数理解析研究所講究録
第1635巻 2009年 77-85

Towards affluent proof theory of LOGCFL and related topics

Satoru Kuroda（黒田覚）
群馬県立女子大学文学部

1 Introduction

LOGCFL is the class of sets which are LOGSPACE reducible to context free languages. This class lies between NL and AC^1 and has several alternative characterizations based on different computational concepts.

Such concepts include Boolean circuits (semi-unbounded fan-in logarithmic depth circuits), algebraic structure (word problem for finite groupoids), database theory (acyclic conjunctive queries) and graph theory (acyclic graph embedding problem).

It had been open whether there exists a theory of weak arithmetic which capture this class, that is, a theory whose provably total functions precisely correspond to functions bitwise computable in **LOGCFL**.

The author gave the first such theory $\mathbf{V}$-$Q^{SAC}(\Sigma_0^B)$ based on the circuit based characterization of **LOGCFL**, while it is still open to construct similar theories based on other characterizations of the class.

Another aspect of complexity theory is the complexity of propositional proofs. Many natural complexity classes are known to correspond to natural propositional proof systems. So it is likely that **LOGCFL** also has a natural counterpart in propositional proof systems.

One such system is the Frege system enhanced to manipulate SAC^1 circuits in place of Boolean formulae. It is known that Frege with polynomial size circuits is equivalent to polynomial size extended Frege system, which corresponds to **P**. Similarly, we can prove that SAC^1-Frege corresponds to the theory $\mathbf{V}$-$Q^{SAC}(\Sigma_0^B)$.

In connection with propositional complexity, the above characterizations of **LOGCFL** might give lots of information. For example, the concept needed to define word problem for groupoids can be translated to a family of propositional formulae. So it is an interesting problem which propositional proof system has or does not have polynomial size proof of such a concept.

This note aims to give a guideline for the research of **LOGCFL** in terms of proof theory as stated above. We give several results obtained so far without mentioning proofs. We will further give some informal ideas for the future research.

2 The class LOGCFL

In this section we overview fundamental properties of the class **LOGCFL**. We will assume that readers are familiar with basic concepts of computational complexity and formal language theory.

Definition 1 **LOGCFL** *is the class of sets which are logspace reducible to some context free language.*

An alternative characterization is known for this class based on Boolean circuits. A semi-unbounded fan-in circuit is a Boolean circuit with basis $\{\wedge, \vee, \neg\}$ where $\wedge$ gates receive at most two inputs and $\neg$ gates appear only at the input level.

Definition 2 $\mathbf{SAC}^1$ *is the class of sets which are decidable by some semi-unbounded fan-in circuit family with* $O(\log n)$ *depth and* $n^{O(1)}$ *size.*

Theorem 1 (Venkateswaran [9]) *A set is in* **LOGCFL** *if and only if it is decidable by an* U_E^**-uniform* $\mathbf{SAC}^1$ *circuit family.*

From this characterization we obtain the following inclusion relation neither of which is known to be proper.

Corollary 1 $\mathbf{NL} \subseteq \mathbf{LOGCFL} \subseteq \mathbf{AC}^1$.

Several complete problems are known for **LOGCFL**.

A *groupoid* is a set with a (not necessarily be associative) binary operation. For a groupoid G, work problem for G is the problem of given a finite word $w = a_1 a_2 \cdots a_l$ over G, decide whether there exists a bracketing of w for which it is multiplied out to an element in given $S \subseteq G$.

Proposition 1 *The word problem for finite groupoids is complete for* **LOGCFL** *under constant depth reductions.*

The following complete problem is also noteworthy.

Theorem 2 (Gottlob et.al. [5]) *The acyclic conjunctive query problem is complete for* **LOGCFL** *under constant depth reductions.*

3 Theories for LOGCFL

Bounded arithmetic gives a logical framework for investigating complexity classes. There are several versions of such theories and two-sort systems gains much popularity recently. In particular, Cook, Kolokolova and Nguyen developed two sort theories which corresponds to complexity classes below **P** (see for example [6]).

In this section we give two theories of two sort which correspond to **LOGCFL**. Let us first give some definitions.

