Exponential Lower Bounds on the Space Complexity of Algorithms on OBDD-Represented Graphs

Daniel Sawitzki

University of Dortmund, Computer Science 2

LATIN 2006, Valdivia

Contents

Motivation

- Many applications produce very large graphs (CAD, Model Checking, WWW, ...)
  - Conflicts with memory limitations
  - Even polynomial algorithms not applicable
- Observation: Real-world graphs are often structured
- One of many heuristic approaches: Implicit algorithms
  - Do not enumerate nodes and edges explicitly
  - Consider graph G as characteristic Boolean function χ_G
  - Represent χ_G by a (hopefully) succinct data structure
  - Solve problems on G in terms of few and efficient operations on χ_G

Motivation

- Many applications produce very large graphs (CAD, Model Checking, WWW, . . .)
  - Conflicts with memory limitations
  - Even polynomial algorithms not applicable
- Observation: Real-world graphs are often structured
- One of many heuristic approaches: Implicit algorithms
  - Do not enumerate nodes and edges explicitly
  - Consider graph G as characteristic Boolean function χ_G
  - Represent χ_G by a (hopefully) succinct data structure
  - Solve problems on G in terms of few and efficient operations on χ_G

Motivation

- Many applications produce very large graphs (CAD, Model Checking, WWW, ...)
  - Conflicts with memory limitations
  - Even polynomial algorithms not applicable
- Observation: Real-world graphs are often structured
- One of many heuristic approaches: Implicit algorithms
  - Do not enumerate nodes and edges explicitly
  - Consider graph G as characteristic Boolean function χ_G
  - Represent χ_G by a (hopefully) succinct data structure
  - Solve problems on G in terms of few and efficient operations on χ_G

Motivation

- Many applications produce very large graphs (CAD, Model Checking, WWW, ...)
  - Conflicts with memory limitations
  - Even polynomial algorithms not applicable
- Observation: Real-world graphs are often structured
- One of many heuristic approaches: Implicit algorithms
  - Do not enumerate nodes and edges explicitly
  - Consider graph G as characteristic Boolean function χ_G
  - Represent χ_G by a (hopefully) succinct data structure
  - Solve problems on G in terms of few and efficient operations on χ_G

Motivation

- Many applications produce very large graphs (CAD, Model Checking, WWW, ...)
  - Conflicts with memory limitations
  - Even polynomial algorithms not applicable
- Observation: Real-world graphs are often structured
- One of many heuristic approaches: Implicit algorithms
  - Do not enumerate nodes and edges explicitly
  - Consider graph G as characteristic Boolean function χ_G
  - Represent χ_G by a (hopefully) succinct data structure
  - Solve problems on G in terms of few and efficient operations on χ_G

Motivation

- Many applications produce very large graphs (CAD, Model Checking, WWW, . . .)
  - Conflicts with memory limitations
  - Even polynomial algorithms not applicable
- Observation: Real-world graphs are often structured
- One of many heuristic approaches: Implicit algorithms
  - Do not enumerate nodes and edges explicitly
  - Consider graph G as characteristic Boolean function χ_G
  - Represent χ_G by a (hopefully) succinct data structure
  - Solve problems on G in terms of few and efficient operations on χ_G

Motivation

- Many applications produce very large graphs (CAD, Model Checking, WWW, ...)
  - Conflicts with memory limitations
  - Even polynomial algorithms not applicable
- Observation: Real-world graphs are often structured
- One of many heuristic approaches: Implicit algorithms
  - Do not enumerate nodes and edges explicitly
  - Consider graph G as characteristic Boolean function χ_G
  - Represent χ_G by a (hopefully) succinct data structure
  - Solve problems on G in terms of few and efficient operations on χ_G

Motivation

- Many applications produce very large graphs (CAD, Model Checking, WWW, ...)
  - Conflicts with memory limitations
  - Even polynomial algorithms not applicable
- Observation: Real-world graphs are often structured
- One of many heuristic approaches: Implicit algorithms
  - Do not enumerate nodes and edges explicitly
  - Consider graph G as characteristic Boolean function χ_G
  - Represent χ_G by a (hopefully) succinct data structure
  - Solve problems on G in terms of few and efficient operations on χ_G

