Sublogarithmic uniform Boolean Proof Nets
DICE 2011 Second Workshop on Developments in Implicit
Computational Complexity

Clément Aubert
aubert@lipn.univ-paris13.fr

Institut Galilée - Université Paris-Nord
99, avenue Jean-Baptiste Clément
93430 Villetaneuse

2nd-3rd April 2011, Saarbrücken
Two models of parallel computation

- Boolean Circuits
- Turing machine
- Proof Nets

Parallel evaluation

Cut-elimination

Implicit complexity framework:
- Size and depth
- Resources needed by the reductions
- Uniformity
Two models of parallel computation

Boolean Circuits

Parallel evaluation

Simulation

Translation

Boolean Proof Nets

Cut-elimination

Implicit complexity framework:
• Size and depth
• Resources needed by the reductions
• Uniformity
Two models of parallel computation

- Boolean Circuits
  - Parallel evaluation

- Boolean Proof Nets
  - Cut-elimination

Simulation
Translation

Implicit complexity framework:
- Size and depth
- Resources needed by the reductions
- Uniformity
Previous works and new results

Previous works

- [Terui, 2004] did not take into account uniformity.
- [Mogbil and Rahli, 2007] did, but translation was in $L$. 
Previous works

- [Terui, 2004] did not took into account *uniformity*.
- [Mogbil and Rahli, 2007] did, but translation was in $L$.

Results

- *Proof Circuits* lighten the size and translate all kind of Boolean Circuits.
- *Translation in AC*⁰ *concerning uniform families*.
- *Proof Nets compute functions under* $L$. 
Map

- First definitions: Boolean Circuits and Boolean Proof Nets
- Simulation of Boolean Proof Nets by Boolean Circuits
- Examples of translation and composition
- Proof Circuits and translation
Examples (Two Boolean Circuits)

Definition (Basis)

\[ B_0 = \{ \neg, \land_2, \lor_2 \} \]

\[ B_1 = \{ \neg, (\land_n)_{n \geq 2}, (\lor_n)_{n \geq 2} \} \].
Examples (Two Boolean Circuits)

Definition (Basis)

$$\mathcal{B}_0 = \{\neg, \land^2, \lor^2\}$$

$$\mathcal{B}_1 = \{\neg, (\land^n)_{n \geq 2}, (\lor^n)_{n \geq 2}\}.$$
Complexity Classes: NC and AC

Definition \((NC^i \text{ resp. } AC^i)\)

For \(i \in \mathbb{N}\), set of boolean functions computable by a uniform family of Boolean Circuits \(C = (C_n)\) where for all \(C_n\)

- its depth is \(O(\log^i n)\),
- its size is polynomial in \(n\),
- its gate are labeled with functions of \(B_0\) resp. \(B_1\).

\[
NC = \bigcup_{i \in \mathbb{N}} NC^i = \bigcup_{i \in \mathbb{N}} AC^i = AC
\]
Definition \((NC^i)\) resp. \((AC^i)\)

For \(i \in \mathbb{N}\), set of boolean functions computable by a uniform family of Boolean Circuits \(C = (C_n)\) where for all \(C_n\)

- its depth is \(O(\log^i n)\),
- its size is polynomial in \(n\),
- its gate are labeled with functions of \(B_0\) resp. \(B_1\).

\[
NC = \bigcup_{i \in \mathbb{N}} NC^i = \bigcup_{i \in \mathbb{N}} AC^i = AC
\]

Theorems

\[
AC^0 \subsetneq NC^1 \subseteq L \subseteq NL \subseteq AC^1 \subseteq NC^1 \subseteq AC^2 \subseteq \ldots
\]

\[
AC^0(UstCONN_2) \subseteq AC^1 \supseteq L
\]
Definition (Rules of MLLu)

\[
\begin{align*}
\frac{}{\vdash A, A^\perp} & \quad \text{ax.} \\
\frac{\vdash \Gamma, A \quad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta} & \quad \text{cut} \\
\frac{\vdash \Gamma_1, A_1 \quad \ldots \quad \vdash \Gamma_n, A_n}{\vdash \Gamma_1, \ldots, \Gamma_n, \otimes^n(A_1, \ldots, A_n)} & \quad \otimes^n \\
\frac{\vdash \Gamma, A_n, \ldots, A_1}{\vdash \Gamma, \otimes^n(A_n, \ldots, A_1)} & \quad \otimes^n
\end{align*}
\]

