A Survey of Complexity Classes and their Associated Propositional Proof Systems and Theories

and a Proof System for Log Space

Stephen Cook

October, 2001

Note: These posted transparencies lack hand-drawn diagrams.

We study triples $\langle C, S, T \rangle$ where

C is a complexity class
S is a propositional proof system
T is a theory

Example: $\langle P, \text{eFrege}, S^1_2 \rangle$

Note that $P[nonuniform]$ = families of circuits

–Lines in a eFrege proof represent circuits

–If S^1_2 proves $\forall x A(x)$ ($\forall \Sigma^b_0$) then $\langle A(x)[n] \rangle$ has polysize eFrege proofs.

–The functions Σ^b_1 definable in S^1_2 comprise precisely FP.

–S^1_2 proves soundness of eFrege. If S^1_2 proves soundness of system S, then eFrege p-simulates S.

$\forall \Sigma^b_0 - S^1_2$ is just a highly uniform version of eFrege.

Class	Proof System	Theory
AC^0	AC^0-Frege	$I\Delta_0(R), V^0, TAC^0$
$AC^0[2]$	$AC^0[2]$-Frege	$TAC^0(2)$
TC^0	TC^0-Frege	TTC^0
NC^1	Frege	AID, T_1, TNC^1
L	L-Frege	Σ_0^p-rec, TLS
DET		$\forall LAP$
NC	NC-Frege	U_1^1, R_2^1, TNC
P	eFrege	QPV, S_2^1, V_1^1
PLS	G_1	T_2^1
PSPACE	G, BoolPr	U_2^1, PSA

Each class has a uniform and a nonuniform version.

Each version has a predicate form and a function form.

Ground objects are binary strings.

Liar Game (Pudlak/Buss)

A is a formula over $\{\wedge, \vee, \neg\}$
PROVER claims A is a tautology
LIAR claims he has a falsifying assignment τ

1. PROVER queries A
2. LIAR asserts $A \rightarrow 0$ (τ falsifies A)

In general PROVER queries a formula B
LIAR replies either $B \rightarrow 0$ or $B \rightarrow 1$

LIAR loses if his answers form a *simple inconsistency*:

$\{A \wedge B \rightarrow 1, A \rightarrow 0\}$, $\{A \wedge B \rightarrow 1, B \rightarrow 0\}$
$\{A \wedge B \rightarrow 0, A \rightarrow 1, B \rightarrow 1\}$
$\{A \vee B \rightarrow 0, A \rightarrow 1\}$, $\{A \vee B \rightarrow 0, B \rightarrow 1\}$
$\{A \vee B \rightarrow 1, A \rightarrow 0, B \rightarrow 0\}$
$\{A \rightarrow u, \neg A \rightarrow u\}, u \in \{0, 1\}$

A proof of A in the LIAR Game is a winning strategy for PROVER.

A winning strategy is given by binary tree, with nodes labeled with PROVER's queries, and edges labeled with LIAR's replies.

Theorem(P/B) The LIAR Game is sound and complete. The minimal number of rounds in the game needed to prove A is proportional to the logarithm of the minimal number of steps in a Frege proof of A.

Corollary: The LIAR Game and Frege p-simulate each other.

Given a connective set (such as $\wedge, \vee, \neg$) the LIAR Game is a *canonical* proof system for formulas with those connective. This is unlike Frege systems, but like Gentzen systems.

The LIAR Game generalizes nicely.

The LIAR Game has a version corresponding to each of many complexity classes.

If a language in a nonuniform class C is specified by a polysize family of objects of type O, then the PROVER for C-Frege presents objects of type O to query.

For AC^0-Frege, the prover specifies a formula with (a fixed) bounded depth.

Similarly for $AC^0[2]$, TC^0, NC.

For P (eFrege) the PROVER specifies boolean circuits, given by straight-line programs.

For PSPACE, the PROVER specifies quantified boolean formulas (or boolean programs).

What about L (= LOGSPACE)?

Branching Programs

A language in nonuniform L is specified by a family of branching programs.

A *branching program* with inputs $\langle x_1, ..., x_n \rangle$ is a rooted DAG such that each nonleaf node is labeled by some x_i and has two out edges labeled 0 and 1. Each leaf is labeled 0 or 1.