Motivation

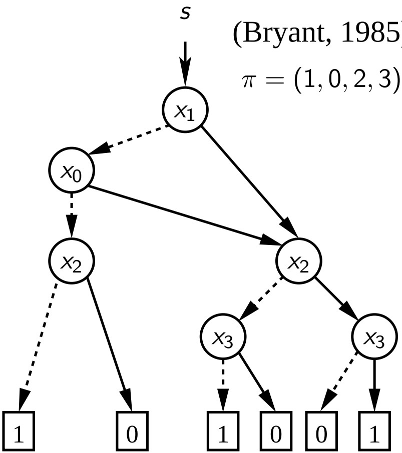
- Many applications produce very large graphs (CAD, Model Checking, WWW, ...)
  - Conflicts with memory limitations
  - Even polynomial algorithms not applicable
- Observation: Real-world graphs are often structured
- One of many heuristic approaches: Implicit algorithms
  - Do not enumerate nodes and edges explicitly
  - Consider graph G as characteristic Boolean function χ_G
  - Represent χ_G by a (hopefully) succinct data structure
  - Solve problems on G in terms of few and efficient operations on χ_G

Ordered Binary Decision Diagrams (OBDDs)

- Data structure for $f\colon \{0,1\}^m \to \{0,1\}$ with vars. $x_0, \dots, x_{m-1} \in \{0,1\}$
- OBDD $\mathcal{G}_f$ is acyclic digraph having inner nodes and sinks
- Inner nodes: Variable label, 0- and 1-edge
- Sink represents value $f(x_0, \dots, x_{m-1})$
- Source pointer s
- Reads vars. w. r. t. $\pi \in \Sigma_m$

(Bryant, 1985)

$\pi = (1, 0, 2, 3)$

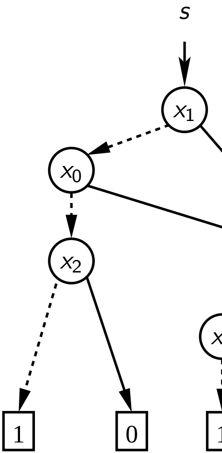


Ordered Binary Decision Diagrams (OBDDs)

- Data structure for $f\colon \{0,1\}^m \to \{0,1\}$ with vars. $x_0, \ldots, x_{m-1} \in \{0,1\}$
- OBDD $\mathcal{G}_f$ is acyclic digraph having inner nodes and sinks
- Inner nodes: Variable label, 0- and 1-edge
- Sink represents value $f(x_0, \ldots, x_{m-1})$
- Source pointer s
- Reads vars. w. r. t. $\pi \in \Sigma_m$

(Bryant, 1985)

$\pi = (1, 0, 2, 3)$

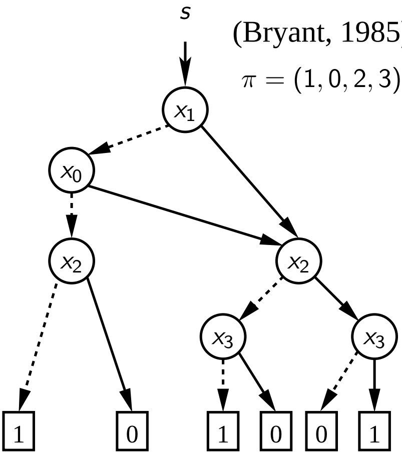


Ordered Binary Decision Diagrams (OBDDs)

- Data structure for $f\colon \{0,1\}^m \to \{0,1\}$ with vars. $x_0, \ldots, x_{m-1} \in \{0,1\}$
- OBDD $\mathcal{G}_f$ is acyclic digraph having inner nodes and sinks
- Inner nodes: Variable label, 0- and 1-edge
- Sink represents value $f(x_0, \ldots, x_{m-1})$
- Source pointer s
- Reads vars. w. r. t. $\pi \in \Sigma_m$

(Bryant, 1985)

$\pi = (1, 0, 2, 3)$

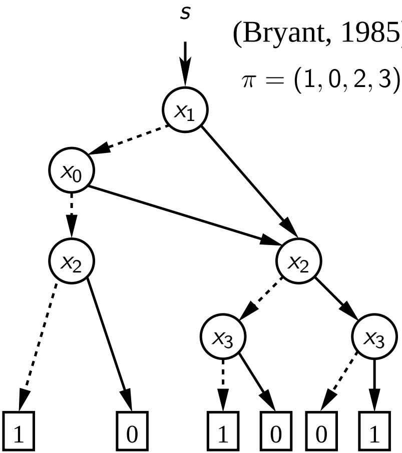


Ordered Binary Decision Diagrams (OBDDs)

