ERATO セミナ 2010 - No. 22

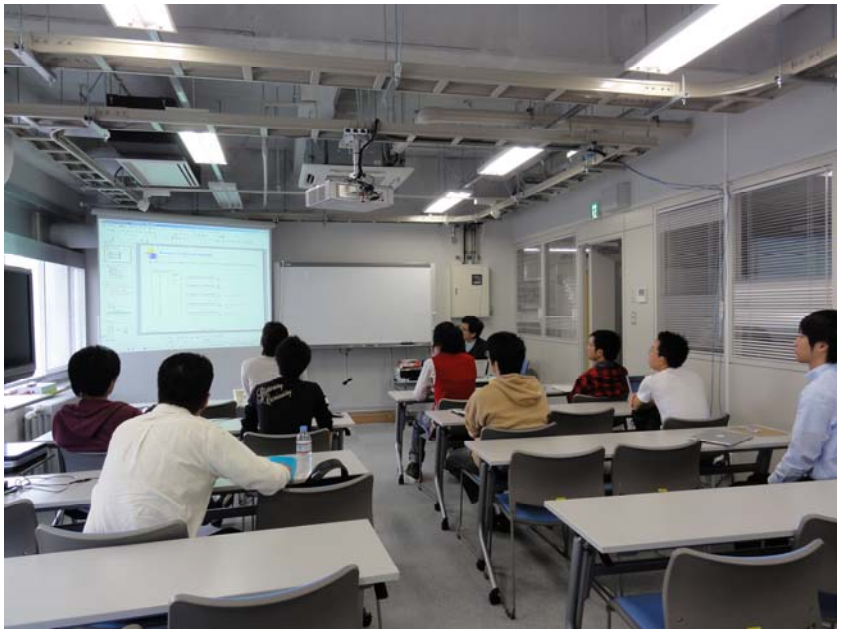
BDD/ZDD を基盤とする種々の離散構造の演算処理と代数系 (algebra) について

湊 真一
北大情報科学研究科/JST ERATO 湊離散構造プロジェクト

2010/10/22

概要

本 ERATO プロジェクトでは、離散構造を統合的に扱う基本処理系として BDD/ZDD を位置づけ，分野横断的な応用を持つ技術体系として再構築することを目指して研究活動を開始している．本講演では，BDD/ZDD を基盤とする離散構造とその演算処理の代数系 (algebra) の例として，BDD による論理関数処理系, ZDD による組合せ集合の処理系，ZDD ベクトル表現による組合せ頻度表の演算処理系，および Sequence BDD による系列集合の演算処理系を取り上げ、それらの代数系を比較するとともに、今後の展望について述べる。




[Minato et al. 2008]
two techniques

Recent Topics on Decision Diagrams and Discrete Structure Manipulation

Shin-ichi Minato
Graduate School of Information Science and Technology
Hokkaido University, Japan.

Background

- BDD-based algorithms have been developed mainly in VLSI logic design area. (since early 1990's.)
  - Equivalence checking for combinational circuits.
  - Symbolic model checking for logic / behavioral designs.
  - Logic synthesis / optimization.
  - Test pattern generation.
- Recently, BDDs are applied for not only VLSI design but also for more general purposes.
  - Data mining (Fast frequent itemset mining) [Minato2005,2008][Loekito,Bailey2006]
  - Computation of Bayesian networks for probabilistic system analysis.[Minato2007]

Contents of this talk

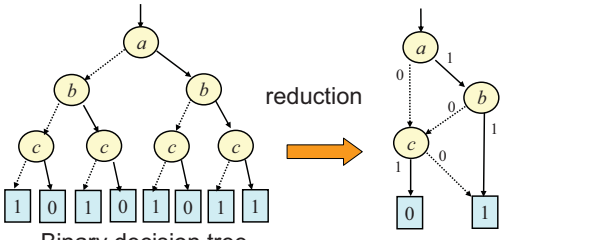
- BDD and ZDD for discrete structure manipulation
  - BDD and Boolean function algebra
  - ZDD and "Family algebra"
- Database analysis based on ZDD manipulation
  - Frequent itemset mining
  - ZDD vector and "Itemset-histogram algebra"
- Sequence BDD for set of sequences
  - Set of sequences and ZDD
  - Sequence BDD and "sequence family algebra"
- Our project for discrete structure manipulation system
  - JST "ERATO" project and current status

Contents of this talk

- BDD and ZDD for discrete structure manipulation
  - BDD and Boolean function algebra
  - ZDD and "Family algebra"
- Database analysis based on ZDD manipulation
  - Frequent itemset mining
  - ZDD vector and "Itemset-histogram algebra"
- Sequence BDD for set of sequences
  - Set of sequences and ZDD
  - Sequence BDD and "sequence family algebra"
- Our project for discrete structure manipulation system
  - JST "ERATO" project and current status