Theorem: A language is in (uniform) L iff it is accepted by some log-space uniform polynomial size family of branching programs.

Syntactic definition of BP's

An n-BP ($n \geq 1$) is a nonempty set α of nodes, where each node has the form

$$v = (l, x_i : r, s)$$

where $l = \text{name}(v) \geq 2$

$1 \leq i \leq n$

$r, s \in \{0, 1\} \cup \{\text{name}(u) \mid u \in \alpha \ \& \ \text{name}(u) > l\}$

$u \neq v \Rightarrow \text{name}(u) \neq \text{name}(v)$

Semantics:

if $x_i = 0$ goto (output) r else goto (output) s

$r, s \in \{0, 1\}$ represent outputs.

$r, s \geq 2$ represent goto's.

L-Frege is the **LIAR Game for n-BPs**

Notation: Suppose $\alpha, \beta, \gamma, \delta$ are n-BPs

$$\alpha = \beta[0 \leftarrow \gamma, 1 \leftarrow \delta] \qquad (1)$$

means $\alpha = \beta' \cup \gamma \cup \delta$
where β' is β with 0 replaced by root(γ)
and 1 replaced by root(δ)
We also allow $\gamma, \delta \in \{0, 1\}$

Definition: Assuming (1) above
and $u \in \{0, 1\}$, the following are *simply inconsistent sets* of BP specs:

$\{\alpha \to u, \beta \to 0, \gamma \to 1 - u\}$
    or $\{\alpha \to u, \beta \to 0\}$ if $\gamma = 1 - u$
$\{\alpha \to u, \beta \to 1, \delta \to 1 - u\}$
    or $\{\alpha \to u, \beta \to 1\}$ if $\delta = 1 - u$
$\{\alpha \to u, \beta \to 1 - u\}$ and NF(α) =NF(β)

where NF(α) results from α by renumbering node names to be consecutive $2, 3, \ldots$ while preserving their order.

Translation of formulas to BPs

Formulas have connectives from $\{\wedge, \vee, \neg\}$ and atoms from $\{x_1, ..., x_n\}$

Notation: $\alpha[+k]$ is α with each name $l \geq 2$ replaced by $l + k$.

Recursive translation of $A \rightsquigarrow \mathsf{BP}(A)$

$\mathsf{BP}(x_j) = \{(2, x_j : 0, 1)\}$

$\mathsf{BP}(A \wedge B) = \mathsf{BP}(A)[0 \leftarrow 0,\ 1 \leftarrow \mathsf{BP}(B)[+k]]$

$\mathsf{BP}(A \vee B) = \mathsf{BP}(A)[0 \leftarrow \mathsf{BP}(B)[+k]],\ 1 \leftarrow 1]$

where $k = |\mathsf{BP}(A)|$

$\mathsf{BP}(\neg A) = \mathsf{BP}(A)[0 \leftarrow 1,\ 1 \leftarrow 0]$

Example: $A = (x_3 \wedge x_5) \vee \neg x_3$

$\mathsf{BP}(A) =$

$\{(2, x_3 : 4, 3), (3, x_5 : 0, 1), (4, x_3 : 1, 0)\}$

Theorem: L-Frege p-simulates Frege.
Proof: Given a prover strategy S for the LIAR Game for Frege, we define a strategy S' for L-Frege, where each time S queries a formula A, S' queries BP(A).

Suppose $A \to u$ and $\neg A \to u, u \in \{0, 1\}$ both occur on a branch of S.
Then $\{\text{BP}(A) \to u,\ \text{BP}(\neg A) \to u\}$ is simply inconsistent.

