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 \( \tau \), 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
The complexity classes for the Comparator Circuit Value Problem

Define a theory for $\text{CC}^*$

Natural complete problems: stable marriage and lex-first maximal matching

Conclusion and open problems
Comparator Circuits

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

**Comparator gate**

- $a \ x \ \bullet \ \min(a, b)$
- $b \ y \ \blacktriangleleft \ \max(a, b)$
Comparator Circuits

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

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

**Comparator gate**
\[
\begin{align*}
a & \quad x \quad \bullet \quad \min(a, b) \\
b & \quad y \quad \bullet \quad \max(a, b)
\end{align*}
\]

**Boolean comparator gate**
\[
\begin{align*}
p & \quad x \quad \bullet \quad p \land q \\
q & \quad y \quad \bullet \quad p \lor q
\end{align*}
\]

**Example**

\[
\begin{array}{cccccccc}
1 & w_0 & 0 & 0 & 0 & 0 \\
1 & w_1 & 0 & 0 & 1 & 0 \\
1 & w_2 & 0 & 1 & 1 & 1 \\
0 & w_3 & 1 & 1 & 1 & 0 \\
0 & w_4 & 1 & 1 & 0 & 1 \\
0 & w_5 & 0 & 0 & 0 & 0
\end{array}
\]
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. \( CC^{Subr} = \{ \text{decision problems log-space many-one-reducible to CCV} \} \)
   - [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. \( \text{CC}^{\text{Subr}} = \{ \text{decision problems log-space many-one-reducible to } Ccv \} \)
   - [Subramanian '90], [Mayr-Subramanian '92]

2. \( \text{CC} = \{ \text{decision problems } AC^0 \text{ many-one-reducible to } Ccv \} \)
   - Complete problems: stable marriage, lex-first maximal matching...

3. \( \text{CC}^* = \{ \text{decision problems } AC^0 \text{ oracle-reducible to } Ccv \} \)
   - Needed when developing a Cook-Nguyen style theory for CC
   - The function class FCC* is closed under composition
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. $CC^{Subr} = \{\text{decision problems log-space many-one-reducible to CCV}\}$
   - [Subramanian ’90], [Mayr-Subramanian ’92]

2. $CC = \{\text{decision problems AC}^0 \text{ many-one-reducible to CCV}\}$
   - Complete problems: stable marriage, lex-first maximal matching...

3. $CC^* = \{\text{decision problems AC}^0 \text{ oracle-reducible to CCV}\}$
   - Needed when developing a Cook-Nguyen style theory for CC
   - The function class FCC$^*$ is closed under composition
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. \(CC^{Subr} = \{\text{decision problems log-space many-one-reducible to } CCV\}\)
   
   - [Subramanian '90], [Mayr-Subramanian '92]

2. \(CC = \{\text{decision problems } AC^0 \text{ many-one-reducible to } CCV\}\)
   
   - Complete problems: stable marriage, lex-first maximal matching...

3. \(CC^* = \{\text{decision problems } AC^0 \text{ oracle-reducible to } CCV\}\)
   
   - Needed when developing a Cook-Nguyen style theory for CC
   - The function class FCC* is closed under composition

\[ NC^1 \subseteq NL \subseteq CC \subseteq CC^{Subr} \subseteq CC^* \subseteq P \]
1. The complexity classes for the Comparator Circuit Value Problem

2. Define a theory for $\text{CC}^*$

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

- Standard model $\mathbb{N}_2 = \langle \mathbb{N}, \text{finite subsets of } \mathbb{N} \rangle$
- $0, 1, +, \cdot, \leq, =$ have usual meaning over $\mathbb{N}$
- $\mid X \mid = \text{length of } X$
- 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 $\mid X \mid, \mid Y \mid, \mid Z \mid, \ldots$
- The only string terms are variable $X, Y, Z, \ldots$
Two-sorted language $L_A^2$ (Zambella '96)

Vocabulary $L_A^2 = [0, 1, +, \cdot, |; |; ; \in, \leq, =_1, =_2]$

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

Note
The natural inputs for Turing machines and circuits are finite strings.
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 $\mathbb{N}$
- $|X| = \text{length of } X$
- 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 $X, Y, Z, \ldots$

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

Definition ($\Sigma^B_0$ 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 $O(\log n)$ with a constant number of alternations.

