Compact-Diagram, a CP propagator for MDDs and Diagrams using bitsets

DDOPT2024

Hélène Verhaeghe

22 November 2024

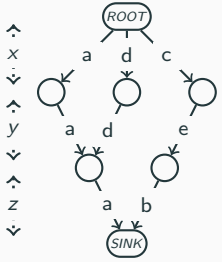
UCLouvain, Louvain-la-Neuve, Belgium
helene.verhaeghe@uclouvain.be

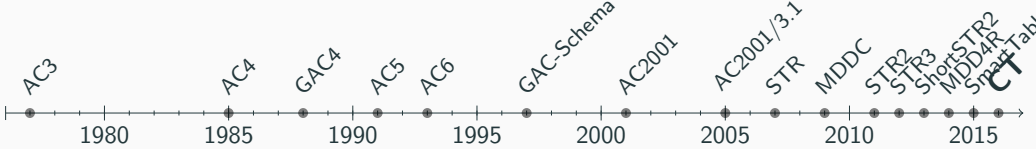
UCLouvain

	x	y	z
τ_1	a	a	a
τ_2	d	d	a
τ_3	c	e	b
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Tables are the oldest most used CP constraints

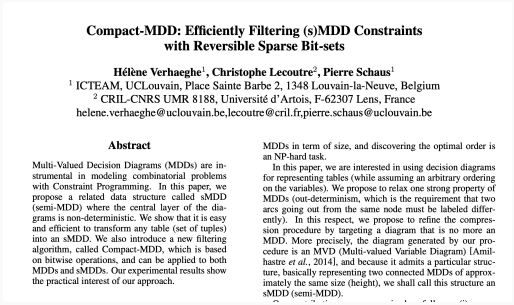
MDDs are equivalent to tables





2016 : New algorithm! Compact-Table [CP2016], based on bitwise operations, completely outperformed existing algorithms

UCLouvain



Compact-MDD: Efficiently Filtering (s)MDD Constraints with Reversible Sparse Bit-sets

Hélène Verhaeghe[1], **Christophe Lecoutre**[2], **Pierre Schaus**[1]
[1] ICTEAM, UCLouvain, Place Sainte Barbe 2, 1348 Louvain-la-Neuve, Belgium
[2] CRIL-CNRS UMR 8188, Université d'Artois, F-62307 Lens, France
helene.verhaeghe@uclouvain.be,lecoutre@cril.fr,pierre.schaus@uclouvain.be

Abstract

Multi-Valued Decision Diagrams (MDDs) are instrumental in modeling combinatorial problems with Constraint Programming. In this paper, we propose a related data structure called sMDD (semi-MDD) where the central layer of the diagrams is non-deterministic. We show that it is easy and efficient to transform any table (set of tuples) into an sMDD. We also introduce a new filtering algorithm, called Compact-MDD, which is based on bitwise operations, and can be applied to both MDDs and sMDDs. Our experimental results show the practical interest of our approach.

MDDs in term of size, and discovering the optimal order is an NP-hard task.

In this paper, we are interested in using decision diagrams for representing tables (while assuming an arbitrary ordering on the variables). We propose to relax one strong property of MDDs (out-determinism, which is the requirement that two arcs going out from the same node must be labeled differently). In this respect, we propose to refine the compression procedure by targeting a diagram that is no more an MDD. More precisely, the diagram generated by our procedure is an MVD (Multi-valued Variable Diagram) [Amilhastre *et al.*, 2014], and because it admits a particular structure, basically representing two connected MDDs of approximately the same size (height), we shall call this structure an sMDD (semi-MDD).

Extending Compact-Diagram to Basic Smart Multi-Valued Variable Diagrams

Hélène Verhaeghe[1], Christophe Lecoutre[2], and Pierre Schaus[1]

[1] UCLouvain, ICTEAM, Place Sainte Barbe 2, 1348 Louvain-la-Neuve, Belgium, {firstname.lastname}@uclouvain.be
[2] CRIL-CNRS UMR 8188, Université d'Artois, F-62307 Lens, France, lecoutre@cril.fr

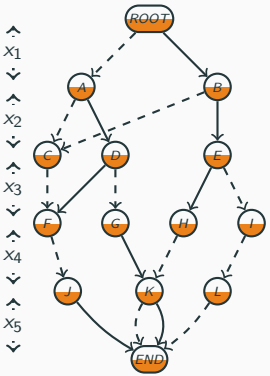
Abstract. Multi-Valued Decision Diagrams (MDDs), and more generally Multi-Valued Variable Diagrams (MVDs), are instrumental in modeling constrained combinatorial problems. This has led to a number of algorithms for filtering constraints such as mddc, MDD4R and CD (Compact-Diagram). Many compressed forms of tables have also been proposed over the years, leading to a 'smart' hybridization between extensional an intentional representations, which was obtained by embedding simple arithmetic constraints in tuples (of tables). Interestingly, the state-of-the-art algorithm CT (Compact-Table) has been recently extended to deal efficiently with *bs*-tables, i.e., 'basic smart' tables containing expressions of the form '$*$', '$\neq v$', '$\leq v$', '$\geq v$' and '$\in S$'. In this paper, we introduce the concept of *bs*-MVDs by enabling arcs of diagrams to be labelled with similar expressions. We show how such diagrams can be naturally derived from ordinary tables and MDDs, and we extend the state-of-the-art algorithm CD in order to handle *bs*-MVDs (and *bs*-MDDs).

Types of diagrams

Types of diagrams: MDD, ...

UCLouvain

MDD **sMDD** **MVD**



in-nd & out-nd in-nd & out-d in-d & out-nd

From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
	1	0	1	1
	1	1	0	1
1	0	0	0	1
	1	0	0	0
	1	1	0	0
	1	1	0	1

From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
		0	1	1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
		1	0	1

From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
			1	1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
			0	1

From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

<table>
<tr><th>x_1</th><th>x_2</th><th>x_3</th><th>x_4</th><th>x_5</th></tr>
<tr><td rowspan="4">0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td rowspan="3">1</td><td rowspan="2">0</td><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td rowspan="4">1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">0</td><td>0</td></tr>
<tr><td>1</td></tr>
</table>

From table to MDD

UCLouvain

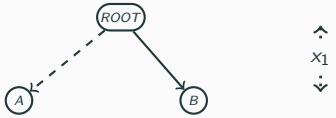
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1

From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

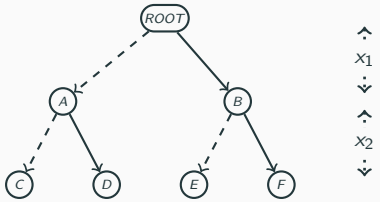
<table>
<tr><th>x_1</th><th>x_2</th><th>x_3</th><th>x_4</th><th>x_5</th></tr>
<tr><td rowspan="4">0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td rowspan="3">1</td><td rowspan="2">0</td><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td rowspan="4">1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">0</td><td>0</td></tr>
<tr><td>1</td></tr>
</table>



From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

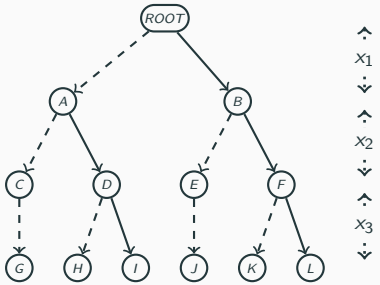
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

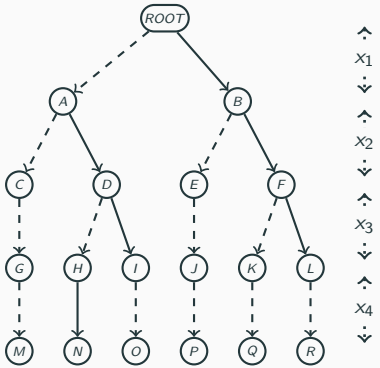
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

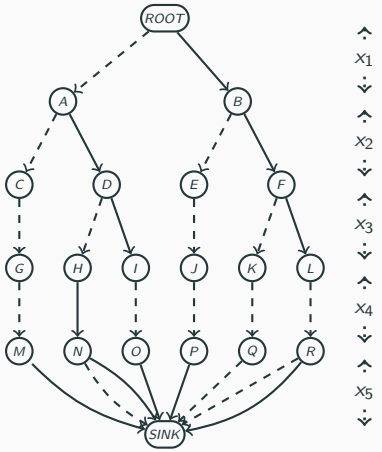
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



UCLouvain

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

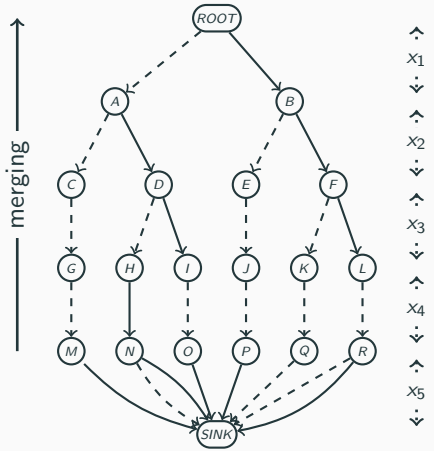
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



