<table>
<thead>
<tr>
<th>項目</th>
<th>内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>タイトル</td>
<td>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</td>
</tr>
<tr>
<td>著者</td>
<td>湊 真一</td>
</tr>
<tr>
<td>発行日</td>
<td>2011-06</td>
</tr>
<tr>
<td>ファイル情報</td>
<td>ERATOセミナ 2010 : No.22. 2010年10月22日</td>
</tr>
</tbody>
</table>

HOKKAIDO UNIVERSITY
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。
Recent Topics on Decision Diagrams and Discrete Structure Manipulation

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

Sep. 17, 2010

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

BDD (Binary Decision Diagram) [Bryant86]

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

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

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Boolean function: \( F = (a \land \neg c) \lor (\neg b \land c) \)

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

- Operations of combinatorial itemsets can be done by BDD-based logic operations.
  - Union of sets \( \lor \) logical OR
  - Intersection of sets \( \land \) logical AND
  - Complement set \( \lnot \) logical NOT

Zero-suppressed BDD (ZDD) [Minato93]

- A variant of BDDs for combinatorial itemsets.
- 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.
  - \( \phi, \{1\} \) Empty and singleton set, \( \{0\} \) 1-terminial
  - \( P, \top \) Returns the item-ID at the top node of \( P \).
  - \( P, \text{offset()} \) Selects the subset of items including or excluding \( v \).
  - \( P, \text{change(v)} \) Switching \( v \) (add / delete) on each itemset.
  - \( P, \cup, \cap \) Returns union, intersection, and difference set.
  - \( P, \text{count} \) Counts number of combinations in \( P \).
  - \( P \times Q \) Cartesian product set of \( P \) and \( Q \).
  - \( P \div Q \) 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
  - Run-length coding

Computability without decompression.

Comparison of BDDs and ZDDs

- Many of real-life problems are likely asymmetric.

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

Frequent itemset mining

- Basic and well-known problem in database analysis.

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.

Frequent itemset mining

<table>
<thead>
<tr>
<th>Record ID</th>
<th>Tuple</th>
<th>Frequency threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a b c</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>a</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>a b c</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>b c</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>a b</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>a b c</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>c</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>a b c</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>a b c</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>a b</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>b c</td>
<td>1</td>
</tr>
</tbody>
</table>
LCM over ZDDs: An example

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

Freq. thres. \( \alpha = 7 \)

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

Performance of LCM over ZDDs

Post Processing after LCM over ZDDs

Itemset-histograms for DB analysis

ZDD Vectors and Multi-Terminal BDD
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.

Example:

<table>
<thead>
<tr>
<th>Tuple</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>abc</td>
<td>5 (101)</td>
</tr>
<tr>
<td>ab</td>
<td>3 (011)</td>
</tr>
<tr>
<td>bc</td>
<td>2 (010)</td>
</tr>
<tr>
<td>c</td>
<td>1 (001)</td>
</tr>
</tbody>
</table>

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

Algebra for itemset-histograms

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

<table>
<thead>
<tr>
<th>Tuple</th>
<th>Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>5</td>
</tr>
<tr>
<td>b</td>
<td>3</td>
</tr>
<tr>
<td>c</td>
<td>2</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
</tr>
</tbody>
</table>

$H = H_1 \cdot 1 + H_0 \cdot 0$

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.

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.
    - \( \{a\}, \{ab, aba, bbc\}, \{a, aa, aaa, aaaa\} \), etc.
  - Here we exclude infinite sequences such as \(a^*\).
- So many real-life applications.
  - Text search and indexing
  - Web (html/xml) data mining
  - Bioinformatics
**Encoded ZDDs for Sets of sequences**

- Pair of (item - position) is considered different symbol.
  - “aaa” → “a1 a2 a3”
  - “aba” → “a1 b2 a3”
- Alphabet size: |Σ|
- Maximum length of sequences: n
- Total encoded symbols: |Σ| × 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>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Returns empty set (0-terminal node)</td>
</tr>
<tr>
<td>(x)</td>
<td>Returns set of only null combination. (1-terminal node)</td>
</tr>
<tr>
<td>P[0]</td>
<td>Returns item 0 at root node of P</td>
</tr>
<tr>
<td>Ponset(x)</td>
<td>Selects the subset of sequences that begin with letter x, and then removes x from the head of each sequence.</td>
</tr>
<tr>
<td>P.offset(x)</td>
<td>Selects the subset of sequences that do not begin with letter x.</td>
</tr>
<tr>
<td>P.push(x)</td>
<td>Appends x to the head of every sequence in P.</td>
</tr>
<tr>
<td>P U Q</td>
<td>Returns union set.</td>
</tr>
<tr>
<td>P \ Q</td>
<td>Returns intersection set.</td>
</tr>
<tr>
<td>P \ Q</td>
<td>Returns difference set. (in P but not in Q.)</td>
</tr>
<tr>
<td>P x Q</td>
<td>Counts number of combinations.</td>
</tr>
<tr>
<td>P x Q</td>
<td>Cartesian product of P and Q. (Concatenations of all pairs in P and in Q)</td>
</tr>
</tbody>
</table>

- 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.
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.

**Discrete structures and applications**

- 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**

- Discrete structure manipulation system
- set theory
- symbolic logic
- inductive proof
- Combinatorics
- graph theory
- probabilistic theory

→ Foundational materials for C.S. and math.

**Technical stance of our project**

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

**Discrete structure manipulation system**

- Computation theory (Science / Mathematics)
- Application (Engineering)
- Knowledge discovery & data mining
- Statistical Analysis & modeling

→ Digital system optimization & verification
→ Application (Engineering)
→ Application (Engineering)

**Direction of our research**

- Current ZDDs
- (Combinatorial)

**Primary output**

- Applications in asymmetric world
- Data mining, Machine learning
- Advanced searching etc.

**Further output**

- Develop special new algebraic operations.

→ Applications with higher data model
- Advanced ZDD-like structure
- Sequence data analysis
- Numerical data processing
- Processing of trees or semi-structured data

**Location of laboratories**

- Main lab located in Hokkaido Univ.
  → Center of attractive city: Sapporo.
  → 300m² 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.

**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.
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.”

Discussion with Prof. Knuth

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.

Summary