Succinct Encodings of Graph Isomorphism

Bireswar Das[1], Patrick Scharpfenecker[2], Jacobo Torán[2]

[1]Indian Institute of Technology
Gandhinagar, India

[2]University of Ulm
Institute of Theoretical Computer Science
Ulm, Germany

11.3.2014

Introduction

Succinct Encodings for directed graph $G = (V, E)$:

- Boolean function $F(\overrightarrow{x}, \overrightarrow{y}) = 1$ iff $(\overrightarrow{x}, \overrightarrow{y}) \in E$
- $V \subseteq \{0, 1\}^n$
- F as circuit, boolean formula or OBDD

x	y	$F = \overline{x} \wedge y$
0	0	0
0	1	1
1	0	0
1	1	0

Introduction

Motivation:

- Many graphs in practice have small circuit-encodings.
- Is there a "nice" hidden structure?
- Graph-Problems not exponentially harder?

Introduction

Succinctly encoded problems are exponentially harder:
[Galperin, Wigderson 83]
[Papadimitriou, Yannakakis 86]
[Wagner 86]
[Balcázar, Lozano, Torán 92]
[Veith 97, 98]

Introduction

Upgrading Theorems for circuits:
Problem A hard for

- L $\Rightarrow$ cirA hard for PSPACE
- P $\Rightarrow$ cirA hard for EXP
- NP $\Rightarrow$ cirA hard for NEXP

Similar with general formulas and OBDDs.

Introduction

Upgrading Theorems for circuits:
Problem A hard for

- L $\Rightarrow$ cirA hard for PSPACE
- P $\Rightarrow$ cirA hard for EXP
- NP $\Rightarrow$ cirA hard for NEXP

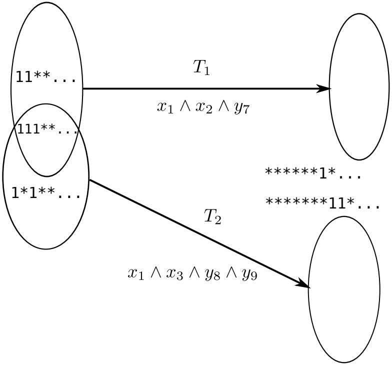
Similar with general formulas and OBDDs.

Our contribution: "new" CNF-, DNF-encodings

CNF and DNF Encodings

DNF $F(x_1, \ldots, x_n, y_1, \ldots, y_n) = (x_1 \wedge x_2 \wedge y_7) \vee (x_1 \wedge x_3 \wedge y_8 \wedge y_9)$:

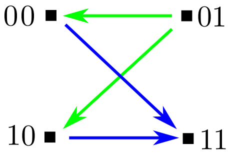
- $V = \{0,1\}^n$
- $(x, y) \in E$ iff $F(x, y) = 1$ iff x, y satisfy some term
- Terms define bicliques: nodes satisfying x part connected to nodes satisfying y part
- Graph is union of all bicliques



CNF and DNF Encodings

Example:

$$F = (\overline{x_1} \land x_2 \land \overline{y_2}) \lor (\overline{x_2} \land y_1 \land y_2)$$



CNF and DNF Encodings

CNF encoding is dual:

- Consider complete graph K_{2^n}.
- Every clause removes biclique violating this clause.

Succinct Gaps

Problem	**CNF encoding**	**DNF encoding**
Dominating Set	NEXP-complete	PP-complete
CNF-SAT	NEXP-complete	NP-complete
STCONN	PSPACE-complete	NL-complete

Succinct Gaps

Problem	**CNF encoding**	**DNF encoding**
Dominating Set	NEXP-complete	PP-complete
CNF-SAT	NEXP-complete	NP-complete
STCONN	PSPACE-complete	NL-complete

But: cirGI $\equiv$ cnfGI $\equiv$ dnfGI

Example: STCONN

CNF encoded:

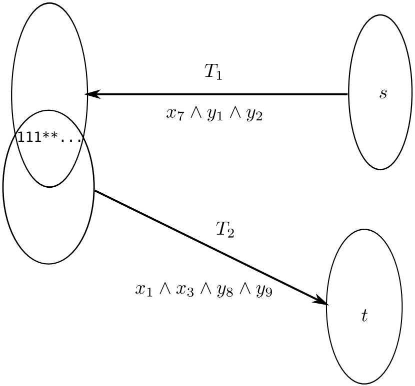
- cirSTCONN is PSPACE-complete
- Following reduction from cirGI to cnfGI maintains connectivity

Example: STCONN

CNF encoded:

- cirSTCONN is PSPACE-complete
- Following reduction from cirGI to cnfGI maintains connectivity

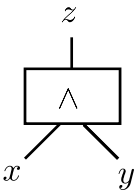
DNF encoded: Navigate through graph via biclique decomposition



Equivalence of Encodings for GI

Tseitin Transformation:

- Reduces Circuit-SAT to 3CNF
- New variable for every gate
- $C(x) = 1 \rightarrow F(x, z) = 1$
- x input, z encodes circuit-evaluation



$$(z \leftrightarrow (x \wedge y))$$
$$= (z \rightarrow x) \wedge (z \rightarrow y) \wedge (z \leftarrow (x \wedge y))$$
$$= (\overline{z} \vee x) \wedge (\overline{z} \vee y) \wedge (z \vee \overline{x} \vee \overline{y})$$

Equivalence of Encodings for GI

GI equal for DNF, CNF and circuits:

- $C(x, y)$ circuit encoding a graph.