UCLouvain

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

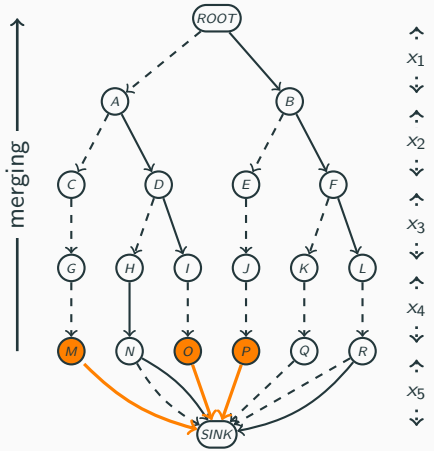
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



UCLouvain

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

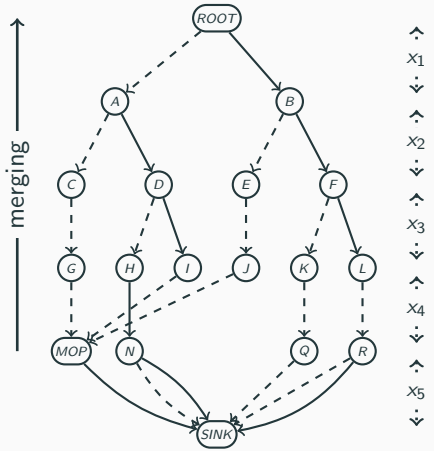
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

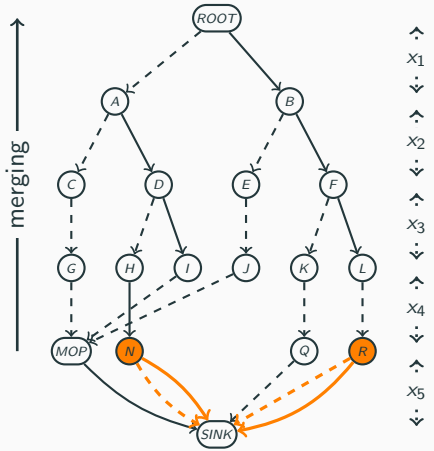
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



UCLouvain

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

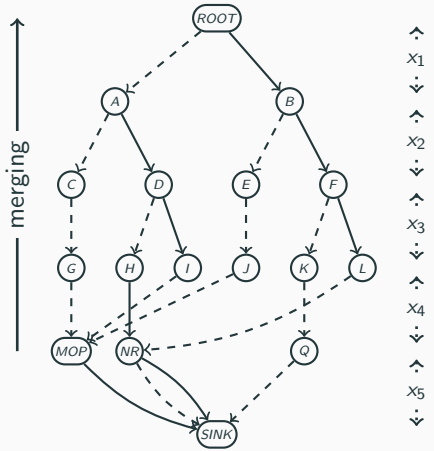
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



UCLouvain

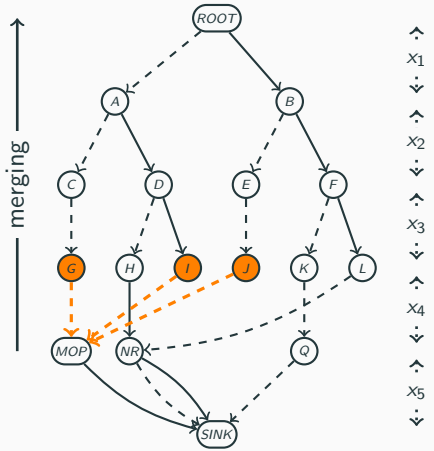
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

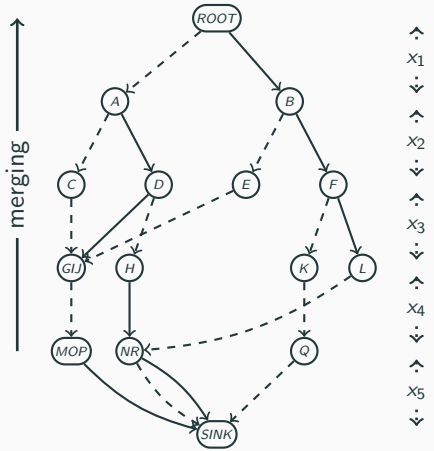
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

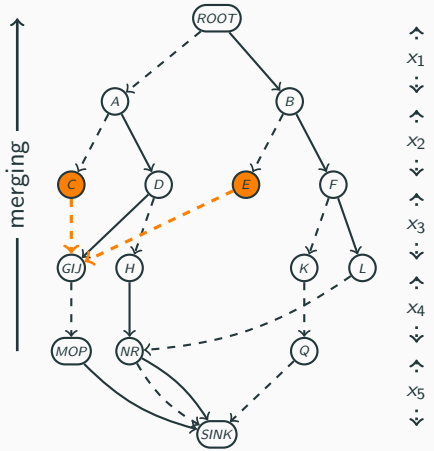
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

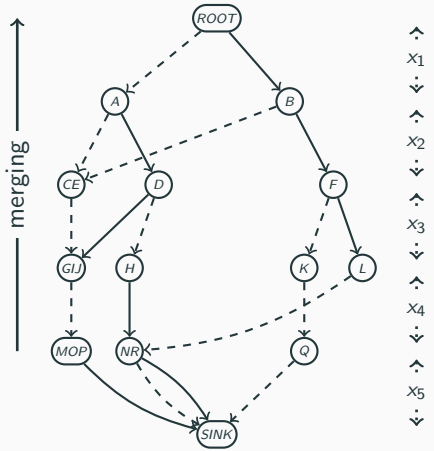
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



From table to MDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

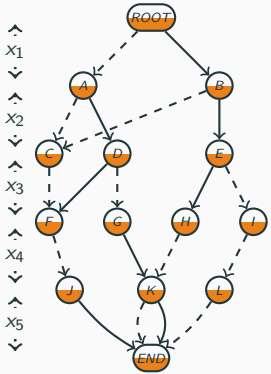
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
	1	0	1	0
				1
		1	0	1
1	0	0	0	1
	1	0	0	0
		1	0	0
				1



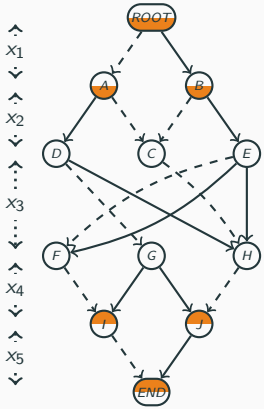
Types of diagrams: MDD, sMDD,...

UCLouvain

MDD



sMDD



MVD

in-nd & out-nd in-nd & out-d in-d & out-nd

From table to sMDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

From table to sMDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
0	1
1	0
1	1

From table to sMDD

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
0	1
1	0
1	1

suffixes

x_4	x_5
0	0
1	0
0	1
1	1

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	0
0	1
1	1

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	

UCLouvain

x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	

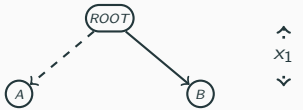
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	



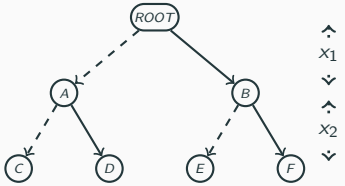
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	



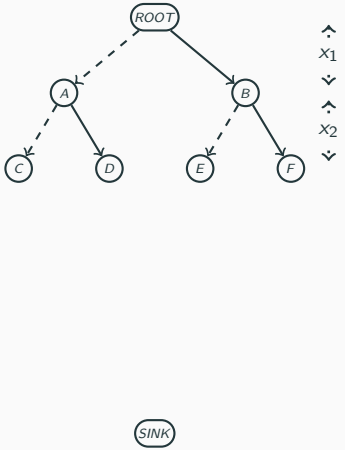
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	



UCLouvain

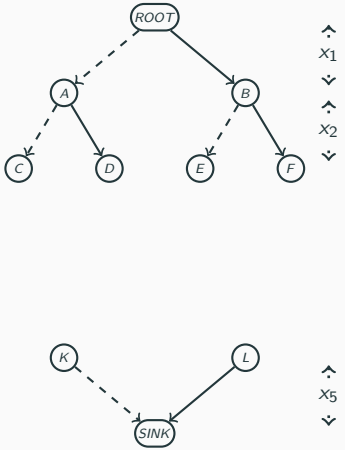
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	



