Complexity Classes and Theories for the Comparator Circuit Value Problem

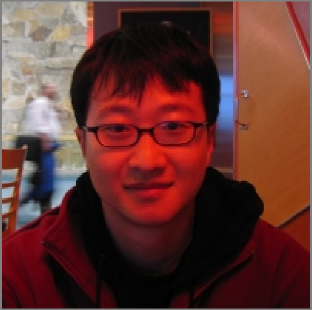
Dai Tri Man Lê

Joint work with Stephen Cook and Yuli Ye

University of Toronto
Canada

Prague Fall Logic School 2011

Stephen Cook ('68)



Yuli Ye

Bounded Reverse Mathematics [Cook-Nguyen '10]

Motivation

Classify theorems according to the computational complexity of concepts needed to prove them.

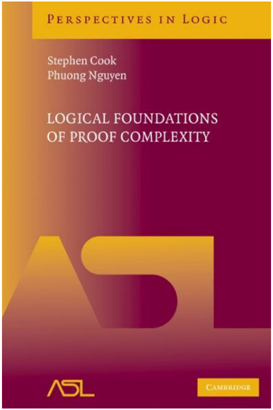
Program in Chapter 9

1. Introduce a general method for associating a canonical minimal theory VC for "nice" complexity classes C

$$AC^0 \subseteq C \subseteq P$$

2. Given a theorem τ, try to find the smallest complexity class C such that

$$VC \vdash \tau$$



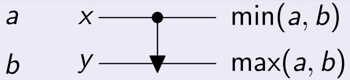
Outline of the talk

1. **The complexity classes for the Comparator Circuit Value Problem**
2. **Define a theory for CC^***
3. **Natural complete problems: stable marriage and lex-first maximal matching**
4. **Conclusion and open problems**

Comparator Circuits

- Originally invented for sorting, e.g.,
  - Ajtai-Komlós-Szemerédi (AKS) $\mathcal{O}(\log n)$-depth sorting networks ('83)
  - Formalized by Jeřábek ('11) in VNC^1_*.

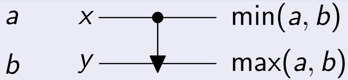
Comparator gate



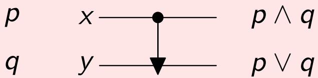
Comparator Circuits

- Originally invented for sorting, e.g.,
  - Ajtai-Komlós-Szemerédi (AKS) $\mathcal{O}(\log n)$-depth sorting networks ('83)
  - Formalized by Jeřábek ('11) in VNC^1_*.
- Can also be seen as boolean circuits.

Comparator gate



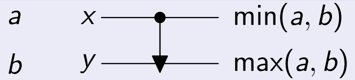
Boolean comparator gate



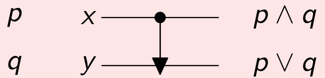
Comparator Circuits

- Originally invented for sorting, e.g.,
  - Ajtai-Komlós-Szemerédi (AKS) $\mathcal{O}(\log n)$-depth sorting networks ('83)
  - Formalized by Jeřábek ('11) in VNC^1_*.
- Can also be seen as boolean circuits.

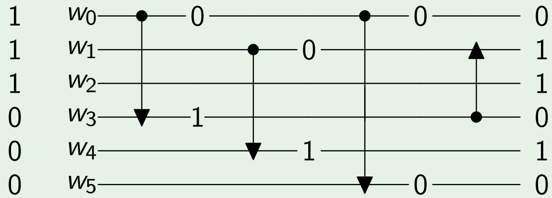
Comparator gate



Boolean comparator gate

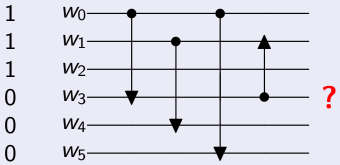


Example



Comparator Circuit Value (CCV) Problem (decision)

Given a comparator circuit with specified Boolean inputs, determine the output value of a designated wire.

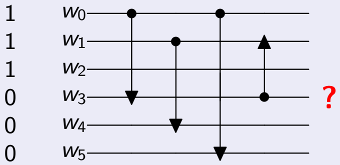


Complexity classes

1. $\mathsf{CC}^{\mathsf{Subr}} = \{$decision problems log-space many-one-reducible to CCV$\}$
   - [Subramanian '90], [Mayr-Subramanian '92]

Comparator Circuit Value (Ccv) Problem (decision)

Given a comparator circuit with specified Boolean inputs, determine the output value of a designated wire.

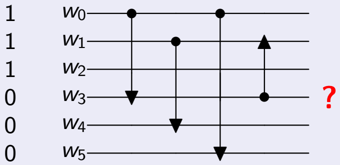


Complexity classes

1. $\text{CC}^{\text{Subr}} = \{$decision problems log-space many-one-reducible to Ccv$\}$
   - [Subramanian '90], [Mayr-Subramanian '92]
2. $\text{CC} = \{$decision problems AC^0 many-one-reducible to Ccv$\}$
   - Complete problems: stable marriage, lex-first maximal matching...
3. $\text{CC}^* = \{$decision problems AC^0 oracle-reducible to Ccv$\}$
   - Needed when developing a Cook-Nguyen style theory for CC
   - The function class FCC^* is closed under compostion

Comparator Circuit Value (CCV) Problem (decision)

Given a comparator circuit with specified Boolean inputs, determine the output value of a designated wire.