The language $\mathcal{L}_2$ of two sort systems consists of the followings:

- number variables : $x, y, z, \ldots$,
- string variables : $X, Y, Z, \ldots$,
- constant : 0,
- function symbols : $s(x) = x + 1$, $x + y$, $x \cdot y$, $|X|$,
- predicate symbols : $x < y$, $x = y$, $x \in X$.

Note that the intended model of $\mathcal{L}_2$ is $(\mathbb{N}, \Sigma^*)$ where $\Sigma = \{0, 1\}$.

Quantifiers of the form $(\forall x)$ and $(\exists x)$ are called number quantifiers and $(\forall X)$ and $(\exists X)$ are called string quantifiers. We define bounded string quantifiers as follows:

$$\begin{aligned}(\forall X < t)\varphi &\equiv (\forall X)(|X| < t \rightarrow \varphi)\\ (\exists X < t)\varphi &\equiv (\exists X)(|X| < t \wedge \varphi)\end{aligned}$$

Definition 3 *Σ_0^B is the set of $\mathcal{L}_2$ formulae in which all quantifiers are bounded number quantifiers. Σ_1^B is the set of $\mathcal{L}_2$ formulae whose quantifiers are either bounded number quantifiers, positive appearances of bounded existential string quantifiers or negative appearances of bounded universal string quantifiers.*

We are interested in the computational complexity of definable functions of a given $\mathcal{L}_2$ theory.

Definition 4 *A number function is a function of the form $f : \mathbb{N}^k \times (\Sigma^*)^l \rightarrow \mathbb{N}$. A string function is a function of the form $f : \mathbb{N}^k \times (\Sigma^*)^l \rightarrow \Sigma^*$.*

Definition 5 *Let T be a $\mathcal{L}_2$ theory and Φ be a class of $\mathcal{L}_2$ formulae. A string function F is Φ definable in T if there exists $\varphi \in \Phi$ such that*

$$T \vdash (\forall \bar{x})(\forall \bar{X})(\exists! Y)\varphi(\bar{x}, \bar{X}, Y)$$

and $(\mathbb{N}, \Sigma^*) \models (\forall \bar{x})(\forall \bar{X})\varphi(\bar{x}, \bar{X}, F(\bar{x}, \bar{X}))$.

3.1 A theory for $\mathbf{SAC}^1$

In [8], the author defined a theory whose provably total functions correspond to $\mathbf{SAC}^1$. Here we briefly overview this system.

Given an input to a Boolean circuit, we call a path an alternating path which gives a witness that the circuit accepts the input. Strictly, alternating paths are defined by induction on the depth of circuits.

For an $\mathcal{L}_2$-formula φ we define the predicate symbol Q_φ^{SAC} by

$$\begin{aligned}Q_\varphi^{SAC}(n, t) \Leftrightarrow\ & \text{there exists an alternating path with the root } t\\ & \text{in the circuit defined by } \varphi \text{ with the input length } n.\end{aligned}$$

and let $\mathbf{V}\text{-}Q^{SAC}(\Sigma_0^B)$ be the system with bit comprehension axioms for $\{Q_\varphi^{SAC}(n, t) : \varphi \in \Sigma_0^B\}$. Then we have

Theorem 3 (Kuroda [8]) $\mathbf{V}$-$Q^{SAC}(\Sigma_0^B)$ *captures LOGCFL.*

The essential part of the proof of Theorem 3 is to show that $Q^{SAC}(\Sigma_0^B)$ is strongly closed and, in particular, it is not obvious that SAC^1 is closed under complementation. Borodin et.al. [2] used the inductive counting to show this and we can formalize their argument inside $\mathbf{V}$-$Q^{SAC}(\Sigma_0^B)$.

3.2 A theory for permutation word problem

Our new theory for **LOGCFL** is based on a version of word problem for groupoids. The idea of defining this version of word problem is that we fix trees denoting the bracketing of a given word w and attach some permutation of w' to leaves of the tree.

