

Optimization problems on clique-width graphs

Shahin Kamali

PhD – Database Group

www.cs.uwaterloo.ca/~s3kamali

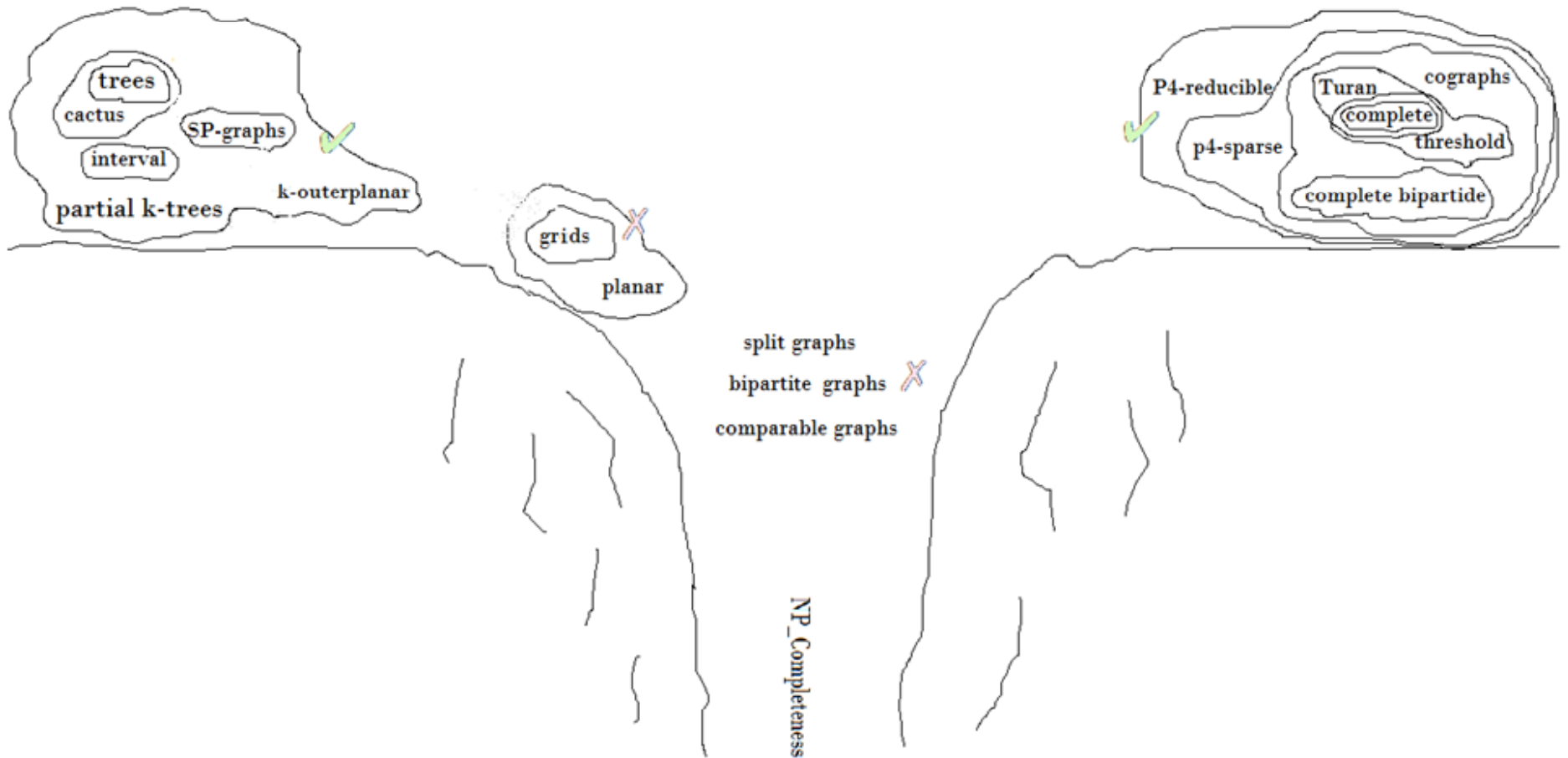
Outline

- Motivation
- A Fundamental Theorem
- Logic
 - Monadic Second-Order Logic
 - Graphs and Problems as logical structures
- Sketch of the proof of the Theorem
- Application in Algorithm Design

Motivation

- Clique-width graphs are extensions of dense graphs:
 - Complete graphs, Complete bipartite, etc..
- What about sparse graphs?
- Theorem: $cw(G) \leq 3 \cdot 2^{tw(G)} - 1$ [Gurski 2000]
 - Ex: for trees we have $cw(T) = 3 \leq 5$
- Partial k-trees are a subset of clique-width graphs