Edge (x, y): $C(x, y) = 1$, gates of C evaluate to z

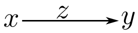
$$x \xrightarrow{z} y$$

Equivalence of Encodings for GI

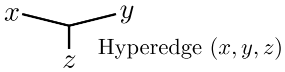
GI equal for DNF, CNF and circuits:

- $C(x, y)$ circuit encoding a graph.

Edge (x, y): $C(x, y) = 1$, gates of C evaluate to z



Via Tseitin Transformation: $F(x, y, z) = 1$

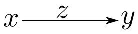


Equivalence of Encodings for GI

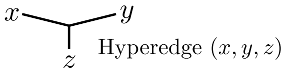
GI equal for DNF, CNF and circuits:

- $C(x, y)$ circuit encoding a graph.

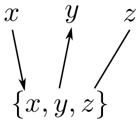
Edge (x, y): $C(x, y) = 1$, gates of C evaluate to z



Via Tseitin Transformation: $F(x, y, z) = 1$



Tranform to structure preserving graph (in CNF)



Equivalence of Encodings for GI

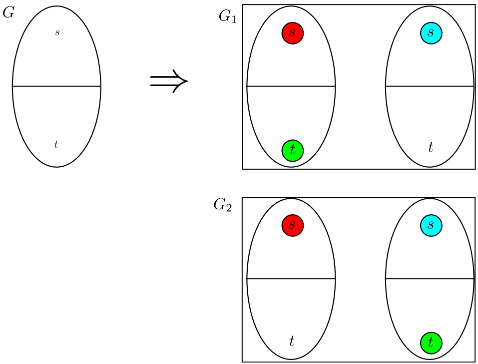
cirGI $\equiv$ cnfGI $\equiv$ dnfGI:

- Transformation showed cirGI $\leq$ cnfGI
- cnfGI $\leq$ cirGI trivial
- cnfGI $\equiv$ dnfGI: GI status remains under complementation

What is exact complexity for cirGI $\in$ *NEXP*?

cirGI is hard for PSPACE

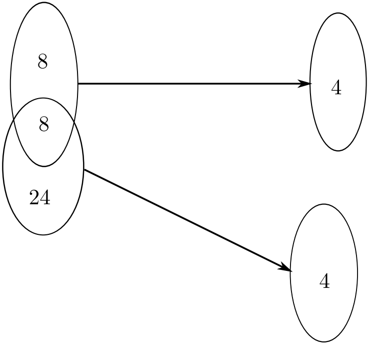
- Succinct versions hard for PSPACE
- Reduce undirected reachability to GI



- Reduction is easy (logtime for each edge)
- Upgrading Theorem proofs PSPACE $\leq_P$ cirGI

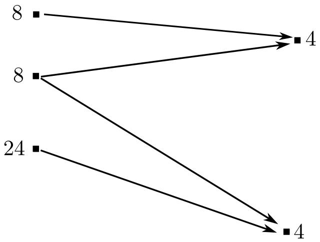
An algorithm for dnfGI

- DNF has s terms
- Graph is union of s bicliques
- Kernelize: 4^s nodes
- solve GI in $2^{\sqrt{s 2^{O(s)}}}$



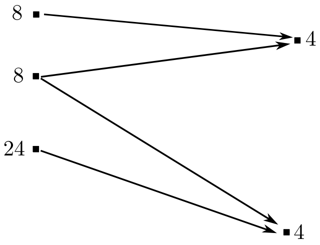
An algorithm for dnfGI

- DNF has s terms
- Graph is union of s bicliques
- Kernelize: 4^s nodes
- solve GI in $2^{\sqrt{s2^{O(s)}}}$



An algorithm for dnfGI

- DNF has s terms
- Graph is union of s bicliques
- Kernelize: 4^s nodes
- solve GI in $2^{\sqrt{s 2^{O(s)}}}$



$\Rightarrow$ Slightly better algorithm for dnf(GI) with o(n) terms

Conclusion

- New succinct encodings do not always give a complexity blow-up
- No difference for GI

Conclusion

- New succinct encodings do not always give a complexity blow-up
- No difference for GI

Additionally:

- PSPACE hardness even for obddGI
- OBDDs are circuits $\Rightarrow$ obddGI $\leq$ dnfGI

Conclusion

- New succinct encodings do not always give a complexity blow-up
- No difference for GI

Additionally:

- PSPACE hardness even for obddGI
- OBDDs are circuits $\Rightarrow$ obddGI $\leq$ dnfGI

Open Problems:

- Is dnfGI $\leq$ obddGI?
- Or is obddGI strictly easier?
- Better algorithms exploiting this structure?