For a node v of a tree let $label(v)$ denote the label assigned to v. We also denote the left and right offsprings of v by $left(v)$ and $right(v)$. The Permutation Word Problem (PWP) is the following decision problem:

Definition 6 *Let $\mathcal{T} = T_1, T_2, \ldots$ be a family of rooted binary trees so that T_n has n leaves. Let G be a finite groupoid and $S \subseteq G$. The permutation word problem for $(G; S)$, denoted by $PWP(\mathcal{T}, G, S)$, is the set of finite words w over G such that there exists a permutation w' of w satisfying the following condition:*

> *if we assign elements of w' to leaves of $T_{|w|}$ and compute each label of non-leaf node v by*
>
> $$label(v) = label(left(v)) \circ_G label(right(v))$$
>
> *then the label of the root of $T_{|w|}$ belongs to S.*

Definition 7 *BPWP is the problem PWP with the restriction that the family $\mathcal{T}$ of binary trees consists of balanced trees.*

Theorem 4 *BPWP is* **LOGCFL** *complete under constant depth reductions.*

Our system for **LOGCFL** is based on the problem BPWP. The essential axiom for the system expresses the computation of the word problem. In order to describe it, we need several functions and predicates which are definable in the base theory.

Definition 8 $\mathbf{V}^0$ *is the $\mathcal{L}_2$ theory with the following axioms:*

- *$BASIC_2$: a finite set of axioms which define symbols in $\mathcal{L}_2$.*
- *Σ_0^B-$COMP$: $(\exists X < a)(\forall y < a)(y \in X \leftrightarrow \varphi(y))$, where $\varphi \in \Sigma_0^B$.*

It is known that we can freely add Σ_1^B definable functions and Δ_1^B definable predicates in $\mathbf{V}^0$. Using this fact we introduce several functions and predicates in $\mathbf{V}^0$.

A two dimensional array X defines a finite function so that

$$X[y] = \min\{z : X(y, z)\}.$$

A finite groupoid is coded by a three dimensional array $n \times n \times n$. We will denote the binary operation $x \circ_G y$ by

$$G[x, y] = \min\{z : G(x, y, z)\}$$

The following formula asserts that any given word can be multiplied out to a single element according to a balanced tree:

$$\begin{aligned}&\delta_{GWP}(a, G, W, Y) \Leftrightarrow \\ &(\forall x < a)\,(Y[x + a] = W[x] \wedge 0 < x \rightarrow (Y[x] = G[Y[2x], Y[2x + 1]]))\,.\end{aligned}$$

Next we define functions and predicates which manipulate permutations. First define

$$\begin{aligned}Perm(F, n) \quad \Leftrightarrow \quad &(\forall x < n)(\exists y! < n)F(x, y) \wedge \\ &(\forall x < n)(\forall y < n)(\forall z < n)(F(x, z) \wedge F(y, z) \rightarrow x = y).\end{aligned}$$

$Perm(X, n)$ asserts that a permutation X of n is a bijection $[n] \rightarrow [n]$ and we have

Proposition 2 *$Perm(X, n)$ is Δ_1^B definable in $\mathbf{V}^0$.*

Next we define the application of a permutation to words.

$$\begin{aligned}&Trans(F, X, n) = Y \Leftrightarrow \\ &(Perm(F, n) \rightarrow (\forall i < n)(Y[F[i]] = X[i])) \wedge (\neg Perm(Fmn) \rightarrow X = Y).\end{aligned}$$

Proposition 3 *$Trans(F, X, n)$ is Σ_1^B definable in $\mathbf{V}^0$.*

Now we can define a single axiom which states that for a given groupoid G and a word W on G there exists a sequence which codes all elements of G so that some permutation of W can be multiplied out to it by a balanced tree. BPWPG (for Balanced Permutation Word Problem for Groupoids) is the following axiom:

$$\begin{aligned}&(\forall G)(\forall W)(\exists S)(\forall i < n) \\ &(S(i) \leftrightarrow (\exists F)(\exists Y)(Perm(F, n) \wedge \delta_{GWP}(n, G, Trans(F, W, n), Y) \wedge Y[n - 1] = i)).\end{aligned}$$

Definition 9 **T-PWP** *is the $\mathcal{L}_2$ theory whose axioms are*

- *all axioms for $\mathbf{V}^0$,*
- *BPWPG.*

Theorem 5 *A function is Σ_1^B definable in* **T-PWP** *if and only if it is bitwise computable in* **LOGCFL**.