- Data structure for $f\colon \{0,1\}^m \longrightarrow \{0,1\}$ with vars. $x_0, \ldots, x_{m-1} \in \{0,1\}$
- OBDD $\mathcal{G}_f$ is acyclic digraph having inner nodes and sinks
- Inner nodes: Variable label, 0- and 1-edge
- Sink represents value $f(x_0, \ldots, x_{m-1})$
- Source pointer s
- Reads vars. w. r. t. $\pi \in \Sigma_m$

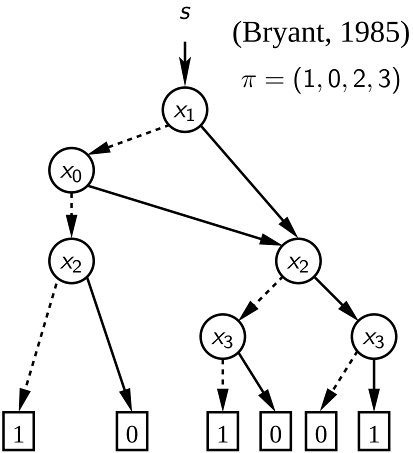
(Bryant, 1985)

$\pi = (1, 0, 2, 3)$



Ordered Binary Decision Diagrams (OBDDs)

- Data structure for $f\colon \{0,1\}^m \to \{0,1\}$ with vars. $x_0, \ldots, x_{m-1} \in \{0,1\}$
- OBDD $\mathcal{G}_f$ is acyclic digraph having inner nodes and sinks
- Inner nodes: Variable label, 0- and 1-edge
- Sink represents value $f(x_0, \ldots, x_{m-1})$
- Source pointer s
- Reads vars. w. r. t. $\pi \in \Sigma_m$

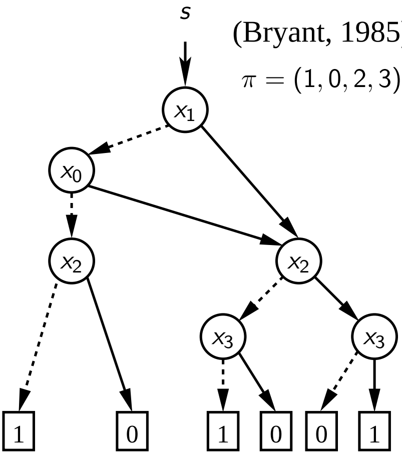


Ordered Binary Decision Diagrams (OBDDs)

- Data structure for $f\colon \{0,1\}^m \to \{0,1\}$ with vars. $x_0, \dots, x_{m-1} \in \{0,1\}$
- OBDD $\mathcal{G}_f$ is acyclic digraph having inner nodes and sinks
- Inner nodes: Variable label, 0- and 1-edge
- Sink represents value $f(x_0, \dots, x_{m-1})$
- Source pointer s
- Reads vars. w. r. t. $\pi \in \Sigma_m$

(Bryant, 1985)

$\pi = (1, 0, 2, 3)$



Implicit Graph Algorithms

- Consider graph $G = (V, E)$ with $V = (v_0, \ldots, v_{2^n-1})$ and $E \subseteq V^2$
- Implicit input: OBDD for $\chi_G \colon \{0,1\}^{2n} \to \{0,1\}$ with

$$\chi_G(x, y) = 1 \;:\Leftrightarrow\; (v_x, v_y) \in E$$

  for $x, y \in \{0,1\}^n$
- Solve problem on G by OBDD operations on χ_G:
  - Synthesis: $\wedge$, $\vee$, $\oplus$, $=$, $\ldots$
  - Quantification: $(\exists/\forall x_i)\, f(x_0, \ldots, x_{n-1})$
  - Satisfiability, equivalence, variable replacement $f_{|x_i=0/1}$
- Process large implicit sets; avoid touching many single nodes/edges

Implicit Graph Algorithms

- Consider graph $G = (V, E)$ with $V = (v_0, \ldots, v_{2^n-1})$ and $E \subseteq V^2$
- Implicit input: OBDD for $\chi_G \colon \{0,1\}^{2n} \to \{0,1\}$ with

$$\chi_G(x, y) = 1 \;:\Leftrightarrow\; (v_x, v_y) \in E$$

  for $x, y \in \{0,1\}^n$
- Solve problem on G by OBDD operations on χ_G:
  - Synthesis: $\wedge$, $\vee$, $\oplus$, $=$, $\ldots$
  - Quantification: $(\exists/\forall x_i)\, f(x_0, \ldots, x_{n-1})$
  - Satisfiability, equivalence, variable replacement $f_{|x_i=0/1}$
- Process large implicit sets; avoid touching many single nodes/edges

