<table>
<thead>
<tr>
<th>項目</th>
<th>内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>タイトル</td>
<td>基調講演 証明論と複雑性</td>
</tr>
<tr>
<td>作者</td>
<td>Kuroda, Satoru</td>
</tr>
<tr>
<td>引用</td>
<td>数理解析研究所講究録 1832: 2-7</td>
</tr>
<tr>
<td>発行年月</td>
<td>2013-04</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/2433/194859">http://hdl.handle.net/2433/194859</a></td>
</tr>
<tr>
<td>型式</td>
<td>Departmental Bulletin Paper</td>
</tr>
<tr>
<td>出版者</td>
<td>Kyoto University</td>
</tr>
</tbody>
</table>
A propositional proof system based on comparator circuits

Satoru Kuroda (黒田覚)
Gunma Prefectural Women's University

1 Introduction

Since the seminal paper by S. Cook [2], there have been many literatures on the connection of complexity classes and proof systems. The most prominent example is the relationships between the class $P$, Buss' theory $S^1_2$ [1] and extended Frege proofs.

In this paper we construct a propositional proof system which corresponds to the class CC. Originally, this class is defined by Subramanian [5] as the set of problems log-space reducible to the comparator circuit value problem. This class has not gained much attention since it was presented. However, recently Cook et.al. [4] shed a new light on the class by defining bounded arithmetic theory VCC and proved that stable marriage problem is definable in the theory. So we believe that our proof system gives a step forward for the investigation of the class.

Here we only give a rough outline of the system and detailed proofs are given in the forthcoming paper.

2 Preliminaries

A comparator gate is a function $C : \{0, 1\}^2 \rightarrow \{0, 1\}^2$ that takes an input pair $(p, q)$ and outputs a pair $(p \wedge q, p \vee q)$. A comparator circuit consists of $n$ wires each having input bits and produces an output. In each layer, two wires are connected by an arrow representing a comparator gate. Formally, a comparator circuit can be represented as a directed acyclic graph with input nodes having indegree 0 and outdegree 2, output nodes with indegree 1 and outdegree 0, and comparator gates with indegree and outdegree 2.

The comparator circuit value problem (CCV) is a decision problem. Given a comparator circuit, an input and a designated output wire, decide whether the circuit outputs one on that wire.

Definition 1 The complexity class $CC$ is the class of problems which are $AC^0$ many-one reducible to $CCV$. 
We formalize CC reasoning in tow sort language. The language $L_2$ comprises number variables $x, y, z, \ldots$ and string variables $X, Y, Z, \ldots$. It also has the following symbols: $Z(x) = 0$, $x + y$, $x \cdot y$, $x \leq y$, $x \in Y$.

The class $\Sigma^B_0$ is the class of $L_2$-formulas in which all quantifiers are bounded number quantifiers $\forall x < t$ or $\exists x < t$ and $\Sigma^B_1$ is the class of formulas of the form

$$\exists X < \hat{t} \varphi(X), \varphi \in \Sigma^B_0.$$ 

We define $L_2$-theory $V^0$ as having the axioms $BASIC_2$ which is a finite set of defining formulas for symbols in $L_2$ together with

$$\Sigma^B_0\text{-IND} : \exists X < a \forall y < a(y \in X \leftrightarrow \varphi(y),$$

where $\varphi \in \Sigma^B_0$ contains no free occurrences of $X$.

The theory $VCC$ is defined the extension of $V^0$ by the axiom expressing CCV. Let $\delta_{CCV}(m, n, X, Y, Z)$ be the following $\Sigma^B_0$ formula:

$$\forall i < m(Y(i) \leftrightarrow Z(0, i) \land \forall i < n \forall x < m \forall y < m \big(Z(i + 1, x) \leftrightarrow (Z(i, x) \land Z(i, y)) \land \forall j < m((j \neq x \land j \neq y) \rightarrow (Z(i + 1, j) \leftrightarrow Z(i, j))) \big).$$

This formula expresses the following properties:

- $X$ encodes a comparator circuit with $m$ wires and $n$ gates as sequence of $n$ pairs $(i, j)$ with $i, j < m$ and $(X)^i$ encodes the $i$-th comparator gate of $X$,
- $Y(i)$ encodes the $i$-th input to $X$,
- $Z$ is an $(n + 1) \times m$ matrix, where $Z(i, j)$ is the value of wire $j$ at layer $i$.

**Definition 2** The theory $VCC$ is the $L_2$ theory which is aximatized by axioms of $V^0$ together with

$$CCV : \exists Z \leq \langle m, n + 1 \rangle + 1 \delta_{CCV}(m, n, X, Y, Z).$$

**Theorem 1** (Cook et.al.) A function is computable in CC if and only if it is $\Sigma^B_1$ definable in VCC.

In the propositional translation, it is easier to work with the universal conservative extension of VCC. Let $L_{CC}$ be the language $L_2$ extended by a single function symbol $F_{CC}$. We denote the $\Sigma^B_0$ formula in the extended language by $\Sigma^B_0(F_{CC})$.

