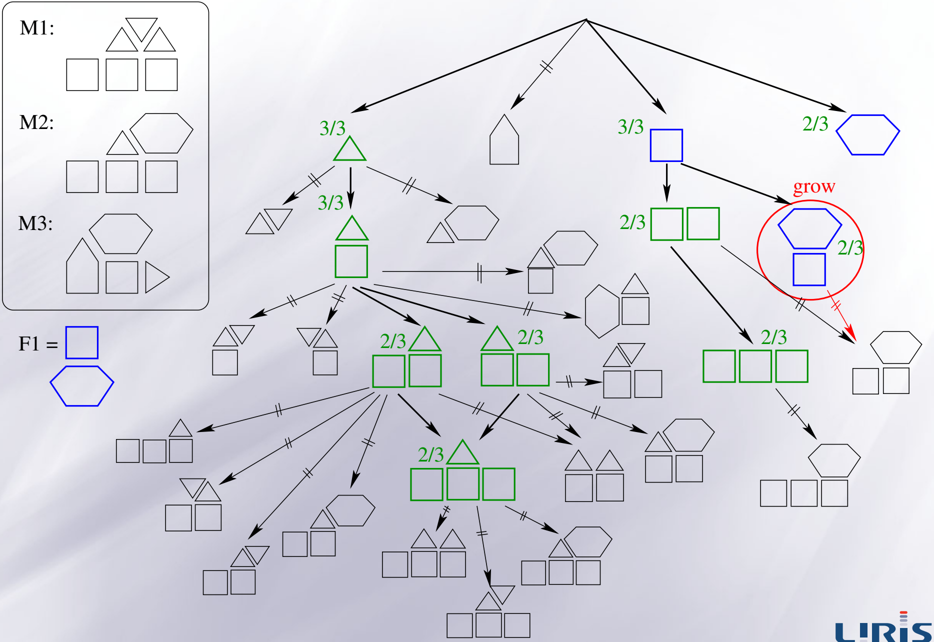

M1:
M2:
M3:
F1 =
3/3
3/3
2/3
3/3
2/3
grow
2/3
2/3

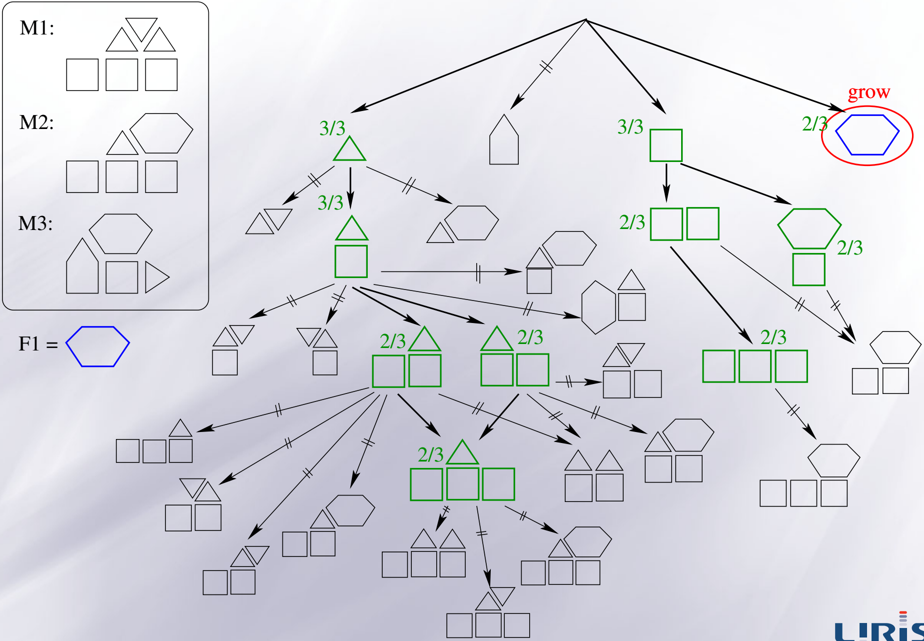

M1:
M2:
M3:
F1 =
3/3
grow
3/3
2/3
3/3
2/3
2/3
2/3
2/3
2/3
2/3

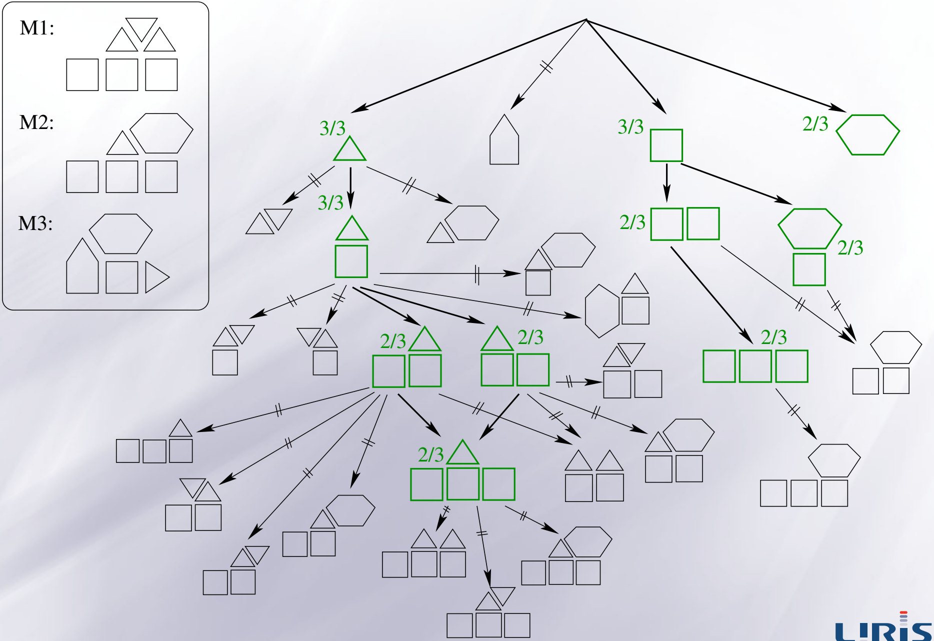


M1:
M2:
M3:
F1 =
3/3
3/3
2/3
3/3
2/3
2/3
2/3
2/3
2/3
2/3
grow


M1:
M2:
M3:
F1 =
3/3
3/3
2/3
3/3
2/3
2/3
2/3
2/3
2/3
grow
2/3
2/3


M1:
M2:
M3:
F1 =
3/3
3/3
3/3
2/3
2/3
grow
2/3
2/3
2/3
2/3
2/3
2/3


M1:
M2:
M3:
F1 =
grow
3/3
3/3
3/3
2/3
2/3
2/3
2/3
2/3
2/3
2/3


M1:
M2:
M3:
3/3
3/3
2/3
3/3
2/3
2/3
2/3
2/3
2/3
2/3
2/3

Plan

Scale-up properties on synthetic databases

Comparison with gSpan [Yan, Han 2002]

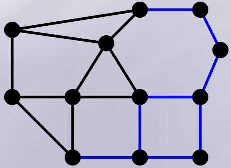
- State-of-the-art for frequent subgraph extraction
- mSpan and gSpan solve different problems
- Extract frequent connected subgraphs from dual graphs

Scale-up properties on synthetic databases

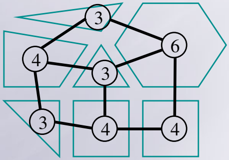
Comparison with gSpan [Yan, Han 2002]

- State-of-the-art for frequent subgraph extraction
- mSpan and gSpan solve different problems
- Extract frequent connected subgraphs from dual graphs
  - Connected subgraph $\rightsquigarrow$ Connected set of faces
  - But face topology is ignored

Map



Primal graph

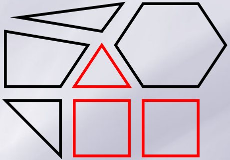


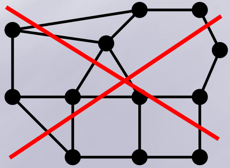
Dual graph

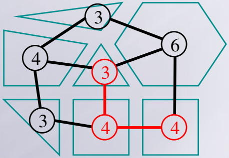
Scale-up properties on synthetic databases

Comparison with gSpan [Yan, Han 2002]

