Applicative theories for logarithmical complexity classes

Sebastian Eberhard

Universität Bern

June 4, 2012

Related research

- Clote, Takeuti: First order bounded arithmetic and small Boolean Circuit Complexity Classes.
- Cantini: Polytime, combinatory logic and positive safe induction
- Kahle, Oitavem: An applicative theory for FPH
- Strahm: Theories with self-application and computational complexity

Important functions within logarithmic complexity classes

- LogTime: $x \mapsto |x|$
- LogTH: Addition

Important functions within logarithmic complexity classes

- LogTime: $x \mapsto |x|$
- LogTH: Addition
- ALogTime: Multiplication

Important functions within logarithmic complexity classes

- LogTime: $x \mapsto |x|$
- LogTH: Addition
- ALogTime: Multiplication

Important functions within logarithmic complexity classes

- LogTime: $x \mapsto |x|$
- LogTH: Addition
- ALogTime: Multiplication

We have LogTime $\subseteq$ LogTH $\subset$ ALogTime $\subseteq$ LogSpace $\subseteq$ PolyTime

Important functions within logarithmic complexity classes

- LogTime: $x \mapsto |x|$
- LogTH: Addition
- ALogTime: Multiplication

We have LogTime $\subseteq$ LogTH $\subset$ ALogTime $\subseteq$ LogSpace $\subseteq$ PolyTime

Function algebra $\mathfrak{W}_1$ for LogTH

Definition

$\mathfrak{W}_1$ contains the following initial functions.

- the constant empty word function.
- the word successor functions.
- projection functions of arbitrary arity.
- the function bit such that $\mathrm{bit}(i, w)$ is the i-th bit of the word w.
- the function abs such that $\mathrm{abs}(w)$ is the length of the word w.
- the function $\times$ where $w \times v$ is the length of v fold concatenation of w with itself.
- the function e where $\mathrm{e}(w)$ is the word w without its leading zeros.

Function algebra for LogTH II

$\mathfrak{W}_1$ is closed under composition and under the following scheme of concatenation recursion on notation *CRN*. We abbreviate $CRN(g, h_0, h_1)$ by f.

$$\begin{array}{lcl} f(\epsilon, \vec{y}) & = & g(\vec{y}) \\ f(\mathsf{S}_0(x), \vec{y}) & = & \mathsf{S}_{\mathsf{BIT}(\epsilon, h_0(x, \vec{y}))}(f(x, \vec{y})) \\ f(\mathsf{S}_1(x), \vec{y}) & = & \mathsf{S}_{\mathsf{BIT}(\epsilon, h_1(x, \vec{y}))}(f(x, \vec{y})) \end{array}$$

Formalise two aspects of *CRN*:

- $f(\mathsf{S}_i(x), \vec{y})$ extends $f(x, \vec{y})$ by one bit.
- Recursion step function is independent of $f(x, \vec{y})$.

Function algebra for LogTH II

$\mathfrak{W}_1$ is closed under composition and under the following scheme of concatenation recursion on notation *CRN*. We abbreviate $CRN(g, h_0, h_1)$ by f.

$$
\begin{array}{lcl}
f(\epsilon, \vec{y}) & = & g(\vec{y}) \\
f(\mathsf{S}_0(x), \vec{y}) & = & \mathsf{S}_{\mathsf{BIT}(\epsilon, h_0(x,\vec{y}))}(f(x, \vec{y})) \\
f(\mathsf{S}_1(x), \vec{y}) & = & \mathsf{S}_{\mathsf{BIT}(\epsilon, h_1(x,\vec{y}))}(f(x, \vec{y}))
\end{array}
$$

Formalise two aspects of *CRN*:

- $f(\mathsf{S}_i(x), \vec{y})$ extends $f(x, \vec{y})$ by one bit.
- Recursion step function is independent of $f(x, \vec{y})$.

Provably total functions

Definition

A function $F : \mathbb{W}^n \to \mathbb{W}$ is called *provably total in an* L *theory* T, if there exists a closed L term t_F such that

(i) $\mathsf{T} \vdash t_F : \mathsf{W}^n \to \mathsf{W}$ and, in addition,

(ii) $\mathsf{T} \vdash t_F \overline{w}_1 \cdots \overline{w}_n = \overline{F(w_1, \ldots, w_n)}$ for all $w_1, \ldots, w_n$ in $\mathbb{W}$.

