Leveraging Structural Analysis for Quantified Boolean Formulae

Reduced Block Triangular Form (RBTF)

Joan Thibault, Khalil Ghorbal

INRIA, Rennes, France

Representing and Manipulating Symbolic Expression : Boolean Function

Applications

- Model Checking
- (Computer Assisted) Hardware/Software Verification/Synthesis
- Other Applications : Mathematics, Physic, Biology, ...

Problems

- SAT/QBF
- (Integer) Linear Programming
- Polynomials
- Constraint Programming

Solvers vs Compilers

Satisfiability : Solver vs Compiler

CNF[1]-DPLL[2]-CDCL[3]-based SAT-solvers

- Efficient
- Proof Size Exponential in the number of 'xor'

Binary Decision Diagrams

- Compilation $\Longrightarrow$ Canonicity and Polynomial Queries.
- Memory Expensive

[1] Conjunctive Normal Form

[2] Davis–Putnam–Logemann–Loveland : algorithm based on case analysis and constant propagation

[3] Conflict-Driven Clause Learning

Satisfiability on Quantified Formulae: Solver vs Compiler

CNF-based QBF-solvers (Quantified Boolean Formula)

- Bad support of quantifier alternation

Binary Decision Diagrams

- Native support of quantifier elimination
- Memory Expensive

Binary Decision Diagram : A Versatile Data Structure

Model Checking / Reactive Synthesis

- Reachability
- Transitive Closure

Combinatoric

- Model Counting
- Parameterized Optimization
- Parameterized Graph Problem (related to HYCOMES' problematic)

Areas outside Computer Science

- Biology
- Physics
- ...

Quantified Boolean Formulae

- A set of **Variables** $(X_i)_i$
- A set of *local*[4] **Constraints** $(C_j)_j$ on these variable

[4] We assume that usually constraints only depends on a small number of variables.

Quantified Boolean Formulae

- A set of **Variables** $(X_i)_i$
- A set of *local*[4] **Constraints** $(C_j)_j$ on these variable
- **Formula** : $\phi = C_1 \wedge \cdots \wedge C_k$.

[4] We assume that usually constraints only depends on a small number of variables.

Quantified Boolean Formulae

- A set of **Variables** $(X_i)_i$
- A set of *local*[4] **Constraints** $(C_j)_j$ on these variable
- **Formula** : $\phi = C_1 \wedge \cdots \wedge C_k$.
- **Quantified Formula** : $\Phi = \exists E_{k'+1} \forall U_{k'} \ldots \forall U_1 \exists E_1 . \phi$.

[4] We assume that usually constraints only depends on a small number of variables.

Compilation Of Quantified Formula

Quantifier Elimination

- $\phi = \phi_1 \wedge \cdots \wedge \phi_k \xrightarrow{\forall U} \phi^{((\forall U))} = (\forall U.\, \phi_1) \wedge \cdots \wedge (\forall U.\, \phi_k)$
- $\phi \xrightarrow{\exists E} \phi^{((\exists E))}$ using `RBTF`.

Compilation Of Quantified Formula

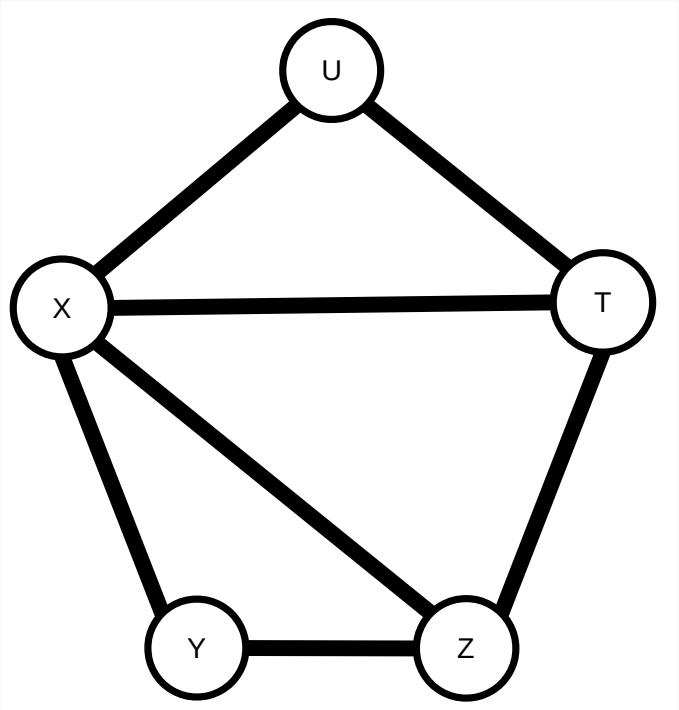
Quantifier Elimination

- $\phi = \phi_1 \wedge \cdots \wedge \phi_k \xrightarrow{\forall U} \phi^{((\forall U))} = (\forall U.\, \phi_1) \wedge \cdots \wedge (\forall U.\, \phi_k)$
- $\phi \xrightarrow{\exists E} \phi^{((\exists E))}$ using RBTF.

$$\exists E_{k+1} \forall U_k \ldots \underbrace{\forall U_1 \underbrace{\exists E_1.\, \phi}_{\phi^{((\exists E_1))}}}_{\phi^{((\exists E_1))((\forall U_1))}}.$$

