Discovering habits with periodic patterns

Alexandre Termier

Université de Rennes 1

Equipe Inria/IRISA Lacodam

M2 SIF – DMV course

Motivation

- Pattern mining : finding regularities in data

- « Habits »
  - Regularity in the actions performed
  - Temporal regularity between occurrences

- Different problem for pattern miner
  - WHAT is repeated => HOW is it repeated

Periodicity

- Pattern P is reapeated (as usual) => has occurrences
- Some temporal property between occurrences
  - Sequencing
  - Timestamps

- Periodicity (naïve version) : constant inter-occurrence delay
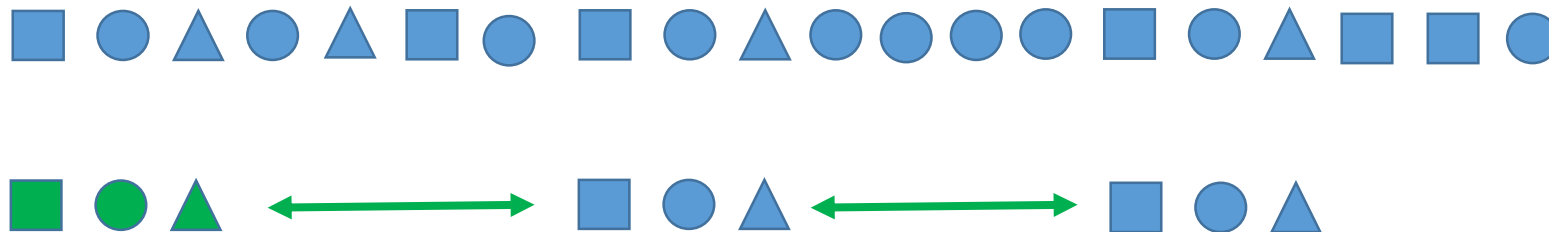
Periodicity in DMV

- Today: exhaustive approach based on *periodic concepts*

- Later, if time allows: pattern set approach based on MDL

- End of course: Sequence segmentation approach
  - Usually during case study, by Peggy

How periodic is your set-top box?
Analyzing the execution of a video decoder

Patricia López Cueva, Aurélie Bertaux, Alexandre Termier, Jean-François Méhaut, Miguel Santana: *Debugging embedded multimedia application traces through periodic pattern mining*. EMSOFT 2012: 13-22

Particia López Cueva, *Debugging Embedded Multimedia Application Execution Traces through Periodic Pattern Mining*, PhD, 2013.

Slides adapted from Patricia Lopez Cueva

Context

- Data : execution traces of set-top boxes
  - System level info : interrupts, context switches,…
  - Applicative info : start/end of (some) high level functions
  - Application : video decoding

- Problem :
  - Understand complex periodic behavior of video decoding software
  - Determine when the periodicity is broken

Data

Execution trace =
Sequence of
timestamped events

Cut into windows

Transform into
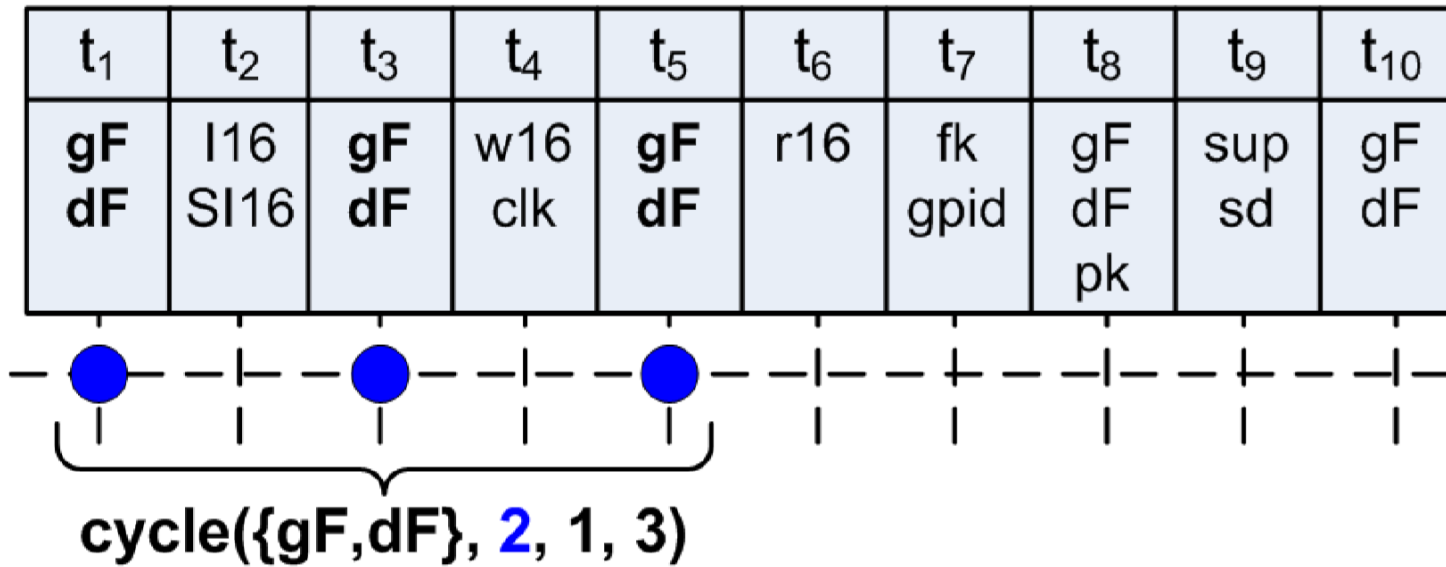sequence of itemsets
(window -> itemset)