BDD (Binary Decision Diagram) [Bryant86]

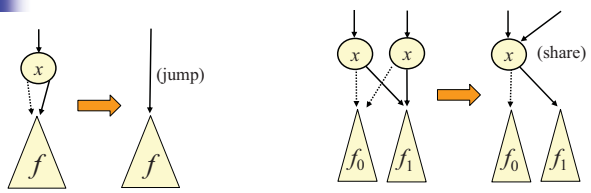
- Graph representation of Boolean function data.
  - Canonical form obtained by applying reduction rules to a binary tree with a fixed variable ordering.



Binary decision tree equivalent to truth table

Reduced Ordered BDD

BDD reduction rules



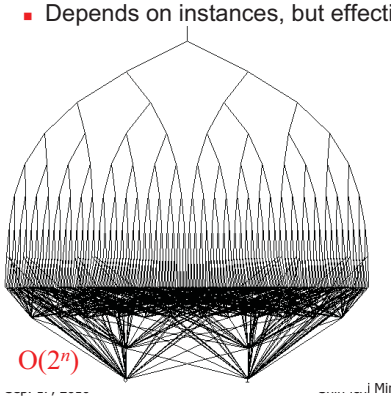
Eliminate all redundant nodes.

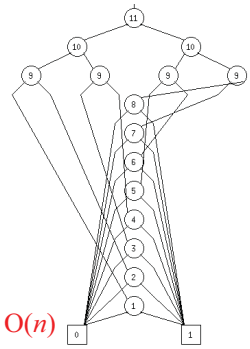
Share all equivalent nodes.

Gives a unique and compressed representation for a given Boolean function under a fixed variable ordering.

Effect of BDD reduction rules

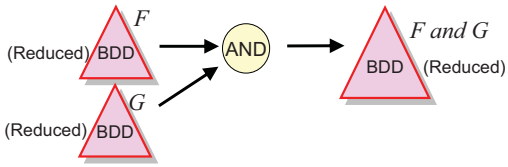
- Exponential advantage can be seen in extreme cases.
  - Depends on instances, but effective for many practical ones.





BDD-based logic operation algorithm

- If we generate BDDs from the binary tree: always requires exponential time & space. (→ impracticable for large number of variables)
- **Innovative BDD synthesis algorithm**
  - **Proposed by R. Bryant in 1986.**
  - Best cited paper for many years in EE&CS areas.



A BDD can be constructed from the two operands of BDDs. (Computation time is linear to BDD size.)

Boolean function and combinatorial itemset

a	b	c	F	
0	0	0	0	
1	0	0	0	
0	1	0	0	
1	1	0	1	→ ab
0	0	1	1	→ c
1	0	1	1	→ ac
0	1	1	0	
1	1	1	0	

Boolean function:
$F = (a\ b \sim c) \vee (\sim b\ c)$

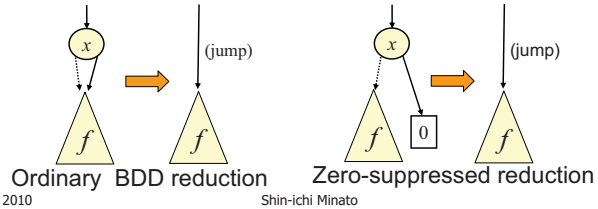
Combinatorial itemset:
$F = \{ab, ac, c\}$

(customer's choice)

- Operations of combinatorial itemsets can be done by BDD-based logic operations.
  - Union of sets → logical OR
  - Intersection of sets → logical AND
  - Complement set → logical NOT

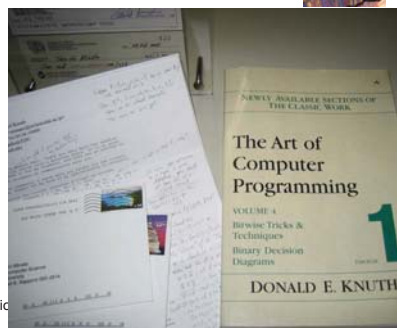
Zero-suppressed BDD (ZDD) [Minato93]

- A variant of BDDs for **combinatorial itemets.**
- Uses a new reduction rule different from ordinary BDDs.
  - Eliminate all nodes whose "1-edge" directly points to 0-terminal.
  - Share equivalent nodes as well as ordinary BDDs.
- If an item x does not appear in any itemset, the ZDD node of x is automatically eliminated.
  - When average appearance ratio of each item is 1%, ZDDs are more compact than ordinary BDDs, up to 100 times.