**Definition 3** The theory $V^0(F_{CC})$ is the $\Sigma^B_0(F_{CC})$ theory which is aximatized by $BASIC_2$, $\Sigma^B_0(F_{CC})\text{-IND}$ and the following defining axiom of $F_{CC}$:

$$F_{CC}(X, Y) = Z \leftrightarrow \delta_{CCV}(\sqrt{|X|}, |Y|, X, Y, Z)$$

where $\sqrt{m}$ is the integer part of the square root of $m$.

It is not difficult to see that

**Theorem 2** $VCC$ and $V^0(F_{CC})$ proves the same $L_2$ theorems.
3 The system CCK

In this section we present a propositional proof system $CCK$ which corresponds to bounded arithmetic theory $VCC$ in the sense that

- $CCK$ has polynomial size proofs for all $\forall \Sigma^B_0$ consequences of $VCC$ and
- $VCC$ proves the reflection principle of $CCK$.

The fundamental idea is to introduce connectives used to construct comparator circuits so that formulas represents circuits. The language of $CCK$ comprises the following symbols:

- propositional variables $x_1, x_2, \ldots$
- connectives $\neg_k, [j, k]$ for $j, k \in \omega$, $\oplus$
- superscripts $(i)$ for $i \in \omega$

We define $CCK$ formulae and a number $w(\varphi)$ for a formula $\varphi$ recursively as follows:

- a propositional variable $x_i$ is a formula and $w(x_i) = 1$,
- if $\varphi$ is a formula and $i, k \leq w(\varphi)$ then so is $(\neg_k \varphi)^{(i)}$ and $w(\neg_k \varphi) = w(\varphi)$,
- if $\varphi$ is a formula and $i, j, k \leq w(\varphi)$ then so is $\varphi[j, k]^{(i)}$ and $w(\varphi[j, k]) = w(\varphi)$
- if $\varphi$ and $\psi$ are formulas and $i \leq w(\varphi) + w(\psi)$ then so is $(\varphi \oplus \psi)^{(i)}$ and $w(\varphi \oplus \psi) = w(\varphi) + w(\psi)$.

The intuitive meaning of the above definition is that, the superscript in $\varphi^{(i)}$ represents its designated output, $\neg_k \varphi$ is $\varphi$ with negation at the top of the $k$-th wire, $\varphi[j, k]$ is obtained from $\varphi$ by placing arrows from $j$ to $k$ at to top, and $\varphi \oplus \psi$ is a juxtaposition of $\varphi$ and $\psi$. Furthermore, the function $w(\varphi)$ represents the number of wires in $\varphi$.

Before we define the proof system $CCK$ we introduce one more important notion. Two $CCK$-formulae are identical if they are of the same form if superscripts are ignored. Thus for instance $(\neg_k \varphi)^{(i)}$ and $(\neg_k \varphi)^{(j)}$ are identical.

**Proposition 1** Checking whether two formulas are identical can be done in $AC^{0}$.

Now we define the system $CCK$. Axioms of $CCK$ are:

\[ \varphi \rightarrow \varphi, \rightarrow T, \perp \rightarrow . \]

Inference rules of $CCK$ are contraction, weakening, exchange, cut and the following logical rules introducing connectives:

\[
\frac{\Gamma \rightarrow \Delta, \varphi^{(i)}}{(\neg \varphi)^{(i)}, \Gamma \rightarrow \Delta} \quad \frac{\varphi^{(j)}, \Gamma \rightarrow \Delta}{(\neg \varphi)^{(j)}, \Gamma \rightarrow \Delta} \quad \neg_i\text{-left}
\]
provided that $\varphi^{(i)}$ and $\varphi^{(j)}$ are identical.

A CCK-proof is a sequence $C_1, \ldots, C_k$ of CCK-formulas such that each $C_i$ is either an axiom or obtained from preceding formulas by one of the inference rules of CCK. The size $\text{size}(P)$ of a CCK-proof $P$ is the number of formulas in it.

It is easy to show that Boolean formulas are expressed by CCK-formulas and any rules of Frege system can be represented by some rule of CCK. So we have the following:

**Proposition 2** CCK proof system $p$-simulates Frege.

As CCK formulas are special cases of Boolean circuits and circuit Frege and extended Frege are $p$-equivalent, we have

**Theorem 3** Extended Frege system $p$-simulates CCK proof system.

### 4 Propositional Translation

In this section we prove that CCK is at least as strong as VCC. More precisely, it is proved that all $\forall\Sigma _{0}^{B}$ theorems of the universal conservative extension of VCC are translated into families of CCK-formulas having polynomial size CCK-proofs.