Application : Boolean Existential Closure

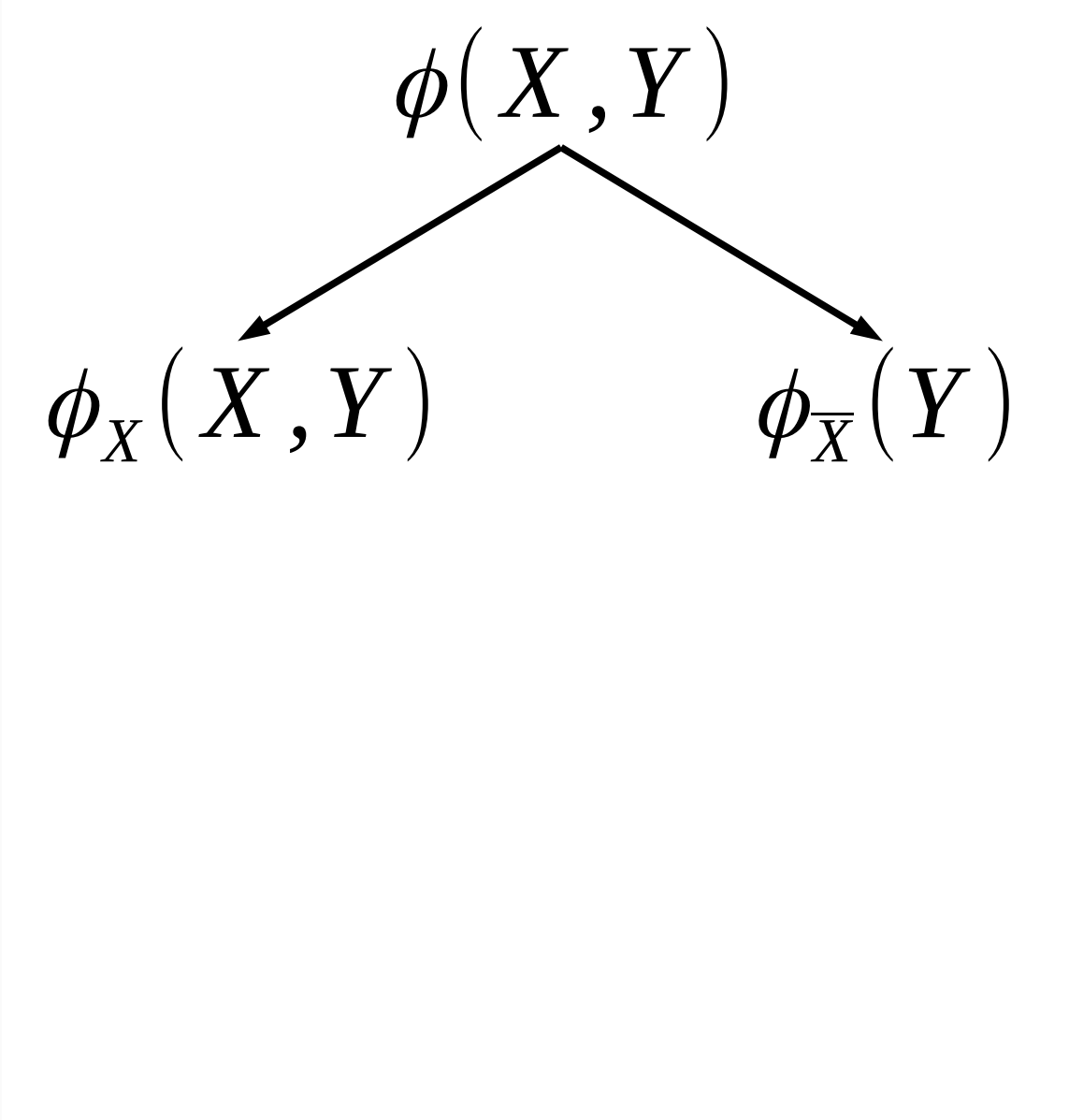
- Variables : X, Y, Z, T, U.
- Constraints : $C_1(X, Y, Z)$, $C_2(X, Z, T)$, $C_3(X, T, U)$.
- Formula :
  $\phi(X, Y, Z, T, U) := C_1(X, Y, Z) \wedge C_2(X, Z, T) \wedge C_3(X, T, U)$.
- Primal Graph :