BDDs/ZDDs in the Knuth's book

- The latest Knuth's book fascicle (Vol. 4-1) includes a BDD section with 140 pages and 236 exercises.
- In this section, Knuth used 30 pages for ZDDs, including more than 70 exercises.
  - I honored to serve proofreading of the draft version of his article.
  - Knuth recommended to use "ZDD" instead of "ZBDD."
  - He reorganized ZDD operations and named "Family Algebra."
  - 2010/05, I visited Knuth's home and discussed the direction of future work.



Algebraic operations for ZDDs

- Knuth evaluated not only the data structure of ZDDs, but more interested in **the algebra on ZDDs**.

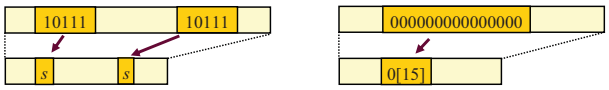
ϕ, {1}	***Empty*** and ***singleton set***. (0/1-terminal)	Basic operations (Corresponds to Boolean algebra)
$P.top$	Returns the ***item-ID*** at the top node of P.	
$P.onset(v)$ $P.offset(v)$	Selects the subset of itemsets including or excluding v.	
$P.change(v)$	Switching v (***add / delete***) on each itemset.	
$\cup$, $\cap$, $\setminus$	Returns ***union, intersection,*** and ***difference set***.	
$P.count$	***Counts number*** of combinations in P.	
$P * Q$	***Cartesian product set*** of P and Q.	**New operations introduced by Minato.**
P / Q	***Quotient set*** of P divided by Q.	
$P \% Q$	***Reminder set*** of P divided by Q.	

Useful for many practical applications.

Formerly I called this "unate cube set algebra," but Knuth reorganized as **"Family algebra."**

Principles for performance improvement

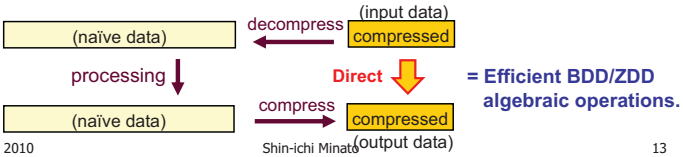
- General principles:
  Data compression & Pruning search space.
- Two basic techniques for data compression:



Dictionary-based coding
(= BDD/ZDD sub-graph sharing)

Run-length coding
(= BDD/ZDD redundant node deletion)

- Computability without decompression.

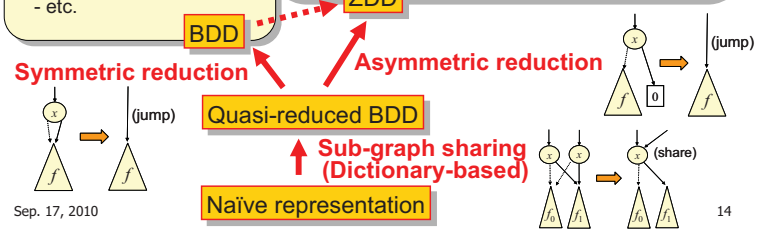


Comparison of BDDs and ZDDs

- Many of real-life problems are likely **asymmetric**.

(Asymmetric world)
- Data mining & Knowledge discovery
- Market data analysis
- Web data analysis
- Formal concept analysis
- Risk analysis
- Calculation with Bayesian networks
- Machine learning & clustering
- etc.

(Symmetric world)
- VLSI Logic Design
- Formal Verification
- etc.



Contents of this talk

- BDD and ZDD for discrete structure manipulation
  - BDD and Boolean function algebra
  - ZDD and "Family algebra"
- Database analysis based on ZDD manipulation
  - Frequent itemset mining
  - ZDD vector and "Itemset-histogram algebra"
- Sequence BDD for set of sequences
  - Set of sequences and ZDD
  - Sequence BDD and "sequence family algebra"
- Our project for discrete structure manipulation system
  - JST "ERATO" project and current status

Frequent itemset mining

- Basic and well-known problem in database analysis.

Record ID	Tuple
1	*a b c*
2	*a b*
3	*a b c*
4	*b c*
5	*a b*
6	*a b c*
7	*c*
8	*a b c*
9	*a b c*
10	*a b*
11	*b c*

Frequency threshold = 10 → *{ b }*

Frequency threshold = 8 → *{ ab, a, b, c }*

Frequency threshold = 7 → *{ ab, bc, a, b, c }*

Frequency threshold = 5 → *{abc, ab, bc, ac, a, b, c }*

Frequency threshold = 1 → *{abc, ab, bc, ac, a, b, c }*

Existing itemset mining algorithms

- Frequent itemset mining is one of the fundamental data mining problems.
- Previous work:
  - ***Apriori*** [Agrawal1993]
    First efficient method of enumerating all frequent patterns.
    Breadth-first search with dynamic programming.
  - ***Eclat*** [Zaki1997]
    Depth-first search algorithm. Less memory consuming.
    In some cases, faster than Apriori.
  - ***FP-growth*** [Han2000]
    Depth-first search using "FP-tree," graph-based data structure.
- ***LCM*** *(Linear time Closed itemset Miner)* [Uno2003]
  - with a theoretical bound as output linear time.
  - known as one of the fastest implementation.