**$\Sigma^B_0$-Representation Theorem [Zambella '96, Cook-Nguyen]**

$R(\vec{x}, \vec{X})$ is in $AC^0$ iff it is represented by a $\Sigma^B_0$-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. **$\Sigma^B_0$-COMP** (Comprehension): for every $\Sigma^B_0$-formula $\varphi(z)$ without $X$,
   $$\exists X \leq y \forall z < y (X(z) \leftrightarrow \varphi(z))$$

Theorem

1. **$\Sigma^B_0$-IND**: for $\varphi \in \Sigma^B_0$
   $$\left[ \varphi(0) \land \forall x (\varphi(x) \rightarrow \varphi(x + 1)) \right] \rightarrow \forall x \varphi(x)$$

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. \( x + 1 = y + 1 \rightarrow x = y \)
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. \( (x \leq y \land y \leq x) \rightarrow x = y \)

B8. \( x \leq x + y \)
B9. \( 0 \leq x \)
B10. \( x \leq y \lor y \leq x \)
B11. \( x \leq y \leftrightarrow x < y + 1 \)
B12. \( x \neq 0 \rightarrow \exists y \leq x (y + 1 = x) \)

L1. \( X(y) \rightarrow y < |X| \)
L2. \( y + 1 = |X| \rightarrow X(y) \)

SE. \([ |X| = |Y| \land \forall i < |X|(X(i) = Y(i)) \] \rightarrow X = Y \)
The theory \( \text{VCC}^* \) for \( \text{CC}^* \)

**Comparator Circuit Value (CCV) Problem (decision)**

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

Recall that \( \text{CC}^* = \{ \text{decision problems AC}^0 \text{ oracle-reducible to CCV} \} \)

The two-sorted theory \( \text{VCC}^* \) [using the Cook-Nguyen method]

- \( \text{VCC}^* \) has vocabulary \( L_A^2 \)
- Axiom of \( \text{VCC}^* = \text{Axiom of V}^0 + \text{one additional axiom asserting the existence of a solution to the CCV problem.} \)
Asserting the existence of a solution to $C_{cv}$

- $X$ 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 $\Sigma^B_0$ formula $\delta_{CCV}(m, n, X, Y, Z)$ states that $Z$ encodes the correct values of all the layers of the $C_{cv}$ instance encoded in $X$ and $Y$:

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

$$(X)^i = \langle x, y \rangle \rightarrow \begin{bmatrix}
Z(i + 1, x) \leftrightarrow (Z(i, x) \land Z(i, y)) \\
\land Z(i + 1, y) \leftrightarrow (Z(i, x) \lor Z(i, y)) \\
\land \forall j < m[j \neq x \land j \neq y \rightarrow (Z(i + 1, j) \leftrightarrow Z(i, j))]
\end{bmatrix}$$

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

- Recall that:

\[
\text{AC}^0 \subseteq \text{TC}^0 \subseteq \text{NC}^1 \subseteq \text{NL} \subseteq \text{CC} \subseteq \text{CC}^{\text{Subr}} \subseteq \text{CC}^* \subseteq \text{P}
\]
Inclusion of theories

- Recall that:

\[ \text{AC}^0 \subseteq \text{TC}^0 \subseteq \text{NC}^1 \subseteq \text{NL} \subseteq \text{CC} \subseteq \text{CC}^{\text{Subr}} \subseteq \text{CC}^* \subseteq \text{P} \]

- We showed in our paper that:

\[ \text{VTC}^0 \subseteq \text{VNC}^1 \subseteq \text{VNL} \subseteq \text{VCC}^* \subseteq \text{VP} \]
Can't talk about reachability!