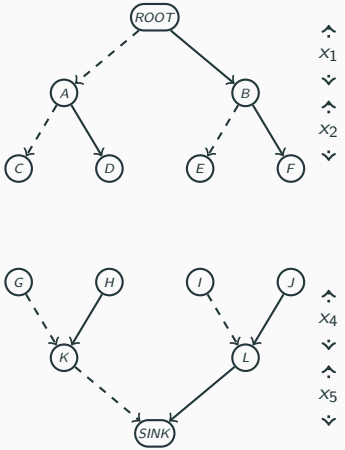
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	



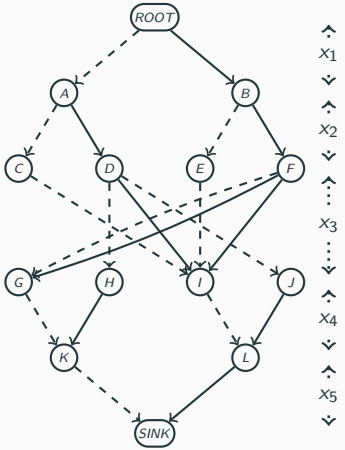
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	



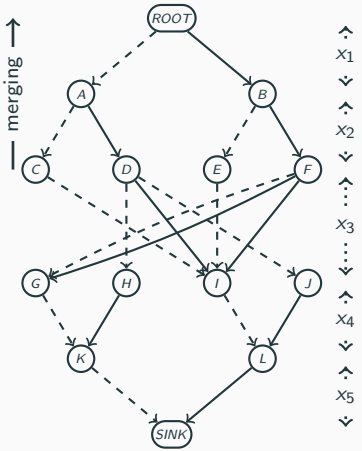
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	



UCLouvain

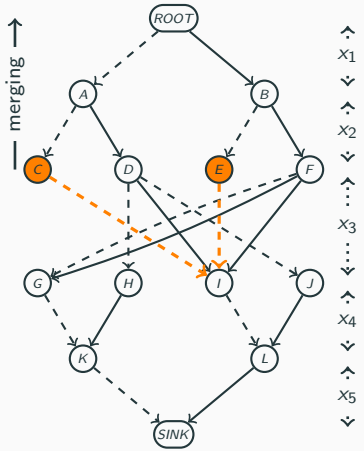
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	



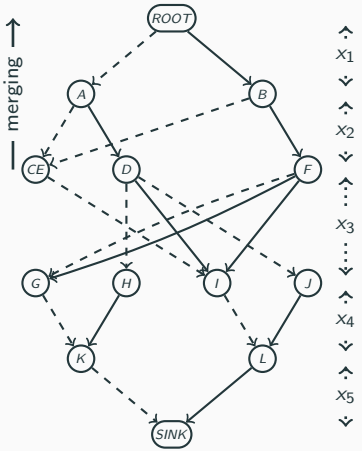
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	



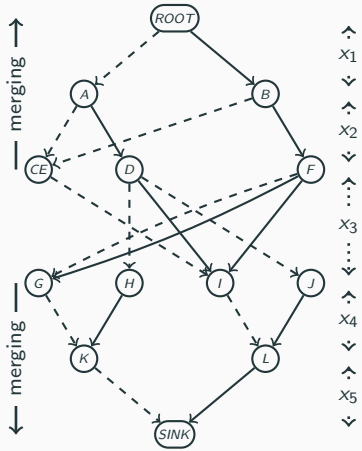
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	



UCLouvain

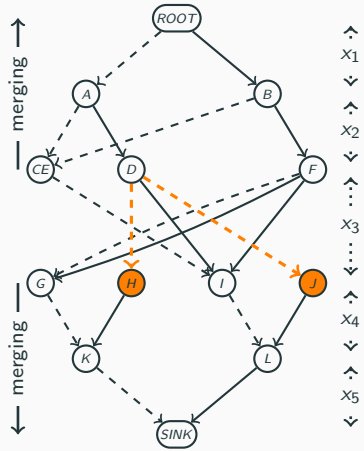
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

suffixes

x_4	x_5
0	0
1	
0	1
1	



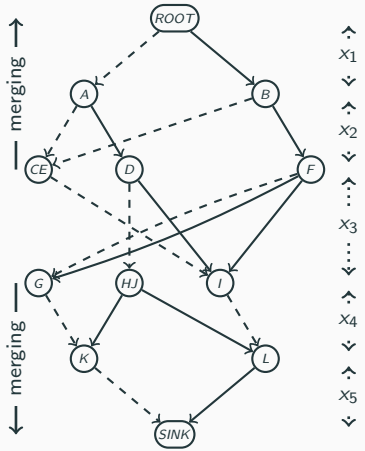
x_1	x_2	x_3	x_4	x_5
0	0	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
1	1	1	0	0
1	1	1	0	1

prefixes

x_1	x_2
0	0
	1
1	0
	1

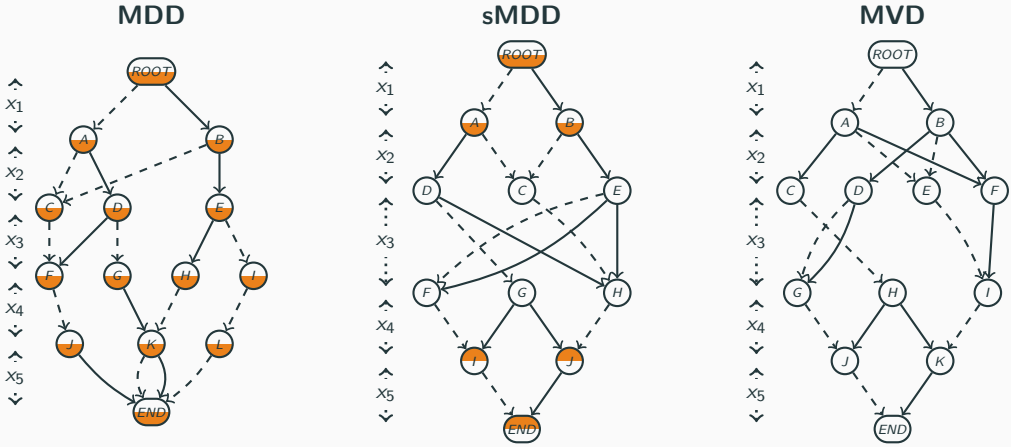
suffixes

x_4	x_5
0	0
1	
0	1
1	

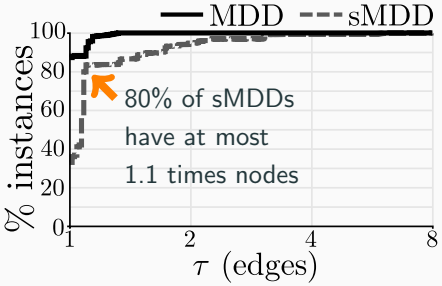


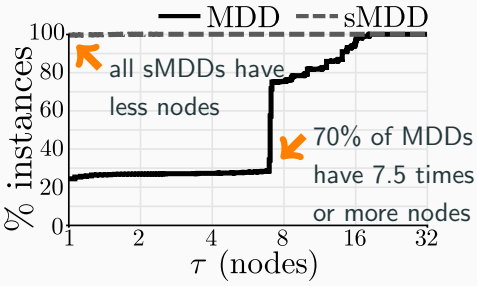
MDD, sMDD and MVD

UCLouvain



Comparison between MDD and sMDD





sMDDs: **Less few** nodes but **a bit more** edges than MDDs

The Compact-MDD algorithm

UCLouvain

	x	y	z
τ_1	0	1	0
τ_2	0	1	1
τ_3	1	0	2
τ_4	2	1	0

- Step 1: update
  - incremental from removed values
  - reset from remaining values
- Step 2: filtering

τ_1	τ_2	τ_3	τ_4
1	1	1	1

supports	τ_1	τ_2	τ_3	τ_4	
(x,0)	✔	1	1	0	0
(x,1)	✔	0	0	1	0
(x,2)	✔	0	0	0	1
(y,0)	✔	0	0	1	0
(y,1)	✔	1	1	0	1
(y,2)	✔	0	0	0	0
(z,0)	✔	1	0	0	1
(z,1)	✔	0	1	0	0
(z,2)	✔	0	0	1	0

	x	y	z
τ_1	0	1	0
τ_2	0	1	1
τ_3	1	0	2
τ_4	2	1	0

- Step 1: update
  - incremental from removed values
  - reset from remaining values
- Step 2: filtering

τ_1	τ_2	τ_3	τ_4
1	1	1	1

supports	τ_1	τ_2	τ_3	τ_4	
(x,0)	✔	1	1	0	0
(x,1)	✗	0	0	1	0
(x,2)	✔	0	0	0	1
(y,0)	✔	0	0	1	0
(y,1)	✔	1	1	0	1
(y,2)	✔	0	0	0	0
(z,0)	✔	1	0	0	1
(z,1)	✔	0	1	0	0
(z,2)	✔	0	0	1	0

	x	y	z
τ_1	0	1	0
τ_2	0	1	1
τ_3	1	0	2
τ_4	2	1	0

- Step 1: update
  - incremental from removed values
  - reset from remaining values
- Step 2: filtering