Clique-width Graphs and Hierarchy classes



Problem Statement

- In partial k-trees almost all problems can be solved in linear time
- Can we extend to clique-width graphs?
- Theorem: [Courcell 1999]

In clique-width graphs almost all problems can be solved in linear time

To prove, we need logic :O

Logic Families

- Propositional logic:

John is Man \rightarrow John is Mortal

- First order logic (predicate logic):

Man(x) \rightarrow x is Mortal

$\forall x \exists y (x + y = 0)$

- Second order logic:

$\forall P \forall x (P(x) \vee \neg(P(x)))$

P(x) unary relation, means $x \in P$

$\forall P \forall x \forall y (P(x,y) \vee \neg(P(x,y)))$

P(x,y) binary relation, means $(x,y) \in P$

Monadic Second Order Logic (MSOL)

- A form of second order logic which allows quantification only on discrete variables and unary relations:

$$\forall P \forall x (P(x) \vee \neg(P(x))) \quad \checkmark$$

$$\forall x \forall y (P(x,y) \vee \neg(P(x,y))) \quad \checkmark$$

$$\forall P \forall x \forall y (P(x,y) \vee \neg(P(x,y))) \quad \times$$

"don't worry... more example will come"

Graphs as Logical Structures

- graph $G \rightarrow$ logical structure $G(\tau_1)$

- Domain: $V: \{u, v, w, x, y\}$

- Vocabulary: binary relation E

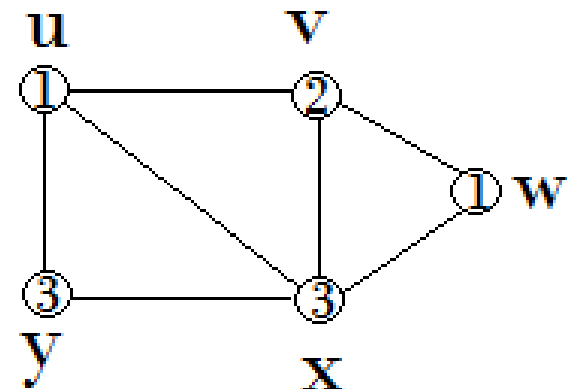
unary relations U_1, \dots, U_p

- $G(\tau_1)$:

$E(u, v) E(u, x) E(u, y) E(v, x)$

$E(y, x) E(v, w) E(x, w) U_1(u)$

$U_2(v) U_3(y) U_3(x) U_1(w)$



Logical Translation of Problems

- In clique-width graphs almost all problems can be solved in linear time
- We need to represent problems in logical format

MSOL problems

- A family of decision problems
- Given a τ -structure A (a graph G)
- Does $A \models \phi$
- ϕ is an MSOL formula over τ

MSOL problems

- Example: 3-colorability:
- Does $G(\tau_1) \models \phi$

$$\varphi \equiv \exists X_1, X_2, X_3 (Partition(X_1, X_2, X_3) \wedge \\ IndSet(X_1) \wedge IndSet(X_2) \wedge IndSet(X_3))$$

$$Partition(X_1, X_2, X_3) \equiv \forall v (X_1(v) \vee X_2(v) \vee X_3(v)) \wedge \\ \neg \exists u ((X_1(u) \wedge X_2(u)) \vee (X_1(u) \wedge X_3(u)) \vee \\ (X_2(u) \wedge X_3(u)))$$

$$Indset(X) \equiv \forall u, v ((X(u) \wedge X(v)) \longrightarrow \neg E(u, v))$$

LinEMSOL Problems

- A family of optimization problems
 - Given τ -structure A (a graph),
 - Given m evaluation functions f_1, \dots, f_m associating values to elements of A (vertices)
- Eligibility constraints
 - Defined by an MSOL formula θ
- Optimality constraints
 - A linear combination of f_i s for a candidate solution

