Construction of Substring Indices Using Sequence BDDs

系列二分決定グラフを用いた部分文字列索引の構築

Shuhei DENZUMI	MC1, Faculty of Eng., Hokkaido University
Hiroki ARIMURA	Grad. School of IST, Hokkaido University
Shin-ichi MINATO	Grad. School of IST, Hokkaido University

String

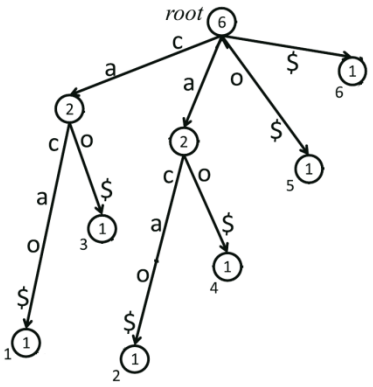
Sets

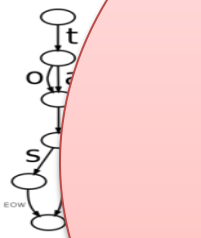
Our Goal

Suffix Tree

ZDD

[McCreight 197 ... 993]





[Blum ...

- Substring
- No operations

- Substring
- Operations

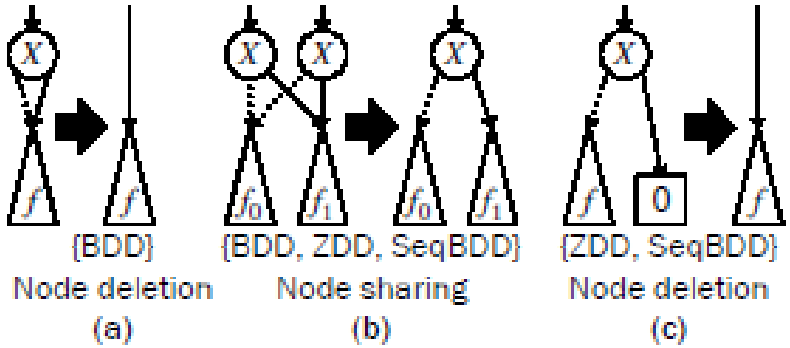
Goal:
We want to devise an **efficient substring index structure** based on SeqBDD

?

SuffixDD

[This work, 2010]

- BDD (Binary Decision Diagram)
  - Bryant, 1986
  - Two reduction rules
- ZDD (Zero-suppressed BDD)
  - Minato, 1993
  - Two reduction rules



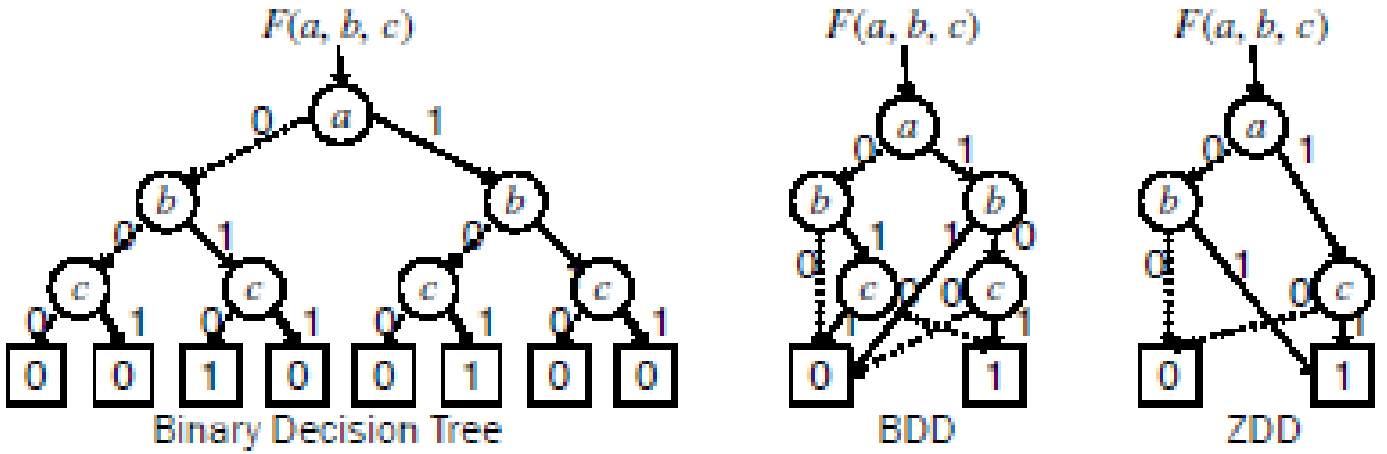
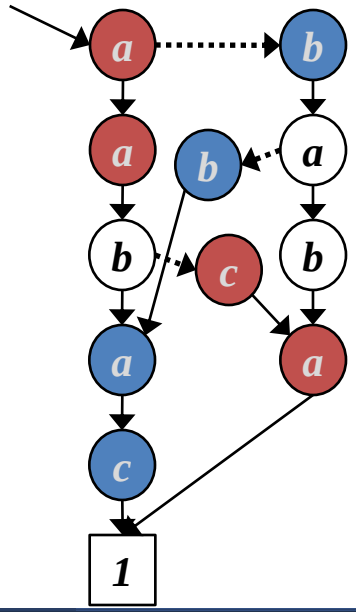