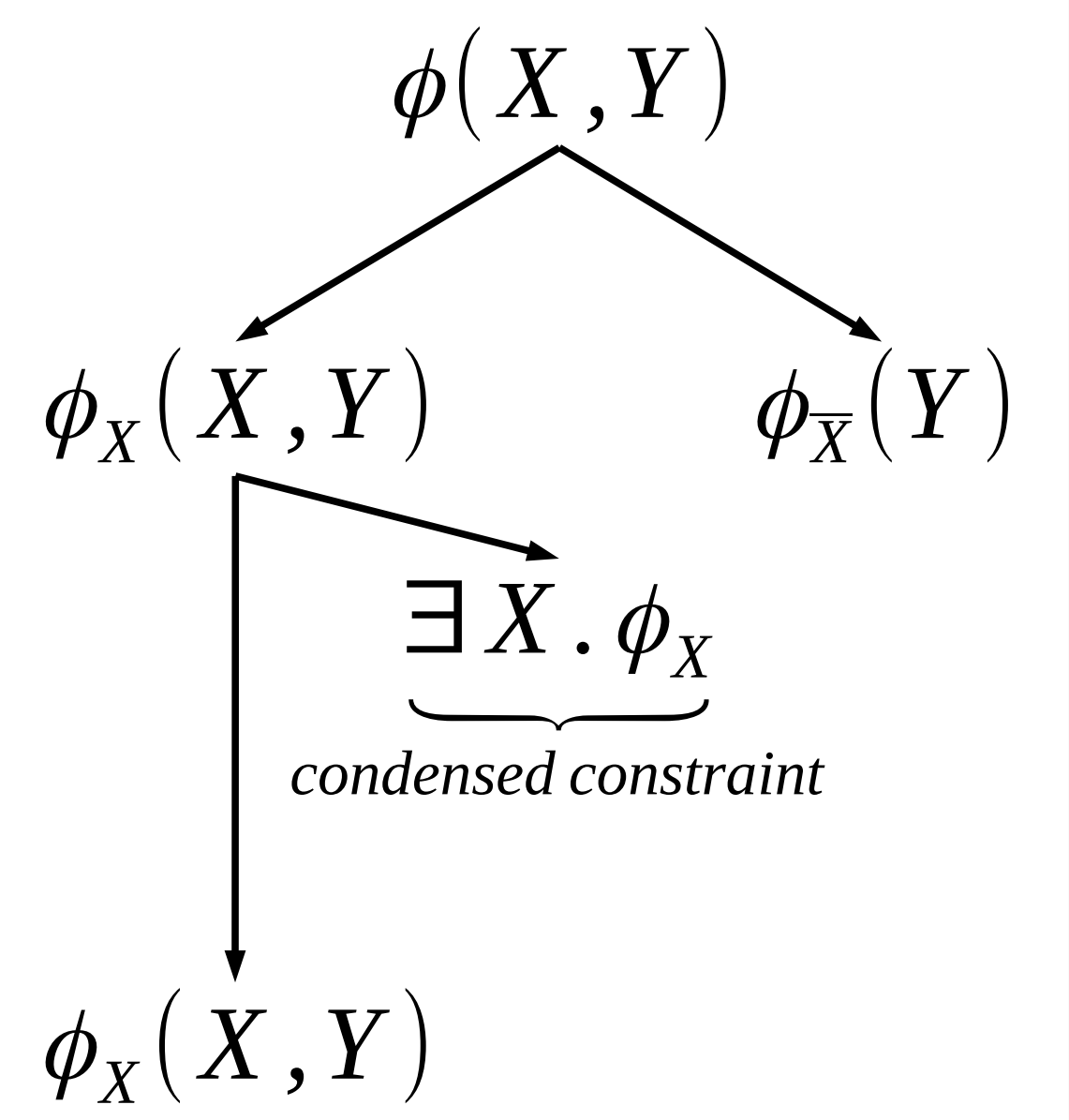
RBTF : Hand Waving Version

$$\phi(X, Y)$$

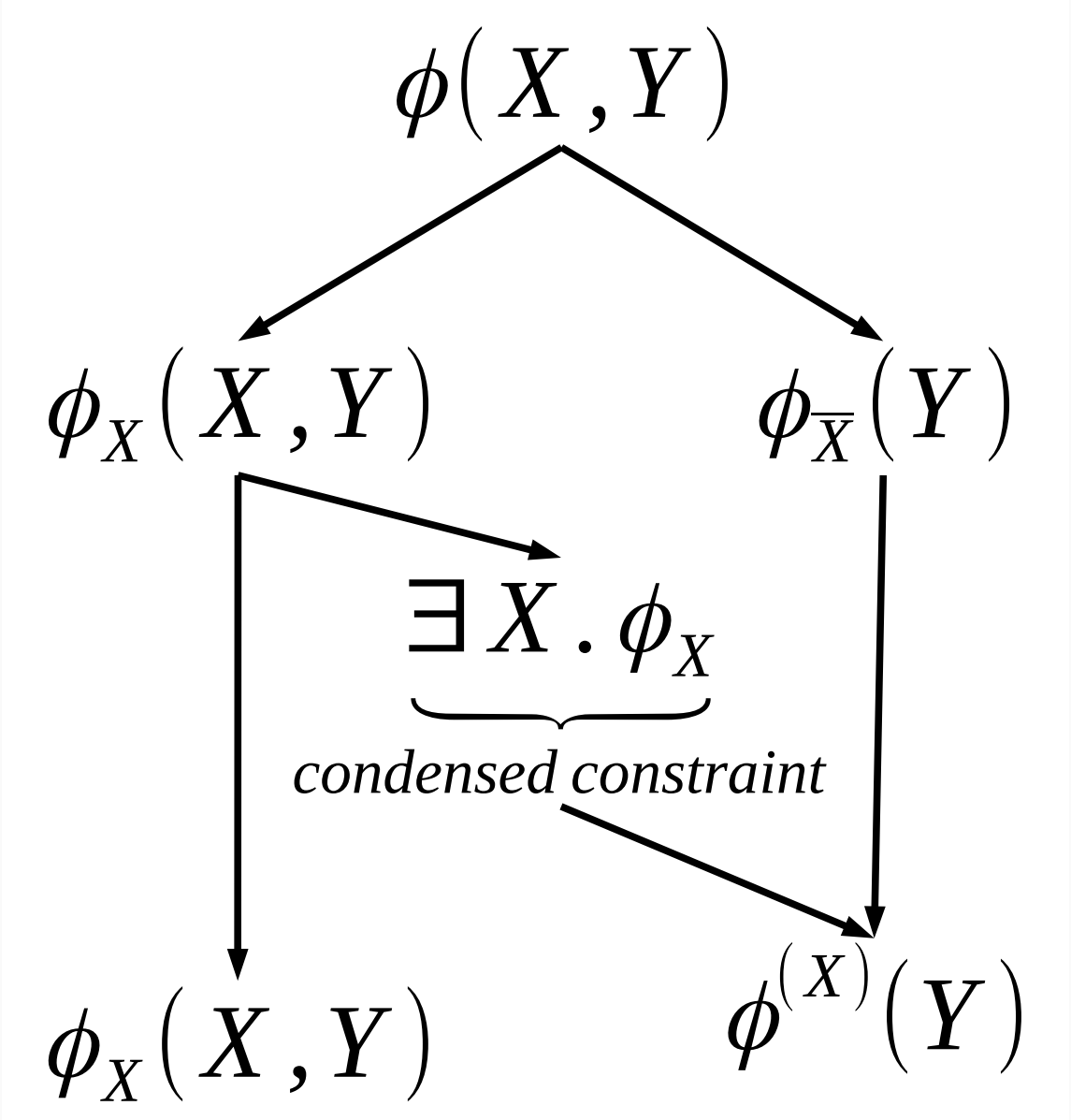
RBTF : Hand Waving Version



RBTF : Hand Waving Version

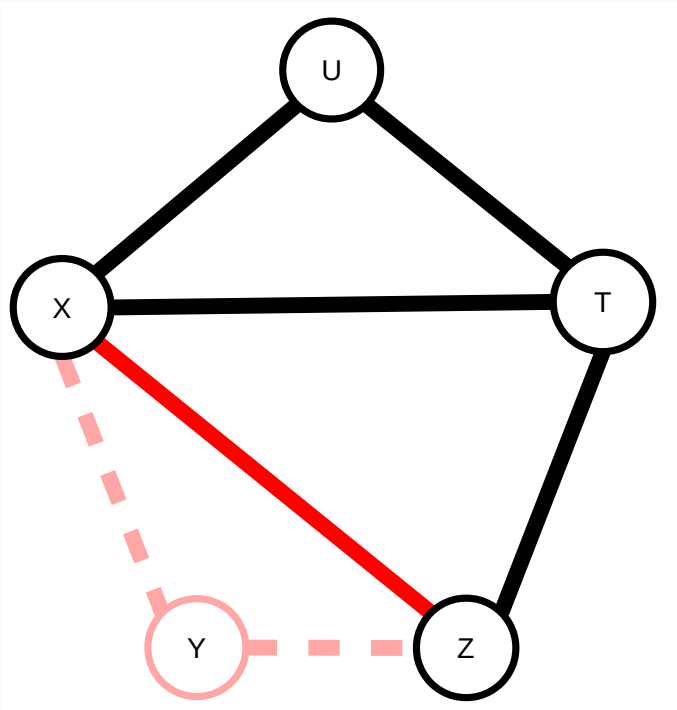


RBTF : Hand Waving Version



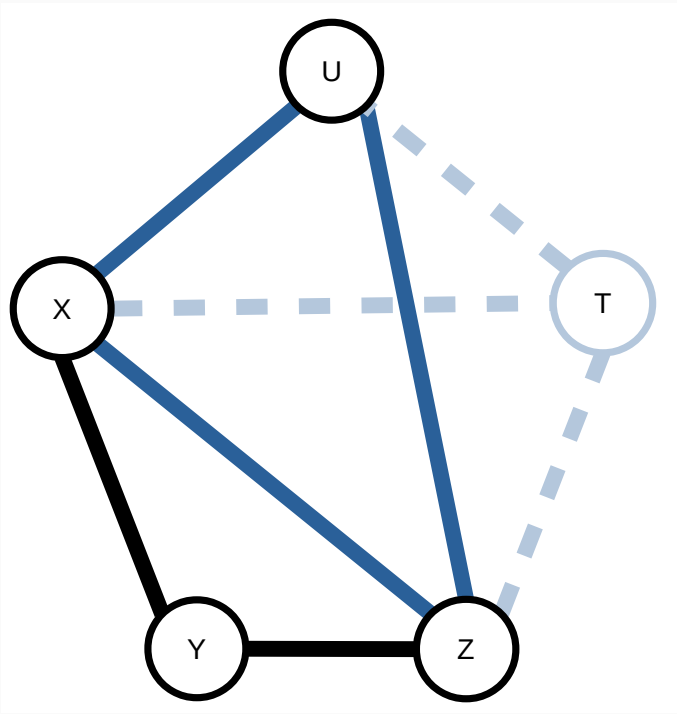
Eliminating A Variable : Y

- Formula :
  $\phi(X, Y, Z, T, U) := C_1(X, Y, Z) \wedge C_2(X, Z, T) \wedge C_3(X, T, U)$.
- Projection : $C_1'(X, Z) := \exists Y, \phi_Y = \exists Y, C_1(X, Y, Z)$.
- WCET : $2^{|\phi_Y|} = 2^3$
- Condensation :
  $\phi^{(Y)}(X, Z, T, U) = C_1'(X, Z) \wedge C_2(X, Z, T) \wedge C_3(X, T, U)$.
- Primal Graph :