τ_1	τ_2	τ_3	τ_4
1	1	0	1

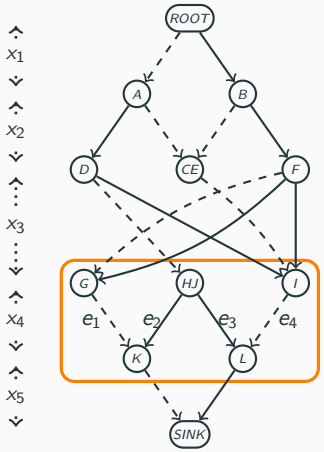
supports	τ_1	τ_2	τ_3	τ_4	
(x,0)	✔	1	1	0	0
(x,1)	✗	0	0	1	0
(x,2)	✔	0	0	0	1
(y,0)	✔	0	0	1	0
(y,1)	✔	1	1	0	1
(y,2)	✔	0	0	0	0
(z,0)	✔	1	0	0	1
(z,1)	✔	0	1	0	0
(z,2)	✔	0	0	1	0

	x	y	z
τ_1	0	1	0
τ_2	0	1	1
τ_3	1	0	2
τ_4	2	1	0

- Step 1: update
  - incremental from removed values
  - reset from remaining values
- Step 2: filtering

τ_1	τ_2	τ_3	τ_4
1	1	0	1

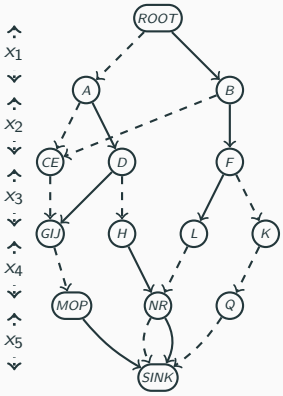
supports	τ_1	τ_2	τ_3	τ_4	
(x,0)	✔	1	1	0	0
(x,1)	✗	0	0	1	0
(x,2)	✔	0	0	0	1
(y,0)	✔	0	0	1	0
(y,1)	✔	1	1	0	1
(y,2)	✔	0	0	0	0
(z,0)	✔	1	0	0	1
(z,1)	✔	0	1	0	0
(z,2)	✗	0	0	1	0



Name	Set	Bit-set
currArcs[x_4]	$\{e_1, e_2, e_3, e_4\}$	[1 1 1 1]
supports[x_4,0]	$\{e_1, \cancel{e_2}, \cancel{e_3}, e_4\}$	[1 0 0 1]
arcsT[HJ,x_4]	$\{\cancel{e_1}, e_2, e_3, \cancel{e_4}\}$	[0 1 0 0]
arcsH[x_4,K]	$\{e_1, e_2, \cancel{e_3}, \cancel{e_4}\}$	[1 1 0 0]

Compact-Diagram: Update

Compact-Diagram: Update



MDD

currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 1 1 1 1]

currArcs[x_4]
[1 1 1 1]

currArcs[x_5]
[1 1 1 1]

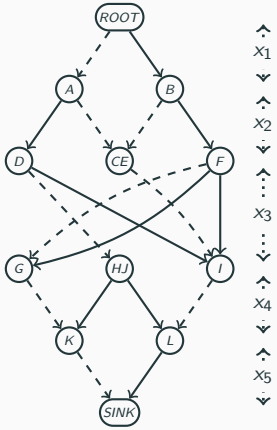
currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 1 1 1 1 1]

currArcs[x_4]
[1 1 1 1]

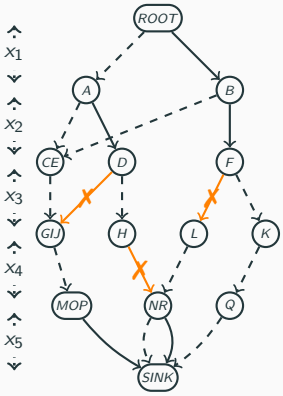
currArcs[x_5]
[1 1]



sMDD

Compact-Diagram: Update

UCLouvain



MDD

currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 1 1 1 1]

currArcs[x_4]
[1 1 1 1]

currArcs[x_5]
[1 1 1 1]

currArcs[x_1]
[1 1]

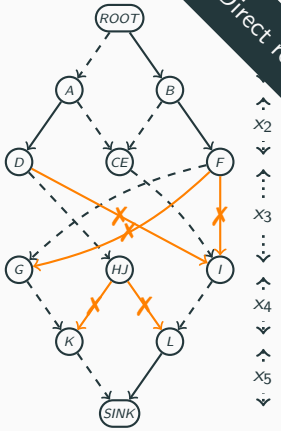
currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 1 1 1 1 1]

currArcs[x_4]
[1 1 1 1]

currArcs[x_5]
[1 1]

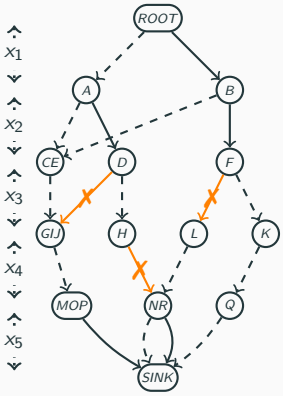
Direct removal



sMDD

Compact-Diagram: Update

UCLouvain



MDD

$\texttt{currArcs}[x_1]$ [1 1]

$\texttt{currArcs}[x_2]$ [1 1 1 1]

$\texttt{currArcs}[x_3]$ [1 **0** 1 **0** 1]

$\texttt{currArcs}[x_4]$ [1 **0** 1 1]

$\texttt{currArcs}[x_5]$ [1 1 1 1]

$\texttt{currArcs}[x_1]$ [1 1]

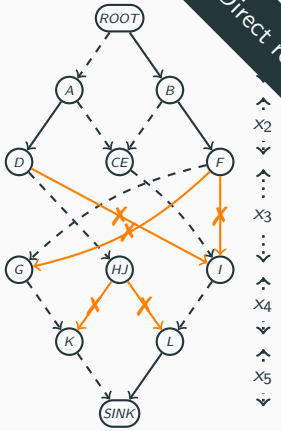
$\texttt{currArcs}[x_2]$ [1 1 1 1]

$\texttt{currArcs}[x_3]$ [1 **0** 1 1 **0 0**]

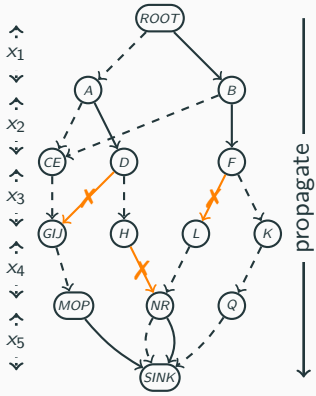
$\texttt{currArcs}[x_4]$ [1 **0 0** 1]

$\texttt{currArcs}[x_5]$ [1 1]

Direct removal



sMDD



propagate

currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 1 1]

currArcs[x_5]
[1 1 1 1]

MDD

currArcs[x_1]
[1 1]

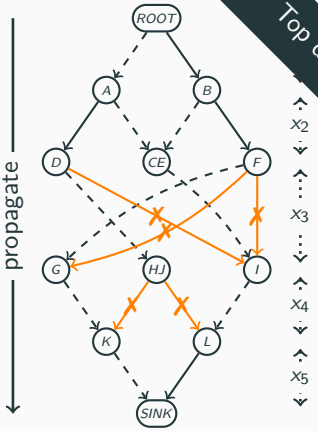
currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 0 1 1 0 0]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 1]

propagate

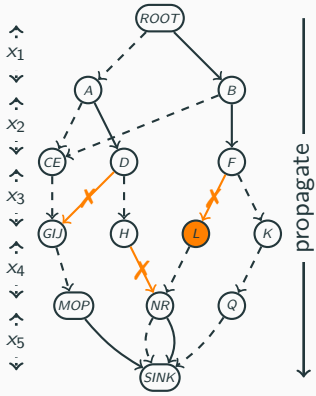


Top down

sMDD

Compact-Diagram: Update

UCLouvain



propagate

currArcs[x_1] [1 1]

currArcs[x_2] [1 1 1 1]

currArcs[x_3] [1 0 1 0 1]

currArcs[x_4] [1 0 1 1]

currArcs[x_5] [1 1 1 1]

MDD

currArcs[x_1] [1 1]

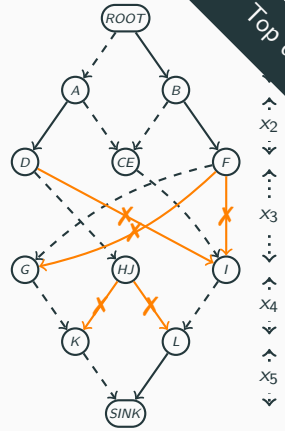
currArcs[x_2] [1 1 1 1]

currArcs[x_3] [1 0 1 1 0 0]

currArcs[x_4] [1 0 0 1]