"LCM over ZDDs" [Minato et al. 2008]

- ***LCM:*** *[Uno2003]*
  Output-linear time algorithm of frequent itemset mining.
- ***ZDD:*** *[Minato93]*
  A compact graph-based representation for large-scale sets of combinations.

Combination of the two techniques

Generates large-scale frequent itemsets on the main memory, with a very small overhead from the original LCM.

(→ Sub-linear time and space to the number of solutions when ZDD compression works well.)

LCM over ZDDs: An example

- The results of frequent itemsets are obtained as ZDDs on the main memory. (not generating a file.)

Record ID	Tuple
1	*a b c*
2	*a b*
3	*a b c*
4	*b c*
5	*a b*
6	*a b c*
7	*c*
8	*a b c*
9	*a b c*
10	*a b*
11	*b c*

LCM over ZDDs

Freq. thres. $\alpha = 7$ →

$\{ ab, bc, a, b, c \}$

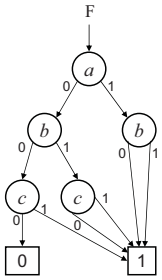
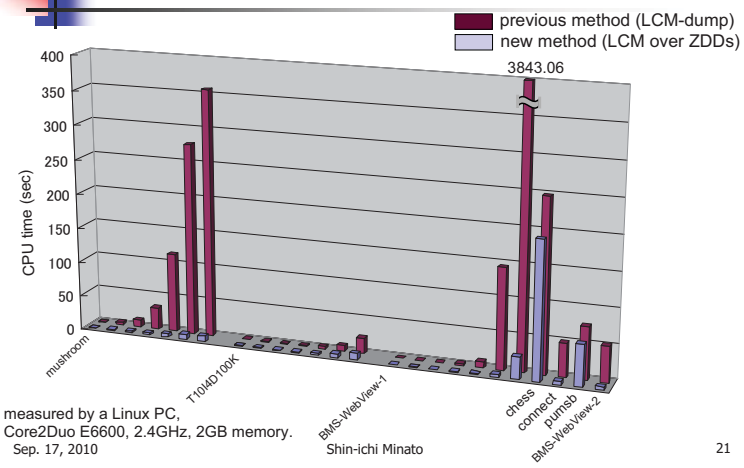


Table 2. Co… CM o… pre… growth

solutions | LCM over ZDDs | Original LCM

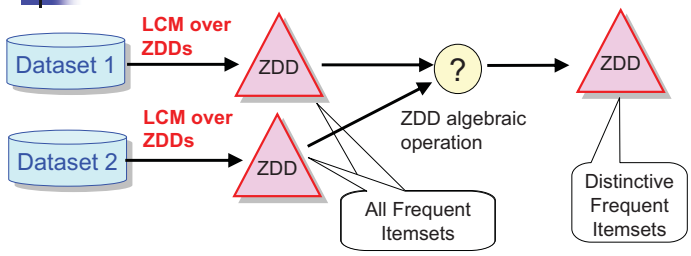
Dataset name: min. support	# solutions: # itemsets	CMov: ZBDD	LCM over ZDDs: Time(s)	pre: Time(s)	Original LCM: Time(s)	growth: Time(s)
mushroom: 1,000	123,287	760	0.50	0.49	0.64	1.78
500	1,442,504	2,254	1.32	1.30	3.29	3.49
300	5,259,786	4,412	2.25	2.22	9.96	5.11
200	18,094,822	6,383	3.21	3.13	31.63	6.24
100	66,076,586	11,584	5.06	4.87	114.21	6.72
70	153,336,056	14,307	7.16	7.08	277.15	6.97
50	198,169,866	17,830	8.17	7.86	357.27	6.39
T10I4D100K: 100	27,533	8,482	0.85	0.85	0.86	209.82
50	53,386	16,872	0.97	0.92	0.98	242.31
20	129,876	58,413	1.13	1.08	1.20	290.78
10	411,366	173,422	1.55	1.36	1.64	332.22
5	1,923,260	628,491	2.86	2.08	3.54	370.54
3	6,169,854	1,576,184	5.20	3.15	8.14	386.72
2	19,561,715	3,270,977	9.68	5.09	22.66	384.60
BMS-WebView-1: 50	8,192	3,415	0.11	0.11	0.12	29.46
40	48,544	10,755	0.18	0.18	0.22	48.54
36	461,522	28,964	0.49	0.42	0.98	67.16
35	1,177,608	38,164	0.80	0.69	2.24	73.64
34	4,849,466	49,377	1.30	1.07	8.58	83.36
33	69,417,074	59,119	3.53	3.13	144.98	91.62
32	1,531,980,298	71,574	31.90	29.73	3,843.06	92.47
chess: 1,000	29,442,849	53,338	197.58	197.10	248.18	1,500.78
connect: 40,000	23,981,184	3,067	5.42	5.40	49.21	212.84
pumsb: 32,000	7,733,322	5,443	60.65	60.42	75.29	4,189.09
BMS-WebView-2: 5	26,946,004	353,091	4.84	3.62	51.28	118.01