MSOL Optimization Problems

- Example: maximum weight clique
 - In a weighted-vertex graphs find a clique with maximum weight
- Given the graph $G(\tau_1)$ and a function $w(v)$
- Eligibility: for a candidate solution X_1 :
$$\theta(X_1) = \forall u, v((X_1(v) \wedge X_1(u) \wedge u \neq v) \longrightarrow E(u, v))$$
- Optimality:

$$\max_{X_1} \sum_{X_1(v)} w(v)$$

Restating the Theorem

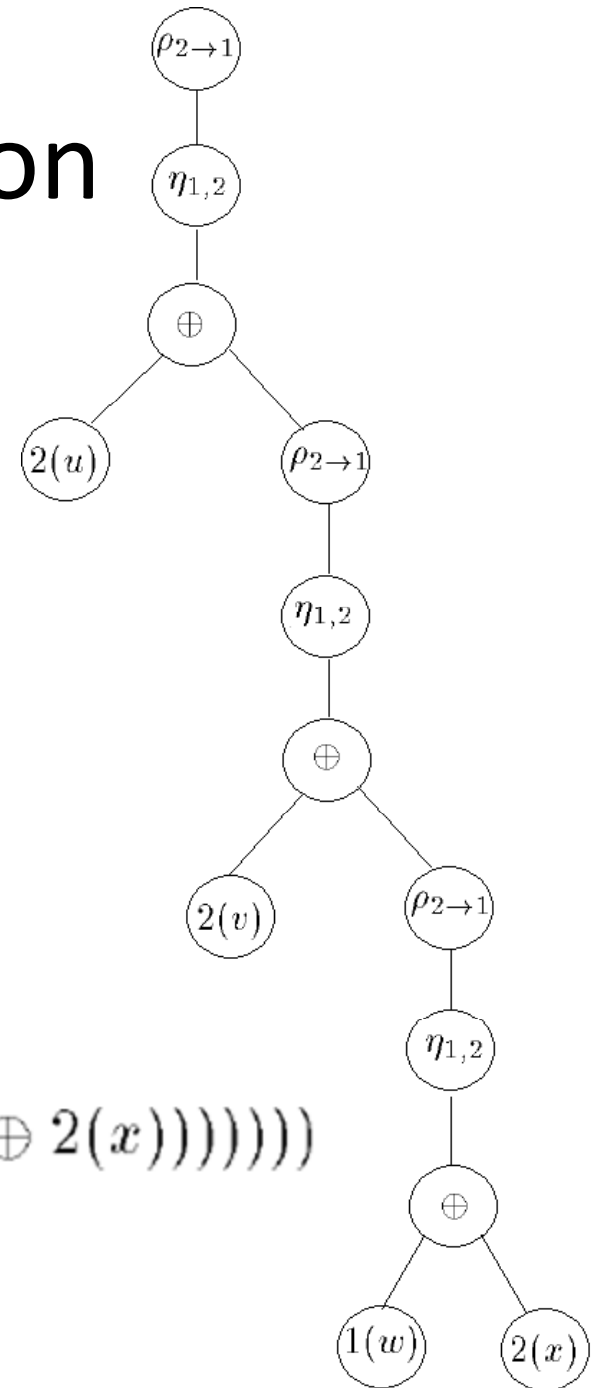
- In clique-width graphs almost all problems can be solved in linear time
- Given a clique-width graph $G(\tau_1)$ and appropriate clique-width decomposition, any LinEMSOL problem defined on G can be solved in Linear time

Clique-width decomposition

- $G \oplus H$: disjoint union
- $\eta_{i,j}(G)$: add edges
- $P_{i \rightarrow j}(G)$: change labels

For K_4 with vertices u, x, y, w

$$\rho_{2 \rightarrow 1}(\eta_{1,2}(2(u) \oplus \rho_{2 \rightarrow 1}(\eta_{1,2}(2(v) \oplus \rho_{2 \rightarrow 1}(\eta_{1,2}(1(w) \oplus 2(x)))))))$$