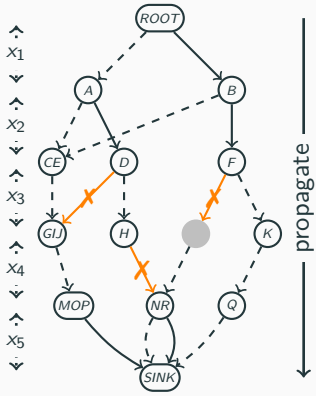
currArcs[x_5] [1 1]

propagate



Top down

sMDD



propagate

currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 1 1]

currArcs[x_5]
[1 1 1 1]

MDD

currArcs[x_1]
[1 1]

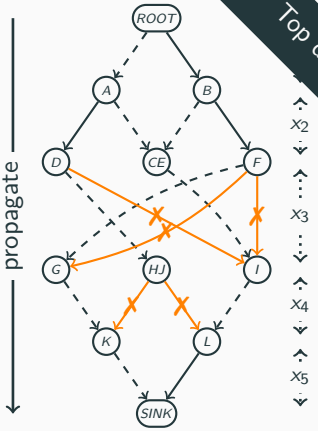
currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 0 1 1 0 0]

currArcs[x_4]
[1 0 0 1]

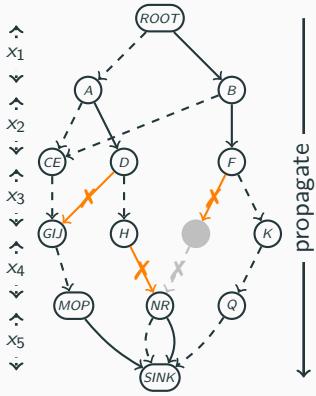
currArcs[x_5]
[1 1]

propagate



Top down

sMDD



propagate

currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 **0** 1]

currArcs[x_5]
[1 1 1 1]

MDD

currArcs[x_1]
[1 1]

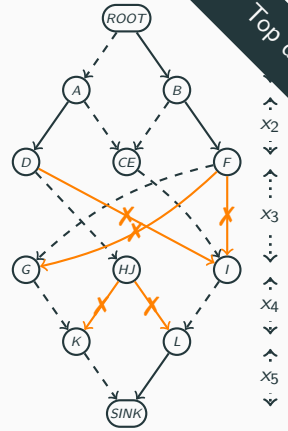
currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 0 1 1 0 0]

currArcs[x_4]
[1 0 0 1]

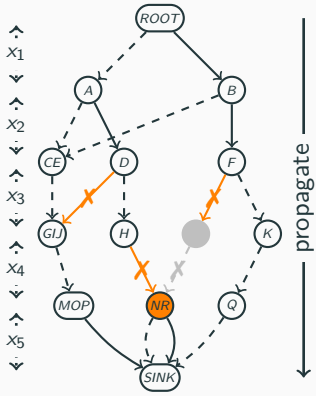
currArcs[x_5]
[1 1]

propagate



Top down

sMDD



propagate

currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 1 1 1]

MDD

currArcs[x_1]
[1 1]

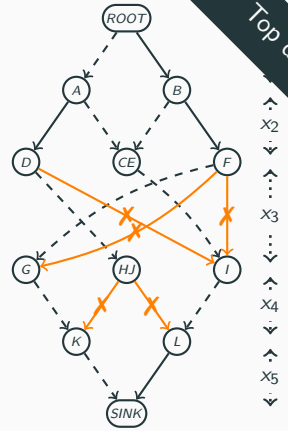
currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 0 1 1 0 0]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 1]

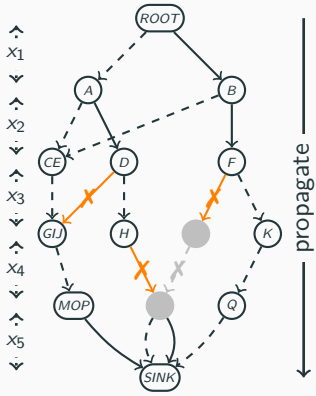
propagate



Top down

sMDD

UCLouvain



propagate

currArcs[x_1] [1 1]

currArcs[x_2] [1 1 1 1]

currArcs[x_3] [1 0 1 0 1]

currArcs[x_4] [1 0 0 1]

currArcs[x_5] [1 1 1 1]

currArcs[x_1] [1 1]

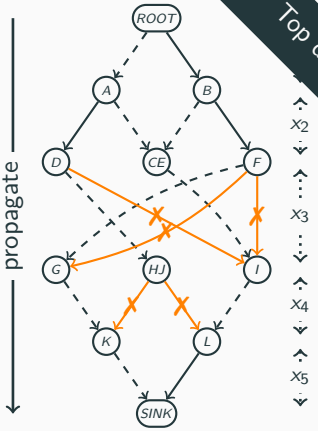
currArcs[x_2] [1 1 1 1]

currArcs[x_3] [1 0 1 1 0 0]

currArcs[x_4] [1 0 0 1]

currArcs[x_5] [1 1]

propagate

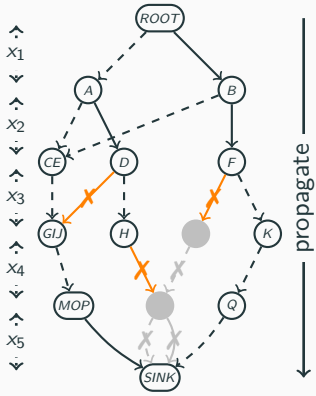


Top down

MDD

sMDD

UCLouvain



propagate

currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 **0 0** 1]

currArcs[x_1]
[1 1]

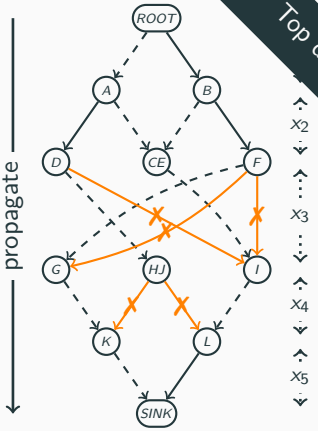
currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 0 1 1 0 0]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 1]

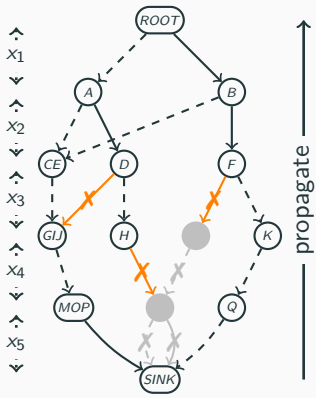
propagate



Top down

MDD

sMDD



propagate

currArcs[x_1] [1 1]

currArcs[x_2] [1 1 1 1]

currArcs[x_3] [1 0 1 0 1]

currArcs[x_4] [1 0 0 1]

currArcs[x_5] [1 0 0 1]

currArcs[x_1] [1 1]

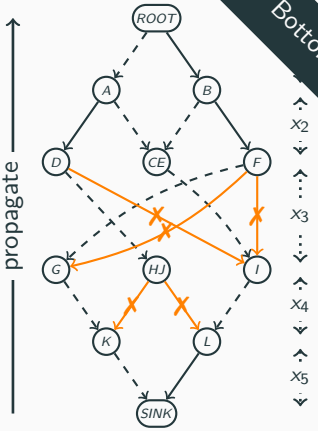
currArcs[x_2] [1 1 1 1]

currArcs[x_3] [1 0 1 1 0 0]

currArcs[x_4] [1 0 0 1]

currArcs[x_5] [1 1]

propagate



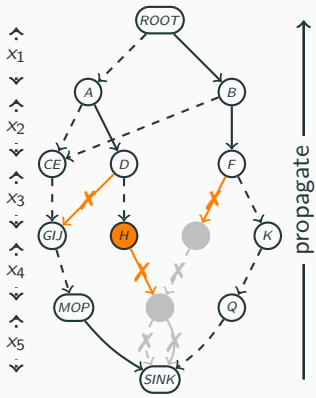
Bottom up

MDD

sMDD

Compact-Diagram: Update

UCLouvain



propagate

currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 0 0 1]

MDD

currArcs[x_1]
[1 1]

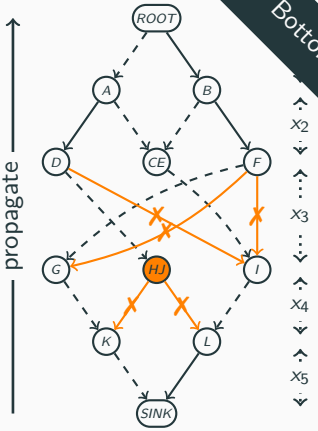
currArcs[x_2]
[1 1 1 1]

currArcs[x_3]
[1 0 1 1 0 0]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 1]