Definition (Boolean Type, [Terui, 2004])

\[
\begin{align*}
\frac{}{\vdash \alpha^\perp, \alpha} & \quad \text{ax.} \\
\frac{}{\vdash \alpha^\perp, \alpha} & \quad \text{ax.} \\
\frac{\vdash \alpha^\perp, \alpha}{\vdash \alpha^\perp, \alpha^\perp, \alpha \otimes \alpha} & \quad \otimes^2 \\
\frac{\vdash \alpha^\perp, \alpha^\perp, \alpha \otimes \alpha}{\vdash \otimes^3(\alpha^\perp, \alpha^\perp, \alpha \otimes \alpha)} & \quad \otimes^3
\end{align*}
\]
Definition (Rules of MLLu)

\[
\frac{\vdash A, A^\perp}{\vdash \Gamma, A, A^\perp} \quad \text{ax.} \quad \frac{\vdash \Gamma_1, A_1, \ldots, \vdash \Gamma_n, A_n}{\vdash \Gamma_1, \ldots, \Gamma_n, \otimes^n(A_1, \ldots, A_n)} \quad \otimes^n \\
\frac{\vdash \Gamma, A \quad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta} \quad \text{cut} \quad \frac{\vdash \Gamma, A_n, \ldots, A_1}{\vdash \Gamma, \otimes^n(A_n, \ldots, A_1)} \quad \otimes^n
\]

Definition (Boolean Type, [Terui, 2004])

\[
\frac{\vdash \alpha^\perp, \alpha^\perp}{\vdash \alpha^\perp, \alpha, \alpha^\perp} \quad \text{ax.} \quad \frac{\vdash \alpha^\perp, \alpha^\perp}{\vdash \alpha^\perp, \alpha^\perp, \alpha \otimes \alpha} \quad \otimes^2 \\
\frac{\vdash \alpha^\perp, \alpha^\perp, \alpha \otimes \alpha}{\vdash \alpha^\perp, \alpha \otimes \alpha} \quad \otimes^3 \\
\frac{\vdash \alpha^\perp, \alpha \otimes \alpha}{\vdash \alpha^\perp, \alpha \otimes \alpha} \quad \otimes^3 \\
\frac{\vdash \alpha \otimes \alpha}{\vdash \alpha \otimes \alpha} \quad \otimes^3
\]
Definition (Rules of MLLu)

\[
\frac{\vdash \Gamma, A \quad \vdash \Delta, A}{\vdash \Gamma, \Delta} \quad \text{cut}
\]

\[
\frac{\vdash \Gamma_1, A_1 \quad \ldots \quad \vdash \Gamma_n, A_n}{\vdash \Gamma_1, \ldots, \Gamma_n, \otimes^n(A_1, \ldots, A_n)}
\]

\[
\frac{\vdash \Gamma, A_n, \ldots, A_1}{\vdash \Gamma, \otimes^n(A_n, \ldots, A_1)}
\]

Definition (Boolean Type, [Terui, 2004])

\[
\frac{\vdash p : \alpha^\perp, p : \alpha \triangleright \text{ax}_p}{\vdash p : \alpha^\perp, q : \alpha \triangleright \text{ax}_q \quad \text{ax.}}
\]

\[
\frac{\vdash p : \alpha^\perp, q : \alpha^\perp, r : \alpha \otimes \alpha \triangleright \text{tensor}^{p,q}_r(ax_p, ax_q)}{\vdash s : \otimes^3(\alpha^\perp, \alpha^\perp, \alpha \otimes \alpha) \triangleright \text{par}^{q,p,r}_s(\text{tensor}^{p,q}_r(ax_p, ax_q))}
\]

\[
\vdash s : \otimes^3(\alpha^\perp, \alpha^\perp, \alpha \otimes \alpha) \triangleright \text{par}^{q,p,r}_s(\text{tensor}^{p,q}_r(ax_p, ax_q))
\]

\[
\vdash s : \otimes^3(\alpha^\perp, \alpha^\perp, \alpha \otimes \alpha) \triangleright \text{par}^{q,p,r}_s(\text{tensor}^{p,q}_r(ax_p, ax_q))
\]
Definition (Boolean Proof Nets, [Terui, 2004])

A Boolean Proof Net $P(\overrightarrow{p})$ with $n$ inputs is a Proof Net of type

$$
\vdash p_1 : B^\perp[A_1], \ldots, p_n : B^\perp[A_n], s : \otimes^{1+m}(B[A], C_1, \ldots, C_m)
$$