Problem 1 *Show that* **T-PWP** *is identical to $\mathbf{V}$-$Q^{SAC}(\Sigma_0^B)$. That is, show that for all $\mathcal{L}_2$ formulae φ* **T-PWP** $\vdash \varphi$ *if and only if $\mathbf{V}$-$Q^{SAC}(\Sigma_0^B) \vdash \varphi$.*

4 Propositional proof system for LOGCFL

It is known that many complexity classes have its counterpart in propositional proofs. A natural candidate for for **LOGCFL** is $\mathbf{SAC}^1$ Frege proof system.

Definition 10 *Circuits C and D are similar (written as $C \simeq D$) if they are unfolded into the same formula.*

Definition 11 *A Circuit Frege (CF) proof system is defined as follows: Let $\mathcal{B}$ be a basis of Boolean formulae and $\mathcal{R}$ be a sound and implicationally complete set of Frege rules over $\mathcal{B}$. A CF proof is a sequence $A_0, \ldots, A_k$ of $\mathcal{B}$-circuits such that each A_i is either*

- *obtained by some $\mathcal{R}$ rule from $A_{j_1}, \ldots A_{j_l}$ for some $j_1, \ldots, j_l < i$, or*
- *$A_j \simeq A_i$ for some $j < i$.*

Definition 12 *A $\mathbf{SAC}^1$-Frege proof is a CF proof where circuits in the proof are restricted to $\mathbf{SAC}^1$ circuits.*

Theorem 6 *There exists a translation $\varphi(\bar{X}) \mapsto \|\varphi(\bar{X})\|_{\bar{m}}$ of Σ_0^B formulae into propositional formulae such that if $\mathbf{V}\text{-}Q^{SAC}(\Sigma_0^B) \vdash (\forall \bar{X})\varphi(\bar{X})$ where $\varphi \in \Sigma_0^B$ then $\|\varphi(\bar{X})\|_{\bar{m}}$ has polynomial size $\mathbf{SAC}^1$-Frege proofs.*

Theorem 7 *$\mathbf{V}\text{-}Q^{SAC}(\Sigma_0^B)$ proves the reflection principle for $\mathbf{SAC}^1$-Frege.*

5 Future research and open problems

In this note we present two theories for **LOGCFL** based on different characterization. Another interesting feature of this class is acyclic conjunctive queries defined in section 2. So it is plausible to define another system based on this characterization.

Problem 2 *Define a natural axiom $\mathbf{A}$ which expresses the concept of acyclic conjunctive queries so that $\mathbf{V}^0 + \mathbf{A}$ captures* **LOGCFL**.

A similar problem can be considered in the context of propositional proofs. Atserias et.al. [1] defined constraint based proof system called CSP-proofs.

Definition 13 *Let σ be a signature for finite models and $\mathcal{A}$ and $\mathcal{B}$ be σ-structures. The homomorphism problem is to decide whether there exists an homomorphism $\mathcal{A} \to \mathcal{B}$.*

It is easily seen that the homomorphism problem is equivalent to constraint satisfaction problem, which is in turn equivalent to conjunctive query problem.

Now the objective of CSP-proof is, given two structures $\mathcal{A}$ and $\mathcal{B}$, show that whether there is no homomorphism $\mathcal{A} \to \mathcal{B}$ in the following manner.

Definition 14 *A constraint is a pair* $(\bar{x}, R^{\mathcal{B}})$ *where* $\bar{x} = x_1, \ldots, x_l$ *is a tuple of elements of* $\mathcal{A}$ *and* $R^{\mathcal{B}}$ *is a relation over* $\mathcal{B}$ *of the arity* l*. For two relations* R, S *we denote their join by* $R \bowtie S$*.*

A CSP$(\mathcal{B})$*-proof from* $\mathcal{A}$ *is a finite sequence of constraints such that each constraint is either one of the followings:*

1. $(\bar{x}, R^{\mathcal{B}})$ *:* $R \in \sigma$ *and* $\bar{x} \in R^{\mathcal{A}}$*,*
2. $(\bar{x} \cup \bar{y}, R \bowtie S)$ *:* $(\bar{x}, R)$ *and* $(\bar{y}, S)$ *appears previously,*
3. $(\bar{x}, \pi_{\bar{x}-y}(R))$ *:* $(\bar{x}, R)$ *appears previously,*
4. $(\bar{x}, S)$ *:* $(\bar{x}, R)$ *appears previously and* $R \subseteq S$*.*

A CSP$(\mathcal{B})$*-refutation of* $\mathcal{A}$ *is a CSP*$(\mathcal{B})$*-proof from* $\mathcal{A}$ *whose last constraint has an empty relation.*

It is easily seen that CSP-refutation is sound and complete. However, this proof system itself cannot be Cook-Reckhow system as it does not satisfy the P-verifiability condition.