First we define the translation.

**Definition 4** For $\varphi(X) \in \Sigma _{0}^{B}(F_{CC})$, we define its propositional translation $\|\varphi(X)\|_a$ inductively as follows:

- if $\varphi$ is an atomic sentence without string variables then
  
  $\|\varphi\| = \begin{cases} 
  \top & \text{if } \varphi \text{ is true,} \\
  \bot & \text{if } \varphi \text{ is false.}
  \end{cases}$
For each string variable $X$ we introduce propositional variables $x_0, \ldots, x_{n-1}$ and let $\|i \in X\|_n = x_i$.

$\|\neg \varphi\|_\overline{n} = \neg k \|\varphi\|_n$ where $k$ is the designated output position of $\|\varphi\|_n$.

$\|x \in F_{CC}(X, Y)\|_{i, m, n} = C_U^{m,n}(p_{\overline{X}}, \overline{p}_{Y})$ where $C_U^{m,n}$ denotes the formula representing universal comparator circuit with a code $X$ for a comparator circuit and $Y$ as its input.

$\|\varphi \wedge \psi\|_{\overline{n}} = (\|\varphi\|_n \oplus \|\psi\|_n)[i, w(\|\varphi\|_n) + j]^{(i)}$,

$\|\varphi \vee \psi\|_{\overline{n}} = (\|\varphi\|_n \oplus \|\psi\|_n)[i, w(\|\varphi\|_n) + j]^{(w(\|\varphi\|_n) + j)}$,

$\|(\forall x < t) \varphi(x)\|_{n} = (\oplus_{x \leq t} \|\varphi(x)\|_n)[i_0, i_1][i_1, i_2] \cdots [i_{t-2}, i_{t-1}]^{(i_0)}$,

$\|(\exists x < t) \varphi(x)\|_{n} = (\oplus_{x \leq t} \|\varphi(x)\|_n)[i_0, i_1][i_1, i_2] \cdots [i_{t-2}, i_{t-1}]^{(i_{t-1})}$.

**Theorem 4** Let $\varphi(\overline{X})$ in $\Sigma^B_0$. If $VCC \vdash (\forall \overline{X}) \varphi(\overline{X})$ then $\{\|\varphi(\overline{X})\|_n\}_{n \in \omega}$ has polynomial size $CCK$-proofs.

(Proof). It suffices to show that axioms of $V^0(F_{CC})$ are translated into $CCK$ formulas having polynomial size proofs. For axioms of $V^0$ it suffices to remark that $CCK$ p-simulates Frege. So it suffices to show that $\Sigma^0_0(F_{CC}$-IND can be simulated by polynomial size $CCK$ proofs. The proof is similar to the one for $VTC^0$ and $TC^0$-Frege.

## 5 Proving the reflection principle

We will show the converse to the argument of the last section; $CCK$ is not stronger than $VCC$.

We will give a rough idea of how formulas, proofs etc. are coded in $L_0$. Assume any reasonable coding of $CCK$ formulas in $L_0$. Then for each $CCK$ formula $\varphi$ we can assign a string $X_\varphi$ which codes an equivalent comparator circuit with negation gates in such a way that $(X_\varphi)^i$ codes the comparator gate or the negation gate on $i$-th level. Although comparator circuit with negation gates is not by definition contained in $VCC$, it can be shown that $VCC$ proves the following result by Cook et.al [3].

**Proposition 3** The circuit value problem for comparator circuits with negation gates is $AC^0$ reducible to $CCV$.

Let $(X, i)$ denote a $CCK$ formula $X$ with the designated output $i$. We can define the $\Sigma^B_0$ formula $Z \models (X, i)$ which states that $(X, i)$ is true on the assignment $Z$. So we have

**Lemma 1** $VCC$ proves that any formula can be evaluated on any assignment.
Let $\text{Prf}^{CCK}(P, X, i)$ be the $L_0$ formula stating that $P$ is a $CCK$-proof of the $CCK$ formula $(X, i)$. Then the following theorem follows by the argument similar to those for other systems.

**Theorem 5** VCC proves that $CCK$ is sound:

$$\forall i, \forall X (\exists P \text{Prf}^{CCK}(P, X, i) \rightarrow \forall Z (Z \models (X, i))).$$

### 6 Concluding Remarks

It is unknown whether the complexity class $CC$ is properly contained in $P$. Furthermore, relations with subclasses of $P$ such as $NL$ is also open. A counterpart to this problem for propositional proof systems is whether $CCK$ p-simulates extended Frege.

Another direction of research is to find a hard tautology for $CCK$ or polynomial size $CCK$ proofs for natural combinatorial principle.

### References