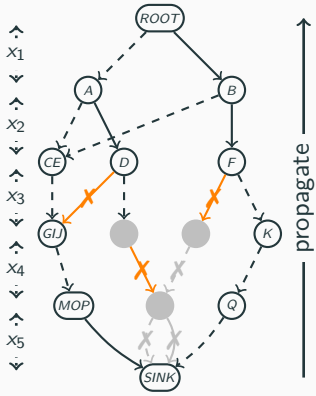
propagate



Bottom up

sMDD

UCLouvain



propagate

currArcs[x_1] [1 1]

currArcs[x_2] [1 1 1 1]

currArcs[x_3] [1 0 1 0 1]

currArcs[x_4] [1 0 0 1]

currArcs[x_5] [1 0 0 1]

currArcs[x_1] [1 1]

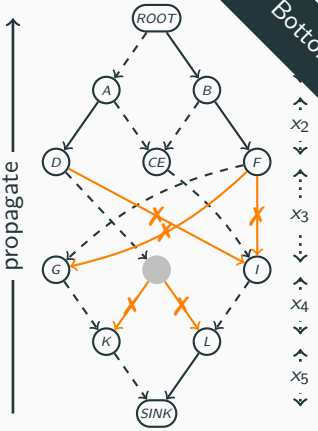
currArcs[x_2] [1 1 1 1]

currArcs[x_3] [1 0 1 1 0 0]

currArcs[x_4] [1 0 0 1]

currArcs[x_5] [1 1]

propagate

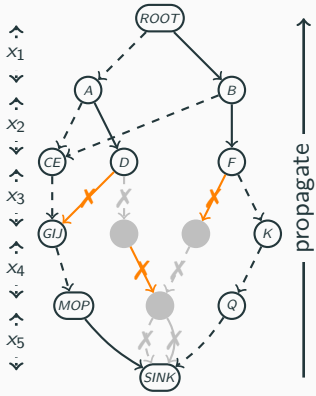


Bottom up

MDD

sMDD

UCLouvain



currArcs[x_1] [1 1]

currArcs[x_2] [1 1 1 1]

currArcs[x_3] [1 0 **0** 0 1]

currArcs[x_4] [1 0 0 1]

currArcs[x_5] [1 0 0 1]

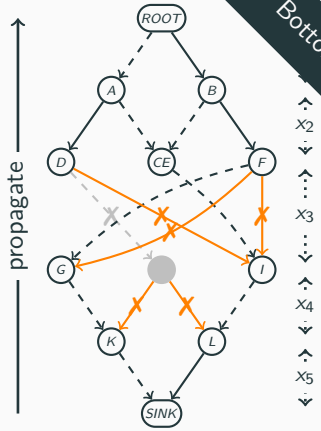
currArcs[x_1] [1 1]

currArcs[x_2] [1 1 1 1]

currArcs[x_3] [**0** 0 1 1 0 0]

currArcs[x_4] [1 0 0 1]

currArcs[x_5] [1 1]



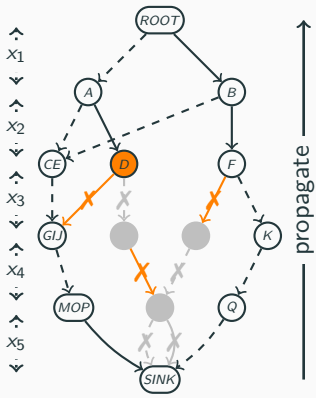
Bottom up

MDD

sMDD

Compact-Diagram: Update

UCLouvain



$\texttt{currArcs}[x_1]$ [1 1]

$\texttt{currArcs}[x_2]$ [1 1 1 1]

$\texttt{currArcs}[x_3]$ [1 0 0 0 1]

$\texttt{currArcs}[x_4]$ [1 0 0 1]

$\texttt{currArcs}[x_5]$ [1 0 0 1]

MDD

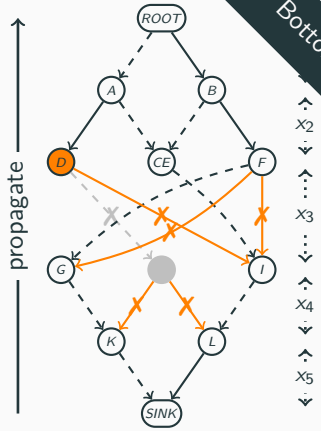
$\texttt{currArcs}[x_1]$ [1 1]

$\texttt{currArcs}[x_2]$ [1 1 1 1]

$\texttt{currArcs}[x_3]$ [0 0 1 1 0 0]

$\texttt{currArcs}[x_4]$ [1 0 0 1]

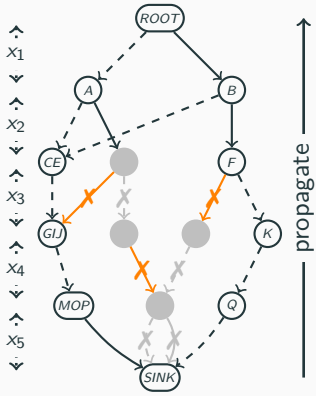
$\texttt{currArcs}[x_5]$ [1 1]



sMDD

Bottom up

UCLouvain



$\texttt{currArcs}[x_1]$ [1 1]

$\texttt{currArcs}[x_2]$ [1 1 1 1]

$\texttt{currArcs}[x_3]$ [1 0 0 0 1]

$\texttt{currArcs}[x_4]$ [1 0 0 1]

$\texttt{currArcs}[x_5]$ [1 0 0 1]

MDD

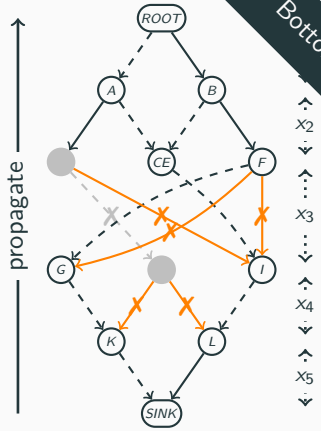
$\texttt{currArcs}[x_1]$ [1 1]

$\texttt{currArcs}[x_2]$ [1 1 1 1]

$\texttt{currArcs}[x_3]$ [0 0 1 1 0 0]

$\texttt{currArcs}[x_4]$ [1 0 0 1]

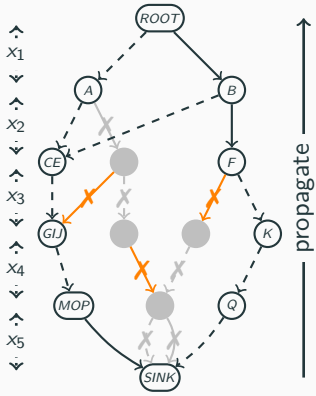
$\texttt{currArcs}[x_5]$ [1 1]



Bottom up

sMDD

UCLouvain



$\texttt{currArcs}[x_1]$ [1 1]

$\texttt{currArcs}[x_2]$ [1 **0** 1 1]

$\texttt{currArcs}[x_3]$ [1 0 0 0 1]

$\texttt{currArcs}[x_4]$ [1 0 0 1]

$\texttt{currArcs}[x_5]$ [1 0 0 1]

MDD

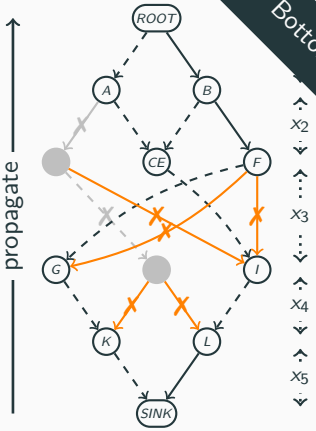
$\texttt{currArcs}[x_1]$ [1 1]

$\texttt{currArcs}[x_2]$ [**0** 1 1 1]

$\texttt{currArcs}[x_3]$ [0 0 1 1 0 0]

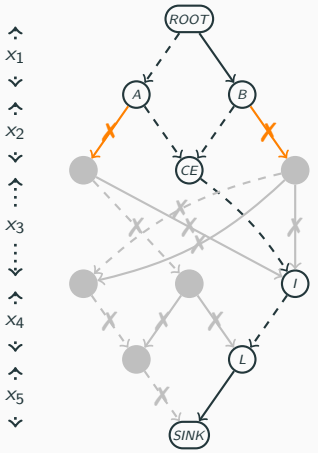
$\texttt{currArcs}[x_4]$ [1 0 0 1]