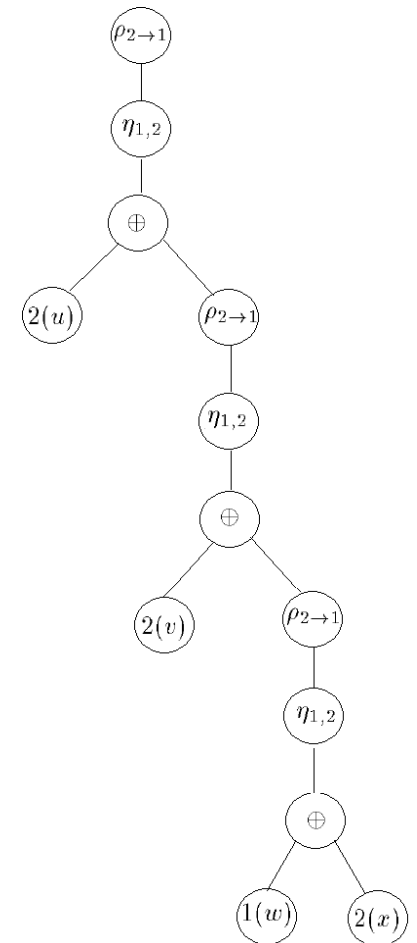
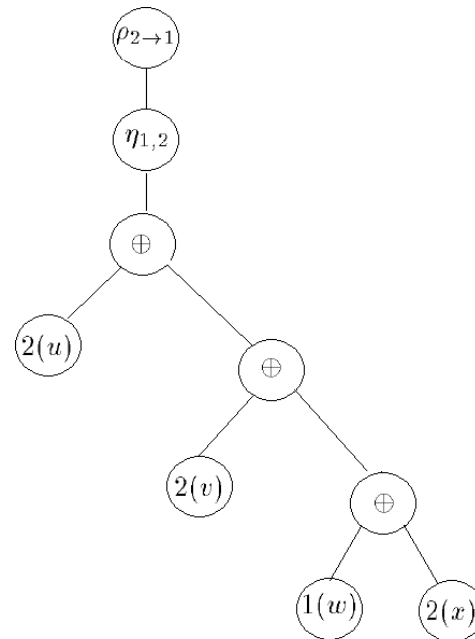


Hint: Tree Size

- Size of $T(G)$ is $O(|V|)$
 - remove redundancies in linear time

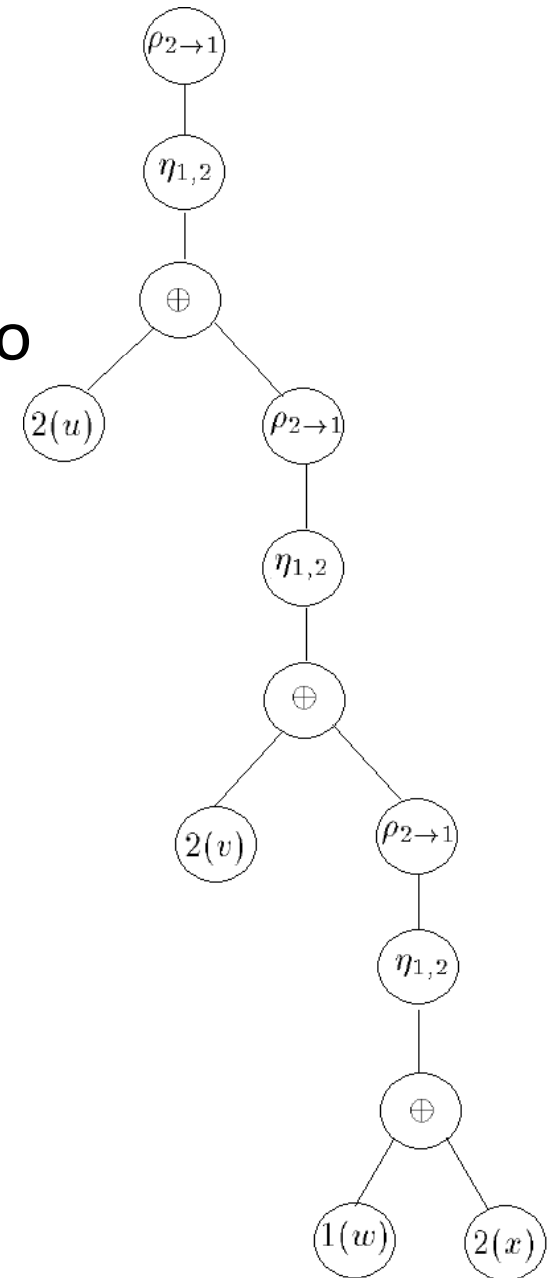
ex: $P_{i \rightarrow j}(P_{j \rightarrow i}(G)) = G$

$$\eta_{i,j}(\eta_{i,j}(G)) = \eta_{i,j}(G)$$



Algorithm: an sketch

- Assign a constant number of formulas to internal nodes (Top-Down manner)
 - Assign θ to the root
 - Use MSOL translation schemes to create formulas for unary nodes ($\eta_{i,j}$, $P_{i \rightarrow j}$)
 - For binary relation \oplus Break formulas to many (but constant) number of formulas
 - Feferman_Vaught theorem adopted to MSOL
- Use formulas to construct optimum solution in a (bottom-up manner)



Application

- Try to put the problem in LinEMSOL format
 - Graph as $G(\tau_1)$ structure
 - Define eligibility constraints with MSOL formula θ
 - No quantification on edges
 - Can associate values to vertices (and not edges)
- Try reduction to such a problem
- Still there are NP-Hard problems

Thank you