Definition ($P(\overrightarrow{b})$)

\[
\overrightarrow{b} \equiv \begin{array}{c}
\begin{array}{c}
\otimes
\end{array}
\end{array}
\]

$P(\overrightarrow{p}) \equiv \begin{array}{c}
\begin{array}{c}
\otimes
\end{array}
\end{array}$
Definition (Boolean Proof Nets, [Terui, 2004])

A Boolean Proof Net $P(\overrightarrow{p})$ with $n$ inputs is a Proof Net of type

$$\vdash p_1 : B_1^\bot[A_1], \ldots, p_n : B_n^\bot[A_n], s : \otimes^{1+m}(B[A], C_1, \ldots, C_m)$$

Definition ($P(\overrightarrow{b})$)

$P(\overrightarrow{b}) \equiv \begin{array}{cccc}
\begin{array}{c}
 b_0 \\
 b_1 \\
 b_1
\end{array} & \cdots & \begin{array}{c}
 b_0 \\
 b_1 \\
 b_1
\end{array}
\end{array}$
Definition ($\rightarrow_m$ and $\rightarrow_a$)

Remark (Critical pair)
Definition ($\rightarrow_m$ and $\rightarrow_a$)

Remark (Critical pair)
Definition ($\rightarrow_m$ and $\rightarrow_a$)

```
\[\cdots \rightarrow_m \cdots\]
```

Remark (Critical pair)

```
\[\rightarrow_a \rightarrow_a\]
```

Cut-elimination
Definition ($\rightarrow_m$ and $\rightarrow_a$)

Remark (Critical pair)

Definition ($t$-reduction)
Map

- First definitions: Boolean Circuits and Boolean Proof Nets
- Simulation of Boolean Proof Nets by Boolean Circuits
- Examples of translation and composition
- Proof Circuits and translation
Theorem ([Terui, 2004])

Every Proof-Net \( P \) normalizes in at most \( 3 \times d(P) \) steps of parallel reduction (\( \Rightarrow_t, \Rightarrow_a, \Rightarrow_m \)).
Theorem ([Terui, 2004])

Every Proof-Net $P$ normalizes in at most $3 \times d(P)$ steps of parallel reduction ($\Rightarrow_t$, $\Rightarrow_a$, $\Rightarrow_m$).

Description of a Boolean Proof Net $P_n$

\[
\begin{aligned}
\downarrow \\
\text{Simulates } \Rightarrow_t \\
\text{Simulates } \Rightarrow_a \\
\text{Simulates } \Rightarrow_m
\end{aligned}
\]
Theorem ([Terui, 2004])

Every Proof-Net \( P \) normalizes in at most \( 3 \times d(P) \) steps of parallel reduction (\( \Rightarrow_t, \Rightarrow_a, \Rightarrow_m \)).
Simulation of $\Rightarrow$ by Boolean Circuits

Theorem ([Terui, 2004])

Every Proof-Net $P$ normalizes in at most $3 \times d(P)$ steps of parallel reduction ($\Rightarrow_t$, $\Rightarrow_a$, $\Rightarrow_m$).

Description of a Boolean Proof Net $P_n$

\[ \text{Constant-depth size } O(|P_n|^3) \]

\[ \begin{align*}
&\text{Simulates } \Rightarrow_t \\
&\text{Simulates } \Rightarrow_a \\
&\text{Simulates } \Rightarrow_m
\end{align*} \]

\[ d(P_n) \text{ times} \]

\[ \text{Establishes the output of } P_n \]
Theorem ([Terui, 2004])

Every Proof-Net $P$ normalizes in at most $3 \times d(P)$ steps of parallel reduction ($\Rightarrow_t, \Rightarrow_a, \Rightarrow_m$).