The language of LogH

- Combinators: s, k
- Pairing and projection: $\mathsf{p}, \mathsf{p}_0, \mathsf{p}_1$
- Empty word: ϵ
- Successor functions: $\mathsf{s}_0, \mathsf{s}_1, \mathsf{s}_\ell$
- Predecessor function: $\mathsf{p}_W, \mathsf{p}_\ell$
- Definition by cases: d_W
- Concatenation: $*$
- Multiplication: $\times$
- Length: abs
- Bit: bit
- Eraser: e

- Unary predicate W for (fully accessible) words
- Unary predicate V for temporary inaccessible words

Axioms of LogH

- Classical logic.
- The usual defining properties for the constants for inputs in W.

- $x \in \mathsf{W} \rightarrow x \in \mathsf{V}$
- $x \in \mathsf{V} \rightarrow \mathsf{s}_i x \in \mathsf{V}$
- $x \in \mathsf{V} \rightarrow \mathsf{p_W} x \in \mathsf{V}$

- $\dfrac{A \rightarrow t \in \mathsf{V}}{A \rightarrow t \in \mathsf{W}}$, where A is positive, and does not contain V.

Induction

(LogH − Ind)

$$(\exists y \in \mathsf{V})A[\epsilon, y])\wedge$$
$$(\forall x \in \mathsf{W})(\forall y \in \mathsf{V})\big(A[x, y] \to (A[\mathsf{s}_i x, \mathsf{s}_0 y] \vee A[\mathsf{s}_i x, \mathsf{s}_1 y])\big) \to$$
$$(\forall x \in \mathsf{W})(\exists y \in \mathsf{V})A[x, y],$$

where A is positive, and W, V and disjunction free.

Effect of allowing disjunctions

Assume
(ALogT − Ind)

$$(\exists y \in \mathsf{V})A[\epsilon, y]) \wedge$$
$$(\forall x \in \mathsf{W})(\forall y \in \mathsf{V})\big(A[x, y] \rightarrow (A[\mathsf{s}_i x, \mathsf{s}_0 y] \vee A[\mathsf{s}_i x, \mathsf{s}_1 y])\big) \rightarrow$$
$$(\forall x \in \mathsf{W})(\exists y \in \mathsf{V})A[x, y],$$

for A positive, W and V free.

$$s \preceq \overline{11 \cdots 11} := s = \epsilon \vee s = \overline{0} \vee s = \overline{1} \vee s = \overline{00} \vee \cdots \vee s = \overline{11 \cdots 11}$$

We can prove

$$\mathsf{LogH} \vdash s \preceq \overline{11 \cdots 11} \leftrightarrow s \in \mathsf{W} \wedge s \leq \overline{11 \cdots 11}.$$

$\Rightarrow$ k bounded recursion can be justified.

Effect of allowing disjunctions

Assume
(ALogT − Ind)

$$(\exists y \in \mathsf{V})A[\epsilon, y])\wedge$$
$$(\forall x \in \mathsf{W})(\forall y \in \mathsf{V})\big(A[x,y] \rightarrow (A[\mathsf{s}_i x, \mathsf{s}_0 y] \vee A[\mathsf{s}_i x, \mathsf{s}_1 y])\big) \rightarrow$$
$$(\forall x \in \mathsf{W})(\exists y \in \mathsf{V})A[x,y],$$

for A positive, W and V free.

$$s \preceq \overline{11\cdots 11} := s = \epsilon \vee s = \overline{0} \vee s = \overline{1} \vee s = \overline{00} \vee \cdots \vee s = \overline{11\cdots 11}$$

We can prove

$$\mathsf{LogH} \vdash s \preceq \overline{11\cdots 11} \leftrightarrow s \in \mathsf{W} \wedge s \leq \overline{11\cdots 11}.$$

$\Rightarrow$ k bounded recursion can be justified.

Proof theoretic strength

Theorem

- The theory LogH (containing $(\mathsf{LogTH}-\mathsf{Ind})$) proves totality exactly for the functions in the logarithmic hierarchy.
- The theory LogH extended by $(\mathsf{ALogT}-\mathsf{Ind})$ proves totality exactly for the functions computable in alternating logarithmic time.