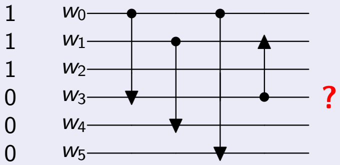


Complexity classes

1. $\mathsf{CC}^{\mathsf{Subr}} = \{$decision problems log-space many-one-reducible to $\mathrm{CCV}\}$
   - [Subramanian '90], [Mayr-Subramanian '92]
2. $\mathsf{CC} = \{$decision problems AC^0 many-one-reducible to $\mathrm{CCV}\}$
   - Complete problems: stable marriage, lex-first maximal matching...
3. $\mathsf{CC}^* = \{$decision problems AC^0 oracle-reducible to $\mathrm{CCV}\}$
   - Needed when developing a Cook-Nguyen style theory for CC
   - The function class FCC^* is closed under compostion

Comparator Circuit Value (Ccv) Problem (decision)

Given a comparator circuit with specified Boolean inputs, determine the output value of a designated wire.



Complexity classes

1. $\mathrm{CC}^{\mathrm{Subr}} = \{$decision problems log-space many-one-reducible to Ccv$\}$
   - [Subramanian '90], [Mayr-Subramanian '92]
2. $\mathrm{CC} = \{$decision problems AC^0 many-one-reducible to Ccv$\}$
   - Complete problems: stable marriage, lex-first maximal matching...
3. $\mathrm{CC}^* = \{$decision problems AC^0 oracle-reducible to Ccv$\}$
   - Needed when developing a Cook-Nguyen style theory for CC
   - The function class FCC^* is closed under compostion

$$\mathrm{NC}^1 \subseteq \mathrm{NL} \subseteq \mathrm{CC} \subseteq \mathrm{CC}^{\mathrm{Subr}} \subseteq \mathrm{CC}^* \subseteq \mathrm{P}$$

1. The complexity classes for the Comparator Circuit Value Problem
2. **Define a theory for CC^***
3. Natural complete problems: stable marriage and lex-first maximal matching
4. Conclusion and open problems

Two-sorted language $\mathcal{L}_A^2$ (Zambella '96)

Vocabulary $\mathcal{L}_A^2 = [0, 1, +, \cdot, |\ |; \in, \leq, =_1, =_2]$

- Standard model $\mathbb{N}_2 = \langle \mathbb{N}, \text{finite subsets of } \mathbb{N} \rangle$
- $0, 1, +, \cdot, \leq, =$ have usual meaning over $\mathbb{N}$
- $|X| =$ length of X
- Set membership $y \in X$

- "number" variables $x, y, z, \ldots$ (range over $\mathbb{N}$)
- "string" variables $X, Y, Z, \ldots$ (range over finite subsets of $\mathbb{N}$)
- Number terms are built from $x, y, z, \ldots, 0, 1, +, \cdot$ and $|X|, |Y|, |Z|, \ldots$
- The only string terms are variable $X, Y, Z, \ldots$

Two-sorted language $\mathcal{L}_A^2$ (Zambella '96)

Vocabulary $\mathcal{L}_A^2 = [0, 1, +, \cdot, |\ |; \in, \leq, =_1, =_2]$

- Standard model $\mathbb{N}_2 = \langle \mathbb{N}, \text{finite subsets of } \mathbb{N} \rangle$
- $0, 1, +, \cdot, \leq, =$ have usual meaning over $\mathbb{N}$
- $|X| =$ length of X
- Set membership $y \in X$

Note

The natural inputs for Turing machines and circuits are finite strings.

- "number" variables $x, y, z, \ldots$ (range over $\mathbb{N}$)
- "string" variables $X, Y, Z, \ldots$ (range over finite subsets of $\mathbb{N}$)
- Number terms are built from $x, y, z, \ldots, 0, 1, +, \cdot$ and $|X|, |Y|, |Z|, \ldots$
- The only string terms are variable $X, Y, Z, \ldots$

Two-sorted language $\mathcal{L}_A^2$ (Zambella '96)

Vocabulary $\mathcal{L}_A^2 = [0, 1, +, \cdot, |\ |; \in, \leq, =_1, =_2]$

- Standard model $\mathbb{N}_2 = \langle \mathbb{N}, \text{finite subsets of } \mathbb{N} \rangle$
- $0, 1, +, \cdot, \leq, =$ have usual meaning over $\mathbb{N}$
- $|X|$ = length of X
- Set membership $y \in X$

Note

The natural inputs for Turing machines and circuits are finite strings.

- "number" variables $x, y, z, \ldots$ (range over $\mathbb{N}$)
- "string" variables $X, Y, Z, \ldots$ (range over finite subsets of $\mathbb{N}$)
- Number terms are built from $x, y, z, \ldots, 0, 1, +, \cdot$ and $|X|, |Y|, |Z|, \ldots$
- The only string terms are variable $X, Y, Z, \ldots$

Definition (Σ_0^B formula)

1. All the number quantifiers are bounded.
2. No string quantifiers (free string variables are allowed)

Two-sorted complexity classes

A two-sorted complexity class consists of relations $R(\vec{x}, \vec{X})$, where

- $\vec{x}$ are number arguments (in unary) and $\vec{X}$ are string arguments

Definition (Two-sorted AC^0)

A relation $R(\vec{x}, \vec{X})$ is in AC^0 iff some alternating Turing machine accepts R in time $\mathcal{O}(\log n)$ with a constant number of alternations.

Σ_0^B-Representation Theorem [Zambella '96, Cook-Nguyen]

$R(\vec{x}, \vec{X})$ is in AC^0 iff it is represented by a Σ_0^B-formula $\varphi(\vec{x}, \vec{X})$.