Performance of LCM over ZDDs



measured by a Linux PC, Core2Duo E6600, 2.4GHz, 2GB memory.

Post Processing after LCM over ZDDs



- We can extract distinctive itemsets by comparing frequent itemsets for multiple sets of databases.
  - Various ZDD algebraic operations can be used for the comparison of the huge number of frequent itemsets.

Itemset-histograms for DB analysis

Itemset DB

Record ID	Itemset
1	a b c
2	a b
3	a b c
4	b c
5	a b
6	a b c
7	c
8	a b c
9	a b c
10	a b
11	b c

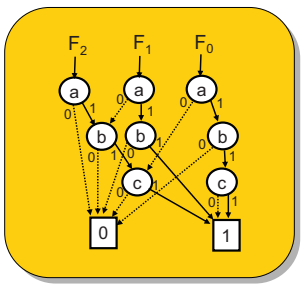
Itemset-histogram

Itemset	Freq.
a b c	5
a b	3
b c	2
c	1

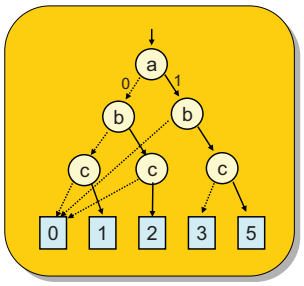
- We sometimes need to represent **not only yes/no** but also the **frequency** (number of occurrence) for each itemset.
  - This is a histogram of itemsets if frequency of all itemsets are generated.
  - Integer-valued Sum-Of-Product form. (VSOP)
  - Function from Combination to Integer. (CtoI)
  - Multi-set of combinations

ZDD Vectors and Multi-Terminal BDD

- The two representation can be considered.
- Just a variable ordering problem of a same BDD.



ZDD Vector



Multi-Terminal BDD (MTBDD)

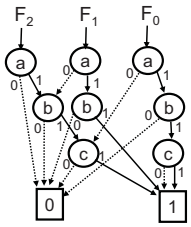
ZDD vector for itemset-histogram

- A ZDD distinguishes only existence of each tuple in the transaction data. (cannot count frequency.)
- We use a binary encoded method with ZDD vectors:
  - Encode frequency numbers into *m*-bit binary code, and represent each bit of combination set using a ZDD.

(例)

itemset	frequency	F_2	F_1	F_0
abc	5 (101)	1	0	1
ab	3 (011)	0	1	1
bc	2 (010)	0	1	0
c	1 (001)	0	0	1

F_0 = {abc, ab, c}
F_1 = {ab, bc}, F_2 = {abc}



Algebra for itemset-histograms

H

Tuple	Freq.
a b c	5
a b	3
b c	2
c	1

H_1=*H*.factor1(*a*)

Tuple	Freq.
b c	5
b	3

H_0=*H*.factor0(*a*)

Tuple	Freq.
b c	2
c	1

$H_1 + H_0$

Tuple	Freq.
b c	7
b	3
c	1

H * *d*

Tuple	Freq.
a b c d	5
a b d	3
b c d	2
c d	1

- Primitive operations:
  - Factoring into two parts by an item.
  - Attaching an item.
  - Sum of two histograms.
  - Counting lines in the table.

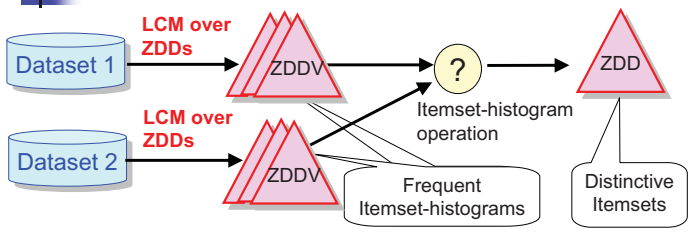
- Each table is compactly represented by ZDDs.
- ZDDs are shared each other in the memory.