Implicit Graph Algorithms

- Consider graph $G = (V, E)$ with $V = (v_0, \ldots, v_{2^n-1})$ and $E \subseteq V^2$
- Implicit input: OBDD for $\chi_G \colon \{0,1\}^{2n} \to \{0,1\}$ with

$$\chi_G(x, y) = 1 \;:\Leftrightarrow\; (v_x, v_y) \in E$$

  for $x, y \in \{0,1\}^n$
- Solve problem on G by OBDD operations on χ_G:
  - Synthesis: $\wedge$, $\vee$, $\oplus$, $=$, $\ldots$
  - Quantification: $(\exists/\forall x_i)\, f(x_0, \ldots, x_{n-1})$
  - Satisfiability, equivalence, variable replacement $f_{|x_i=0/1}$
- Process large implicit sets; avoid touching many single nodes/edges

Implicit Graph Algorithms

- Consider graph $G = (V, E)$ with $V = (v_0, \ldots, v_{2^n-1})$ and $E \subseteq V^2$
- Implicit input: OBDD for $\chi_G \colon \{0,1\}^{2n} \to \{0,1\}$ with

$$\chi_G(x, y) = 1 \ :\Leftrightarrow \ (v_x, v_y) \in E$$

  for $x, y \in \{0,1\}^n$
- Solve problem on G by OBDD operations on χ_G:
  - Synthesis: $\wedge$, $\vee$, $\oplus$, $=$, ...
  - Quantification: $(\exists/\forall x_i)\ f(x_0, \ldots, x_{n-1})$
  - Satisfiability, equivalence, variable replacement $f_{|x_i=0/1}$
- Process large implicit sets; avoid touching many single nodes/edges

Implicit Graph Algorithms

- Consider graph $G = (V, E)$ with $V = (v_0, \ldots, v_{2^n-1})$ and $E \subseteq V^2$
- Implicit input: OBDD for $\chi_G \colon \{0,1\}^{2n} \to \{0,1\}$ with

$$\chi_G(x, y) = 1 \;:\Leftrightarrow\; (v_x, v_y) \in E$$

  for $x, y \in \{0,1\}^n$
- Solve problem on G by OBDD operations on χ_G:
  - Synthesis: $\wedge$, $\vee$, $\oplus$, $=$, ...
  - Quantification: $(\exists/\forall x_i)\ f(x_0, \ldots, x_{n-1})$
  - Satisfiability, equivalence, variable replacement $f_{|x_i=0/1}$
- Process large implicit sets; avoid touching many single nodes/edges

Implicit Graph Algorithms

- Consider graph $G = (V, E)$ with $V = (v_0, \ldots, v_{2^n-1})$ and $E \subseteq V^2$
- Implicit input: OBDD for $\chi_G \colon \{0,1\}^{2n} \to \{0,1\}$ with

$$\chi_G(x, y) = 1 \;:\Leftrightarrow\; (v_x, v_y) \in E$$

  for $x, y \in \{0,1\}^n$
- Solve problem on G by OBDD operations on χ_G:
  - Synthesis: $\wedge$, $\vee$, $\oplus$, $=$, ...
  - Quantification: $(\exists/\forall x_i)\, f(x_0, \ldots, x_{n-1})$
  - Satisfiability, equivalence, variable replacement $f_{|x_i=0/1}$
- Process large implicit sets; avoid touching many single nodes/edges

Implicit Graph Algorithms

- Consider graph $G = (V, E)$ with $V = (v_0, \dots, v_{2^n-1})$ and $E \subseteq V^2$
- Implicit input: OBDD for $\chi_G \colon \{0,1\}^{2n} \to \{0,1\}$ with

$$\chi_G(x, y) = 1 \;:\Leftrightarrow\; (v_x, v_y) \in E$$

  for $x, y \in \{0,1\}^n$
- Solve problem on G by OBDD operations on χ_G:
  - Synthesis: $\wedge$, $\vee$, $\oplus$, $=$, ...
  - Quantification: $(\exists/\forall x_i)\ f(x_0, \dots, x_{n-1})$
  - Satisfiability, equivalence, variable replacement $f_{|x_i=0/1}$
- Process large implicit sets; avoid touching many single nodes/edges

Results

So far:

- Upper bounds on structured instances
- FPT-intractability unless P=PSPACE
- New results: Concrete lower bounds on OBDD sizes

Theorem

Maximum Flow, Shortest Paths, and Restricted Reachability on OBDD-represented graphs have exponential space complexity.