Known fact:

\[ VTC_0 \subseteq VNC_1 \subseteq VCC^* \]

We prove the correctness of this construction using only counting.
VNL $\subseteq$ VCC*

- Can’t talk about reachability!
- Known fact:
  \[ VTC^0 \subseteq VNC^1 \subseteq VCC^* \]
- We prove the correctness of this construction using only counting.
1. The complexity classes for the Comparator Circuit Value Problem

2. Define a theory for $\text{CC}^*$

3. Natural complete problems: stable marriage and lex-first maximal matching

4. Conclusion and open problems
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**

<table>
<thead>
<tr>
<th>Men:</th>
<th>$a$</th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$b$</td>
<td>$y$</td>
<td>$x$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Women:</th>
<th>$x$</th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$y$</td>
<td>$a$</td>
<td>$b$</td>
</tr>
</tbody>
</table>
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:
\[
\begin{array}{c|ccc}
    & a & x & y \\
\hline
    b & y & x \\
\end{array}
\]

Women:
\[
\begin{array}{c|ccc}
    & x & a & b \\
\hline
    y & a & b \\
\end{array}
\]

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

<table>
<thead>
<tr>
<th>Men:</th>
<th>a</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>b</td>
<td>y</td>
<td>x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Women:</th>
<th>x</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>y</td>
<td>a</td>
<td>b</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Men:</th>
<th>$a$</th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$b$</td>
<td>$y$</td>
<td>$x$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Women:</th>
<th>$x$</th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$y$</td>
<td>$a$</td>
<td>$b$</td>
</tr>
</tbody>
</table>

Stable marriage

![Stable marriage diagram](image1)

Unstable marriage

![Unstable marriage diagram](image2)

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:  
\[
\begin{array}{ccc}
  a & x & y \\
  b & y & x \\
\end{array}
\]

Women:  
\[
\begin{array}{ccc}
  x & a & b \\
  y & a & b \\
\end{array}
\]
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

$$
\begin{array}{c}
\text{a} \\
\text{x}
\end{array} \quad \begin{array}{c}
\text{b} \\
\text{y}
\end{array} \quad \begin{array}{c}
\text{c} \\
\text{z}
\end{array} \quad \begin{array}{c}
\text{w}
\end{array}
$$
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

![Diagram of a bipartite graph with nodes $a$, $b$, $c$, $x$, $y$, $z$, $w$ and edges between them]
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

![Graph](image-url)
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.

![Graph Example](image)
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

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

\[ vL_{\text{FMM}} \quad \text{Ccv} \quad 3vL_{\text{FMM}} \]

\[ \text{Ccv} \quad \text{Ccv}\neg \quad 3L_{\text{FMM}} \quad L_{\text{FMM}} \]
Reducing vLFMM to Ccv
Reducing \( \text{CV} \) to \( \text{VLFMM} \)

Remark: Bipartite graphs with degree \( \leq 3 \) suffice.
Reducing \( C_{CV} \) to \( vLFMM \)

\[
\begin{align*}
p_0 & \quad 1 \quad 1 \quad p_1 \\
q_0 & \quad 1 \quad 1 \quad q_1
\end{align*}
\]
Reducing \( \text{CCV} \) to \( vLFMM \)

Bipartite graphs with degree \( \leq 3 \) suffice.
Reducing $CCV$ to $VLfmm$
Reducing \textbf{CCV to VLFMN}

\begin{align*}
\begin{array}{c}
p_0 & 0 & 1 & p_1 \\
q_0 & 1 & 0 & q_1
\end{array}
\end{align*}

Remark: Bipartite graphs with degree \( \leq 3 \) suffice.
Reducing CCV to vLFMM

Remark
Bipartite graphs with degree $\leq 3$ suffice.
A bigger example

\[
\begin{align*}
0 & & a & & 1 \\
1 & & b & & 1 \\
1 & & c & & 0 \\
\end{align*}
\]

\[
\begin{align*}
0 & & 1 & & 2 \\
\end{align*}
\]
Summary of the reductions

\[
\begin{align*}
  &vL\text{FMM} \\
  \rightarrow & \quad C\text{cv} \\
  \rightarrow & \quad C\text{cv}^{-} \\
  \rightarrow & \quad L\text{FMM} \\
  \rightarrow & \quad 3L\text{FMM} \\
  \rightarrow & \quad 3vL\text{FMM} \\
  \rightarrow & \quad vL\text{FMM}
\end{align*}
\]
Summary of the reductions

\[ \text{vLFMM} \xleftarrow{} \text{CcV} \xrightarrow{} 3\text{vLFMM} \]

\[ \text{CcV} \xrightarrow{} \text{CcV}^\rightarrow \xleftarrow{} 3\text{LFMM} \xleftarrow{} \text{LFMM} \]
Summary of the reductions