Basic operations of Itemset-histogram algebra

Table 2: Algebraic operations for itemset-histgrams.

c ($c \in N$)	Returns the histogram of only λ with occurrence c. ($c = 0$ means empty.)
$P.top$	Returns item-ID of the highest order involved in P.
$P.offset(v)$	Returns sub-histogram of itemsets not including item v.
$P.onset(v)$	Gets $P - P.offset(v)$ and then deletes v from each itemset.
$P.change(v)$	Inverts existence of v (add / delete) on each itemset.
$Max(P, Q)$	Histogram with the greater occurrences between P and Q.
$Min(P, Q)$	Histogram with the less occurrences between P and Q.
$P - Q$	Difference of the two histograms. (limited to only positive occurrences.)
$P.count$	Counts the variety of itemsets in P.
$P.upperbound$	Returns the highest occurrence number among all itemsets in P.
$P.unitset$	Returns the histogram with occurrence = 1 for all itemsets in P.
$P * Q$	Cartesian product of P and Q.
P/Q	Quotient of P divided by Q.
$P\%Q$	Reminder of P divided by Q.
$P + Q$	Addition of number of occurrences.

Post Processing after LCM over ZDD(V)s



- The result can be analyzed flexibly using itemset-histogram operations.
  - Extracting long/short frequent patterns.
  - Comparison of two sets of frequent patterns.
  - Calculating statistical data (e.g. confidence, support)
  - Finding disjoint sub-factors in frequent patterns.

Contents of this talk

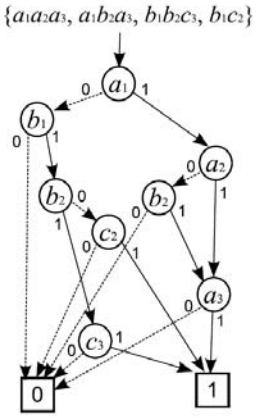
- BDD and ZDD for discrete structure manipulation
  - BDD and Boolean function algebra
  - ZDD and "Family algebra"
- Database analysis based on ZDD manipulation
  - Frequent itemset mining
  - ZDD vector and "Itemset-histogram algebra"
- Sequence BDD for set of sequences
  - Set of sequence and ZDD
  - Sequence BDD and "sequence family algebra"
- Our project for discrete structure manipulation system
  - JST "ERATO" project and current status

Sets of sequences (sets of strings)

- Sets of combinations:
  - Don't consider order and duplication of items
  - "abcc" and "bca" are the same.
- Sets of sequences:
  - Distinguishes all finite sequences.
  - { λ }, { ab, aba, bbc }, { a, aa, aaa, aaaa }, etc.
  - Here we exclude infinite sequences such as { a* }.
- So many real-life applications.
  - Text search and indexing
  - Web (html/xml) data mining
  - Bio informatics

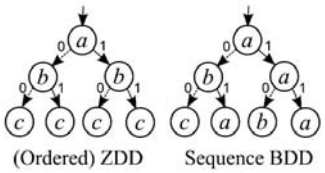
Encoded ZDDs for Sets of sequences

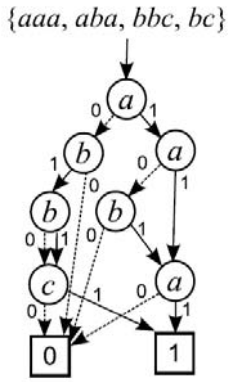
- Pair of (Item - position) is considered different symbol.
  "aaa" → "a1 a2 a3"
  "aba" → "a1 b2 a3"
- Alphabet size: $|\Sigma|$
  Maximum length of sequences: n
  Total encoded symbols: $|\Sigma| \times n$
- Not very efficient.
  - Many symbols needed.
  - We need to put a fixed maximum length of sequences.



Sequence BDD (SeqBDD)

- Loekito, Bailey, and Pei (2009)
  - Same as ZDD reduction rule.
  - **Only 0-edges** keep variable ordering.
  - 1-edges has no restriction.
  - Still unique representation for a given set of sequences.
  - Each path from root to 1-terminal corresponds to a sequence.