Consider OBDD $\chi_G(x, y)$ for $G = (V, E)$

Definition

Restricted Reachability: Compute $\{\chi_{R_i}(x)\}_i$ for

$$\chi_{R_i}(x) = 1 :\Leftrightarrow v_x \text{ reachable from } s \in V \text{ via at most } 2^i \text{ edges}$$

Results

So far:

- Upper bounds on structured instances
- FPT-intractability unless P=PSPACE
- New results: Concrete lower bounds on OBDD sizes

Theorem

Maximum Flow, Shortest Paths, and Restricted Reachability on OBDD-represented graphs have exponential space complexity.

Consider OBDD $\chi_G(x, y)$ for $G = (V, E)$

Definition

Restricted Reachability: Compute $\{\chi_{R_i}(x)\}_i$ for

$$\chi_{R_i}(x) = 1 :\Leftrightarrow v_x \text{ reachable from } s \in V \text{ via at most } 2^i \text{ edges}$$

Results

So far:

- Upper bounds on structured instances
- FPT-intractability unless P=PSPACE
- New results: Concrete lower bounds on OBDD sizes

Theorem

Maximum Flow, Shortest Paths, and Restricted Reachability on OBDD-represented graphs have exponential space complexity.

Consider OBDD $\chi_G(x, y)$ for $G = (V, E)$

Definition

Restricted Reachability: Compute $\{\chi_{R_i}(x)\}_i$ for

$$\chi_{R_i}(x) = 1 :\Leftrightarrow v_x \text{ reachable from } s \in V \text{ via at most } 2^i \text{ edges}$$

Results

So far:

- Upper bounds on structured instances
- FPT-intractability unless P=PSPACE
- New results: Concrete lower bounds on OBDD sizes

Theorem

Maximum Flow, Shortest Paths, and Restricted Reachability on OBDD-represented graphs have exponential space complexity.

Consider OBDD $\chi_G(x, y)$ for $G = (V, E)$

Definition

Restricted Reachability: Compute $\{\chi_{R_i}(x)\}_i$ for

$$\chi_{R_i}(x) = 1 :\Leftrightarrow v_x \text{ reachable from } s \in V \text{ via at most } 2^i \text{ edges}$$

Results

So far:

- Upper bounds on structured instances
- FPT-intractability unless P=PSPACE
- New results: Concrete lower bounds on OBDD sizes

Theorem

Maximum Flow, Shortest Paths, and Restricted Reachability on OBDD-represented graphs have exponential space complexity.

Consider OBDD $\chi_G(x, y)$ for $G = (V, E)$

Definition

Restricted Reachability: Compute $\{\chi_{R_i}(x)\}_i$ for

$$\chi_{R_i}(x) = 1 :\Leftrightarrow v_x \text{ reachable from } s \in V \text{ via at most } 2^i \text{ edges}$$

Contents

An Instance with Small OBDD Size

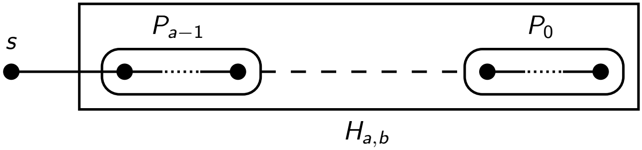
- Consider instance $G_m := (V_m, E_m)$
- G_m consists of 2^{2m} components $H_{a,b}$, $a, b \in \{0, \ldots, 2^m - 1\}$
- $H_{a,b} := P_{a-1} \ldots P_0$ for $P_i := (v_{i,b-1}, \ldots, v_{i,0})$
- $\Rightarrow H_{a,b}$ is path of length $a \cdot b - 1$
- G_m has OBDD χ_{G_m} of size $\mathcal{O}(m)$
- Uses interleaved variable order π_m on $8(m+1)$ variables

An Instance with Small OBDD Size

- Consider instance $G_m := (V_m, E_m)$
- G_m consists of 2^{2m} components $H_{a,b}$, $a, b \in \{0, \ldots, 2^m - 1\}$
- $H_{a,b} := P_{a-1} \ldots P_0$ for $P_i := (v_{i,b-1}, \ldots, v_{i,0})$
- $\Rightarrow H_{a,b}$ is path of length $a \cdot b - 1$
- G_m has OBDD χ_{G_m} of size $\mathcal{O}(m)$
- Uses interleaved variable order π_m on $8(m+1)$ variables