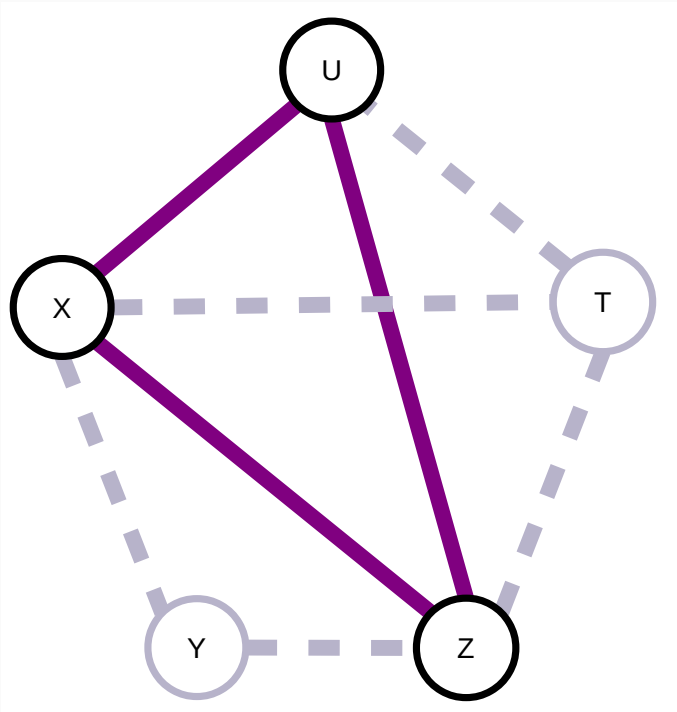
Eliminating A Variable : T

- Formula :
  $\phi(X, Y, Z, T, U) := C_1(X, Y, Z) \wedge C_2(X, Z, T) \wedge C_3(X, T, U)$.
- Projection : $C_2'(X, Z, U) := \exists T, \phi_T = \exists T, C_2 \wedge C_3$.
- WCET : $2^{|\phi_T|} = 2^4$
- Condensation : $\phi^{(T)}(X, Y, Z, U) = C_1(X, Y, Z) \wedge C_2'(X, Z, U)$.
- Primal Graph :



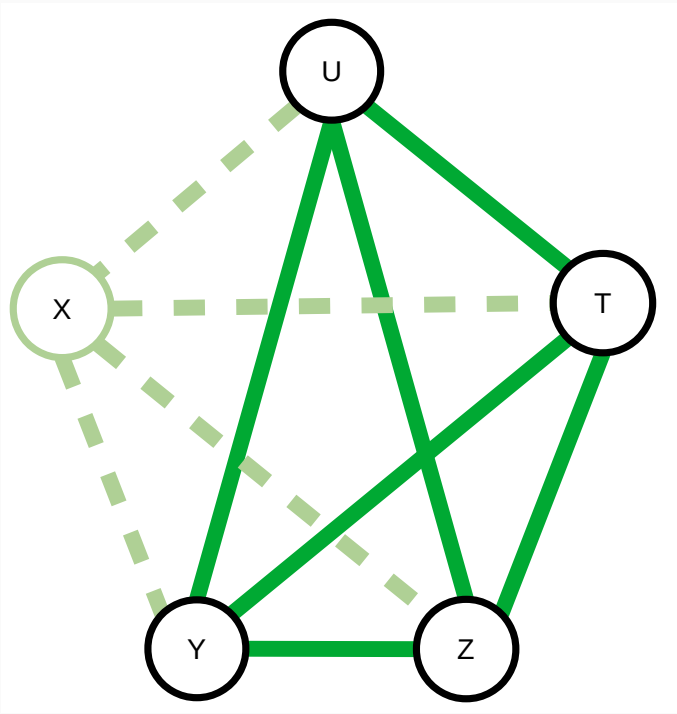
Eliminating Several Variables : $\{Y, T\}$

- Formula :
  $\phi(X, Y, Z, T, U) := C_1(X, Y, Z) \wedge C_2(X, Z, T) \wedge C_3(X, T, U)$.
- Projection :
  $C_1'(X, Z, U) := \exists(Y, T), \phi_{\{Y,T\}} = \exists(Y, T), C_1 \wedge C_2 \wedge C_3$.
- WCET : $2^{|\phi_{\{Y,T\}}|} = 2^5$
- Condensation : $\phi^{(\{Y,T\})}(X, Z, U) = C_1'(X, Z, U)$.
- Primal Graph :



Bad Selection of Variable : X

- Formula :
  $\phi(X, Y, Z, T, U) := C_1(X, Y, Z) \wedge C_2(X, Z, T) \wedge C_3(X, T, U)$.
- Projection : $C_1'(Y, Z, T, U) := \exists X, \phi_X = \exists X, C_1 \wedge C_3 \wedge C_3$.
- WCET : $2^{|\phi_X|} = 2^5$
- Condensation : $\phi^{(X)}(X, Z, T, U) = C_1'(Y, Z, T, U)$.
- Primal Graph :



Constraint Propagation : Eliminating All Variables

- Basic Operation : $\mathtt{S}(\phi(\mathbf{X}, \mathbf{Y}), \mathbf{X}) := \phi_{\mathbf{X}}(\mathbf{X}, \mathbf{Y}) \wedge \phi^{(\mathbf{X})}(\mathbf{Y})$

Constraint Propagation : Eliminating All Variables

- Basic Operation : $\mathtt{S}(\phi(\mathbf{X},\mathbf{Y}),\mathbf{X}) := \phi_{\mathbf{X}}(\mathbf{X},\mathbf{Y}) \wedge \phi^{(\mathbf{X})}(\mathbf{Y})$
- Basic Operation Cost : $\mathtt{S}^{\mathtt{c}}(\phi,\mathbf{X}) := 2^{|\phi_{\mathbf{X}}|}$
  - For any formula ϕ, we denote $\mathtt{supp}(\phi)$ the set of variables appearing in ϕ, we denote $|\phi| := |\mathtt{supp}(\phi)|$ its cardinal, i.e., the number of variables in ϕ.

Constraint Propagation : Eliminating All Variables

- Basic Operation : $\mathtt{S}(\phi(\mathbf{X},\mathbf{Y}),\mathbf{X}) := \phi_{\mathbf{X}}(\mathbf{X},\mathbf{Y}) \wedge \phi^{(\mathbf{X})}(\mathbf{Y})$
- Basic Operation Cost : $\mathtt{S}^{\mathtt{c}}(\phi,\mathbf{X}) := 2^{|\phi_{\mathbf{X}}|}$
  - For any formula ϕ, we denote $\mathtt{supp}(\phi)$ the set of variables appearing in ϕ, we denote $|\phi| := |\mathtt{supp}(\phi)|$ its cardinal, i.e., the number of variables in ϕ.
- $\phi(\mathbf{X_1},\cdots,\mathbf{X_k}) \equiv \phi_{\mathbf{X_1}} \wedge \phi^{(\mathbf{X_1})}_{\mathbf{X_2}} \wedge \phi^{(\mathbf{X_1},\mathbf{X_2})}_{\mathbf{X_3}} \wedge \cdots \wedge \phi^{(\mathbf{X_1},\cdots,\mathbf{X_{k-1}})}_{\mathbf{X_k}} \wedge \underbrace{\phi^{(\mathbf{X_1},\cdots,\mathbf{X_k})}}_{\text{constant}}$.