Execution Trace ($s.\mu s$)

68.770630	getFrame
68.770697	displayFrame
68.770741	int16
68.770768	swint16
68.770869	displayFrame
68.770913	getFrame
68.770959	write16
68.770982	cpu_clock
68.771032	getFrame
68.771099	displayFrame
68.771150	read16
68.771235	fork
68.771324	get_pid
68.771346	getFrame
68.771372	displayFrame
68.771402	printk
68.771456	sem_up
68.771487	sem_down
68.771540	getFrame
68.771586	displayFrame

0.1 ms

Preprocessing

Transactional Database

t_1	getFrame, displayFrame
t_2	int16, swint16
t_3	displayFrame, getFrame
t_4	write16, cpu_clock
t_5	getFrame, displayFrame
t_6	read16
t_7	fork, get_pid
t_8	getFrame, displayFrame, printk
t_9	sem_up, sem_down
t_{10}	getFrame, displayFrame

Pattern building block : the **cycle**

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF **dF**	I16 SI16	**gF** **dF**	w16 clk	**gF** **dF**	r16	fk gpid	gF dF pk	sup sd	gF dF

cycle({gF,dF}, **2**, 1, 3)

Repeated events   Period   Start offset   #repetition

Periodic pattern
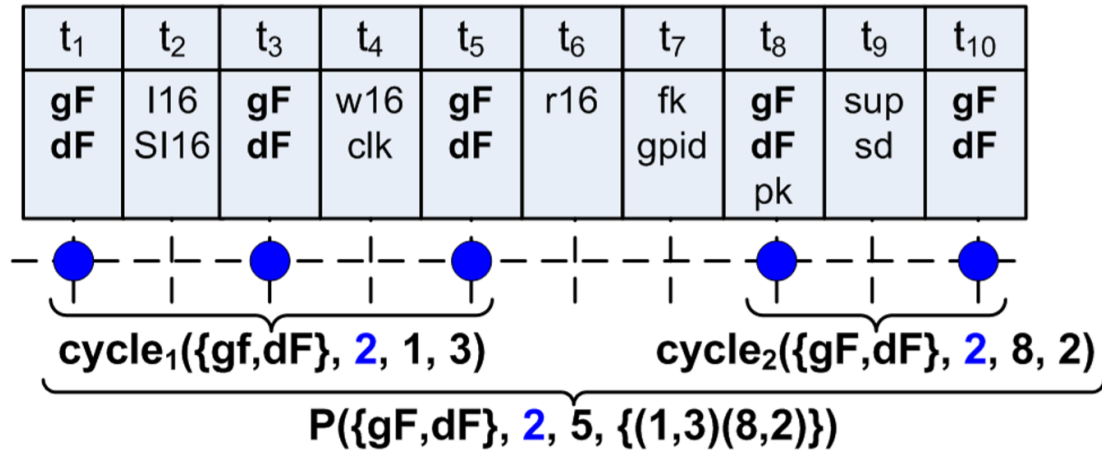
Many redundancies

Support

Sum of all *cycles* lengths:
$cycles = \{(o_1, l_1), ..., (o_k, l_k)\}$

$$support = \sum_{i=1}^{k} l_i$$

Frequent Periodic Pattern
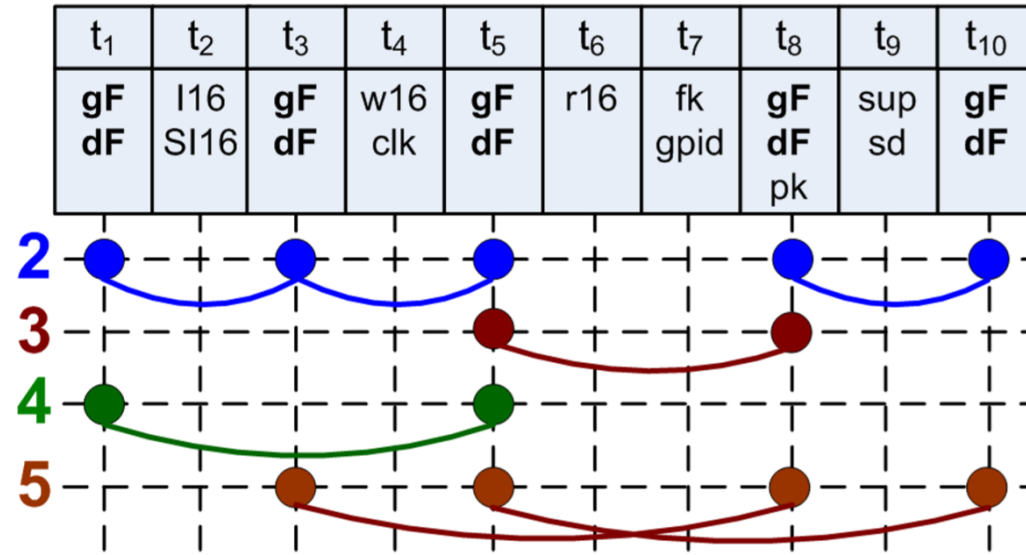
Given a minimum support threshold (min_sup), a pattern is frequent if

$$support \geq min\_sup$$

Towards a condensed representation

- Too many redundant patterns -> condensed representation
  - Closed periodic patterns ?

- Pb : cannot compute classic closure with (Itemset, Period, Transactions)

- Solution : move from diadic to triadic !

Triadic representation

	t₁	t₂	t₃	t₄	t₅	t₆	t₇	t₈	t₉	t₁₀
