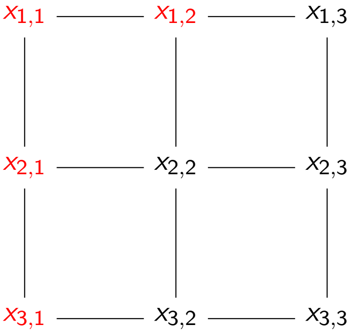
FPT Algorithms for Knowledge Compilation

Florent Capelli

CRIStAL, Université de Lille & Inria & CNRS France

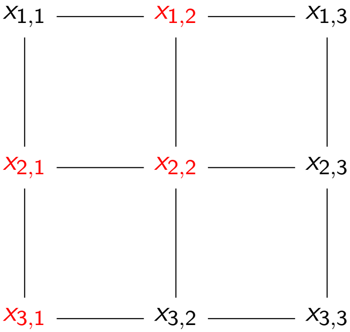
Research School on Knowledge Compilation, ENS Lyon, December 4^{th}-8^{th}

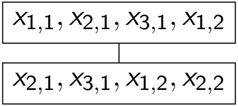
Treewidth of grids



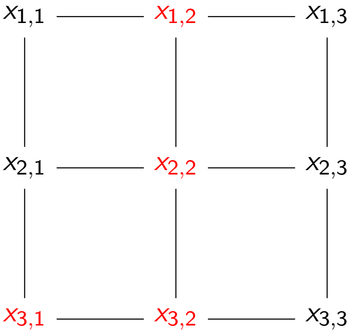
$x_{1,1}, x_{2,1}, x_{3,1}, x_{1,2}$

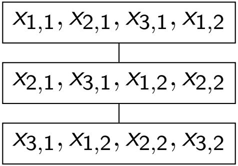
Treewidth of grids



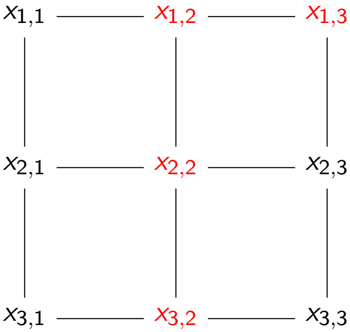


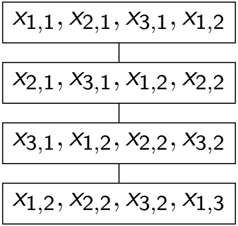
Treewidth of grids



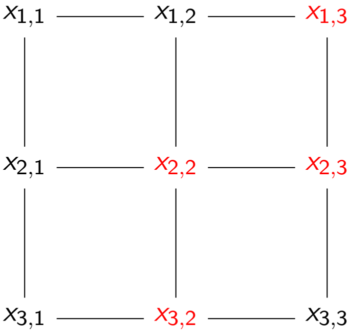


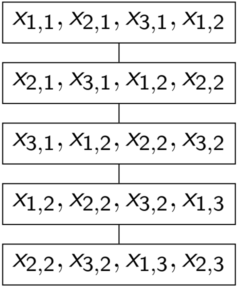
Treewidth of grids



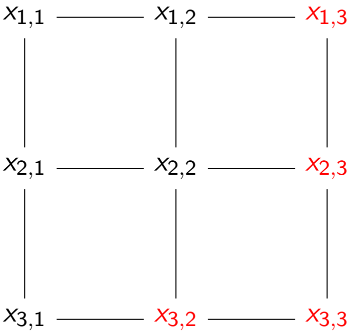


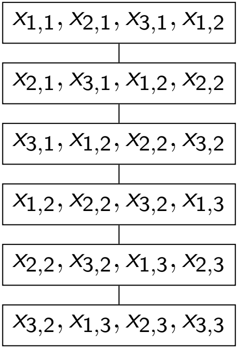
Treewidth of grids





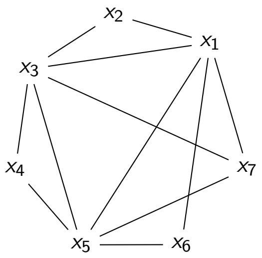
Treewidth of grids





Primal graph

Vertices are variables of the formula F.
Edge (x, y) iff x and y occur in the same clause.



$$(x_1 \vee x_2 \vee x_3) \wedge (x_3 \vee x_4 \vee \neg x_5) \wedge (x_1 \vee x_5 \vee x_6) \wedge (x_1 \vee \neg x_3 \vee x_5 \vee \neg x_7)$$