- State-of-the-art for frequent subgraph extraction
- mSpan and gSpan solve different problems
- Extract frequent connected subgraphs from dual graphs
  - Connected subgraph $\rightsquigarrow$ Connected set of faces
  - But face topology is ignored


Map


Primal graph

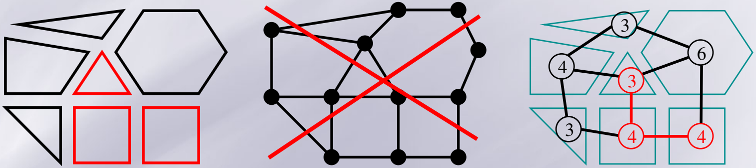


Dual labeled graph

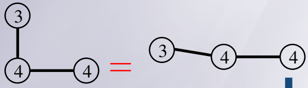
Scale-up properties on synthetic databases

Comparison with gSpan [Yan, Han 2002]

- State-of-the-art for frequent subgraph extraction
- mSpan and gSpan solve different problems
- Extract frequent connected subgraphs from dual graphs
  - Connected subgraph $\rightsquigarrow$ Connected set of faces
  - But face topology is ignored



Map | Primal graph | Dual labeled graph

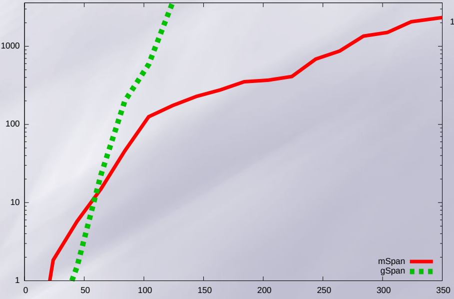


Scale-up properties w.r.t. map sizes

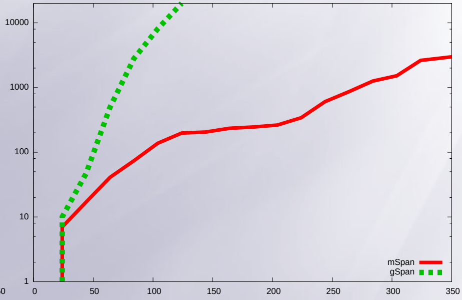
Datasets

- Each dataset = 1000 randomly generated connected maps
- Increase the number of faces from 4 to 350
- Set σ to 0.9

Time wrt map size



Nb of frequent patterns wrt map size

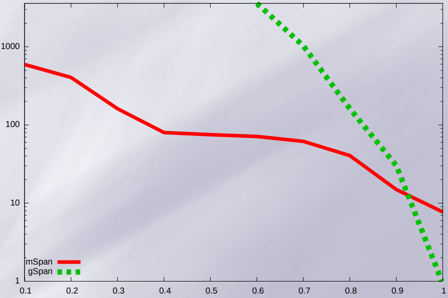


Scale-up properties w.r.t. σ

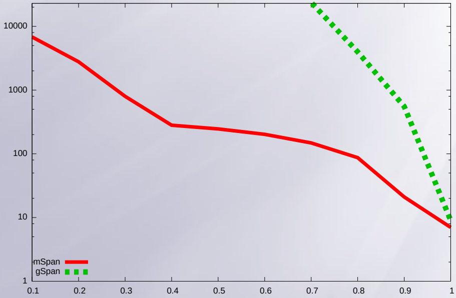
Datasets

- Each dataset = 1000 randomly generated connected maps
- Set the number of faces to 60
- increase σ from 0.1 to 1

Time wrt σ



Number of frequent patterns wrt σ



Example of application: Image classification

Modelling images by means of frequent patterns

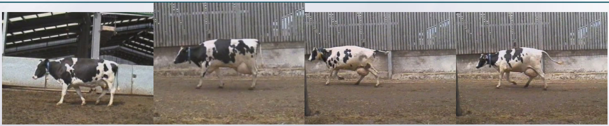
- Segment images and associate image regions to map faces
- Extract frequent patterns
- Model each image by a vector V:
  $\rightsquigarrow V[i]$ number of occurrences of the ith pattern

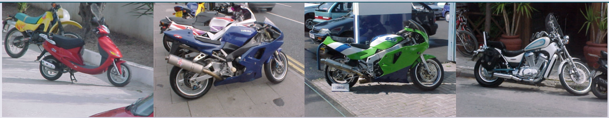
Use classification/clustering tools defined on vector spaces