	gF **dF**	I16 SI16	**gF** **dF**	w16 clk	**gF** **dF**	r16	fk gpid	**gF** **dF** pk	sup sd	**gF** **dF**

Itemsets	Periods	2										3									
	Transactions	t₁	t₂	t₃	t₄	t₅	t₆	t₇	t₈	t₉	t₁₀	t₁	t₂	t₃	t₄	t₅	t₆	t₇	t₈	t₉	t₁₀
gF		X		X		X			X		X					X			X		
dF		X		X		X			X		X					X			X		
...																					

Itemsets	Periods	4										5									
	Transactions	t₁	t₂	t₃	t₄	t₅	t₆	t₇	t₈	t₉	t₁₀	t₁	t₂	t₃	t₄	t₅	t₆	t₇	t₈	t₉	t₁₀
gF		X				X								X		X			X		X
dF		X				X								X		X			X		X
...																					11

Itemsets	Periods	2										3									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X		X		X			X		X					X			X		
dF		X		X		X			X		X					X			X		
...																					

Itemsets	Periods	4										5									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X				X								X		X			X		X
dF		X				X								X		X			X		X
...																					

Triples
$(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5\})$

Itemsets	Periods	2										3									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X		X		X			X		X					X			X		
dF		X		X		X			X		X					X			X		
...																					

Itemsets	Periods	4										5									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X				X								X		X			X		X
dF		X				X								X		X			X		X
...																					

Periodic Concepts
$T_1(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5, t_8, t_{10}\})$

13

Itemsets	Periods	2										3									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X		X		X			X		X					X			X		
dF		X		X		X			X		X					X			X		
...																					

Itemsets	Periods	4										5									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X				X								X		X			X		X
dF		X				X								X		X			X		X
...																					