$\texttt{currArcs}[x_5]$ [1 1]



sMDD

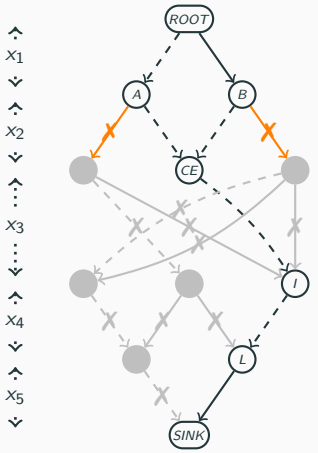
Ccompact-Diagram: Filtering



$\Delta_{x_2} = \{1\}$

x_1	x_2	x_3	x_4	x_5
$\{0,1\}$	$\{0\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$

(x,v)	currArcs[x]	supports[x,v]	$\cap$
$(x_1,0)$	11	10	10
$(x_1,1)$	11	01	01
$(x_3,0)$	001000	101100	001000
$(x_3,1)$	001000	010011	000000
$(x_4,0)$	0001	1001	0001
$(x_4,1)$	0001	0110	0000
$(x_5,0)$	01	10	00
$(x_5,1)$	01	01	01



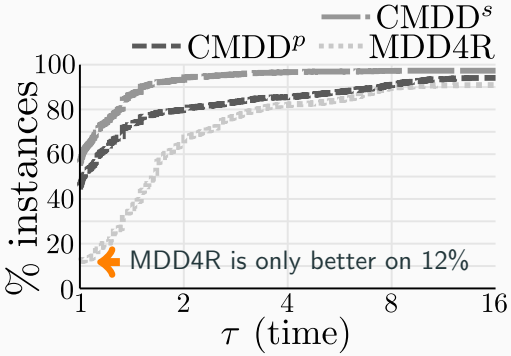
$\Delta_{x_2} = \{1\}$

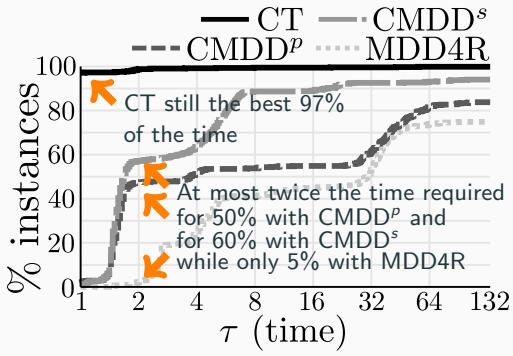
x_1	x_2	x_3	x_4	x_5
$\{0,1\}$	$\{0\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$

(x,v)	currArcs[x]	supports[x,v]	$\cap$
$(x_1,0)$	11	10	10
$(x_1,1)$	11	01	01
$(x_3,0)$	001000	101100	001000
$(x_3,1)$	001000	010011	000000
$(x_4,0)$	0001	1001	0001
$(x_4,1)$	0001	0110	0000
$(x_5,0)$	01	10	00
$(x_5,1)$	01	01	01

Ccompact-Diagram: Results

Results (4000 instances, 10 min T.O., 10 Go)



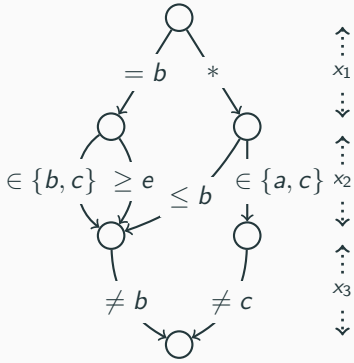


CMDD more efficiant than MDD4R

Reduction of gap between table contraint (CT) and layered graph

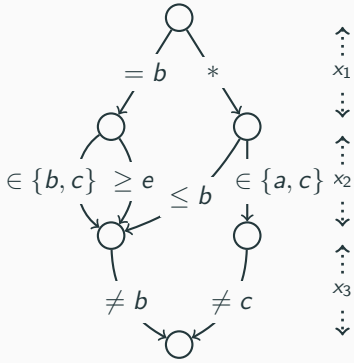
Basic Smart Diagrams

A basic smart MVD



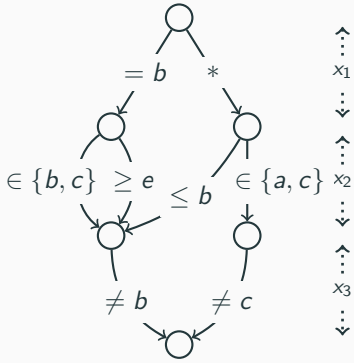
A basic smart MVD labeled with Smart Elements representing multiples values

a b c d e f



Basic Smart MVD

A basic smart MVD



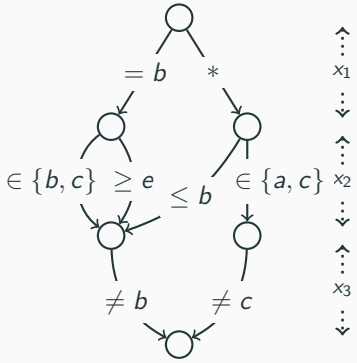
labeled with Smart Elements

single value: $= e$

representing multiples values

a	b	c	d	e	f
✗	✗	✗	✗	✔	✗

A basic smart MVD



labeled with Smart Elements

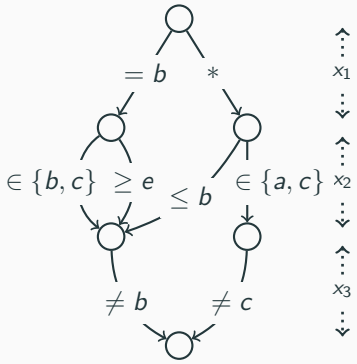
single value: $= e$

universal value: $*$

representing multiples values

a	b	c	d	e	f
✗	✗	✗	✗	✔	✗
✔	✔	✔	✔	✔	✔

A basic smart MVD



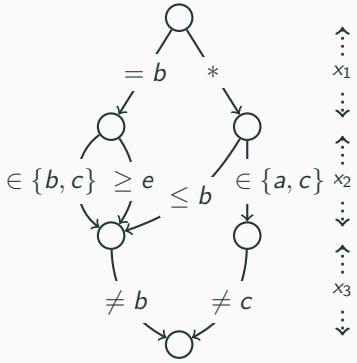
labeled with Smart Elements

representing multiples values

	a	b	c	d	e	f
single value: $= e$	✗	✗	✗	✗	✔	✗
universal value: $*$	✔	✔	✔	✔	✔	✔
exclusion: $\neq e$	✔	✔	✔	✔	✗	✔

UCLouvain

A basic smart MVD



labeled with Smart Elements

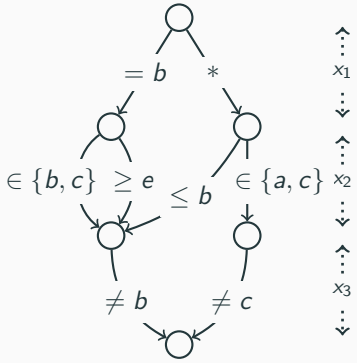
representing multiples values

	a	b	c	d	e	f
single value: $= e$	✗	✗	✗	✗	✔	✗
universal value: $*$	✔	✔	✔	✔	✔	✔
exclusion: $\neq e$	✔	✔	✔	✔	✗	✔
upper bound: $\leq c$	✔	✔	✔	✗	✗	✗

Basic Smart MVD

UCLouvain

A basic smart MVD



labeled with Smart Elements

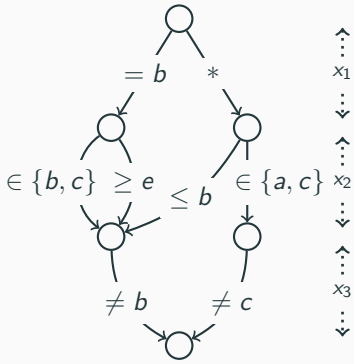
representing multiples values

	a	b	c	d	e	f
single value: $= e$	✗	✗	✗	✗	✔	✗
universal value: $*$	✔	✔	✔	✔	✔	✔
exclusion: $\neq e$	✔	✔	✔	✔	✗	✔
upper bound: $\leq c$	✔	✔	✔	✗	✗	✗
lower bound: $\geq c$	✗	✗	✔	✔	✔	✔

Basic Smart MVD

UCLouvain

A basic smart MVD



labeled with Smart Elements

representing multiples values

	a	b	c	d	e	f
single value: $= e$	✗	✗	✗	✗	✔	✗
universal value: $*$	✔	✔	✔	✔	✔	✔
exclusion: $\neq e$	✔	✔	✔	✔	✗	✔
upper bound: $\leq c$	✔	✔	✔	✗	✗	✗
lower bound: $\geq c$	✗	✗	✔	✔	✔	✔
set: $\in \{a, c, d\}$	✔	✗	✔	✔	✗	✗

- Step 1: update
  - incremental from removed values
  - reset from remaining values (generic)
  - from lower/upper bounds
- Step 2: filtering