Constraint Propagation : Eliminating All Variables

- Basic Operation : $\mathtt{S}(\phi(\mathbf{X},\mathbf{Y}),\mathbf{X}) := \phi_{\mathbf{X}}(\mathbf{X},\mathbf{Y}) \wedge \phi^{(\mathbf{X})}(\mathbf{Y})$
- Basic Operation Cost : $\mathtt{S}^{\mathtt{c}}(\phi,\mathbf{X}) := 2^{|\phi_{\mathbf{X}}|}$
  - For any formula ϕ, we denote $\mathtt{supp}(\phi)$ the set of variables appearing in ϕ, we denote $|\phi| := |\mathtt{supp}(\phi)|$ its cardinal, i.e., the number of variables in ϕ.
- $\phi(\mathbf{X_1},\cdots,\mathbf{X_k}) \equiv \phi_{\mathbf{X_1}} \wedge \phi^{(\mathbf{X_1})}_{\mathbf{X_2}} \wedge \phi^{(\mathbf{X_1},\mathbf{X_2})}_{\mathbf{X_3}} \wedge \cdots \wedge \phi^{(\mathbf{X_1},\cdots,\mathbf{X_{k-1}})}_{\mathbf{X_k}} \wedge \underbrace{\phi^{(\mathbf{X_1},\cdots,\mathbf{X_k})}}_{\text{constant}}$.
- We call this process : *Forward Reduction Process* (FRP).
- We term this representation the *weakly Reduced Block Triangular Form* (weak-RBTF)

Constraint Saturation : Reduced Block Triangular Form

- $\phi \equiv \underbrace{\phi_{\mathbf{X_1}}}_{=\phi'_0} \wedge \underbrace{\phi^{(\mathbf{X_1})}_{\mathbf{X_2}}}_{=\phi'_1} \wedge \underbrace{\phi^{(\mathbf{X_1},\mathbf{X_2})}_{\mathbf{X_3}}}_{=\phi'_2} \wedge \cdots \wedge \underbrace{\phi^{(\mathbf{X_1},\cdots,\mathbf{X_{k-1}})}_{\mathbf{X_k}}}_{=\phi'_{k-1}} \wedge \underbrace{\phi^{(\mathbf{X_1},\cdots,\mathbf{X_k})}}_{=\phi'_k}$.
- One may show that, using a similar process, the final constraint may be propagated back into the structure leading to canonical representation (up to an ordered partition of the variables).
- $\phi \equiv \phi'' = \bigwedge_{0 \leq i \leq k} \phi''_i$
  - with $\phi''_i \equiv \phi_{|\mathtt{supp}(\phi'_i)}$
  - with, for any formula $\psi(\mathbf{X}, \mathbf{Y})$, $\psi_{|\mathbf{X}}(\mathbf{X}) := \exists \mathbf{Y}, \psi(\mathbf{X}, \mathbf{Y})$.
- We term this process the *Backward Propagation Process* (BPP)
- We term this representation *Reduced Block Triangular Form* (RBTF)

Weighted Adjacency Propagation (WAP) problem

Models `FRP` on the formula's Primal Graph

- project variables $\Rightarrow$ remove vertices : X
- add back a new constraint $\Rightarrow$ add a clique on these vertices' neighborhood : $N_G(X)$.
- time/memory $\Rightarrow$ cost function $2^{|N_G(X)|}$.

Weighted Adjacency Propagation (WAP) problem

Models `FRP` on the formula's Primal Graph

- project variables $\Rightarrow$ remove vertices : X
- add back a new constraint $\Rightarrow$ add a clique on these vertices' neighborhood : $N_G(X)$.
- time/memory $\Rightarrow$ cost function $2^{|N_G(X)|}$.

WAP is strongly related to tree-decomposition, tree-width and chordal completion.

Weighted Adjacency Propagation (WAP) problem

Models `FRP` on the formula's Primal Graph

- project variables $\Rightarrow$ remove vertices : X
- add back a new constraint $\Rightarrow$ add a clique on these vertices' neighborhood : $N_G(X)$.
- time/memory $\Rightarrow$ cost function $2^{|N_G(X)|}$.

WAP is strongly related to tree-decomposition, tree-width and chordal completion.

Heuristic

- Quotient the graph by *true-twins*
  - vertices are weighted
  - single vertex selection
- Select a vertex with lightest neighborhood.

Preliminary Experimental Results : SAT competition 2018

mchess_n : UNSAT

n	$\|X\|$	$\|C\|$	BLOCK	CADICAL	RBTF
15	420	1391	31	140s	4s
16	480	1596	32	2m24s	4s
17	544	1815	34	67m34s	21s
18	612	2048	39	10m28s	26s
19	684	2295	41	8h43m	36s
20	760	2556	47	8h31m	2m30s

where BLOCK : number of variable in the biggest sub-problem.

otherwise : timeout or memout for RBTF.

conclusion : failure if BLOCK $>$ 40-50 variables

Preliminary Experimental Results : SAT competition 2018

mchess_n : UNSAT

n	$\|X\|$	$\|C\|$	BLOCK	CADICAL	RBTF
15	420	1391	31	140s	4s
16	480	1596	32	2m24s	4s
17	544	1815	34	67m34s	21s
18	612	2048	39	10m28s	26s
19	684	2295	41	8h43m	36s
20	760	2556	47	8h31m	2m30s

where BLOCK : number of variable in the biggest sub-problem.