Periodic Concepts
$T_1(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5, t_8, t_{10}\})$
$T_2(\{gF, dF\}, \{2, 4\}, \{t_1, t_5\})$

14

Itemsets	Periods	2										3									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X		X		X			X		X					X			X		
dF		X		X		X			X		X					X			X		
...																					

Itemsets	Periods	4										5									
	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF		X				X								X		X			X		X
dF		X				X								X		X			X		X
...																					

Periodic Concepts

$T_1(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5, t_8, t_{10}\})$
$T_2(\{gF, dF\}, \{2, 4\}, \{t_1, t_5\})$
$T_3(\{gF, dF\}, \{2, 5\}, \{t_3, t_5, t_8, t_{10}\})$

Itemsets	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
Periods						2										3					
gF		X		X		X			X	X						X			X		
dF		X		X		X			X	X						X			X		
...																					

Itemsets	Transactions	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
Periods						4										5					
gF		X				X								X		X			X		X
dF		X				X								X		X			X		X
...																					

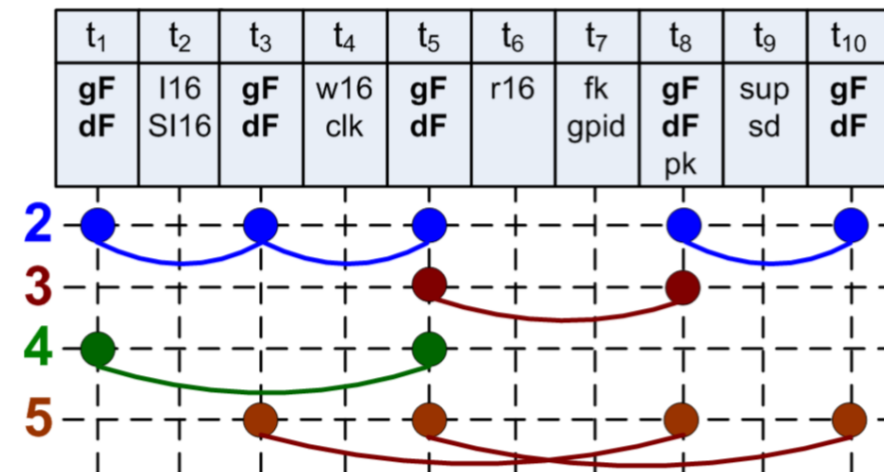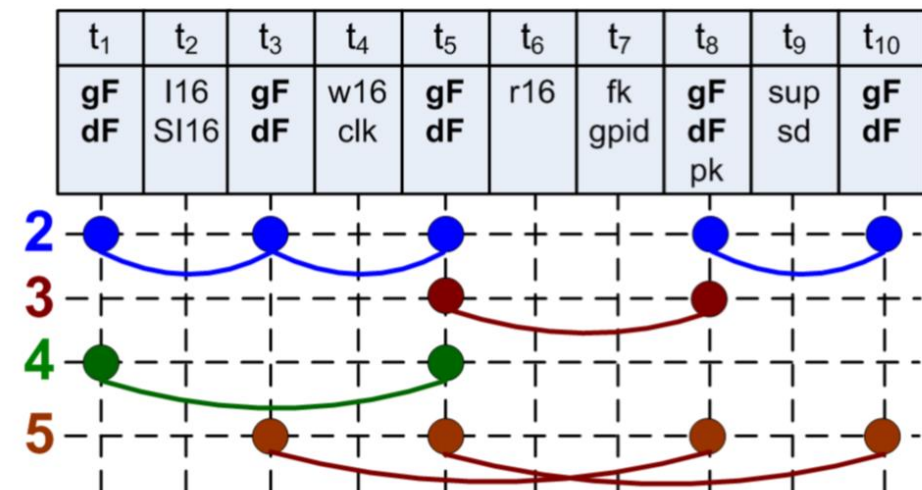| Periodic Concepts |
|---|
| $T_1(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5, t_8, t_{10}\})$ |
| $T_2(\{gF, dF\}, \{2, 4\}, \{t_1, t_5\})$ |
| $T_3(\{gF, dF\}, \{2, 5\}, \{t_3, t_5, t_8, t_{10}\})$ |
| $T_4(\{gF, dF\}, \{2, 3, 5\}, \{t_5, t_8\})$ |

16

Core Periodic Concept [EMSoft 2012]

Core Periodic Concept

A periodic concept (I, P, T) is a **core periodic concept** if there does not exist any other periodic concept (I', P', T') such that $I = I'$, $P' \subset P$ and $T' \supset T$.