Useful consequences

1. Don't need to work with uniform circuit families or alternating Turing machines when defining AC^0 functions or relations.
2. Useful when working with AC^0-reductions

The theory V^0 for AC^0 reasoning

The theory V^0

1. **2-BASIC axioms**: essentially the axioms of Robinson arithmetic plus
   - the defining axioms for $\leq$ and the string length function $|\;|$
   - the axiom of extensionality for finite sets (bit strings).
2. Σ_0^B**-COMP** (Comprehension): for every Σ_0^B-formula $\varphi(z)$ without X,

$$\exists X \leq y \forall z < y \big(X(z) \leftrightarrow \varphi(z)\big)$$

Theorem

1. Σ_0^B**-IND:** *for* $\varphi \in \Sigma_0^B$

$$\Big[\varphi(0) \wedge \forall x \big(\varphi(x) \rightarrow \varphi(x+1)\big)\Big] \rightarrow \forall x \varphi(x)$$

2. *The provably total functions in* V^0 *are precisely* FAC^0.

Note: Theories, developed using Cook-Nguyen method, extend V^0.

The 2-BASIC axioms

B1. $x + 1 \neq 0$

B2. $x + 1 = y + 1 \to x = y$

B3. $x + 0 = x$

B4. $x + (y + 1) = (x + y) + 1$

B5. $x \cdot 0 = 0$

B6. $x \cdot (y + 1) = (x \cdot y) + x$

B7. $(x \leq y \wedge y \leq x) \to x = y$

B8. $x \leq x + y$

B9. $0 \leq x$

B10. $x \leq y \vee y \leq x$

B11. $x \leq y \leftrightarrow x < y + 1$

B12. $x \neq 0 \to \exists y \leq x\,(y + 1 = x)$

L1. $X(y) \to y < |X|$

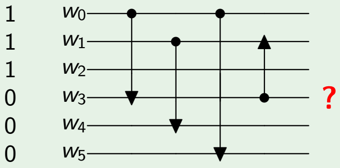
L2. $y + 1 = |X| \to X(y)$

SE. $\Big[|X| = |Y| \wedge \forall i < |X|\big(X(i) = Y(i)\big)\Big] \to X = Y$

The theory VCC* for CC*

Comparator Circuit Value (Ccv) Problem (decision)

- Given a comparator circuit with specified Boolean inputs
- Determine the output value of a designated wire.

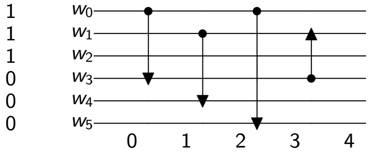


Recall that $\mathrm{CC}^* = \{$decision problems AC^0 oracle-reducible to Ccv$\}$

The two-sorted theory VCC* [using the Cook-Nguyen method]

- VCC^* has vocabulary $\mathcal{L}_A^2$
- Axiom of VCC^* = Axiom of V^0 + one additional axiom asserting the existence of a solution to the Ccv problem.

Asserting the existence of a solution to Ccv



- X encodes a comparator circuit with m wires and n gates
- Y encodes the input sequence
- Z is an $(n+1) \times m$ matrix, where column i of Z encodes values layer i

The following Σ_0^B formula $\delta_{\mathrm{CCV}}(m, n, X, Y, Z)$ states that Z encodes the correct values of all the layers of the Ccv instance encoded in X and Y:

$$\forall k < m\big(Y(k) \leftrightarrow Z(0,k)\big) \land \forall i < n \forall x < m \forall y < m,$$

$$(X)^i = \langle x, y\rangle \rightarrow \left[\begin{array}{ll} & Z(i+1,x) \leftrightarrow \big(Z(i,x) \land Z(i,y)\big) \\ \land & Z(i+1,y) \leftrightarrow \big(Z(i,x) \lor Z(i,y)\big) \\ \land & \forall j < m\Big[(j \neq x \land j \neq y) \rightarrow \big(Z(i+1,j) \leftrightarrow Z(i,j)\big)\Big] \end{array}\right]$$

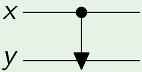
$$\mathrm{VCC}^* = \mathrm{V}^0 + \exists Z \leq \langle m, n+1\rangle + 1,\ \delta_{\mathrm{CCV}}(m, n, X, Y, Z)$$

Inclusion of theories

- Recall that:

$$\boxed{AC^0 \subseteq TC^0 \subseteq NC^1 \subseteq NL \subseteq CC \subseteq CC^{Subr} \subseteq CC^* \subseteq P}$$

Comparator gate

p		$p \wedge q$
q		$p \vee q$

Inclusion of theories

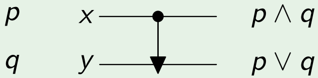
- Recall that:

$$\mathsf{AC}^0 \subseteq \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{NL} \subseteq \mathsf{CC} \subseteq \mathsf{CC}^{\mathsf{Subr}} \subseteq \mathsf{CC}^* \subseteq \mathsf{P}$$

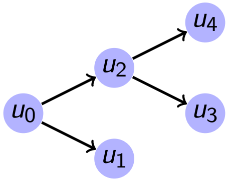
- We showed in our paper that:

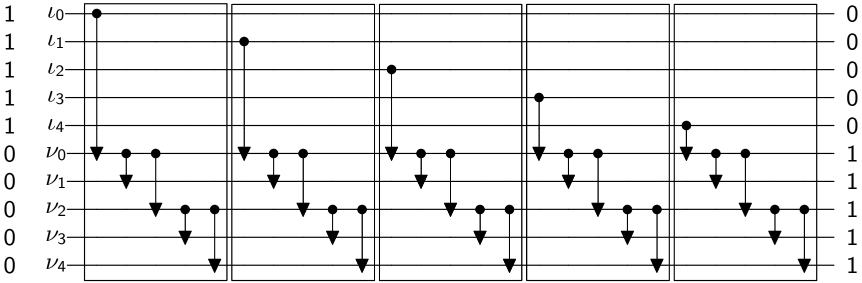
$$\mathsf{VTC}^0 \subseteq \mathsf{VNC}^1 \subseteq \mathsf{VNL} \subseteq \mathsf{VCC}^* \subseteq \mathsf{VP}$$

