



Title	ZDD Rubyライブラリ
Author(s)	中元, 政一; 羽室, 行信
Citation	2010年度科学技術振興機構ERATO湊離散構造処理系プロジェクト講究録. p.445-446.
Issue Date	2011-06
Doc URL	http://hdl.handle.net/2115/48341
Type	conference presentation
Note	ERATO湊離散構造処理系プロジェクト: 2010年度初冬のワークショップ (ERATO合宿) . 2010年11月29日 (月) ~ 12月1日 (水) . 札幌北広島クラッセホテル.
File Information	15.nakamoto.pdf



[Instructions for use](#)

ZDD Rubyライブラリ

統計・マイニング応用グループ技術員 中元政一
関西学院大学 羽室行信

2011年11月30日

2010年11月30日火曜日

最初に

ZDDを使用した整数値組合せ集合計算プログラム(VSOP)をrubyで使えるようにし、さらに既存のVSOPでは大変だった繰返実行や条件判定を簡単に行えるようにした。

2010年11月30日火曜日

目次

- ZDDのruby拡張パッケージ
 - 動作環境
 - 実装の概要
 - 利用例

2010年11月30日火曜日

動作環境

- mac or linux(fedora8 ,debian 5)
- ruby 1.8.6 or 1.8.7 (1.9以降は未確認)
- rubygems 1.3.5

ソースから作成する場合は以下も必要

- ruby-devel

2010年11月30日火曜日

実装の概要1

- Ruby拡張ライブラリとして実装する。
 - zdd_so.bundle:
拡張ライブラリ本体 (拡張子がMacでは.bundle,linuxでは.so)
 - zdd.rb:
文字列の"*"演算子専用オーバーライドメソッド。rubyスクリプトから requireされる。
- オリジナルのソースは、基本的には手を付けず、置換プログラムで新たにソースを生成している。
- パッケージはrubygemsを利用。

2010年11月30日火曜日

実装概要2

- ZDD Classオブジェクトの初期化は"文字列".v,数値.vにより行う。
- アイテム(シンボル)名として任意の文字列を指定できる
- アイテムの宣言はZDD.symbolメソッドでおこなう。
- ZDDオブジェクトに対する演算において、Ruby整数や Ruby文字列は自動的にキャストされる。
- オリジナルのVSOPで実装されているメソッドはほとんど利用可能
 - Plotメソッドは利用できない。
 - 追加メソッド: csvout,hashout, same(==).diff,each, delta

2010年11月30日火曜日

利用例1

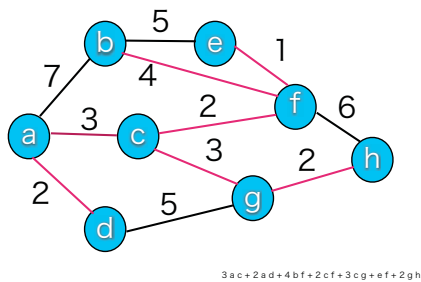
Rubyソース	解説	VSOP記述
<code>#!/usr/bin/env ruby</code>	シェルにインタプリタを教える	
<code>require 'rubygems'</code> <code>require 'zdd'</code>	gemパッケージ管理ライブラリのインクルード	
<code>ZDD.symbol("a")</code>		symbol a
<code>ZDD.symbol("b")</code>	"a","b","c"という名前のアイテム(シンボル)の宣言	symbol b
<code>ZDD.symbol("c")</code>		symbol c
<code>a="a".v</code>	"a"というアイテム集合のZDDを生成しaというruby変数に格納	A = a
<code>ab="a".v * "b".v</code>	"a b"という2アイテム集合のZDDオブジェクト。	AB = a b
<code>b="b".v</code>	"b"というアイテム集合のZDDオブジェクト	B = b
<code>f=(a+1)*("b+1")*ab</code>	ZDD演算の結果をfというruby変数に格納する	F = (A+1)*(B+1)*AB
<code>f.Show</code>	横和形式で標準出力に出力	print F
<code>4 a b</code>	結果	

2010年11月30日火曜日

利用例2(追加メソッドで)

最小連結木を求めるプログラム	
<pre> rubyソース require 'rubygems' require 'zdd' v = "a".v + "b".v + "c".v + "d".v + "e".v + "f".v + "g".v + "h".v e = "a".v*"b".v + "a".v*"c".v + "b".v*"d".v + "b".v*"e".v + "b".v*"f".v + "b".v*"g".v + "c".v*"d".v + "c".v*"e".v + "c".v*"f".v + "f".v*"h".v + "g".v*"h".v r, e = 0 r.v = "a".v while 1 i1 = e.restrict(r.v) while 1 i2 = i1.termEQ(i1.minval) break if i2==0 i2.each{ x @r_e_tmp = r, e + x @r_v_tmp = (@r_e_tmp.items > 0) break if r.v.dff7(@r_v_tmp) } break if r.v.dff7(@r_v_tmp) i1 = i1-i2 end r, e = @r_e_tmp r.v = @r_v_tmp break if r.v==v end e.show </pre>	<p>別図のノードを表すZDD(a+b+c+d+f+g+h)</p> <p>別図のエッジ(重み)を表すZDD $(7ab + 3ac + 2ad + 5be + 4bf + 3cg + 2cf + 5dg + ef + 6fh + 2gh)$</p> <p>連結可能なエッジを見つける</p> <p>重みが最小である新たなエッジを見つける</p> <p>新たなエッジを追加して ノードの候補を調べる ノードが追加されれば、連結可能最小</p> <p>今回の調べたものは条件を満たさないののですてもう一度</p> <p>すべてのノードが含められれば終了</p>

2010年11月30日火曜日



2010年11月30日火曜日

最後に

今後欲しい機能

- ・データ読み込み機能
- ・ZDDのexportにsymbol名を含めたい

詳細マニュアル

http://www-erato.ist.hokudai.ac.jp/~kansai/vsop_ruby/doc/

2010年11月30日火曜日