- \( vLFMM \) → \( Ccv \) → \( 3LFMM \)
- \( Ccv \) → \( 3LFMM \)
- \( LFMM \) → \( Ccv \) → \( vLFMM \)
Reducing $C_{CV} \neg \rightarrow$ to $C_{CV}$ (using “double-rail” logic)
Reducing **L_{FMM} to C_{CV}**

![Diagram showing a reduction process from L_{FMM} to C_{CV} with nodes a, b, c, x, y, a', b', c', x', y' and the corresponding truth values in a table.](image)
Summary

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

\[ \text{NC}^1 \subseteq \text{NL} \subseteq \text{CC} \subseteq \text{CC}^\text{Subr} \subseteq \text{CC}^* \subseteq \text{P} \]
Summary

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

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

2. Introduce the new two-sorted theory $\text{VCC}^*$ that “captures” $\text{CC}^*$. We show that

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

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

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

2. Introduce the new two-sorted theory $\text{VCC}^*$ that “captures” $\text{CC}^*$. We show that

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

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

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

   \[
   \text{NC}^1 \subseteq \text{NL} \subseteq \text{CC} \subseteq \text{CC}^\text{Subr} \subseteq \text{CC}^* \subseteq \text{P}
   \]

2. Introduce the new two-sorted theory $\text{VCC}^*$ that “captures” $\text{CC}^*$. We show that

   \[
   \text{VNC}^1 \subseteq \text{VNL} \subseteq \text{VCC}^* \subseteq \text{VP}
   \]

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

4. Prove the correctness of the above reductions within $\text{VCC}^*$.
Summary

1. New classes \( \text{CC} \) and \( \text{CC}^* \): \( \text{AC}^0 \)-many-one-closure and \( \text{AC}^0 \)-oracle-closure of \( \text{CCV} \).
   \[
   \text{NC}^1 \subseteq \text{NL} \subseteq \text{CC} \subseteq \text{CC}^{\text{Subr}} \subseteq \text{CC}^* \subseteq \text{P}
   \]

2. Introduce the new two-sorted theory \( \text{VCC}^* \) that “captures” \( \text{CC}^* \). We show that
   \[
   \text{VNC}^1 \subseteq \text{VNL} \subseteq \text{VCC}^* \subseteq \text{VP}
   \]

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

4. Prove the correctness of the above reductions within \( \text{VCC}^* \).

5. Promote the use of \( \Sigma^B_0 \)-formulas when working with \( \text{AC}^0 \) functions or relations.
Summary

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

\[
\text{NC}^1 \subseteq \text{NL} \subseteq \text{CC} \subseteq \text{CC}^{\text{Subr}} \subseteq \text{CC}^* \subseteq \text{P}
\]

2. Introduce the new two-sorted theory $\text{VCC}^*$ that “captures” $\text{CC}^*$. We show that

\[
\text{VNC}^1 \subseteq \text{VNL} \subseteq \text{VCC}^* \subseteq \text{VP}
\]

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

4. Prove the correctness of the above reductions within $\text{VCC}^*$.

5. Promote the use of $\Sigma^B_0$-formulas when working with $\text{AC}^0$ functions or relations.
Summary

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

   \[
   \text{NC}^1 \subseteq \text{NL} \subseteq \text{CC} \subseteq \text{CC}^{\text{Subr}} \subseteq \text{CC}^* \subseteq \text{P}
   \]

2. Introduce the new two-sorted theory $\text{VCC}^*$ that “captures” $\text{CC}^*$. We show that

   \[
   \text{VNC}^1 \subseteq \text{VNL} \subseteq \text{VCC}^* \subseteq \text{VP}
   \]

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

4. Prove the correctness of the above reductions within $\text{VCC}^*$.

5. Promote the use of $\Sigma^B_0$-formulas when working with $\text{AC}^0$ functions or relations.

Open Problems

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