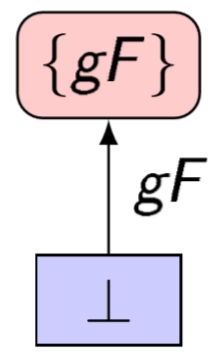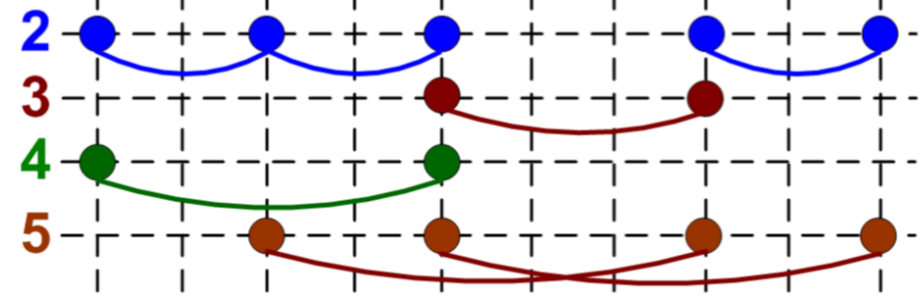| Periodic Concepts |
|---|
| $T_1(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5, t_8, t_{10}\})$ |
| $T_2(\{gF, dF\}, \{2, 4\}, \{t_1, t_5\})$ |
| $T_3(\{gF, dF\}, \{2, 5\}, \{t_3, t_5, t_8, t_{10}\})$ |
| $T_4(\{gF, dF\}, \{2, 3, 5\}, \{t_5, t_8\})$ |

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF dF	I16 SI16	gF dF	w16 clk	gF dF	r16	fk gpid	gF dF pk	sup sd	gF dF

Core Periodic Concept [EMSoft 2012]

Core Periodic Concept

A periodic concept (I, P, T) is a **core periodic concept** if there does not exist any other periodic concept (I', P', T') such that $I = I'$, $P' \subset P$ and $T' \supset T$.

	Core Periodic Concepts
$T_1(\{gF, dF\}, \{2\}, \{t_1, t_3, t_5, t_8, t_{10}\})$	
$T_2(\{gF, dF\}, \{2, 4\}, \{t_1, t_5\})$	
$T_3(\{gF, dF\}, \{2, 5\}, \{t_3, t_5, t_8, t_{10}\})$	
$T_4(\{gF, dF\}, \{2, 3, 5\}, \{t_5, t_8\})$	

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF dF	I16 SI16	gF dF	w16 clk	gF dF	r16	fk gpid	gF dF pk	sup sd	gF dF