Comparator gate

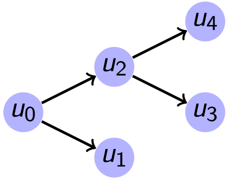


VNL ⊆ VCC*


u_0
u_1
u_2
u_3
u_4


ι_0
ι_1
ι_2
ι_3
ι_4
ν_0
ν_1
ν_2
ν_3
ν_4
1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1

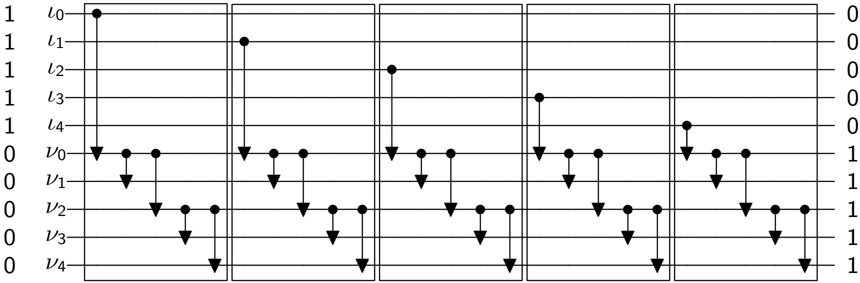
VNL ⊆ VCC*



- Can't talk about reachability!
- Known fact:

$$\mathsf{VTC}^0 \subseteq \mathsf{VNC}^1 \subseteq \mathsf{VCC}^*$$

- We prove the correctness of this construction using only counting.



Stable Marriage Problem (search version) (Gale-Shapley '62)

- Given n men and n women together with their preference lists
- Find a stable marriage between men and women, i.e.,
  1. a perfect matching
  2. satisfies the stability condition: no two people of the opposite sex like each other more than their current partners

Preference lists

Men:

a	x	y
b	y	x

Women:

x	a	b
y	a	b

Stable Marriage Problem (search version) (Gale-Shapley '62)

- Given n men and n women together with their preference lists
- Find a stable marriage between men and women, i.e.,
  1. a perfect matching
  2. satisfies the stability condition: no two people of the opposite sex like each other more than their current partners

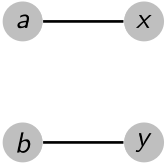
Preference lists

Men:

a	x	y
b	y	x

Women:

x	a	b
y	a	b



stable marriage

Stable Marriage Problem (search version) (Gale-Shapley '62)

- Given n men and n women together with their preference lists
- Find a stable marriage between men and women, i.e.,
  1. a perfect matching
  2. satisfies the stability condition: no two people of the opposite sex like each other more than their current partners

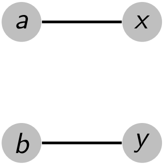
Preference lists

Men:

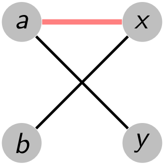
a	x	y
b	y	x

Women:

x	a	b
y	a	b



stable marriage



unstable marriage

Stable Marriage Problem (search version) (Gale-Shapley '62)

- Given n men and n women together with their preference lists
- Find a stable marriage between men and women, i.e.,
  1. a perfect matching
  2. satisfies the stability condition: no two people of the opposite sex like each other more than their current partners

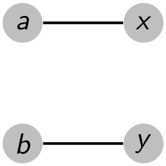
Preference lists

Men:

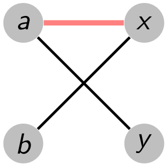
a	x	y
b	y	x

Women:

x	a	b
y	a	b



stable marriage



unstable marriage

Stable Marriage Problem (decision version)

Is a given pair of (m, w) in the man-optimal (woman-optimal) stable marriage?

The stable marriage problem is in CC

- Based on Subramanian '90
- We use three-valued logic
- We formalize in VCC*

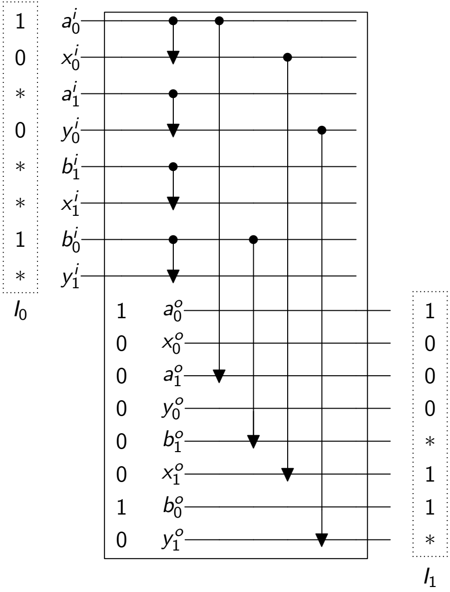
Preference lists

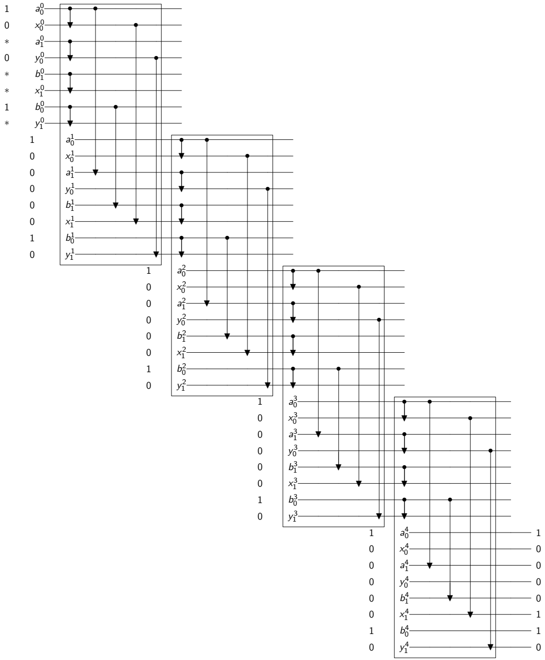
Men:

a	x	y
b	y	x

Women:

x	a	b
y	a	b



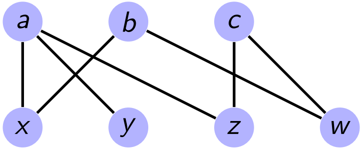


1 0 * 0 * * 1 *
a_0^0 x_0^0 a_1^0 y_0^0 b_1^0 x_1^0 b_0^0 y_1^0
1 0 0 0 0 0 1 0
a_0^1 x_0^1 a_1^1 y_0^1 b_1^1 x_1^1 b_0^1 y_1^1
1 0 0 0 0 0 1 0
a_0^2 x_0^2 a_1^2 y_0^2 b_1^2 x_1^2 b_0^2 y_1^2
1 0 0 0 0 0 1 0
a_0^3 x_0^3 a_1^3 y_0^3 b_1^3 x_1^3 b_0^3 y_1^3
1 0 0 0 0 0 1 0
a_0^4 x_0^4 a_1^4 y_0^4 b_1^4 x_1^4 b_0^4 y_1^4
1 0 0 0 0 1 1 0

Lex-first maximal matching problem

Lex-first maximal matching

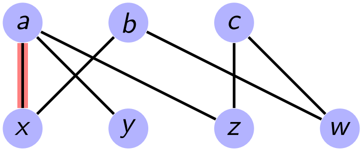
- Let G be a bipartite graph.
- Successively match the bottom nodes $x, y, z, \ldots$ to the least available top node



Lex-first maximal matching problem

Lex-first maximal matching

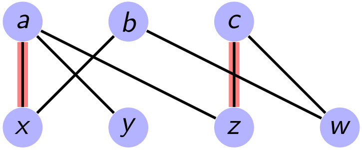
- Let G be a bipartite graph.
- Successively match the bottom nodes $x, y, z, \dots$ to the least available top node



Lex-first maximal matching problem

Lex-first maximal matching

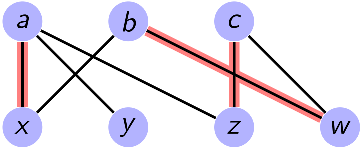
- Let G be a bipartite graph.
- Successively match the bottom nodes $x, y, z, \ldots$ to the least available top node



Lex-first maximal matching problem

Lex-first maximal matching

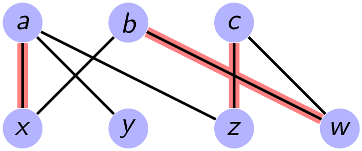
- Let G be a bipartite graph.
- Successively match the bottom nodes $x, y, z, \ldots$ to the least available top node



Lex-first maximal matching problem

Lex-first maximal matching

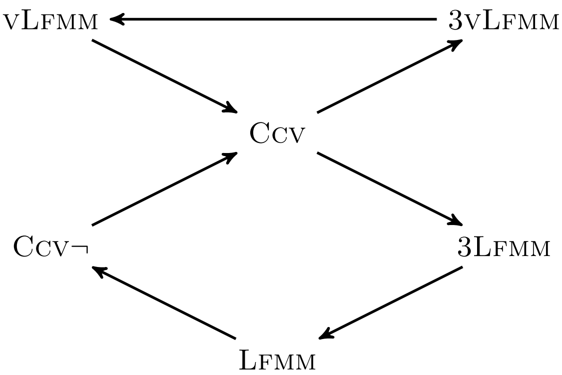
- Let G be a bipartite graph.
- Successively match the bottom nodes $x, y, z, \ldots$ to the least available top node



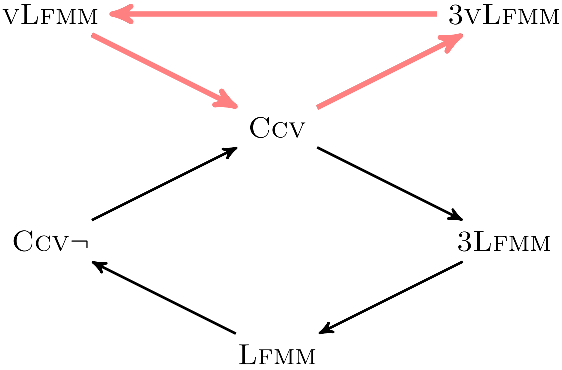
Lex-first maximal matching decision problems

- LFMM: Is a given edge $\{u, v\}$ in the lex-first maximal matching?
- vLFMM: Is a top node v matched in the lex-first maximal matching?

