Automata Processing
Applications

Ke Wang[1] (Research Scientist of CAP)
Matt Tanner[2]
Kevin Skadron[1] (Center Director of CAP)
Mircea Stan[1] (Assoc. Director of CAP)
[1] University of Virginia
[2] Micron Technology Inc.

THE CENTER FOR AUTOMATA PROCESSING

AUTOMATA PROCESSING

Application I: Brill Tagging Micron Automata Processor

Keira Zhou, Jeffrey J. Fox, Ke Wang, Donald E. Brown, Kevin Skadron

University of Virginia
Center for Automata Processing

UNIVERSITY of VIRGINIA

Micron
Focused on Memory | Engineered for Innovation

Motivation

- Semantic analysis often uses a pipeline of Natural Language Processing (NLP) tools, one common piece of which is part-of-speech (POS) tagging
- Provide speed-up for certain tasks within NLP
  - Brill tagging
  - Rule-based NLP tasks
- Combine new architecture and traditional CPU to accelerate current implementation

Background: Brill Tagging

- A two-stage tagging technique[3]
  - Stage 1: Baseline tagging
  - Stage 2: Update tags based on some rules
- 218 context-based rules trained from training corpus publicly available
- Maximum span: 3 words ahead or 3 words after

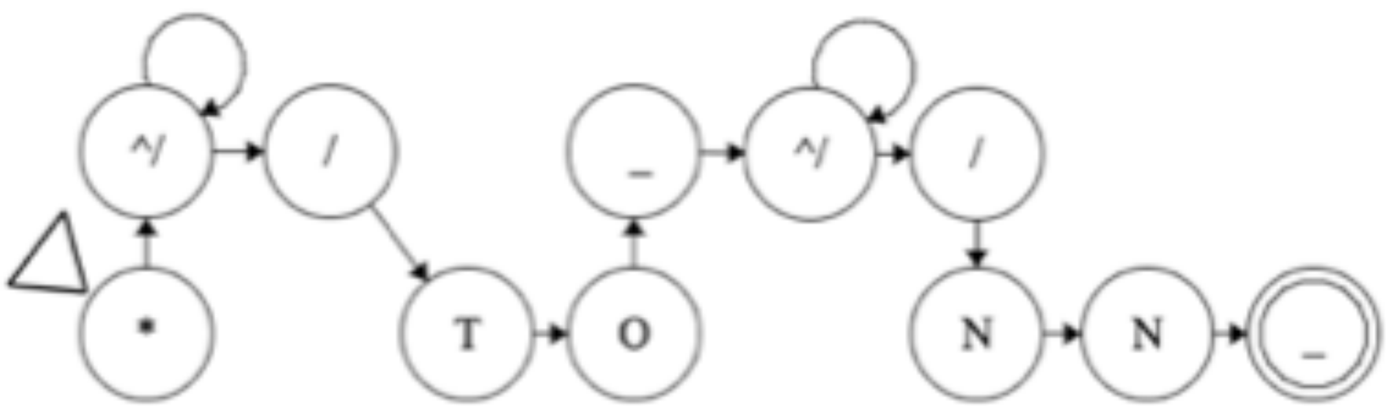
[3] Brill, Eric. "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging." Computational linguistics 21.4 (1995): 543-565.

Approach: The Implementation

- Update tags based on some rules (AP)
  - NN VB PREVTAG TO:

If "**..WORD1/TO WORD2/NN..**", then update into "**..WORD1/TO WORD2/VB..**"