Ex.: classification of the Pascal dataset with decision trees (C4.5)

Car

100 images

Cow

111 images

Moto

115 images

Actual class \ Classed to	Car	Cow	Moto
Car	86	5	9
Cow	9	94	8
Moto	21	13	81

Plan

Contributions

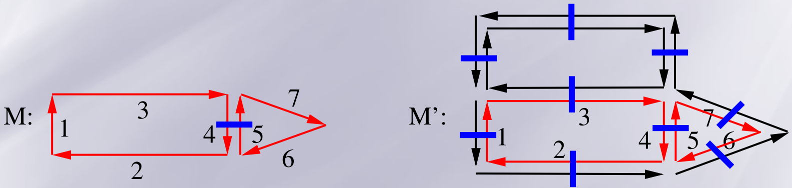
- Algorithm for frequent submap extraction
  - Incremental polynomial-time
- Experimental results on synthetic databases
  - Topology allows mSpan to extract less patterns than gSpan
- First experimental results on images
  - Frequent submaps $\rightsquigarrow$ model images by numerical vectors
  - Numerous analysis tools defined on vector spaces

Further work

- On the algorithmic side:
  - Design incremental signatures
- On the applicative side:
  - Image: Combine frequent submaps with other image features
  - Other applications: Aperiodic tillings, 3D and 3D+t objects, ...

Submap isomorphism [CVIU 2011]

- Input: a connected pattern map M and a target map M'
- Output: decide if there exists a copy of M in M'
- Basic principle:
  - choose a dart of M
  - for each dart of M':
    - Perform a traversal of M/M' and match corresponding darts
    - Answer yes if the matching preserves β_i functions
  - Answer No
- Time complexity: $\mathcal{O}(|D| \cdot |D'|)$



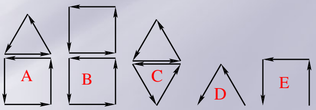
Map signatures [TCS 2011]

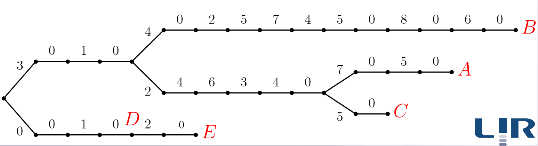
Construction of the signature of a database of maps

- Input: a set S of k maps of at most t darts
- Output: lexicographic tree of the signatures of the maps of S
- Complexity: time in $\mathcal{O}(k \cdot t^2)$ and space in $\mathcal{O}(k \cdot t)$

Search for a map in a database of maps

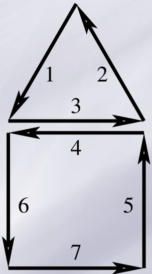
- Input:
  - lex. tree T of the signatures of the maps of a database
  - a new map M
- Output: decide if there is a map in T isomorphic to M
- Time complexity: $\mathcal{O}(t^2)$ where t = number of darts of M



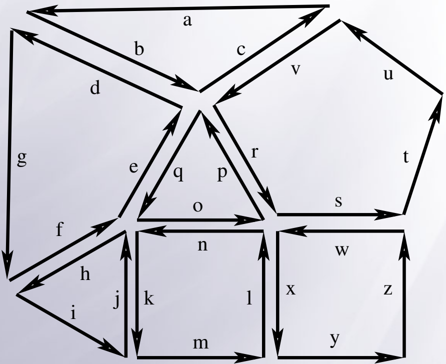


Storing all occurrences of a pattern

Store the dart associated with dart 1 for each occurrence
$\rightsquigarrow \mathcal{O}(k)$ where $k = \#$ occurrences



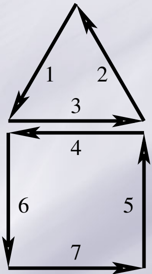
Pattern P



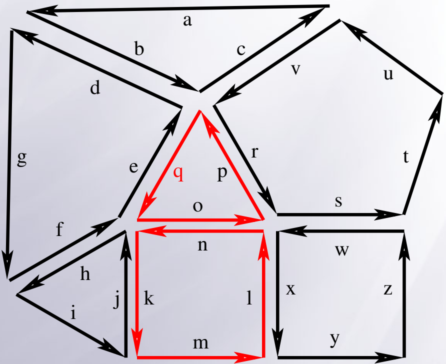
Occurrences of P =

Storing all occurrences of a pattern

Store the dart associated with dart 1 for each occurrence
$\rightsquigarrow \mathcal{O}(k)$ where $k = \#$ occurrences