Fig. 1. Binary Decision Tree, BDDs and ZDDs

SeqBDD (Sequence BDD)

- Loekito, Bailey and Pei, 2009
  - Variant of ZDD
  - 0-edges are ordered (variable order is fixed)
  - 1-edges are not ordered
  - A letter is allowed to occur multiple times in a path

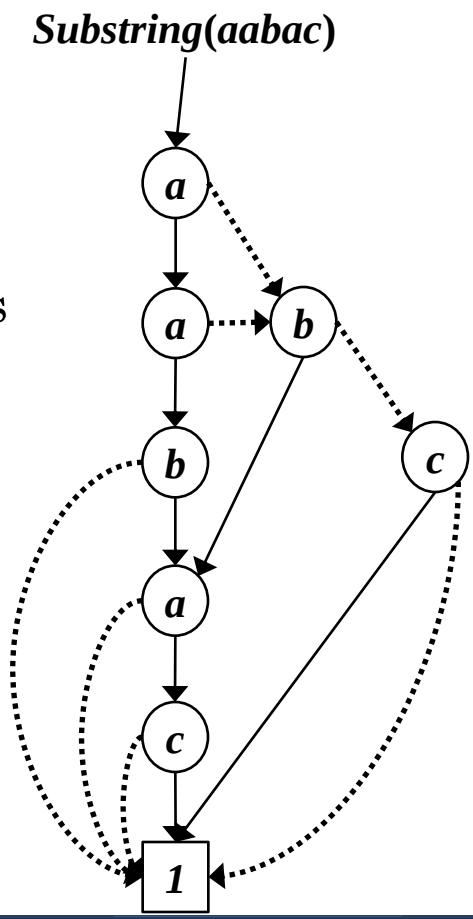
{aabac, aaca, baba, bbac}



(0-terminal node is omitted)

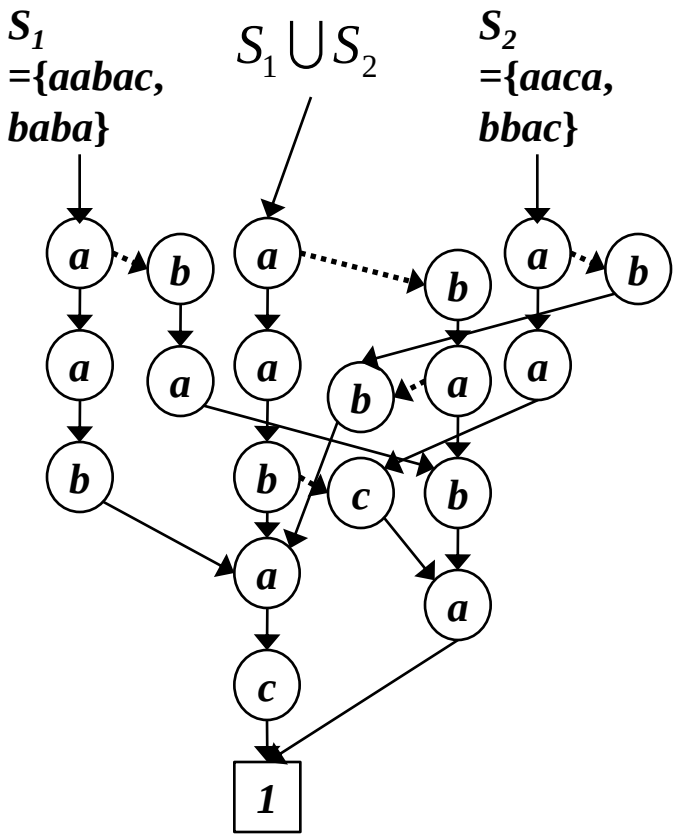
Our proposal: SuffixDD

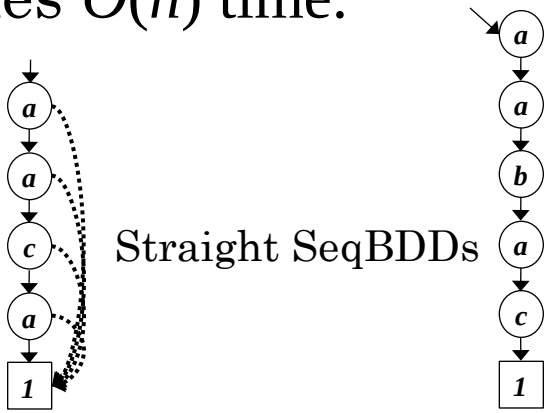
- *Suffix Decision Diagram* (This work)
  - Substring index on SeqBDD
  - Represents the sets of all substrings of a text
  - The number of nodes
    - $n+1$ in the best case
    - $3n-2$ in the worst case
  - The number of edges is twice as mush as nodes