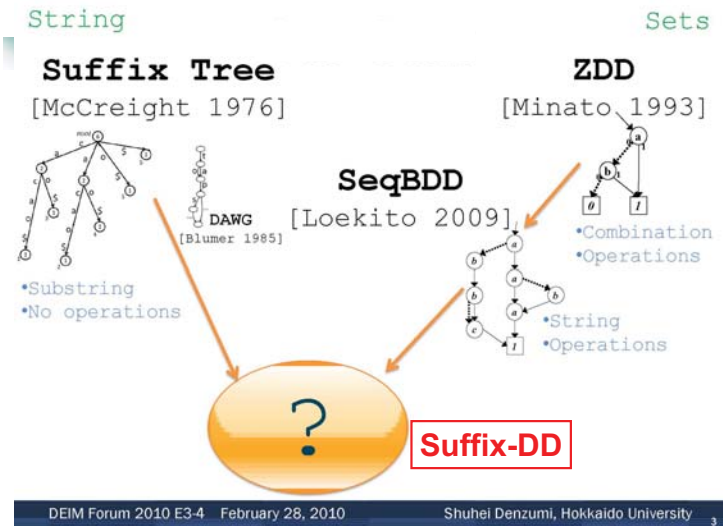


Basic operations of sequence family algebra

Table 3: Algebraic operations for SeqBDDs

$\emptyset$	Returns empty set. (0-termial node)
$\{\lambda\}$	Returns the set of only null-combination. (1-terminal node)
$P.top$	Returns item-ID at root node of P.
$P.onset(x)$	Selects the subset of sequences that begin with letter x, and then removes x from the head of each sequence.
$P.offset(x)$	Selects the subset of sequences that do not begin with letter x.
$P.push(x)$	Appends x to the head of every sequence in P.
$P \cup Q$	Returns union set.
$P \cap Q$	Returns intersection set.
$P \setminus Q$	Returns difference set. (in P but not in Q.)
$P.count$	Counts number of combinations.
$P * Q$	Cartesian product of P and Q. (Concatenations of all pairs in P and in Q)

- ZDD-like algebraic operations.
  - *onset, offset*, and *push* operations are different.
  - Other operations are almost same.



Contents of this talk

- BDD and ZDD for discrete structure manipulation
  - BDD and Boolean function algebra
  - ZDD and "Family algebra"
- Database analysis based on ZDD manipulation
  - Frequent itemset mining
  - ZDD vector and "Itemset-histogram algebra"
- Sequence BDD for set of sequences
  - Set of sequences and ZDD
  - Sequence BDD and "sequence family algebra"
- Our project for discrete structure manipulation system
  - JST "ERATO" project and current status

ERATO Projects

- **Top projects** for scientific research in Japan.
  - Executed by JST (Japan Science and Technology Agency).
  - 5 projects / Year are accepted from all scientific subjects. (Computer Science: 0 or 1 project / Year.)
  - 100 projects have been accepted in 30 years.
  - Each project has 5 years long, total 1 billion Yen. about 10 PD researchers and 3 admin staffs.
- This project is accepted on Oct. 2009.
  - Research activities started from April 2010, until March 2015.

Discrete structures and applications

Many problems solved by computers can be decomposed as a type of **discrete structures** using simple primitive operations.
→ Often needs **a huge amount of enumerative operations**.

design automation | data mining / knowledge discovery | fault analysis | bio informatics | machine learning / classification | constraint satisfaction problem | web data analysis

So many applications
→ Great effects for the society.

Our main subject

Performance Improvement (10–100x)

Discrete structure manipulation system

set theory | symbolic logic | inductive proof | combinatorics | graph theory | probabilistic theory

Foundational materials for C.S. and math.

37

Technical stance of our project

Digital system optimization & verification | Knowledge discovery & data mining | Statistical Analysis & modeling

Application-specific technical areas. → Application (Engineering) | Application (Engineering) | Application (Engineering)

Our objective layer:
- **Not only concept / theory, but also efficiency of implementation.**
- **Beauty of simplicity / universality.**

→ **Discrete structure manipulation system ("Art" / algebraic architecture)**

Basis of CS & math. (conceptual, theoretical) → Computation theory (Science / Mathematics)

38

Direction of our research

Current ZDDs → **Primary output**: **Applications in asymmetric world**
Data mining, Machine learning
Advanced searching etc.

Still many applications remains where ZDDs would be effective.

(Combinatorial)

(Higher model)

Further outputs

- **Multisets**
- **Sequences**
- **Permutations**
- **Partitions**
- **Trees, DAGs**
- **Networks**
- **etc.**

Advanced ZDD-like structure

Develop special new algebraic operations.