Input: *… to/TO conflict/**NN** with/IN …*



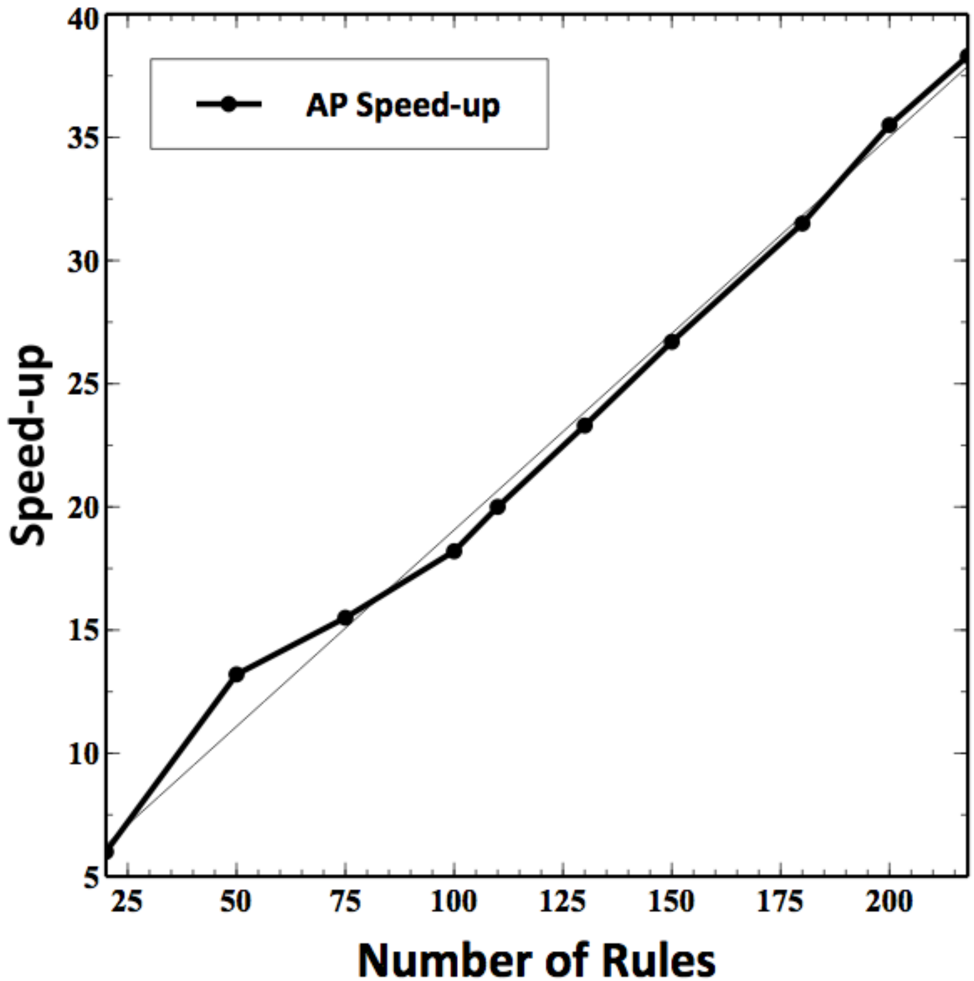
Results

- Results for the implementation
  - Comparing against C version developed by Brill [5]
  - Tested on a file size 99KB (File size will NOT impact the speed-up)

(time in microsecs)	20 rules	75 rules	130 rules	180 rules	218 rules
CPU time	13687	48167	81187	113435	141810
AP time	2288	3104	3481	3601	3707
Speed-up	6.0X	15.5X	23.3X	31.5X	38.3X

[5] Brill, Eric. Brill's code: http://www.tech.plym.ac.uk/soc/staff/guidbugm/software/RULE_BASED_TAGGER_V.1.14.tar.Z

Results (Cont'd)



- Linear Speed-up with No. of Rules
  - Processing all rules in parallel
  - Complexity: nKR for CPU vs. n for AP
    - n: Input size
    K: window-span
    R: No. of Rules
    - The speed-up is independent of the size of the corpus
- Known ruleset size: 1729
  - Projected speed-up: 276X

Application II:
String Kernel

Chunkun Bo, Ke Wang, Yanjun Qi, Kevin Skadron

University of Virginia
Center for Automata Processing

Motivations

- String Kernel (SK), a widely used kernel in machine learning and text mining
- SK testing phase is computationally expensive
- ***Feature vector mapping*** is the current performance bottleneck, which involves a lot of pattern matching
- Micron's Automata Processor (AP) can match complex regular expressions in massive parallelism