Operations

- Various operations of SeqBDD can be used freely, which are again inherited from ZDD
  - Union(∪)
  - Intersection(∩)
  - Difference(＼)
- For a *n* length straight SeqBDD and any SeqBDD, union operation takes $O(n)$ time.





Experimental Setting

- PC
  - Intel Corei7, 2.67 GHz
  - 3.25 GB main memory
  - 1.5 GB used
  - Windows XP (SP3)
- Implementation
  - Erlang language
    - Version: 5.7.3
    - Distributed mining server
  - Functional language
  - 300 lines


```
-module(sequencebdd).

-export([set_table/0, delete_table/0]).
-export([store/1, clean/1]).
-export([online_build_subword/1, naive_online_build_subword/1]).
-export([union/2, intersect/2, minus/2]).

set_table() ->
    ets:new(uniquetable, [private, named_table]),
    ets:new(nodememory, [private, named_table]),
    ets:insert(uniquetable, {{false, 0, 0}, 0}),
    ets:insert(uniquetable, {{true, 0, 0}, 1}),
    ets:insert(nodememory, {0, {false, 0, 0}}),
    ets:insert(nodememory, {1, {true, 0, 0}}),
    ets:insert(nodememory, {node_number, 1}),
    ets:new(cache, [private, named_table]),
    ok.

delete_table() ->
    ets:delete(uniquetable),
    ets:delete(nodememory),
    ets:delete(cache),
    erlang:garbage_collect(),
    ok.

store({_, 0, N0}) ->
    N0;
store(Node) ->
    Object = ets:lookup(uniquetable, Node),
    case Object of
      [] ->
        Number = ets:update_counter(nodememory, node_number, {2, 1}),
        ets:insert(uniquetable, {Node, Number}),
        ets:insert(nodememory, {Number, Node}),
        Number;
      [{_, Number}] ->
        Number
    end.
```

Exp1: SuffixDD/SeqBDDの圧縮効果

- Calgary Corpusの6つの英文テキストファイル
- すべての部分文字列集合をBDDで構築
- テキスト，SuffixDD，部分文字列集合のサイズを比較

File	File size (B)	SuffixDD size	#Substrings	#letters	Time (ms)
paper1	53,161	102,025	1.41×10^9	2.50×10^{13}	25,323
paper2	82,199	157,398	3.38×10^9	9.26×10^{13}	43,391
paper3	46,526	89,941	1.08×10^9	1.68×10^{13}	22,344
paper4	13,286	26,078	88,196,012	3.91×10^{11}	4,443
paper5	11,954	23,243	71,392,689	2.85×10^{11}	4,297
paper6	38,105	73,989	725,674,256	9.22×10^{12}	16,261
Sum	245,231	472,674	6.76×10^9	1.44×10^{14}	–
Union	–	470,534	6.76×10^9	1.44×10^{14}	2,079
Intersection	–	2,397	5,280	24,409	521

Exp2: 対話的クエリの実行例

- テキストマイニングにおけるアドホック問合せのシナリオ
- ６つのファイルから共通文字列で最長なものを出力したい
- データサイズ240KB. 計算時間：SuffixDD構築＝数十秒．ユニオン＝2秒くらい

```
Eshell V5.7.3  (abort with ^G)
1> seqbdd:set_table().                                  ==> ok
2> S1 = sdd:suffixdd(seq:read("paper1")).   ==>  4379855
3> S2 = sdd:suffixdd(seq:read("paper2")).   ==>  11555546
4> S3 = sdd:suffixdd(seq:read("paper3")).   ==>  15431702
5> S4 = sdd:suffixdd(seq:read("paper4")).   ==>  16299568
6> S5 = sdd:suffixdd(seq:read("paper5")).   ==>  17134036
7> S6 = sdd:suffixdd(seq:read("paper6")).   ==>  20018804
8> I = sdd:intersect(sdd:intersect(S1,S2),S3).   ==>  20042241
9> U = sdd:union(sdd:union(S4,S5),S6).      ==>  20089690
10> D = sdd:difference(I, U).                ==>  20094751
11> sdd:longest(D).  ==>
   "\n.sp2\n.ce4\nDepartment of Computer Science\nThe University
   of Calgary\n2500 University Drive NW\nCalgary, Canada T2N 1N4
   \n.sp2\n."
```