Incidence graph

By associating a graph to a formula:

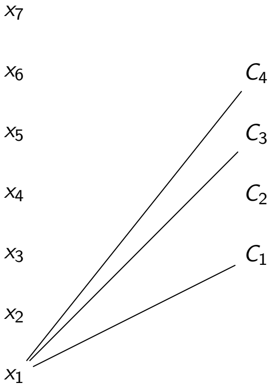


FIGURE – $(x_1 \vee x_2 \vee x_3) \wedge (x_3 \vee x_4 \vee \neg x_5) \wedge (x_1 \vee x_5 \vee x_6) \wedge (x_1 \vee \neg x_3 \vee x_5 \vee \neg x_7)$

Incidence graph

By associating a graph to a formula:

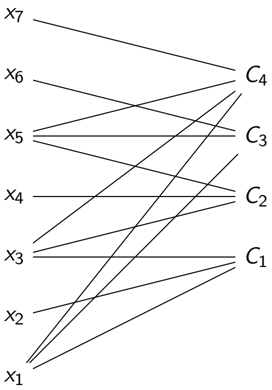


FIGURE –
$(x_1 \vee x_2 \vee x_3) \wedge (x_3 \vee x_4 \vee \neg x_5) \wedge (x_1 \vee x_5 \vee x_6) \wedge (x_1 \vee \neg x_3 \vee x_5 \vee \neg x_7)$

Structure of formulas

- **Primal treewidth** of F: $ptw(F)$ (tw of the primal graph)
- **Incidence treewidth** of F: $itw(F)$ (tw of the incidence graph).

Theorem

- $itw(F) \leq ptw(F) + 1.$
- *There exists* F_n *with* $ptw(F_n) = n$ *and* $itw(F_n) = 1.$

Incidence treewidth is strictly more general than primal treewidth.

Let's sketch a proof on the blackboard.

Exploiting the structure of formulas

Warm up:

Theorem

#SAT parametrised by ptw is FPT.

More precisely, we can count the number of solution of F in time $2^{O(k)} \cdot poly(|F|)$ where $k = ptw(F)$.

Exploiting the structure of formulas

Warm up:

Theorem
#SAT parametrised by ptw is FPT.

More precisely, we can count the number of solution of F in time $2^{O(k)} \cdot poly(|F|)$ where $k = ptw(F)$.

Actually, our algorithm is a compilation algorithm to FPT-size decision-DNNF.

Let's do better

Theorem

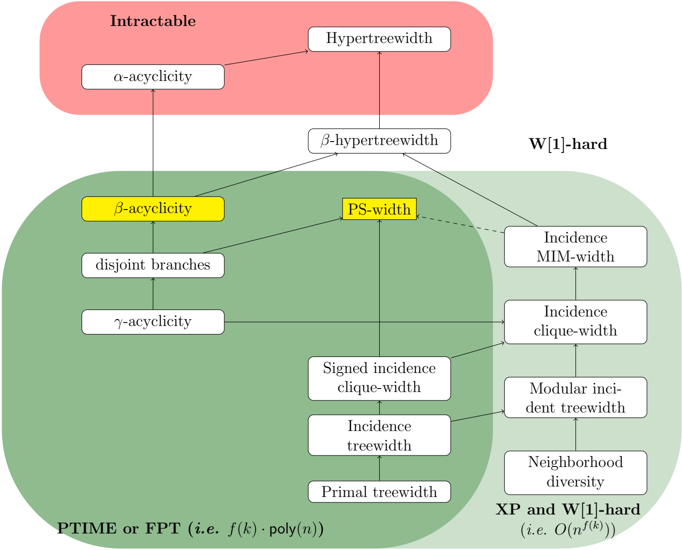
A CNF formula F can be compiled into an FPT-size d-DNNF parametrised by itw.

Let's do better

Theorem
A CNF formula F can be compiled into an FPT-size d-DNNF parametrised by itw.

You want more?

Zoology!



Intractable
Hypertreewidth
α-acyclicity
β-hypertreewidth
W[1]-hard
β-acyclicity
PS-width
Incidence MIM-width
disjoint branches
γ-acyclicity
Incidence clique-width
Signed incidence clique-width
Modular inci- dent treewidth
Incidence treewidth
Neighborhood diversity
Primal treewidth
XP and W[1]-hard
($i.e.$ $O(n^{f(k)})$)
PTIME or FPT (*i.e.* $f(k) \cdot \mathsf{poly}(n)$)