otherwise : timeout or memout for RBTF.

conclusion : failure if BLOCK $>$ 40-50 variables

future work : **structural analysis** and **underlying representation**

Conclusion on RBTF

Reduction Process : $\phi \xrightarrow{\text{FRP}}$ **weak-RBTF** $\xrightarrow{\text{BPP}}$ **RBTF**

- generic
- compatible with symbolic representation
- preserves and exploit tree-like structures
- encouraging preliminary experimental results

Limitations (Orthogonal Future Work)

- structural analysis : *WAP*
- underlying representation : λDD[5]

Future Work

- implement cascading RBTF (compilation of Quantified Formulae)
- unit propagation, reset, multi-layer analysis, parallelism
- co-design with a solver

[5] J. Thibault and K. Ghorbal, Functional Decision Diagrams: A Unifying Data Structure For Binary Decision Diagrams

Contraint Propagation

Basic Properties
For any formula $\phi(\mathbf{X}, \mathbf{Y})$ where $\mathbf{X}$ and $\mathbf{Y}$ are disjoint sets of variables.

- $\phi(\mathbf{X}, \mathbf{Y}) \iff \exists \mathbf{X}, \phi(\mathbf{X}, \mathbf{Y})$

Contraint Propagation

Basic Properties
For any formula $\phi(\mathbf{X}, \mathbf{Y})$ where $\mathbf{X}$ and $\mathbf{Y}$ are disjoint sets of variables.

- $\phi(\mathbf{X}, \mathbf{Y}) \iff \exists \mathbf{X}, \phi(\mathbf{X}, \mathbf{Y})$
- $\phi(\mathbf{X}, \mathbf{Y}) \equiv \phi(\mathbf{X}, \mathbf{Y}) \wedge \exists \mathbf{X}, \phi(\mathbf{X}, \mathbf{Y})$

Contraint Propagation

Basic Properties
For any formula $\phi(\mathbf{X}, \mathbf{Y})$ where $\mathbf{X}$ and $\mathbf{Y}$ are disjoint sets of variables.

- $\phi(\mathbf{X}, \mathbf{Y}) \iff \exists \mathbf{X}, \phi(\mathbf{X}, \mathbf{Y})$
- $\phi(\mathbf{X}, \mathbf{Y}) \equiv \phi(\mathbf{X}, \mathbf{Y}) \wedge \exists \mathbf{X}, \phi(\mathbf{X}, \mathbf{Y})$
- $\phi(\mathbf{X}, \mathbf{Y}) = \underbrace{C_1(\mathbf{X}, \mathbf{Y}) \wedge \cdots \wedge C_k(\mathbf{X}, \mathbf{Y})}_{=\phi_{\mathbf{X}}(\mathbf{X}, \mathbf{Y})} \wedge \underbrace{D_1(\mathbf{Y}) \wedge \cdots \wedge D_{k'}(\mathbf{Y})}_{=\phi_{\setminus \mathbf{X}}(\mathbf{Y})}.$

Contraint Propagation

Basic Properties

For any formula $\phi(\mathbf{X}, \mathbf{Y})$ where $\mathbf{X}$ and $\mathbf{Y}$ are disjoint sets of variables.

- $\phi(\mathbf{X}, \mathbf{Y}) \iff \exists \mathbf{X}, \phi(\mathbf{X}, \mathbf{Y})$
- $\phi(\mathbf{X}, \mathbf{Y}) \equiv \phi(\mathbf{X}, \mathbf{Y}) \wedge \exists \mathbf{X}, \phi(\mathbf{X}, \mathbf{Y})$
- $\phi(\mathbf{X}, \mathbf{Y}) = \underbrace{C_1(\mathbf{X}, \mathbf{Y}) \wedge \cdots \wedge C_k(\mathbf{X}, \mathbf{Y})}_{=\phi_{\mathbf{X}}(\mathbf{X}, \mathbf{Y})} \wedge \underbrace{D_1(\mathbf{Y}) \wedge \cdots \wedge D_{k'}(\mathbf{Y})}_{=\phi_{\setminus \mathbf{X}}(\mathbf{Y})}.$

Variable Set Elimination

- $\phi(\mathbf{X}, \mathbf{Y}) = \underbrace{\phi_{\mathbf{X}}(\mathbf{X}, \mathbf{Y})}_{\equiv \phi_{\mathbf{X}}(\mathbf{X}, \mathbf{Y}) \wedge \exists \mathbf{X}, \phi_{\mathbf{X}}(\mathbf{X}, \mathbf{Y})} \wedge \phi_{\setminus \mathbf{X}}(\mathbf{Y})$

Contraint Propagation

Basic Properties

For any formula $\phi(\mathbf{X}, \mathbf{Y})$ where $\mathbf{X}$ and $\mathbf{Y}$ are disjoint sets of variables.