We use the AP to accelerate String Kernel Testing

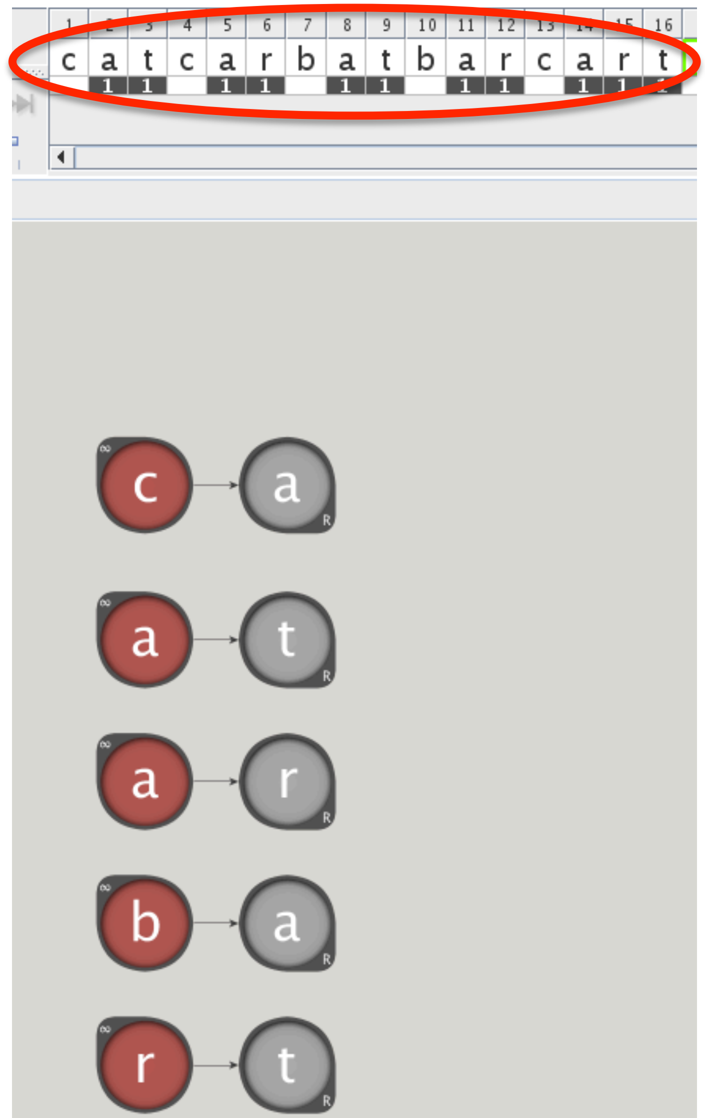
Design in AP

- Exact Match Kernel (K = 2)
- Input: cat, car, bat, bar, cart
- Kernel Function Results

k(bat, car) = 0

k(cat, car) = 1

	ca	at	ar	ba	rt
cat	1	1	0	0	0
car	1	0	1	0	0
bat	0	1	0	1	0
bar	0	0	1	1	0
cart	1	0	1	0	1