Description of a Boolean Proof Net $P_n$

- Simulates $\Rightarrow_t$ $d(P_n)$ times
- Simulates $\Rightarrow_a$ $d(P_n)$ times
- Simulates $\Rightarrow_m$ $d(P_n)$ times

Constant-depth size $O(|P_n|^3)$

Establishes the output of $P_n$

Depth $d(P_n)$ size $O(|P_n|^4)$ over $\mathcal{B}_1(UstCONN_2)$
Map

- First definitions: Boolean Circuits and Boolean Proof Nets
- Simulation of Boolean Proof Nets by Boolean Circuits
- Examples of translation and composition
- Proof Circuits and translation
First example: negation
First example: negation
First example: negation
First example: negation
First example: negation
Second example: conditional
Second example: conditional
Second example: conditional

Composition:
\[ \sqrt[4]{\ref{Terui, 2004}} \]

Translation in [Terui, 2004]
\[\propto^4\] translated

Translation in [Aubert, 2010]
Map

- First definitions: Boolean Circuits and Boolean Proof Nets
- Simulation of Boolean Proof Nets by Boolean Circuits
- Examples of translation and composition
- Proof Circuits and translation
\[ b_0 \equiv \begin{array}{c}
\text{(Diagram of}\ b_0) \\
\end{array} \]

\[ b_1 \equiv \begin{array}{c}
\text{(Diagram of}\ b_1) \\
\end{array} \]

\[ \text{COND} \equiv \begin{array}{c}
\text{(Diagram of}\ \text{COND}) \\
\end{array} \]

\[ \text{NEG} \equiv \begin{array}{c}
\text{(Diagram of}\ \text{NEG}) \\
\end{array} \]
$CONJ^i \equiv$

\[
\begin{align*}
&b_0 \quad e_1 \\
&\quad b_0 \\
&\quad e_2 \\
&g_1 \\
&\quad g_1 \\
&\quad g_{j-1} \\
&\quad g_{j-1} \\
&\quad g_{i-1} \\
&\quad s \\
&i - 3 \text{ times}
\end{align*}
\]

\[
\begin{align*}
&b_0 \\
&\quad b_0 \\
&\quad e_j \\
&\quad e_j \\
&\quad e_i \\
&\quad e_i
\end{align*}
\]
$DUPL^i \equiv \begin{array}{cc}
\begin{array}{ll}
\ldots & \ldots \\
\times & \times \\
b_0 & b_0 \\
\end{array}
&
\begin{array}{ll}
\ldots & \ldots \\
\times & \times \\
b_1 & b_1 \\
\end{array}
\end{array}
\begin{array}{cc}
n & \begin{array}{ll}
S_1 \\
S_2 \\
\ldots \\
S_{i-1} \\
S_i \\
\end{array}
\end{array}$
Definition (Proof Circuits)

Theorem ([Aubert, 2010])
Every Proof Circuits is a Boolean Proof Net.
Definition (Proof Circuits)

\[ P_1 \quad e_1 \quad e_2 \quad s \quad P_2 \quad e_1 \quad e_2 \quad e_3 \quad s \quad P_3 \quad e_1 \quad e_2 \quad S \]

Theorem ([Aubert, 2010])
Every Proof Circuits is a Boolean Proof Net.
Definition (Proof Circuits)

\[
\begin{align*}
P_1 & \\ s & \quad e_1 & \quad e_2 & \\

P_2 & \\ s & \quad e_1 & \quad e_2 & \quad e_3 & \\

P_3 & \\ s & \quad e_1 & \quad e_2 & \\
\end{align*}
\]

Theorem ([Aubert, 2010])
Every Proof Circuits is a Boolean Proof Net.
Definition (Proof Circuits)

\[ \mathcal{P}_1 \]
\[ \mathcal{P}_2 \]
\[ \mathcal{P}_3 \]

Theorem [Aubert, 2010]
Every Proof Circuits is a Boolean Proof Net.
Definition (Proof Circuits)

Proof Circuits

Theorem ([Aubert, 2010])
Every Proof Circuits is a Boolean Proof Net.
Definition (Proof Circuits)

Theorem ([Aubert, 2010])

Every Proof Circuits is a Boolean Proof Net.
Definition \((CCP^i [Aubert, 2010])\)

For \(i \in \mathbb{N}\), set of boolean functions computable by a uniform family of Proof Circuits \(P = (P_n)\) where for all \(P_n\)

- its depth is \(O(\log^i n)\),
- its size is polynomial in \(n\).

\[
CCP = \bigcup_{i \in \mathbb{N}} CCP^i
\]
Definition \((CCPi \ [\text{Aubert, 2010}])\)

For \(i \in \mathbb{N}\), set of boolean functions computable by a uniform family of Proof Circuits \(P = (P_n)\) where for all \(P_n\)

- its depth is \(O(\log^i n)\),
- its size is polynomial in \(n\).

\[
CCP = \bigcup_{i \in \mathbb{N}} CCPi
\]

Theorem (Simulation)

For all \(i \in \mathbb{N}\),

\[
CCPi \subseteq AC^i(\text{UstCONN}_2)
\]
Definition (Translation from $AC^i$ to $CCP^i$)

Input: Description of a uniform family of Boolean Circuits $C = (C_n)$ in $AC^i$.