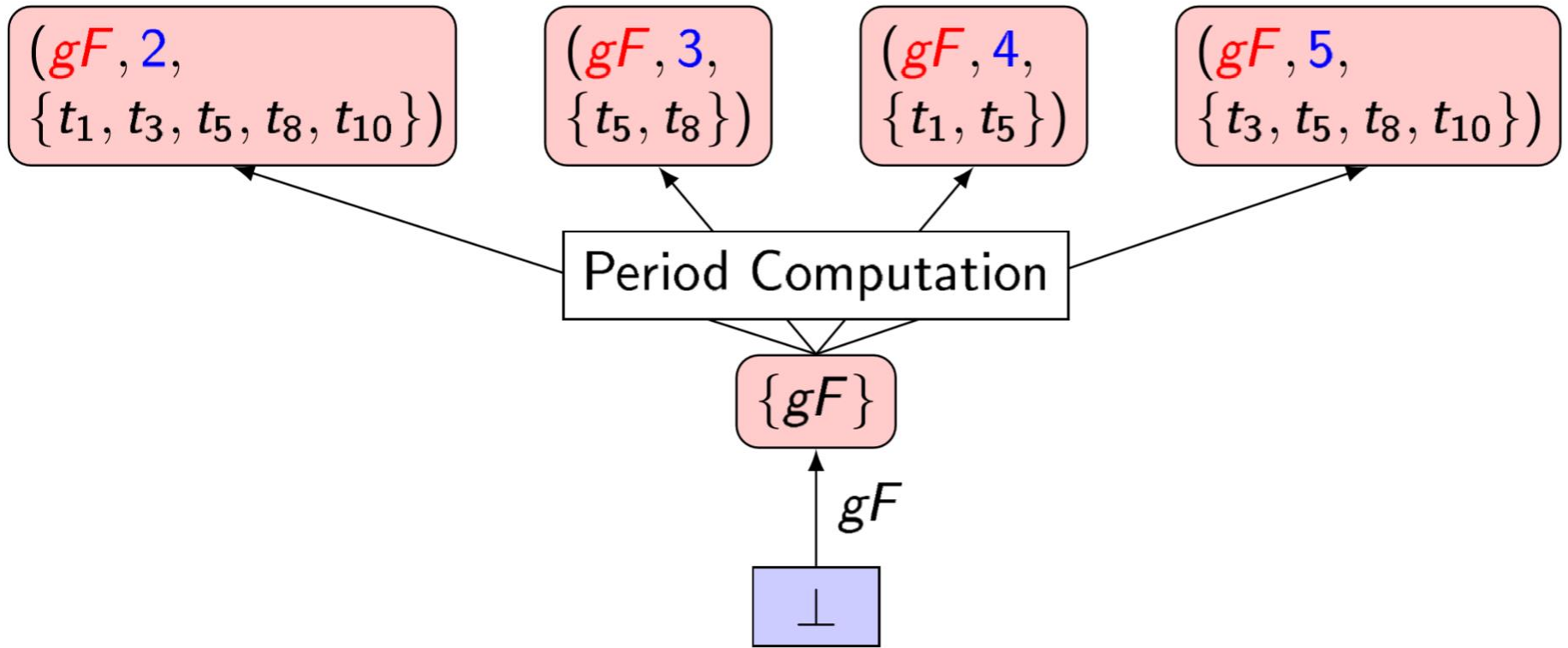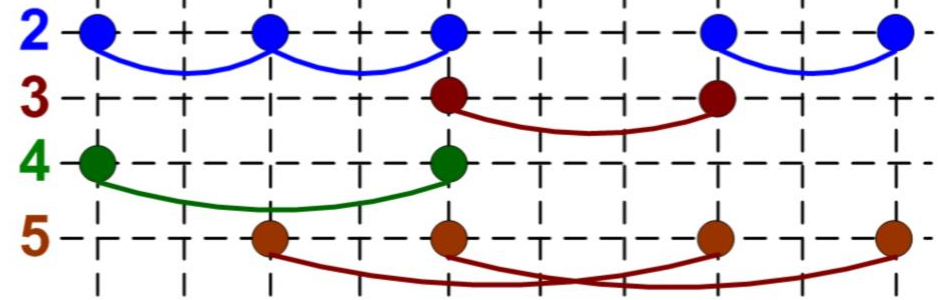CPC = condensed representation of all periodic concepts