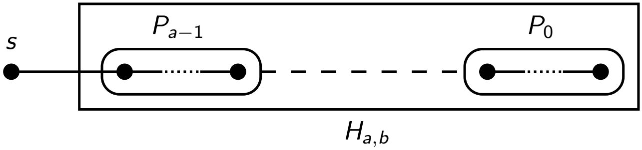
An Instance with Small OBDD Size

- Consider instance $G_m := (V_m, E_m)$
- G_m consists of 2^{2m} components $H_{a,b}$, $a, b \in \{0, \ldots, 2^m - 1\}$
- $H_{a,b} := P_{a-1} \ldots P_0$ for $P_i := (v_{i,b-1}, \ldots, v_{i,0})$
- $\Rightarrow H_{a,b}$ is path of length $a \cdot b - 1$
- G_m has OBDD χ_{G_m} of size $\mathcal{O}(m)$
- Uses interleaved variable order π_m on $8(m+1)$ variables



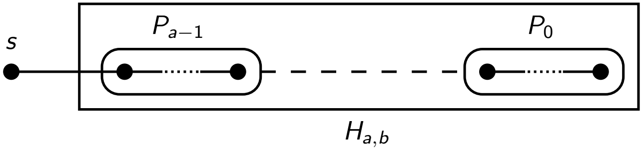
An Instance with Small OBDD Size

- Consider instance $G_m := (V_m, E_m)$
- G_m consists of 2^{2m} components $H_{a,b}$, $a, b \in \{0, \ldots, 2^m - 1\}$
- $H_{a,b} := P_{a-1} \ldots P_0$ for $P_i := (v_{i,b-1}, \ldots, v_{i,0})$
- $\Rightarrow H_{a,b}$ is path of length $a \cdot b - 1$
- G_m has OBDD χ_{G_m} of size $\mathcal{O}(m)$
- Uses interleaved variable order π_m on $8(m+1)$ variables



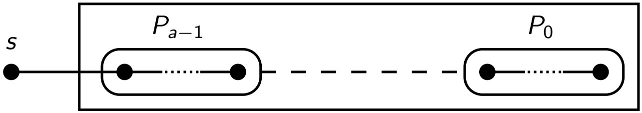
An Instance with Small OBDD Size

- Consider instance $G_m := (V_m, E_m)$
- G_m consists of 2^{2m} components $H_{a,b}$, $a, b \in \{0, \ldots, 2^m - 1\}$
- $H_{a,b} := P_{a-1} \ldots P_0$ for $P_i := (v_{i,b-1}, \ldots, v_{i,0})$
- $\Rightarrow H_{a,b}$ is path of length $a \cdot b - 1$
- G_m has OBDD χ_{G_m} of size $\mathcal{O}(m)$
- Uses interleaved variable order π_m on $8(m+1)$ variables



An Instance with Small OBDD Size

- Consider instance $G_m := (V_m, E_m)$
- G_m consists of 2^{2m} components $H_{a,b}$, $a, b \in \{0, \ldots, 2^m - 1\}$
- $H_{a,b} := P_{a-1} \ldots P_0$ for $P_i := (v_{i,b-1}, \ldots, v_{i,0})$
- $\Rightarrow H_{a,b}$ is path of length $a \cdot b - 1$
- G_m has OBDD χ_{G_m} of size $\mathcal{O}(m)$
- Uses interleaved variable order π_m on $8(m+1)$ variables



A Lower Bound for an Output OBDD

- Consider result function
  $\chi_{R_{2m-1}}(a,b,i,j) = 1 \Leftrightarrow s \stackrel{2^{2m-1}}{\longrightarrow} v_{a,b,i,j}$
- Let f_m be subfunction of $\chi_{R_{2m-1}}$ for $i = j := 0$

$$f_m(a,b) = 1 \Leftrightarrow a \cdot b \leq 2^{2m-1}$$

- $\Rightarrow$ $MUL_{m,2m-1}$ is at most polynomially larger than f_m

$$MUL_{m,2m-1}(a,b) := \lfloor (a \cdot b)/2^{2m-1} \rfloor \Leftrightarrow a \cdot b \geq 2^{2m-1}$$