- Step 1: update
  - incremental from removed values ($=$, $\neq$, $*$)
  - reset from remaining values (generic)
  - from lower/upper bounds
- Step 2: filtering

- Step 1: update
  - incremental from removed values ($=$, $\neq$, $*$)
  - reset from remaining values (generic)
  - from lower/upper bounds ($\leq$, $\geq$)
- Step 2: filtering

- Step 1: update
  - incremental from removed values ($=$, $\neq$, $*$)
  - reset from remaining values (generic, $\in$, $\notin$)
  - from lower/upper bounds ($\leq$, $\geq$)
- Step 2: filtering

Reorganization of supports bitsets

word 0				word 1				word 2			
w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	-	-
$=$	$\leq$	$\geq$	$\in$	$\neq$	$>$	$\notin$	$<$	$\neq$	$*$		

Reorganization of supports bitsets

word 0				word 1				word 2			
w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	-	-
$=$	$\leq$	$\geq$	$\in$	$\neq$	$>$	$\notin$	$<$	$\neq$	$*$		

$\Downarrow$

word 0				word 1				word 2				word 3			
w_0	w_4	w_8	w_9	w_3	w_6	-	-	w_1	w_7	-	-	w_2	w_5	-	-
$=$	$\neq$	$\neq$	$*$	$\in$	$\notin$			$\leq$	$<$			$\geq$	$>$		

Update by categories

word 0				word 1				word 2				word 3			
w_0	w_4	w_8	w_9	w_3	w_6	-	-	w_1	w_7	-	-	w_2	w_5	-	-
$=$	$\neq$	$\neq$	$*$	$\in$	$\notin$			$\leq$	$<$			$\geq$	$>$		

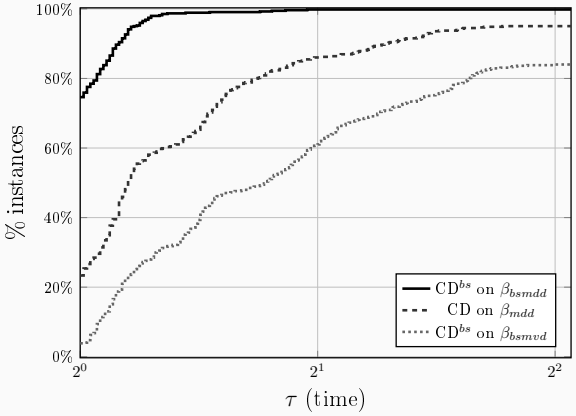
Incremental or Reset update (depending if $|\Delta(x)| < |dom(x)|$)

Reset update

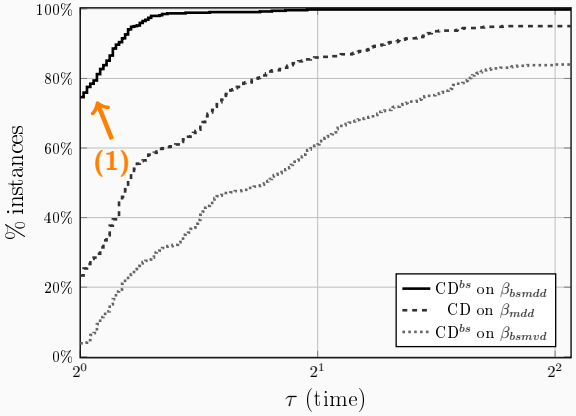
Lower bound update

Upper bound update

Results for basis smart diagrams (4000 instances)

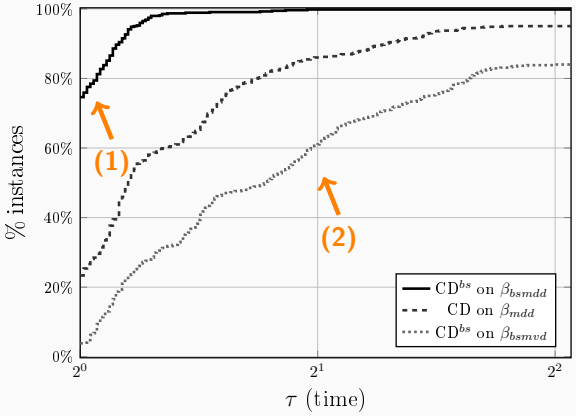


Results for basis smart diagrams (4000 instances)



(1) CD^{bs} on bs-MDDs (fewer arcs) best 80% of the time

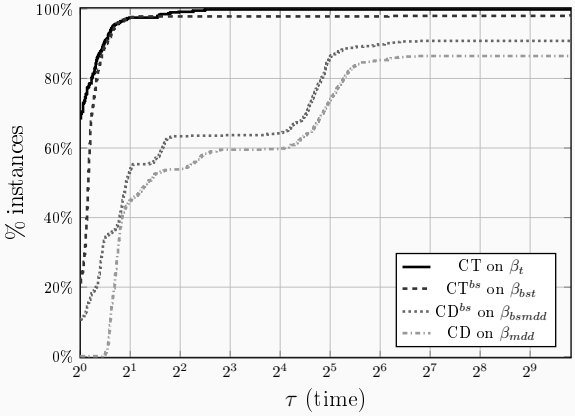
Results for basis smart diagrams (4000 instances)



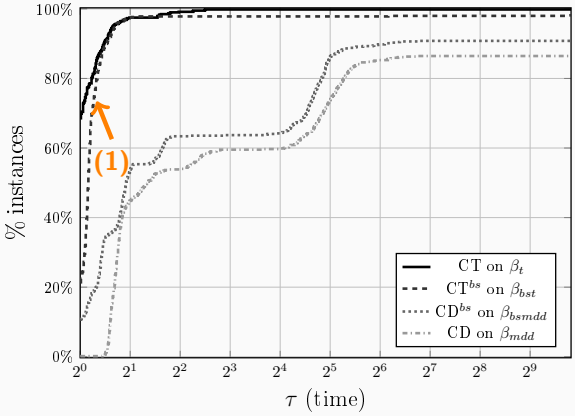
(1) CD^{bs} on bs-MDDs (fewer arcs) best 80% of the time

(2) CD^{bs} on bs-MVDs (more nodes) worst

Results for basis smart diagrams (4000 instances)

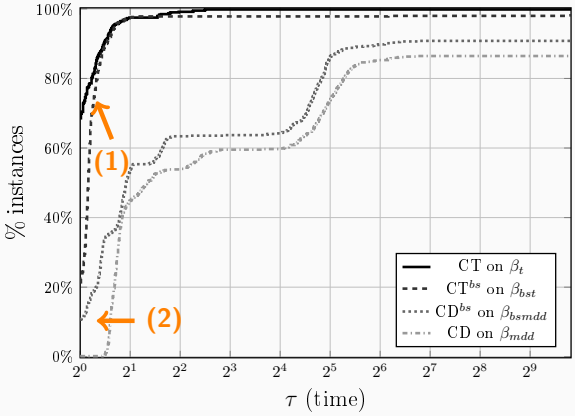


Results for basis smart diagrams (4000 instances)



(1) CT and CT^{bs} still dominating

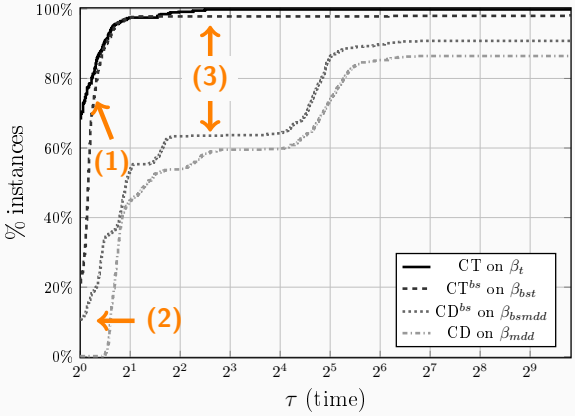
Results for basis smart diagrams (4000 instances)



(1) CT and CT^{bs} still dominating

(2) CD^{bs} becomes efficient when compression is high

Results for basis smart diagrams (4000 instances)



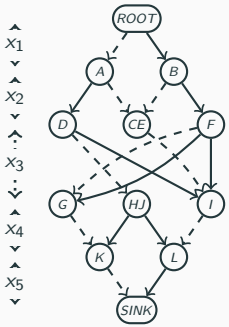
(1) CT and CT^{bs} still dominating

(2) CD^{bs} becomes efficient when compression is high

(3) gap reduced

Conclusion

- New type of layered graph (sMDD) allowing **less nodes**
- Compression of the edges of diagrams using basic smart edges allowing **less edges**
- **More efficiant** layered graph based propagator (Compact-Diagram)
- **Gap reduction** between table based (CT) and layered graph based (CD) propagator
- **First** propagator for basic smart MVDs (CDbs)



Thank you for listening!

Any questions?