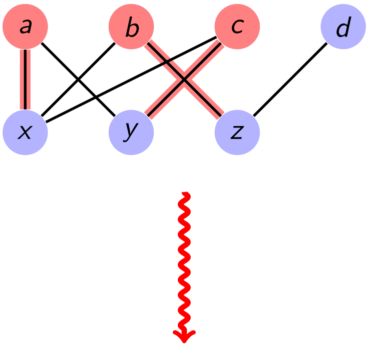
Overview of the reductions

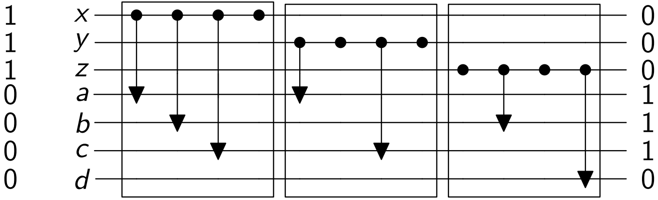


Overview of the reductions

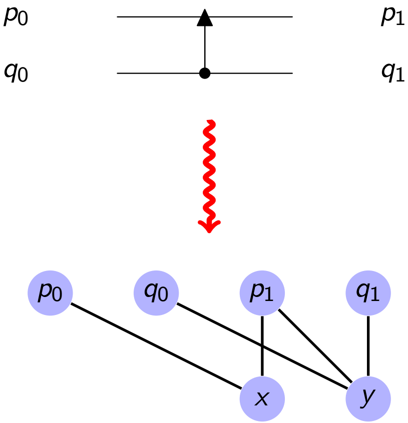


Reducing vLfmm to Ccv

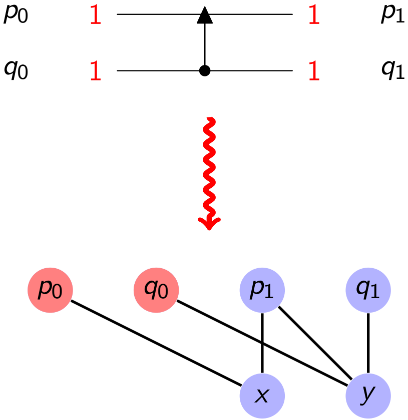




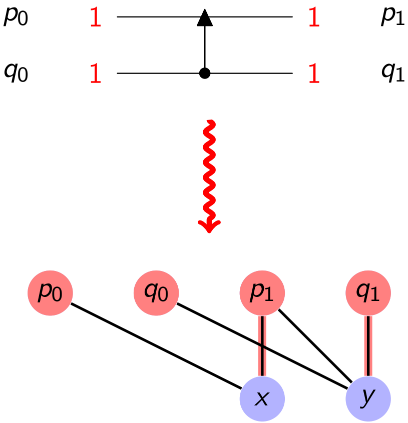
Reducing CCV to vLFMM



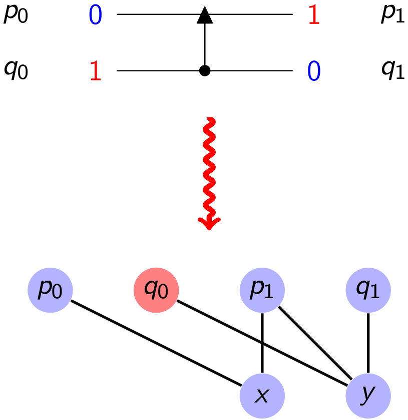
Reducing CCV to vLFMM



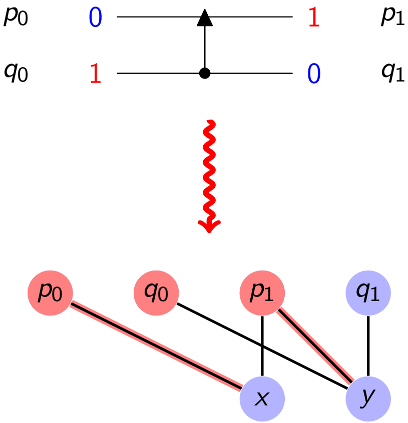
Reducing Ccv to vLfmm



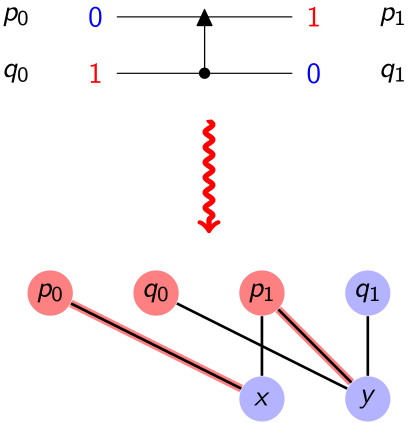
Reducing CCV to vLFMM



Reducing CCV to vLFMM



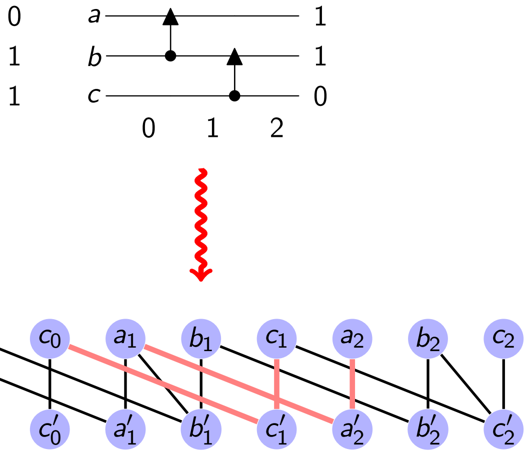
Reducing CCV to vLFMM



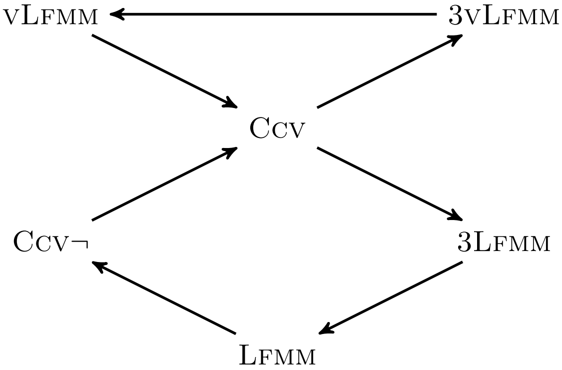
Remark

Bipartite graphs with degree ≤ 3 suffice.