Design in AP

- Mismatch kernel

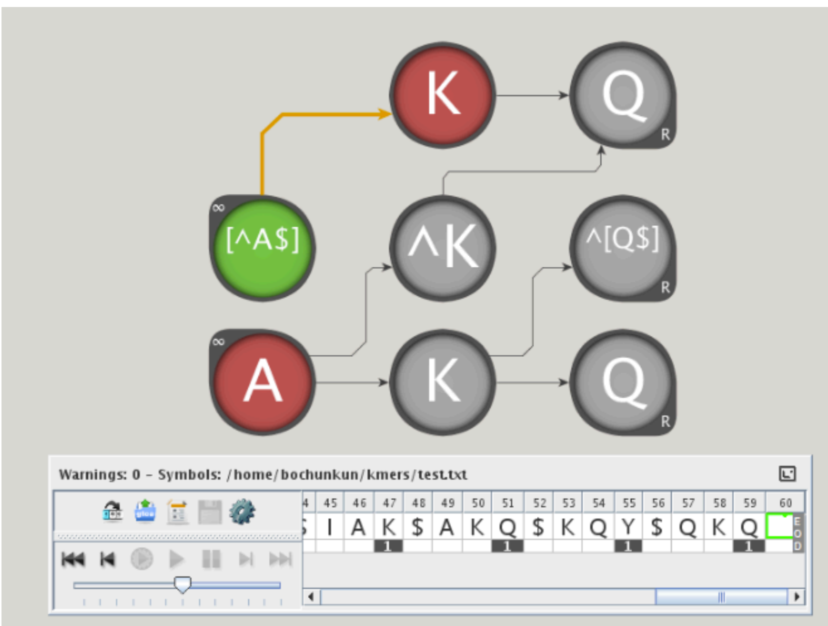
K=3

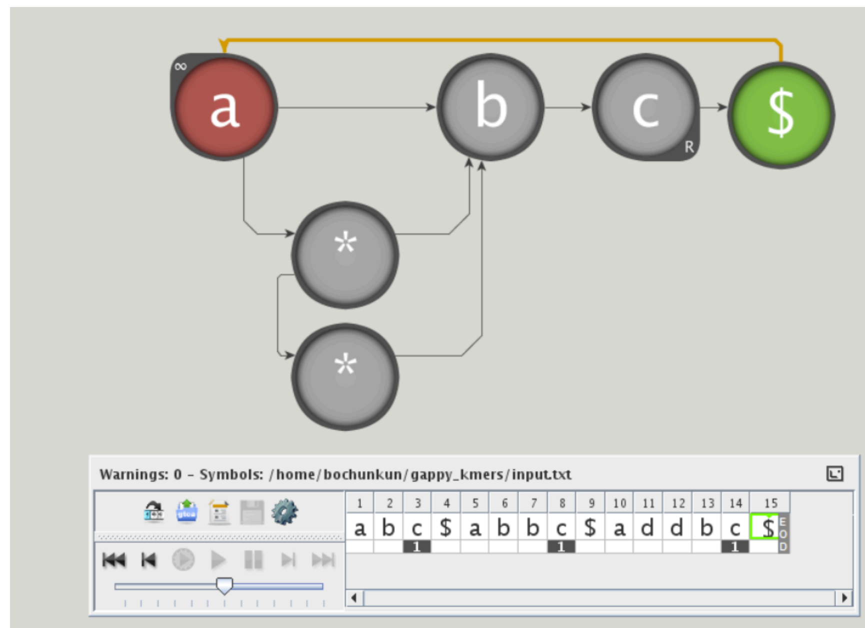
Hamming distance =0, 1

- Gappy kernel

K=3,

gaps <= 2





Design in AP

- Spatial Kernel

t=2, k=1, d < 5

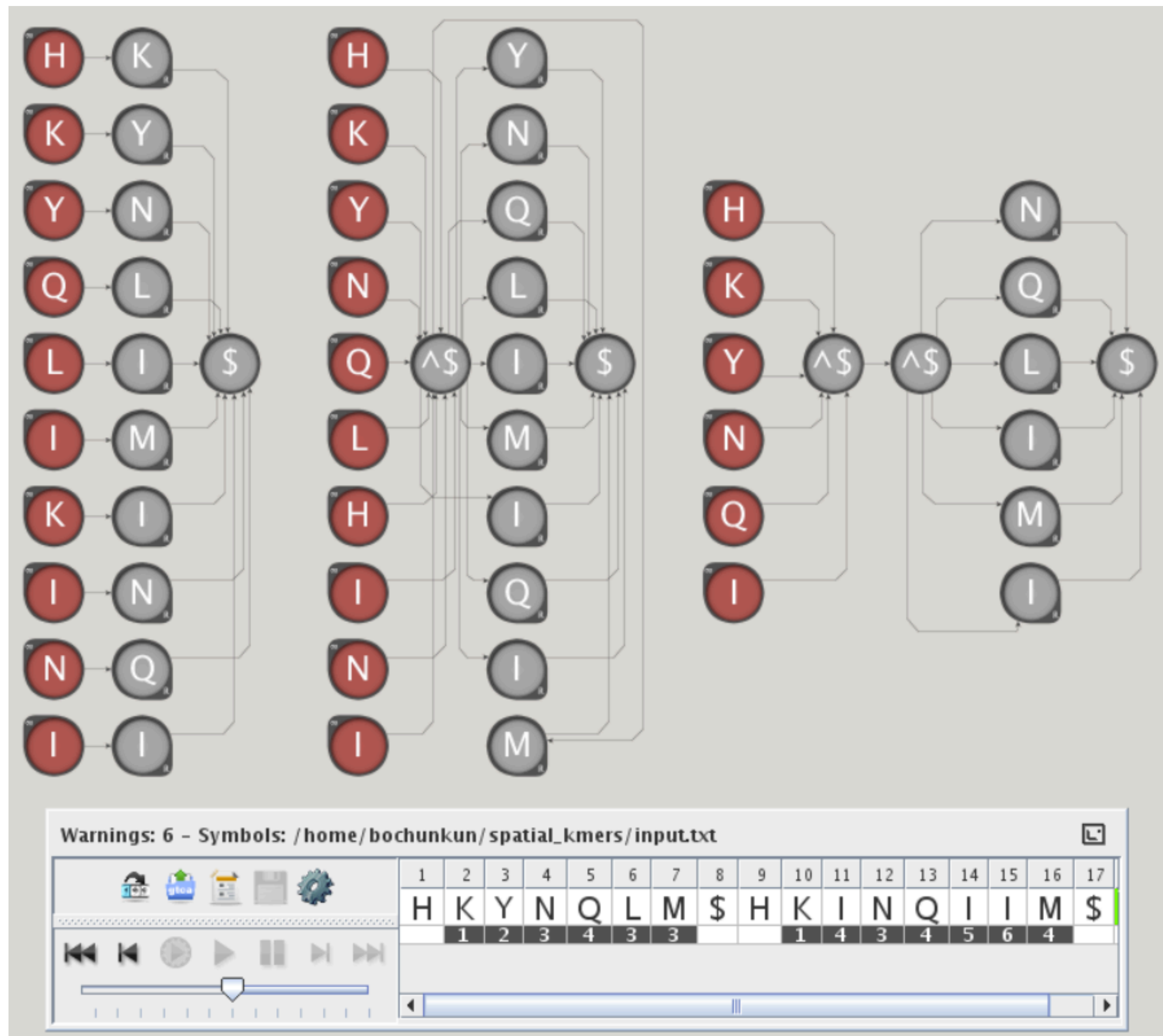
Input1=HKYNQLM

Input2=HKINQIIM

HK	H_Y	H__N	H___Q	H____L
KY	K_N	K__Q	K___L	K____I
YN	Y_Q	Y__L	Y___I	Y____M
NQ	N_L	N__I	N___M	
QL	Q_I	Q__M		
LI	L_M			
IM				

HK	H_I	H__N	H___Q	H____I
KI	K_N	K__Q	K___I	K____I
IN	I_Q	I__I	I___I	I____M
NQ	N_I	N__I	N___M	
QI	Q_I	Q__M		
II	I_M			
IM				

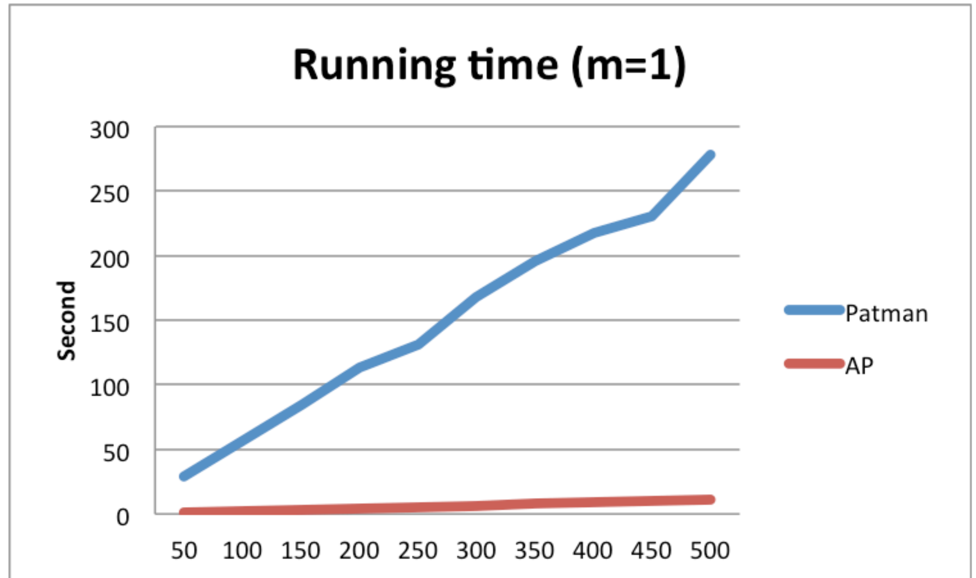
d = 0 d = 1 d = 2

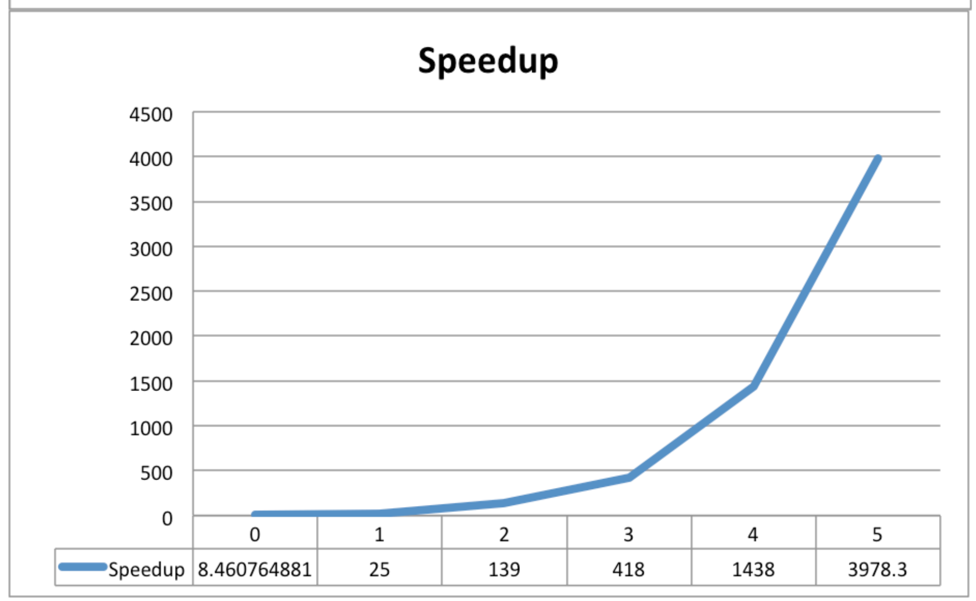


Performance Evaluation

- Both AP and PatMaN time increase linearly as input size increases
- PatMaN increases much more severely
- Different mismatch distances: similar trends

- Speedups increases exponentially





Application III: Association Rule Mining

Ke Wang, Yanjun Qi, Jeffrey J. Fox,
Mircea Stan, Kevin Skadron

University of Virginia
Center for Automata Processing

Association Rule Mining

Association rule mining (ARM, or frequent itemset mining, FIM):

- Identify ***strong rules*** discovered in databases
- The order of items within a transaction doesn't matter

- Web usage mining
- Traffic accident analysis
- Intrusion detection
- Market basket analysis
- Bioinformatics

Trans.	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer, Coke
5	Bread, Milk, Diaper, Coke

Itemset

K−Itemset

Support: number of transactions which contain this itemset

sup({Diaper, Milk})= 3

Minimum Support: threshold to tell frequent or not

AP Accelerated ARM – Concept

ARM	AP implementation
Item	Symbol 8-bit or 16-bit
Itemset	NFA by STEs
Transactions	Input Stream (Connecting by a special symbol)
Frequency counting	Counter Element

AP Accelerated ARM - Flowchart

Data preprocessing:

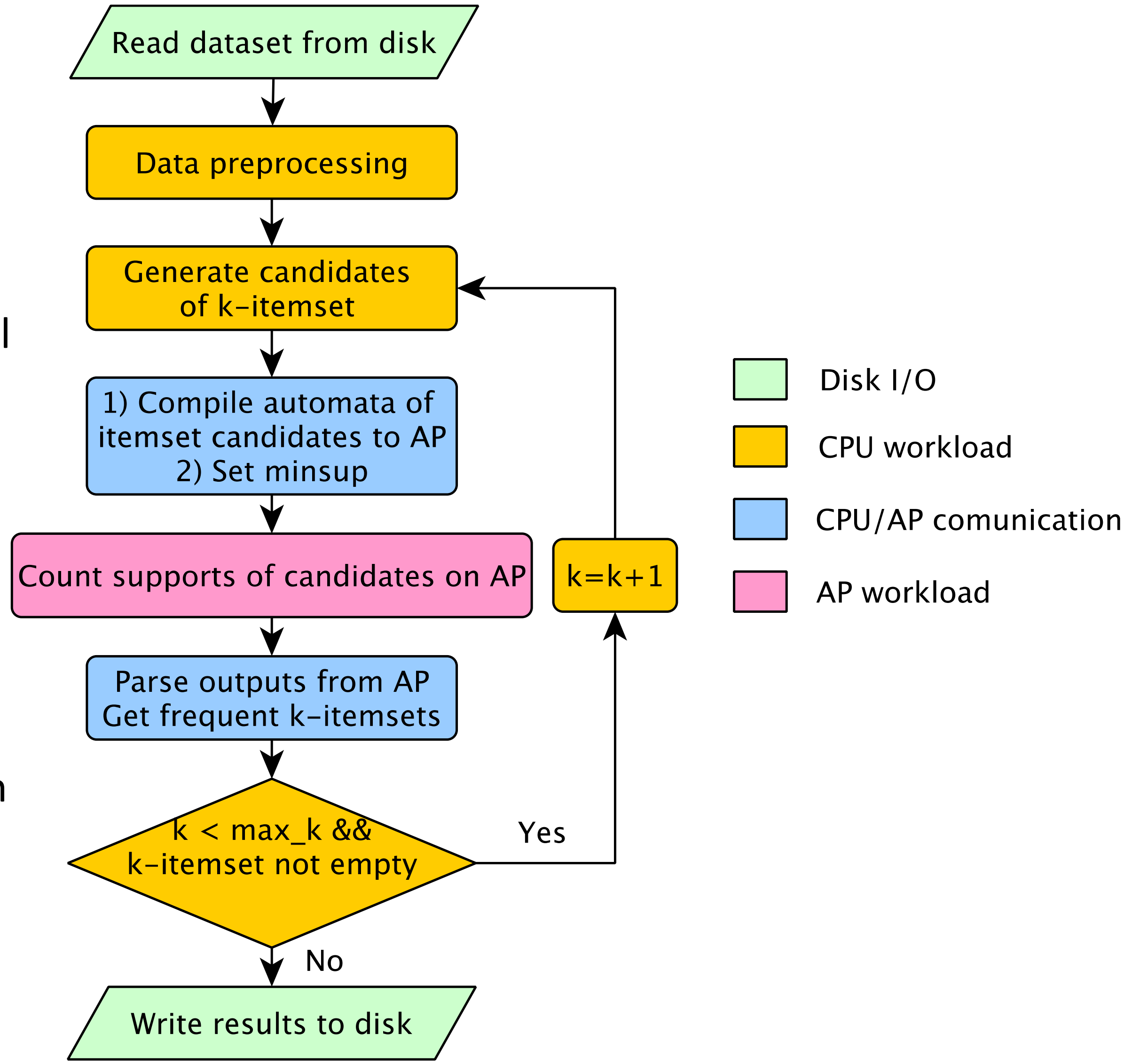
1) Filter out infrequent items
2) Recode -> 8-bit / 16-bit symbols
3) Recode transactions
4) Sort items in transactions
5) Connect transactions by a special symbol（\x255）

Encoding:

freq_item# <255: 8-bit
254< freq_item# <64516:16-bit

Sorting:

Descending sorting according to item frequency [1]



[1] Christian Borgelt, "Efficient implementations of Apriori and Eclat," in Proc. FIMI '03 , 2003