- Exponential lower bound on OBDD size of f_m follows from

A Lower Bound for an Output OBDD

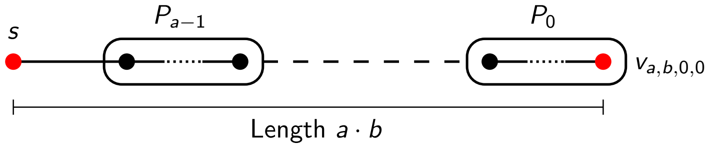
- Consider result function
  $\chi_{R_{2m-1}}(a,b,i,j) = 1 \ \Leftrightarrow\ s \overset{2^{2m-1}}{\longrightarrow} v_{a,b,i,j}$
- Let f_m be subfunction of $\chi_{R_{2m-1}}$ for $i = j := 0$

$$f_m(a,b) = 1 \ \Leftrightarrow\ a \cdot b \le 2^{2m-1}$$

- $\Rightarrow$ $MUL_{m,2m-1}$ is at most polynomially larger than f_m

$$MUL_{m,2m-1}(a,b) := \lfloor (a \cdot b)/2^{2m-1} \rfloor \ \Leftrightarrow\ a \cdot b \ge 2^{2m-1}$$

- Exponential lower bound on OBDD size of f_m follows from



A Lower Bound for an Output OBDD

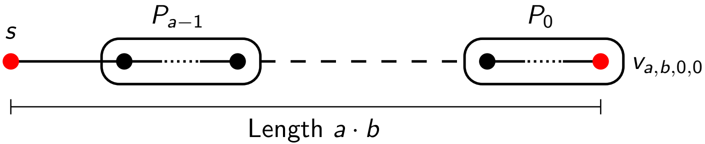
- Consider result function
  $\chi_{R_{2m-1}}(a,b,i,j) = 1 \;\Leftrightarrow\; s \overset{2^{2m-1}}{\longrightarrow} v_{a,b,i,j}$
- Let f_m be subfunction of $\chi_{R_{2m-1}}$ for $i = j := 0$

$$f_m(a,b) = 1 \;\Leftrightarrow\; a \cdot b \leq 2^{2m-1}$$

- $\Rightarrow$ $MUL_{m,2m-1}$ is at most polynomially larger than f_m

$$MUL_{m,2m-1}(a,b) := \lfloor (a \cdot b)/2^{2m-1} \rfloor \;\Leftrightarrow\; a \cdot b \geq 2^{2m-1}$$

- Exponential lower bound on OBDD size of f_m follows from



A Lower Bound for an Output OBDD

- Consider result function
  $\chi_{R_{2m-1}}(a,b,i,j) = 1 \Leftrightarrow s \overset{2^{2m-1}}{\longrightarrow} v_{a,b,i,j}$
- Let f_m be subfunction of $\chi_{R_{2m-1}}$ for $i = j := 0$

$$f_m(a,b) = 1 \Leftrightarrow a \cdot b \leq 2^{2m-1}$$

- $\Rightarrow MUL_{m,2m-1}$ is at most polynomially larger than f_m

$$MUL_{m,2m-1}(a,b) := \lfloor (a \cdot b)/2^{2m-1} \rfloor \Leftrightarrow a \cdot b \geq 2^{2m-1}$$

- Exponential lower bound on OBDD size of f_m follows from

Theorem

The π_m-OBDD size of $MUL_{m,2m-1}$ is at least $2^{(m-5)/6}$.

Contents

Lower Bound Technique

For $a, b \in \{0,1\}^m$:

$$MUL_{m,2m-1}(a,b) = 1 \;:\Leftrightarrow\; a \cdot b \geq 2^{2m-1}$$

- Let $m = 6k$,
  $a_L := a \bmod 2^k$, $a_H := \lfloor a/2^k \rfloor$,
  $b_L := b \bmod 2^k$, $b_H := \lfloor b/2^k \rfloor$
- Fix $b_L := 0$
- Lower bound $\#a_L$-subfunctions

Lower Bound Technique

For $a, b \in \{0,1\}^m$:

$$MUL_{m,2m-1}(a,b) = 1 \;:\Leftrightarrow\; a \cdot b \geq 2^{2m-1}$$

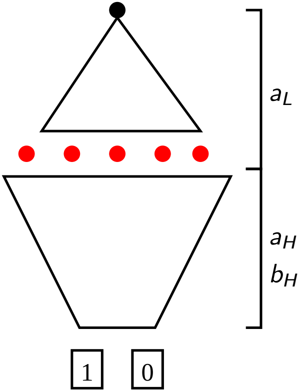
- Let $m = 6k$,
  $a_L := a \bmod 2^k$, $a_H := \lfloor a/2^k \rfloor$,
  $b_L := b \bmod 2^k$, $b_H := \lfloor b/2^k \rfloor$
- Fix $b_L := 0$
- Lower bound $\#a_L$-subfunctions