- $\phi(\mathbf{X}, \mathbf{Y}) \iff \exists \mathbf{X}, \phi(\mathbf{X}, \mathbf{Y})$
- $\phi(\mathbf{X}, \mathbf{Y}) \equiv \phi(\mathbf{X}, \mathbf{Y}) \wedge \exists \mathbf{X}, \phi(\mathbf{X}, \mathbf{Y})$
- $\phi(\mathbf{X}, \mathbf{Y}) = \underbrace{C_1(\mathbf{X}, \mathbf{Y}) \wedge \cdots \wedge C_k(\mathbf{X}, \mathbf{Y})}_{=\phi_{\mathbf{X}}(\mathbf{X}, \mathbf{Y})} \wedge \underbrace{D_1(\mathbf{Y}) \wedge \cdots \wedge D_{k'}(\mathbf{Y})}_{=\phi_{\setminus \mathbf{X}}(\mathbf{Y})}.$

Variable Set Elimination

- $\phi(\mathbf{X}, \mathbf{Y}) = \underbrace{\phi_{\mathbf{X}}(\mathbf{X}, \mathbf{Y})}_{\equiv \phi_{\mathbf{X}}(\mathbf{X}, \mathbf{Y}) \wedge \exists \mathbf{X}, \phi_{\mathbf{X}}(\mathbf{X}, \mathbf{Y})} \wedge \phi_{\setminus \mathbf{X}}(\mathbf{Y})$
- $\phi(\mathbf{X}, \mathbf{Y}) \equiv \phi_{\mathbf{X}}(\mathbf{X}, \mathbf{Y}) \wedge \underbrace{(\overbrace{(\exists \mathbf{X}, \phi_{\mathbf{X}}(\mathbf{X}, \mathbf{Y}))}^{=D_0(\mathbf{Y})} \wedge \phi_{\setminus \mathbf{X}}(\mathbf{Y}))}_{= \phi^{(\mathbf{X})}(\mathbf{Y}) \equiv \exists \mathbf{X}, \phi(\mathbf{X}, \mathbf{Y})}$

Weighted Adjacency Propagation (WAP) : Definition

- $\mathbb{S}(G, \mathbf{X}) := (V', E')$ with :
  - $V' := V \backslash \mathbf{X}$
  - $E' := \left(E \bigcup N_G(\mathbf{X})^2\right) \bigcap V'^2$

Weighted Adjacency Propagation (WAP) : Definition

- $\mathbb{S}(G, \mathbf{X}) := (V', E')$ with :
  - $V' := V \backslash \mathbf{X}$
  - $E' := \left(E \bigcup N_G(\mathbf{X})^2\right) \bigcap V'^2$
- $\mathbb{S}^c(G, \mathbf{X}) := 2^{|N_G(\mathbf{X})|}$

Weighted Adjacency Propagation (WAP) : Definition

- $\mathbb{S}(G, \mathbf{X}) := (V', E')$ with :
  - $V' := V \backslash \mathbf{X}$
  - $E' := \left(E \bigcup N_G(\mathbf{X})^2\right) \bigcap V'^2$
- $\mathbb{S}^c(G, \mathbf{X}) := 2^{|N_G(\mathbf{X})|}$
- $\mathbb{S}(G, \mathbf{X_0}, \ldots, \mathbf{X_k}) = \mathbb{S}(\mathbb{S}(G, \mathbf{X_0}), \mathbf{X_1}, \ldots, \mathbf{X_k})$

Weighted Adjacency Propagation (WAP) : Definition

- $\mathbb{S}(G, \mathbf{X}) := (V', E')$ with :
  - $V' := V \backslash \mathbf{X}$
  - $E' := \left(E \bigcup N_G(\mathbf{X})^2\right) \bigcap V'^2$
- $\mathbb{S}^c(G, \mathbf{X}) := 2^{|N_G(\mathbf{X})|}$
- $\mathbb{S}(G, \mathbf{X_0}, \ldots, \mathbf{X_k}) = \mathbb{S}(\mathbb{S}(G, \mathbf{X_0}), \mathbf{X_1}, \ldots, \mathbf{X_k})$
- $\mathbb{S}^c(G, \mathbf{X_0}, \ldots, \mathbf{X_k}) = \mathbb{S}^c(G, \mathbf{X_0}) + \mathbb{S}^c(\mathbb{S}(G, \mathbf{X_0}), \mathbf{X_1}, \ldots, \mathbf{X_k})$

Weighted Adjacency Propagation (WAP) : Definition

- $\mathbb{S}(G, \mathbf{X}) := (V', E')$ with :
  - $V' := V \backslash \mathbf{X}$
  - $E' := \left(E \bigcup N_G(\mathbf{X})^2\right) \bigcap V'^2$
- $\mathbb{S}^c(G, \mathbf{X}) := 2^{|N_G(\mathbf{X})|}$
- $\mathbb{S}(G, \mathbf{X_0}, \ldots, \mathbf{X_k}) = \mathbb{S}(\mathbb{S}(G, \mathbf{X_0}), \mathbf{X_1}, \ldots, \mathbf{X_k})$
- $\mathbb{S}^c(G, \mathbf{X_0}, \ldots, \mathbf{X_k}) = \mathbb{S}^c(G, \mathbf{X_0}) + \mathbb{S}^c(\mathbb{S}(G, \mathbf{X_0}), \mathbf{X_1}, \ldots, \mathbf{X_k})$
- Goal : find a partition $(\mathbf{X}_i)_i$ of $G.V$ such that $\mathbb{S}^c(G, (\mathbf{X_i})_i)$ is minimal.

Known Properties

- Adding unerasable vertices allows to solve parametrized problem (e.g., QBF and parametrized `argmax`).
- WAP to H-WAP reduction
  - vertices are weighted (1 by default).
    - $\omega(\mathbf{X}) := \sum_{x \in \mathbf{X}} \omega(x)$
    - $\mathrm{s}^{\mathrm{c}}(G, \mathbf{X}) := 2^{\omega(N_G(\mathbf{X}))}$.
  - true-twins vertices are merged.
    - two vertices x and y are said true-twins iff $N_G(x) = N_G(y)$
  - only one vertex is erased at a time
- optimal elimination of pendant H-vertices
  - optimal elimination of H-trees
- H-Simplicial Elimination
- Similarity with treewidth (TW)
- Clique Seperators are WAP-Separators