Suppose $A \wedge B \to 0, A \to 1,\ B \to 1$ all occur on a branch of S. Then S' adds the query

$$\text{BP}(B)[+k],\ (\text{where } k = |\text{BP}(A)|)$$

Note that BP(B) and BP(B)[$+k$]) each have the same normal form.

e

Theorem: L-Frege is sound and complete.
Proof: Soundness is easy. For completeness:
Lemma: If α is an n-BP and $\{\alpha \to v\} \cup C$ is semantically inconsistent for $v \in \{0,1\}$, where $C = \{x_{i_1} \to u_1, ..., x_{i_k} \to u_k\}$, then there is an L-Frege refutation of $\alpha \to v$ from assumptions $\mathrm{BP}(x_{i_1}) \to u_1$, ..., $\mathrm{BP}(x_{i_k}) \to u_k$.
Proof: Induction on the depth of α.
Basis: depth$(\alpha) = 1$. Then $\{\mathrm{NF}(\alpha) \to v, \mathrm{BP}(x_{i_j}) \to u_j\}$ is simply inconsistent for some j.
Ind Step: $\alpha = \beta[0 \leftarrow \gamma,\ 1 \leftarrow \delta]$ where
$\beta = \{(l, x_j : 0, 1)\}$

Then both $\{\gamma \to v, x_j \to 0\} \cup C$ and $\{\delta \to v, x_j \to 1\} \cup C$ are semantically inconsistent. Query β and $\mathrm{NF}(\beta)$ and apply the induction hypothesis.

Zambella's Log Space Theory ZL
(or $\Sigma_0^p - rec$)

Second-order (i.e. two-sorted) theory
$x, y, z, ...$ range over **N**
$X, Y, Z...$ range over finite subsets of **N**
Function symbols $0, 1, +, \cdot, |X|$
Predicate symbols $=, \leq, \in$

$|X| = 1 +$ largest element in X
$|\emptyset| = 0$
$X(y)$ stands for $y \in X$

Think of X as a bit string of length $|X|$
$X(0)X(1)...X(n-1), n = |X|$

Σ_0^B formulas have bounded number quantifiers and no string quantifiers.
Σ_0^B formulas represent AC^0 relations in their free string variables.

Axioms of ZL:

$\forall$ sentences representing simple properties of $0, 1, +, \cdot, \leq$

Predecessor: $x \neq 0 \rightarrow \exists y(y + 1 = x)$

Least upper bound:

$X(y) \rightarrow y < |X|$ and

$y + 1 = |X| \rightarrow X(y)$

Two axiom schemes: Σ_0^B-comp

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

$\Sigma_0^B - rec$:

$\forall x < a \exists y < a \phi(x, y) \rightarrow \exists Z \forall w < b \phi(Z(w), Z(w+1))$

where $Z(x)$ is the value at x of the function coded by the set Z.

This asserts that there are arbitrarily long paths through any given $\mathbf{\Sigma_0^B}$-definable directed graph without leaves.

Theorem: ZL proves the Σ_0^B-IND axioms.

Theorem: A function $f : \{0,1\}^* \to \{0,1\}^*$ is Σ_1^B definable in ZL iff f is log space computable.

Proof: $\Longleftarrow$: Use Σ_0^B-rec to define a log space computation.

$\Longrightarrow$: Use Buss witnessing for Σ_1^B formulas. The interesting case is the $S0B$ scheme.

Translating Σ_0^B formulas

We use the Paris-Wilkie method to translate a Σ_0^B formula $A(X)$ to a polysize bounded-depth family $\langle A(X)[n]\rangle$, with atoms corresponding to the n bits of X.

This in turn is translated into a Branching Program family $\langle \mathrm{BP}(A(X)[n])$.

CLAIM: If ZL $\vdash \forall X A(X)$ then $\langle \mathrm{BP}(A(X)[n])$ has polysize L-Frege proofs.

The idea is to witness Σ_1^B formulas using branching programs.

CLAIM: ZL proves the soundness of L-Frege.

Conclusion

Separating the $\forall\Sigma_0^B$ fragments of the logical theories is as interesting (and maybe easier) than separating the corresponding proof systems.

This is like separating the uniform versions of the complexity classes (e.g. P $\neq$ NP), as opposed to separating the non-uniform classes (e.g. circuit size lower bounds).

If theory T corresponds to proof system S, then T proves "S is sound", but this is probably not provable by the weaker theories. PROBLEM: Find natural equivalent statements.

Example: Avigad's combinatorial DAG principle separates the $\forall\Sigma_0^b$ fragments of S_2^1 and AID, if anything does.

Does $(AB = I \to BA = I)$ play such a role in linear algebra theory?

Define a canonical theory for each class.