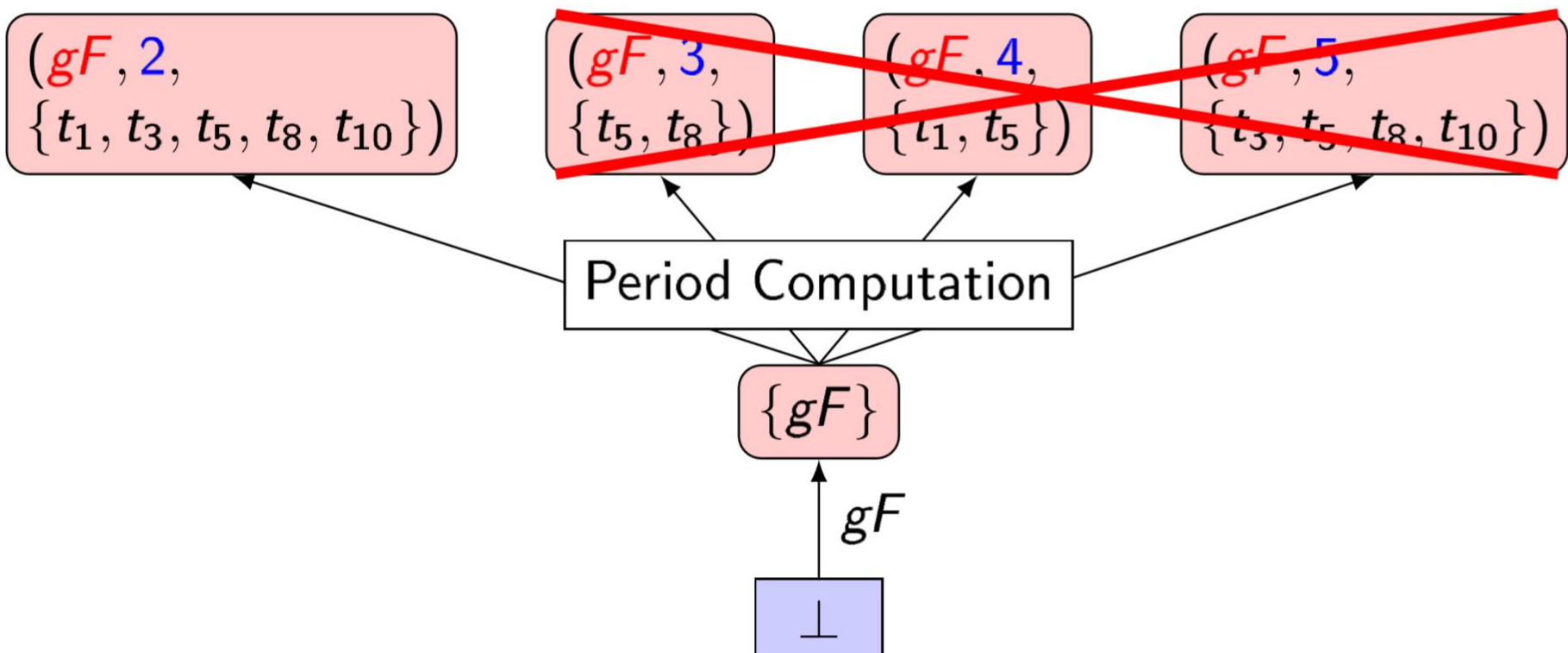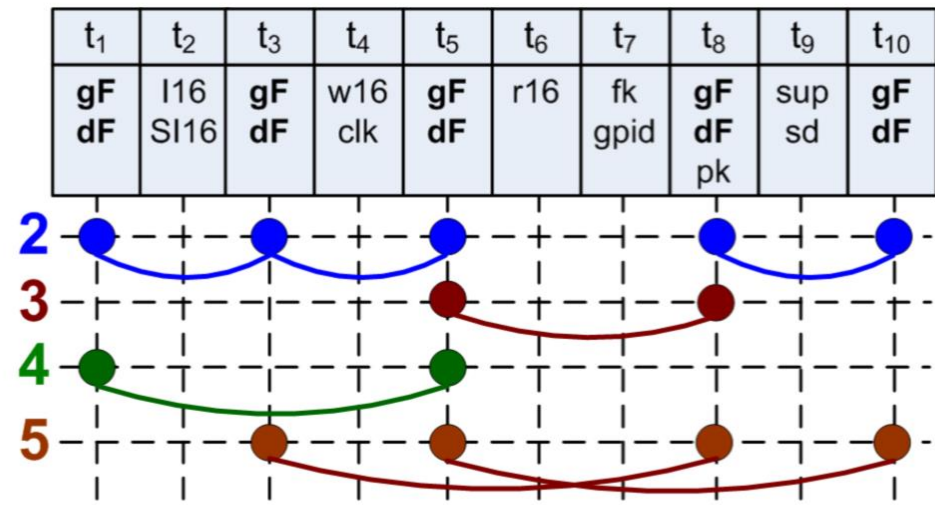
Mining Core Periodic Concepts