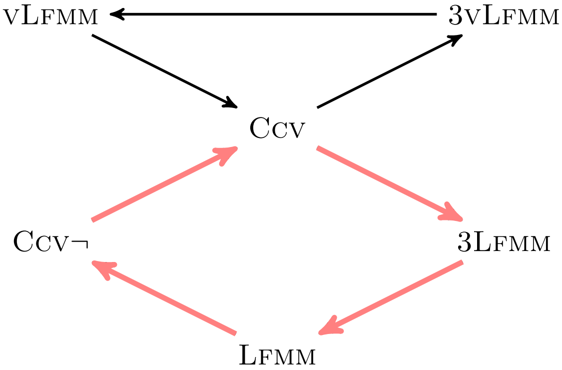
A bigger example



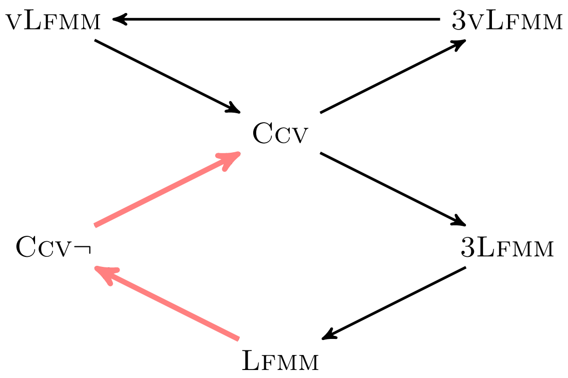
Summary of the reductions



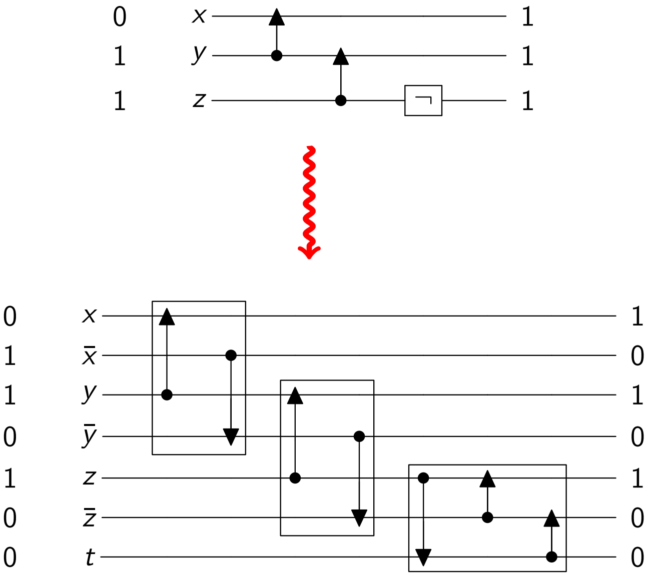
Summary of the reductions



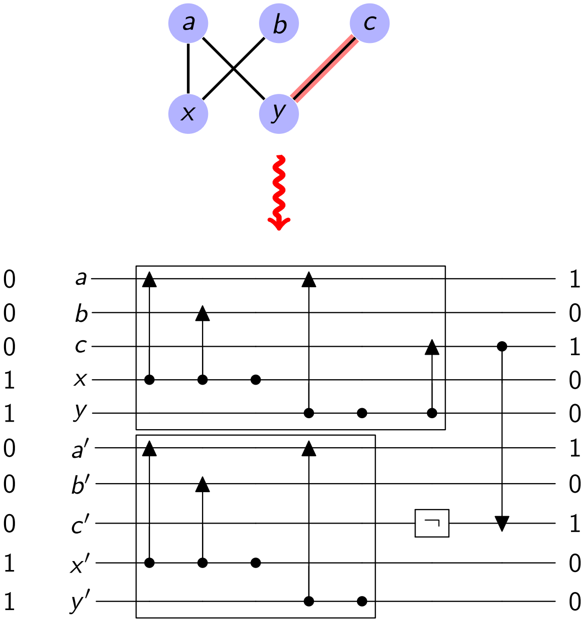
Summary of the reductions



Reducing CCV¬ to CCV (using "double-rail" logic)



Reducing LFMM to CCV¬



Summary

1. New classes CC and CC^*: AC^0-many-one-closure and AC^0-oracle-closure of $\mathrm{C{\scriptsize CV}}$.

$$\boxed{NC^1 \subseteq NL \subseteq CC \subseteq CC^{Subr} \subseteq CC^* \subseteq P}$$

Summary

1. New classes CC and CC*: AC^0-many-one-closure and AC^0-oracle-closure of $\mathrm{C{\scriptstyle CV}}$.

$$\boxed{\mathsf{NC}^1 \subseteq \mathsf{NL} \subseteq \mathsf{CC} \subseteq \mathsf{CC}^{\mathsf{Subr}} \subseteq \mathsf{CC}^* \subseteq \mathsf{P}}$$

2. Introduce the new two-sorted theory VCC* that "captures" CC*. We show that

$$\boxed{\mathsf{VNC}^1 \subseteq \mathsf{VNL} \subseteq \mathsf{VCC}^* \subseteq \mathsf{VP}}$$