Applications with higher data model
Sequence data analysis
Numerical data processing
Processing of trees or semi-structured data

39

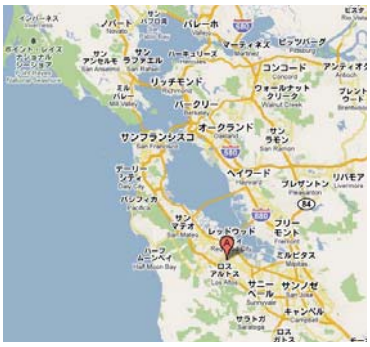
Location of laboratories

- Main lab is located in Hokakido Univ.
  - Center of attractive city: Sapporo.
  - 300m^2 space devoted for the project.
  - Convenient access from the airport and the station.
  - **→ Good environment for the members to concentrate into research.**
- Satellite labs located in Tokyo and Osaka.
  - Collaboration with many univ and companies.
  - **Connecting by high-quality tele-conf. system.**

25

Visit to Knuth's home

- May, 2010, during my trip to US for attending SIAM Data Mining Conference
- Visited Knuth's home in Stanford Univ. campus.
  - Greetings and thanks for personal check giving for me and our students who found error of the Book.
  - Discussion on future work on BDD/ZDD and other discrete structures.



Outside of Knuth's house

- In the faculty's house area of Stanford campus on an 80 years' lease (1968～2048)

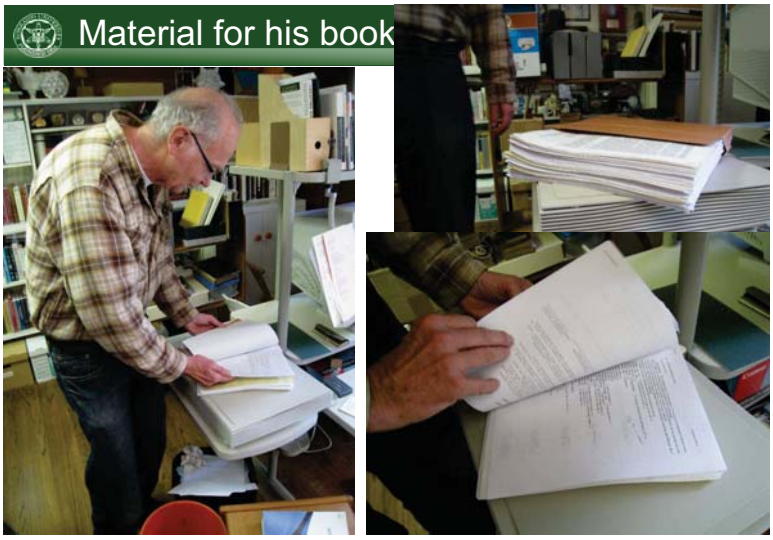
He has a room in Univ., but he mainly works in his home.

No educational work. Only attends annual special lecture And weekly faculty's lunch.

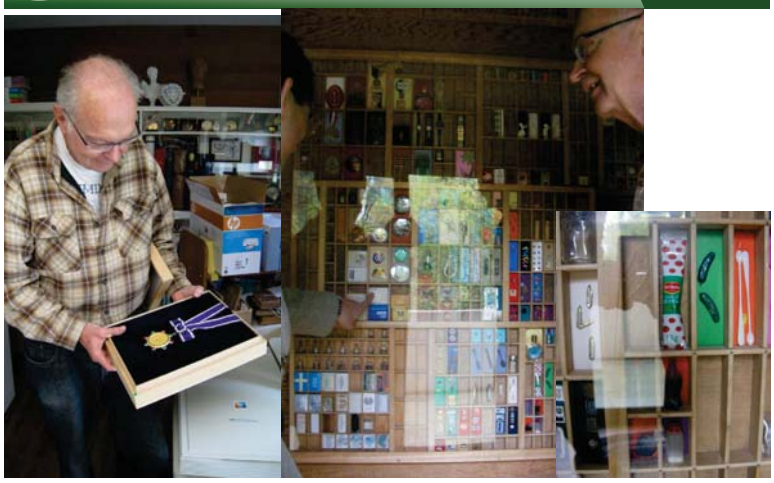
Knuth's pipe-organ and his work room

Bookshelves

Material for his book



Kyoto-award medal and display frame



Discussion with Prof. Knuth

- Sequence BDD and applications.
  - He is very interested in the new data structure.
- I told him that I got too busy to manage the project since ZDD is written in Knuth book.
  - He recognizes his writing may have significant effect, and he said:
    "I'm partly responsible to make your life change. So, let's discuss future work of your project."
  - Knuth's proposals have same direction as our ERATO project: **"Finding and organizing higher-level algebraic structures based on BDD/ZDD."**

49

Summary

- Focus on "discrete structure manipulation system."
  - Fundamentals for various practical applications.
- Based on BDDs/ZDDs.
  - Representing "logic" and "set," primitive models of discrete structures.
  - We will consider higher-level algebraic structures.
- Technical stance of the project.
  - Producing "Art" to connect Science and Engineering.
- Two objectives of the project.
  - Organizing new algebraic structures / operations.
  - Providing implementation with deep knowhow.
- Best use of ERATO framework
  - Organize an active research group with synergistic effect.
  - Contribution to the society to provide good research results as well as good researchers.

50