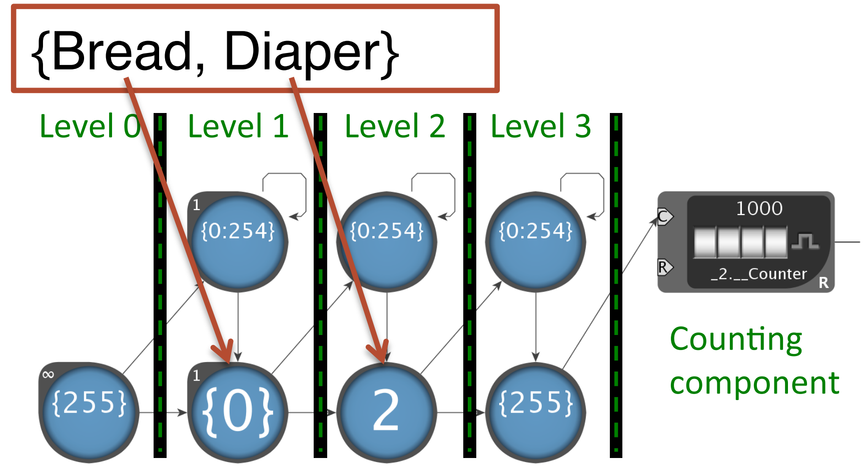
AP Accelerated ARM – Automata Design

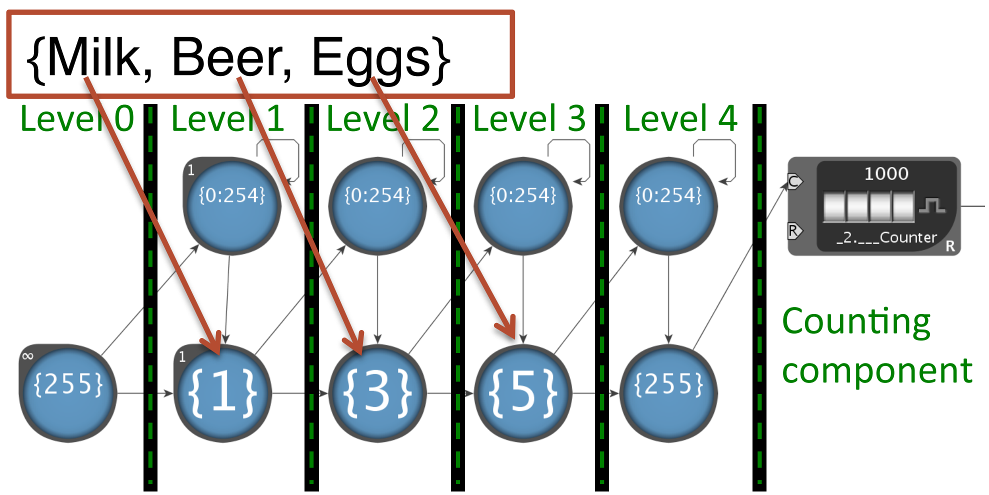
Trans.	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer, Coke
5	Bread, Milk, Diaper, Coke

Item	Code
Bread	0
Milk	1
Diaper	2
Beer	3
Coke	4
Eggs	5
Separator	255(\xFF)

Transaction stream:

01\xFF0235\xFF1234\xFF01234\xFF0124





Performance Evaluation - Datasets

- Four real-world datasets

Table I: **Real-World Datasets**

Name	Trans#	Aver. Len.	Item#	Size (MB)
Pumsb	49046	74	2113	16
Accidents	340183	33.8	468	34
Webdocs	1692082	177.2	5267656	1434
ENWiki	11507383	70.3	6322092	2997.5

Pumsb, Accidents and Webdocs are from *Frequent itemset mining dataset repository,"* http://fimi.ua.ac.be/data/.
ENWiki was generated English Wikipedia 2014

- Three synthetic datasets

Table II: **Synthetic Datasets**

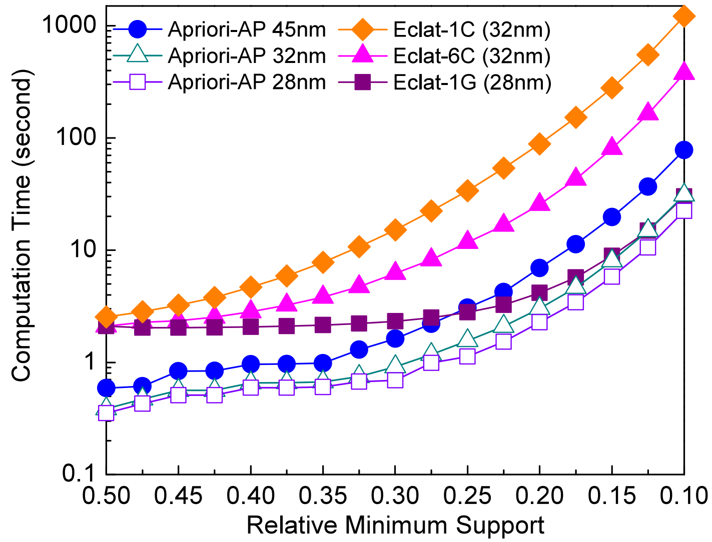
Name	Trans#	Aver. Len.	Item#	ALMP	Size (MB)
T40D500K	500K	40	100	15	49
T100D20M	20M	100	200	25	6348.8
Webdocs5X	8460410	177.2	5267656	N/A	7168

T40D500K and *T100D20M* ware generated from IBM Market-Basket Synthetic Data Generator
Webdocs5X is generated by duplicating transactions of Webdocs 5 times