Summary

1. New classes CC and CC*: AC^0-many-one-closure and AC^0-oracle-closure of $\mathrm{C{\scriptstyle CV}}$.

$$\mathsf{NC}^1 \subseteq \mathsf{NL} \subseteq \mathsf{CC} \subseteq \mathsf{CC}^{\mathsf{Subr}} \subseteq \mathsf{CC}^* \subseteq \mathsf{P}$$

2. Introduce the new two-sorted theory VCC* that "captures" CC*. We show that

$$\mathsf{VNC}^1 \subseteq \mathsf{VNL} \subseteq \mathsf{VCC}^* \subseteq \mathsf{VP}$$

3. Sharpen and simplify Subramanian's results: we show the following problems are CC-complete (under many-one AC^0-reduction)
   - lex-first maximal matching decision problems (even with degree ≤ 3)
   - stable-marriage (man-opt, woman-opt and search version)
   - three-valued $\mathrm{C{\scriptstyle CV}}$ (showing the completeness of stable marriage)

Summary

1. New classes CC and CC^*: AC^0-many-one-closure and AC^0-oracle-closure of $\textsc{Ccv}$.

$$\mathsf{NC}^1 \subseteq \mathsf{NL} \subseteq \mathsf{CC} \subseteq \mathsf{CC}^{\mathsf{Subr}} \subseteq \mathsf{CC}^* \subseteq \mathsf{P}$$

2. Introduce the new two-sorted theory VCC^* that "captures" CC^*. We show that

$$\mathsf{VNC}^1 \subseteq \mathsf{VNL} \subseteq \mathsf{VCC}^* \subseteq \mathsf{VP}$$

3. Sharpen and simplify Subramanian's results: we show the following problems are CC-complete (under many-one AC^0-reduction)
   - lex-first maximal matching decision problems (even with degree ≤ 3)
   - stable-marriage (man-opt, woman-opt and search version)
   - three-valued $\textsc{Ccv}$ (showing the completeness of stable marriage)
4. Prove the correctness of the above reductions within VCC^*.

Summary

1. New classes CC and CC*: AC^0-many-one-closure and AC^0-oracle-closure of $\textsc{Ccv}$.

   $$\mathsf{NC}^1 \subseteq \mathsf{NL} \subseteq \mathsf{CC} \subseteq \mathsf{CC}^{\mathsf{Subr}} \subseteq \mathsf{CC}^* \subseteq \mathsf{P}$$

2. Introduce the new two-sorted theory VCC* that "captures" CC*. We show that

   $$\mathsf{VNC}^1 \subseteq \mathsf{VNL} \subseteq \mathsf{VCC}^* \subseteq \mathsf{VP}$$

3. Sharpen and simplify Subramanian's results: we show the following problems are CC-complete (under many-one AC^0-reduction)
   - lex-first maximal matching decision problems (even with degree ≤ 3)
   - stable-marriage (man-opt, woman-opt and search version)
   - three-valued $\textsc{Ccv}$ (showing the completeness of stable marriage)
4. Prove the correctness of the above reductions within VCC*.
5. Promote the use of Σ_0^B-formulas when working with AC^0 functions or relations.

Summary

1. New classes CC and CC^*: AC^0-many-one-closure and AC^0-oracle-closure of CCV.

$$\mathsf{NC}^1 \subseteq \mathsf{NL} \subseteq \mathsf{CC} \subseteq \mathsf{CC}^{\mathsf{Subr}} \subseteq \mathsf{CC}^* \subseteq \mathsf{P}$$

2. Introduce the new two-sorted theory VCC^* that "captures" CC^*. We show that

$$\mathsf{VNC}^1 \subseteq \mathsf{VNL} \subseteq \mathsf{VCC}^* \subseteq \mathsf{VP}$$

3. Sharpen and simplify Subramanian's results: we show the following problems are CC-complete (under many-one AC^0-reduction)
   - lex-first maximal matching decision problems (even with degree ≤ 3)
   - stable-marriage (man-opt, woman-opt and search version)
   - three-valued CCV (showing the completeness of stable marriage)
4. Prove the correctness of the above reductions within VCC^*.
5. Promote the use of Σ_0^B-formulas when working with AC^0 functions or relations.

Summary

1. New classes CC and CC^*: AC^0-many-one-closure and AC^0-oracle-closure of Ccv.

$$\mathsf{NC}^1 \subseteq \mathsf{NL} \subseteq \mathsf{CC} \subseteq \mathsf{CC}^{\mathsf{Subr}} \subseteq \mathsf{CC}^* \subseteq \mathsf{P}$$

2. Introduce the new two-sorted theory VCC^* that "captures" CC^*. We show that

$$\mathsf{VNC}^1 \subseteq \mathsf{VNL} \subseteq \mathsf{VCC}^* \subseteq \mathsf{VP}$$

3. Sharpen and simplify Subramanian's results: we show the following problems are CC-complete (under many-one AC^0-reduction)
   - lex-first maximal matching decision problems (even with degree ≤ 3)
   - stable-marriage (man-opt, woman-opt and search version)
   - three-valued Ccv (showing the completeness of stable marriage)
4. Prove the correctness of the above reductions within VCC^*.
5. Promote the use of Σ_0^B-formulas when working with AC^0 functions or relations.

Open Problems

1. $\mathsf{CC} = \mathsf{CC}^{\mathsf{Subr}} = \mathsf{CC}^*$? Do universal comparator circuits exist?
2. $\mathsf{CC}^* = \mathsf{P}$?
3. Do the complete problems in CC have NC or RNC algorithms?
4. Can we prove the correctness of the Gale-Shapley algorithm in CC^*?