In order to overcome this problem, the notion of representation class is introduced:

Definition 15 *Let* B *be a finite set. A representation class for Boolean valued functions with domain* B^k *is the tuple* $\mathcal{R} = (Q, I, S)$ *where*

- Q *is the set of representations,*
- I *is a mapping* $Q \to \{f : B^k \to \{0,1\}\}$ *called interpretation,*
- S *is a mapping* $Q \to \mathbb{N}$ *called size function.*

Taking some computation model as the set of representation Q we obtain P-verifiable versions of CSP-proof system.

Definition 16 *Let* π *be an order on Boolean variables. A* π*-OBDD is a binary decision diagram in which all paths have labels which are consistent with* π*.*

Definition 17 *Let* π *be an order on variables. Then* π*-OBDD proof system is the CSP proof system where the representation class for constraints is* π *Ordered Binary Decision Diagram (OBDD).*

Since basic operations for OBDDs (such as the equivalence of two OBDDs etc) are polynomial time computable (see [3]), this proof system is known to be a Cook-Reckhow system.

However Krajíček [7] proved an exponential lower bounds to OBDD-proofs. So we ask the following question:

Problem 3 *Is there a representation class for which CSP-proof corresponds to* **LOGCFL**, **P** *etc?*

Apart from the relation to **LOGCFL**, CSP proof itself is interesting. We close this note by posing a question about CSP proofs.

Definition 18 *A Free BDD (FBDD) is a BDD such that in each path from the root to an end node, each variable is checked at most once.*

Given two FBDDs it is not known to be polynomial time computable to check whether they are identical. However, by introducing the notion of type (see [4]), we can check in polynomial time the equivalence of two given FBDDs of the same type. This leads to the definition of FBDD proofs as follows:

Definition 19 *Let τ be a type of Free BDDs. A τ-FBDD is a FBDD with type τ. A τ-FBDD proof is a CSP-proof whose representation class is τ-FBDDs.*

As models of computations, FBDD have exponential speed-up over OBDD. So the author conjectures

Conjecture 1 *FBDD-proofs have exponential speed-up over OBDD-proofs.*

参考文献

[1] Atserias, A., Ph.G.Kolaitis, and M.Y.Vardi. Constraint Propagation as a Proof System, in 10th International Conference on Principles and Practice of Constraint Programming (CP), vol.3258 Lecture Notes in Computer Science, (2004) pp.77–91.

[2] Borodin, A., S. A. Cook, P. W. Dymond, W. L. Ruzzo, and M. Tompa, Two applications of inductive counting for complementation problems, SIAM Journal of Computing, 18(3), (1989), pp.559–578.

[3] Bryant, R. E. Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams, ACM Computing Surveys, Vol. 24,No.3, (1992) pp. 293-318.

[4] Gergov,J., C. Meinel, Efficient Boolean Manipulation With OBDD's can be Extended to FBDD's. IEEE Trans. Computers 43(10): (1994), pp.1197–1209

[5] Gottlob, G., N. Leone, F. Scarcello. The Complexity of Acyclic Conjunctive Queries. Journal of the ACM, 48:3, (2001) pp431-498.

[6] Kolokolova, A., Systems of Bounded Arithmetic from Descriptive Complexity. Ph.D Dissertation, Toronto University, (2005).

[7] Krajíček, J., An exponential lower bound for a constraint propagation proof system based on ordered binary decision diagrams, J. of Symbolic Logic, 73(1), (2008), pp. 227-237.

[8] Kuroda, S., Generalized quantifier and a bounded arithmetic theory for LOGCFL, Archve for Mathematical Logic. 46(5-6), pp.489–516. (2007)

[9] Venkateswaran, H., Properties that characterize LOGCFL, Journal of Computer and System Sciences, 42 (1991), pp. 380-404.