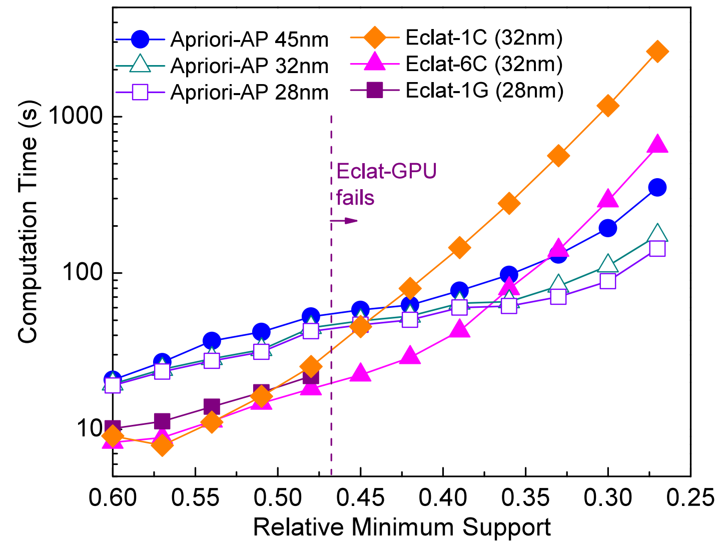
THE CENTER FOR AUTOMATA PROCESSING

UNIVERSITY of VIRGINIA

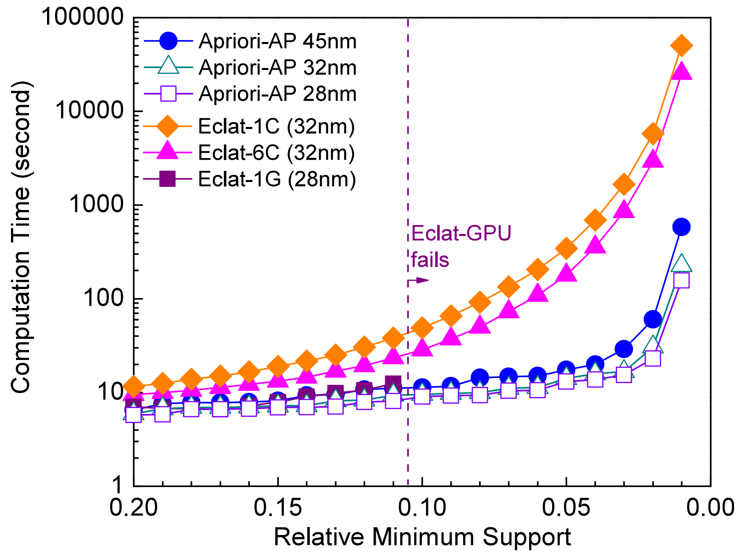
Performance Evaluation – vs. Eclat



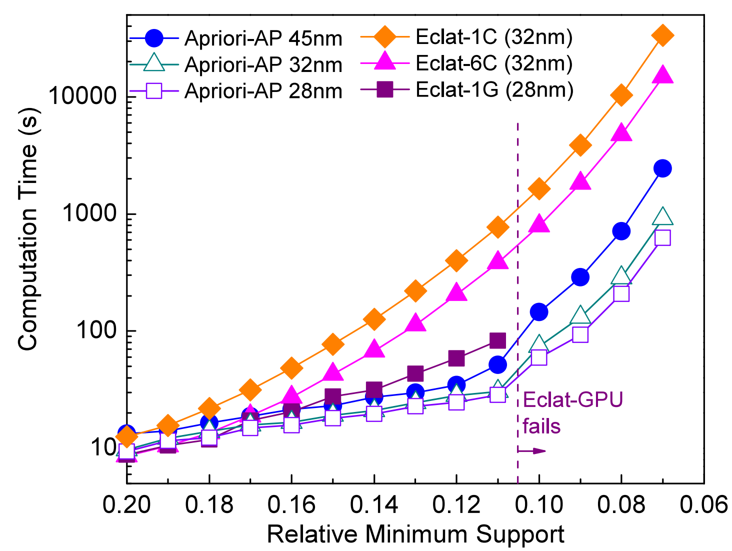
Accidents (34MB)



T100D20M (6.3GB)



ENWiki(3.0GB)



Webdocs5X (7.1GB)

Micron

Reference

- K. Zhou, J. J. Fox, K. Wang, D. E. Brown, and K. Skadron. "Brill Tagging on the Micron Automata Processor." In Proc. ICSC'15
- C. Bo, K. Wang, Y. Qi and K. Skadron. "String Kernel Testing Acceleration using the Micron Automata Processor". The 1st International Workshop of Computer Architecture for Machine learning. (In conjunction with ISCA'15)
- K. Wang, J. Qi, J. J. Fox, M. R. Stan, and K. Skadron. "Association Rule Mining with the Micron Automata Processor." In Proc. IPDPS'15