Function algebra for logspace

- Initial functions
- Concatenation recursion
- Sharply bounded recursion:

$$
\begin{aligned}
f(\epsilon, \vec{y}) &= g(\vec{y})|\,|b(\epsilon, \vec{y})| \\
f(\mathsf{S}_i(x), \vec{y}) &= h_i(x, \vec{y}, f(x, \vec{y})))|\,|b(\mathsf{S}_i(x), \vec{y})|
\end{aligned}
$$

$\Rightarrow$ allow to access a sharply bounded initial segment.

Function algebra for logspace

- Initial functions
- Concatenation recursion
- Sharply bounded recursion:

$$\begin{aligned} f(\epsilon, \vec{y}) &= g(\vec{y})|\,|b(\epsilon, \vec{y})| \\ f(\mathsf{S}_i(x), \vec{y}) &= h_i(x, \vec{y}, f(x, \vec{y})))|\,|b(\mathsf{S}_i(x), \vec{y})| \end{aligned}$$

$\Rightarrow$ allow to access a sharply bounded initial segment.

Axioms for V in LogSp

- $x \in \mathsf{W} \rightarrow x \in \mathsf{V}$
- $x \in \mathsf{V} \rightarrow \mathsf{s}_i x \in \mathsf{V}$
- $x \in \mathsf{V} \rightarrow \mathsf{p}_\mathsf{W} x \in \mathsf{V}$
- $\dfrac{A \rightarrow t \in \mathsf{V}}{A \rightarrow t \in \mathsf{W}}$, where A is positive, and does not contain V.

New axiom to access sharply bounded initial segment:

- $x \in \mathsf{W} \wedge y \in \mathsf{V} \rightarrow y|_{|x|} \in \mathsf{W}$,
- $A[\epsilon] \wedge (\forall x \in \mathsf{W})(A[x] \rightarrow A[\mathsf{s}_i x]) \rightarrow (\forall x \in \mathsf{W})A[x]$,
  for A positive and W free.

Two sorted function algebra $\mathcal{A}$ corresponding to LogSp

$\mathcal{A}$ is the analogon of Bellantoni's BC. It is given as follows:

- Initial functions ϵ , s_0, s_1, p_W with safe output and safe input. Initial functions ABS, BIT with normal input and normal output. Initial functions $\pi_i^{n,m}$ (projections) with safe output and both normal and safe inputs. The initial function $init.seg(x; y) = y|_{|x|}$ with normal output.
- Closure under ordinary composition

  $$f(\vec{x}; \vec{y}) = h(\vec{g}(\vec{x}; \vec{y}); \vec{j}(\vec{x}; \vec{y})),$$

  where the g_i have normal output. f has the same sort of outputs as h.
- Closure under safe recursion on notation defined as follows

  $$\begin{aligned} f(\vec{x}, \epsilon; \vec{y}) &:= g(\vec{x}; \vec{y}) \\ f(\vec{x}, \mathsf{s}_i w; \vec{y}) &:= h_i(\vec{x}, w; f(\vec{x}, w; \vec{y}), \vec{y}), \end{aligned}$$

  where g, h_0, h_1 are elements of Alg with safe output. The f has safe output.
- Closure under raising: from $f(\vec{x};)$ with safe output obtain $f^\nu(\vec{x};)$ with normal output.

Treatment of $\mathcal{A}$ in LogSp

Let $F(\vec{x};\vec{y})$ be a function in $\mathcal{A}$ with normal/safe output. Then there exists a closed term t_F such that

- $\mathsf{LogSp} \vdash \vec{x} \in \mathsf{W}, \vec{y} \in \mathsf{V} \Rightarrow t_F\vec{x}\vec{y} \in \mathsf{W}/\mathsf{V}$
- $\mathsf{LogSp} \vdash t_F\overline{w}_1 \cdots \overline{w}_n = \overline{F(w_1, \ldots, w_n)}$ for all $w_1, \ldots, w_n$ in $\mathbb{W}$.

Treatment of LogSp in $\mathcal{A}$

It is possible to realise LogSp within $\mathcal{A}$.

- Realisers of formulas *with* occurrence of W are *normal* arguments of realisation functions.
- Realisers of formulas *without* occurrence of W are *safe* arguments of realisation functions.
- Safe inputs have to be inserted component-wise into realisation functions (similarly as in Cantini's treatment of B).

Proof theoretic strength

Theorem

The theory LogSp proves totality exactly for the functions computable in logarithmic space.