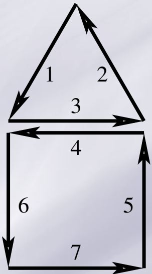
Pattern P



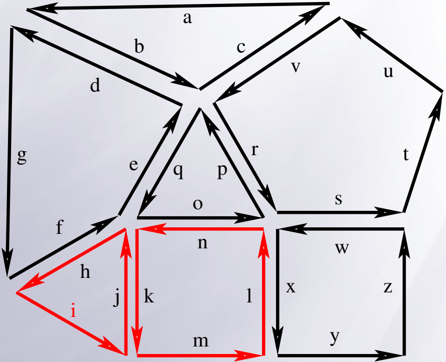
Occurrences of P = q

Storing all occurrences of a pattern

Store the dart associated with dart 1 for each occurrence
$\rightsquigarrow \mathcal{O}(k)$ where $k = \#$ occurrences



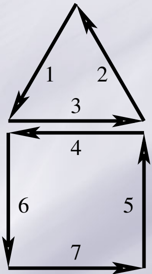
Pattern P



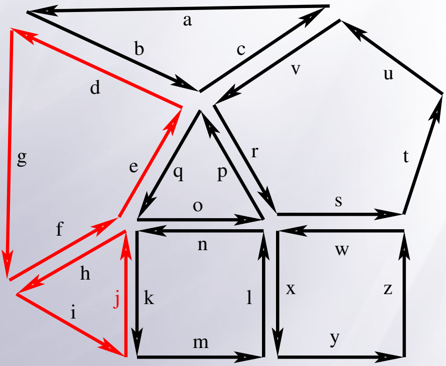
Occurrences of P = q,i

Storing all occurrences of a pattern

Store the dart associated with dart 1 for each occurrence
$\rightsquigarrow \mathcal{O}(k)$ where $k = \#$ occurrences



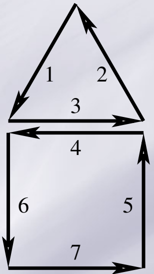
Pattern P



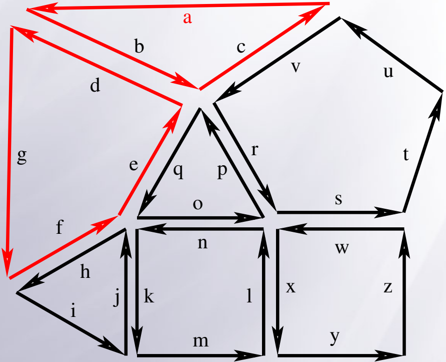
Occurrences of P = q,i,j

Storing all occurrences of a pattern

Store the dart associated with dart 1 for each occurrence
$\rightsquigarrow \mathcal{O}(k)$ where $k = \#$ occurrences



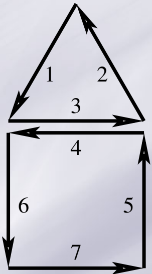
Pattern P



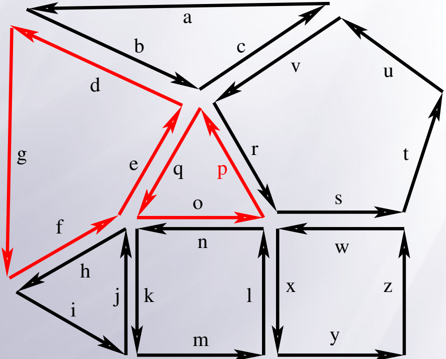
Occurrences of P = q,i,j,a

Storing all occurrences of a pattern

Store the dart associated with dart 1 for each occurrence
$\rightsquigarrow \mathcal{O}(k)$ where $k = \#$ occurrences



Pattern P



Occurrences of P = q,i,j,a,p