Lower Bound Technique

For $a, b \in \{0,1\}^m$:

$$MUL_{m,2m-1}(a,b) = 1 \;:\Leftrightarrow\; a \cdot b \geq 2^{2m-1}$$

- Let $m = 6k$,
  $a_L := a \bmod 2^k$, $a_H := \lfloor a/2^k \rfloor$,
  $b_L := b \bmod 2^k$, $b_H := \lfloor b/2^k \rfloor$
- Fix $b_L := 0$
- Lower bound $\#a_L$-subfunctions

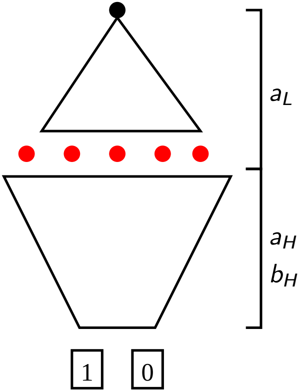


Lower Bound Technique

For $a, b \in \{0,1\}^m$:

$$MUL_{m,2m-1}(a,b) = 1 \;:\Leftrightarrow\; a \cdot b \geq 2^{2m-1}$$

- Let $m = 6k$,
  $a_L := a \bmod 2^k$, $a_H := \lfloor a/2^k \rfloor$,
  $b_L := b \bmod 2^k$, $b_H := \lfloor b/2^k \rfloor$
- Fix $b_L := 0$
- Lower bound $\#a_L$-subfunctions



Multiplication Lemma

Lemma

For $a_L < a'_L$ there are a_H and b_H such that $a \cdot b < 2^{2m-1} \leq a' \cdot b$.

- $\Rightarrow$ For $a_L \neq a'_L$ it is

$$MUL_{m,2m-1}(a_L) \neq MUL_{m,2m-1}(a'_L)$$

- $\Rightarrow$ Fixing a_L yields $2^k = 2^{m/6}$ different subfunctions

Theorem

The π_m-OBDD size of $MUL_{m,2m-1}$ is at least $2^{(m-5)/6}$.

Theorem

OBDD-based Restricted Reachability has exponential space complexity.

Multiplication Lemma

Lemma

For $a_L < a'_L$ there are a_H and b_H such that $a \cdot b < 2^{2m-1} \leq a' \cdot b$.

- $\Rightarrow$ For $a_L \neq a'_L$ it is
$$MUL_{m,2m-1}(a_L) \neq MUL_{m,2m-1}(a'_L)$$
- $\Rightarrow$ Fixing a_L yields $2^k = 2^{m/6}$ different subfunctions

Theorem

The π_m-OBDD size of $MUL_{m,2m-1}$ is at least $2^{(m-5)/6}$.

Theorem

OBDD-based Restricted Reachability has exponential space complexity.

Multiplication Lemma

Lemma

For $a_L < a'_L$ there are a_H and b_H such that $a \cdot b < 2^{2m-1} \leq a' \cdot b$.

- $\Rightarrow$ For $a_L \neq a'_L$ it is

$$MUL_{m,2m-1}(a_L) \neq MUL_{m,2m-1}(a'_L)$$

- $\Rightarrow$ Fixing a_L yields $2^k = 2^{m/6}$ different subfunctions

Theorem

The π_m-OBDD size of $MUL_{m,2m-1}$ is at least $2^{(m-5)/6}$.

Theorem

OBDD-based Restricted Reachability has exponential space complexity.

Multiplication Lemma

Lemma

For $a_L < a'_L$ there are a_H and b_H such that $a \cdot b < 2^{2m-1} \leq a' \cdot b$.

- $\Rightarrow$ For $a_L \neq a'_L$ it is
$$MUL_{m,2m-1}(a_L) \neq MUL_{m,2m-1}(a'_L)$$
- $\Rightarrow$ Fixing a_L yields $2^k = 2^{m/6}$ different subfunctions

Theorem

The π_m-OBDD size of $MUL_{m,2m-1}$ is at least $2^{(m-5)/6}$.

Theorem

OBDD-based Restricted Reachability has exponential space complexity.

Multiplication Lemma

Lemma

For $a_L < a'_L$ there are a_H and b_H such that $a \cdot b < 2^{2m-1} \leq a' \cdot b$.

- $\Rightarrow$ For $a_L \neq a'_L$ it is
$$MUL_{m,2m-1}(a_L) \neq MUL_{m,2m-1}(a'_L)$$
- $\Rightarrow$ Fixing a_L yields $2^k = 2^{m/6}$ different subfunctions

Theorem

The π_m-OBDD size of $MUL_{m,2m-1}$ is at least $2^{(m-5)/6}$.

Theorem

OBDD-based Restricted Reachability has exponential space complexity.

“That’s all Folks!”