- Solution 1: [EMSoft 2012]
  - Use DataPeeler (Cerf et al., 2009) to get triadic patterns
  - Postprocess to filter CPC

- Solution 2: [López Cueva PhD, 2013]
  - Direct mining of CPC
  - Based on LCM/CbO enumeration strategy
  - Proven poly-delay time, poly space

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
gF **dF**	I16 SI16	**gF** **dF**	w16 clk	**gF** **dF**	r16	fk gpid	**gF** **dF** pk	sup sd	**gF** **dF**

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
	gF	l16	**gF**	w16	**gF**	r16	fk	**gF**	sup	**gF**
	dF	SI16	**dF**	clk	**dF**		gpid	**dF**	sd	**dF**
								pk		
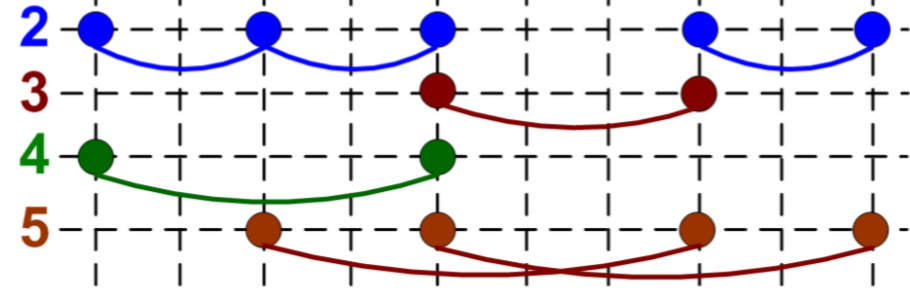
$(gF, 2, \{t_1, t_3, t_5, t_8, t_{10}\})$

$(gF, 3, \{t_5, t_8\})$

$(gF, 4, \{t_1, t_5\})$

$(gF, 5, \{t_3, t_5, t_8, t_{10}\})$

Period Computation

$\{gF\}$

gF

$\perp$

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
	gF	l16	**gF**	w16	**gF**	r16	fk	**gF**	sup	**gF**
	dF	SI16	**dF**	clk	**dF**		gpid	**dF**	sd	**dF**
								pk		

$(gF, 2,$
$\{t_1, t_3, t_5, t_8, t_{10}\})$

$(gF, 3,$
$\{t_5, t_8\})$

$(gF, 4,$
$\{t_1, t_5\})$

$(gF, 5,$
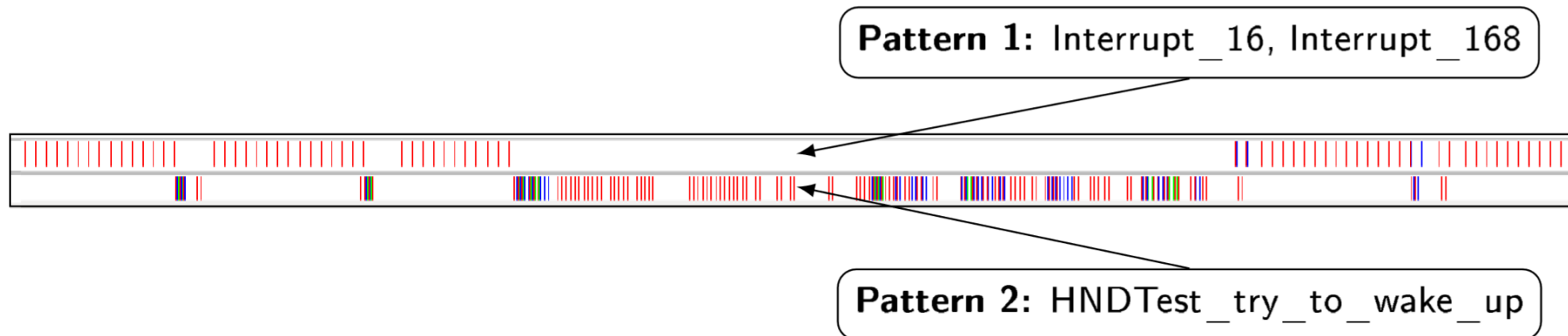$\{t_3, t_5, t_8, t_{10}\})$

Period Computation

$\{gF\}$

gF

$\bot$

22

Application on real execution trace

Discovered conflict between the application and the system (USB port)

- Interrupt_16: processor clock interrupt.
- Interrupt_168: USB interrupt.
- HNDTest_try_to_wake_up: system call (try_to_wake_up).

Pattern 1: Interrupt_16, Interrupt_168

Pattern 2: HNDTest_try_to_wake_up

25

Conclusion

- Three approaches for mining temporal regularities presented
  - Quite strict cycles, gaps allowed between cycles, transaction data, condensed representation
  - Tolerant + nested cycles, sequence data, MDL
  - Segmentation, transaction data, optimisation/Pareto/MDL

- Many other interesting problems await

- Surprisingly few people in that research area (since 1999)

Perspectives

- Robustness, robustness, robustness
    - Most periodic pattern definitions break to easily
    - -> prevent the discovery of more general/covering patterns

- Take into account domain knowledge

- Provide easy to use implementations

Thank you for your attention!