Output: Description of a family of Proof Circuits $P = (P_n)$ in $CCP^i$ so that for all $k$, for all $\overrightarrow{b}$, $P_k(\overrightarrow{b}) \rightarrow_{ev} b_j$ iff $C_k(\overrightarrow{b})$ evaluates to $j$. 

Theorem
Translation from $AC^i$ to $CCP^i$ belongs to $AC^0$. 

$1 \land 2 \lor 3 \neg 3$
Complexity of the translation

Definition (Translation from $AC^i$ to $CCP^i$)

Input: Description of a uniform family of Boolean Circuits $C = (C_n)$ in $AC^i$.

Output: Description of a family of Proof Circuits $P = (P_n)$ in $CCP^i$ so that for all $k$, for all $\vec{b}$, $P_k(\vec{b}) \rightarrow_{ev} b_j$ iff $C_k(\vec{b})$ evaluates to $j$.

Theorem

Translation from $AC^i$ to $CCP^i$ belongs to $AC^0$. 
Theorem

Translation from $AC^i$ to $CCP^i$ belongs to $AC^0$. 

$$
\begin{align*}
&\neg \lor ^3 \\
\lor ^3 \\
\land ^3 \\
\end{align*}
$$
Theorem

Translation from $AC^i$ to $CCP^i$ belongs to $AC^0$. 

Complexity of the translation
Lemmas ([Aubert, 2010])

• For $n, m \in \mathbb{N}$, a gate of fan-in $n$ and fan-out $m$ labeled by a function of $\mathcal{B}_1$ is translated with a piece of size $9(n + m)$. 
Lemmas ([Aubert, 2010])

- For $n, m \in \mathbb{N}$, a gate of fan-in $n$ and fan-out $m$ labeled by a function of $\mathcal{B}_1$ is translated with a piece of size $9(n + m)$.
- The depth of a piece and the arity of the function it represents are independent.
Lemmas ([Aubert, 2010])

- For $n, m \in \mathbb{N}$, a gate of fan-in $n$ and fan-out $m$ labeled by a function of $\mathcal{B}_1$ is translated with a piece of size $9(n + m)$.
- The depth of a piece and the arity of the function it represents are independent.

Theorem ([Aubert, 2010])

For all $i \in \mathbb{N}$, if $C = (C_n) \in AC^i$ (resp. $NC^i$), then its translation $P = (P_n)$ is such that for all $j \in \mathbb{N}$

- the size of $P_j$ is linear (resp. quadratic) in the size of $C_j$,
Size and depth of Proof Circuits

Lemmas ([Aubert, 2010])

- For $n, m \in \mathbb{N}$, a gate of fan-in $n$ and fan-out $m$ labeled by a function of $\mathcal{B}_1$ is translated with a piece of size $9(n + m)$.
- The depth of a piece and the arity of the function it represents are independent.

Theorem ([Aubert, 2010])

For all $i \in \mathbb{N}$, if $C = (C_n) \in AC^i$ (resp. $NC^i$), then its translation $P = (P_n)$ is such that for all $j \in \mathbb{N}$
- the size of $P_j$ is linear (resp. quadratic) in the size of $C_j$,
- the depth of $P_j$ is linear in the depth of $C_j$. 
Corollaries

- Uniformity is preserved
- Translation is lighter and takes NC into account
- For all $i \in \mathbb{N}$, $AC^i \subseteq PCC^i \subseteq AC^i(UstConn_2)$

Future work:
- $UstConn_2 \in \mathbf{L}$, is there a lighter function?
- Link between Proof Nets and Alternating Turing Machines?
Corollaries

- Uniformity is preserved
- *Translation is lighter and takes NC into account*
- For all $i \in \mathbb{N}$, $AC^i \subseteq PCC^i \subseteq AC^i(UstConn_2)$

Future work:

- $UstConn_2 \in L$, is there a lighter function?
- Link between Proof Nets and Alternating Turing Machines?
Aubert, C. (2010).
Réseaux de preuves booléens sous-logarithmiques.
Mémoire de M2 L.M.F.I., Paris VII, L.I.P.N.

Uniform circuits, & Boolean proof nets.

Proof Nets and Boolean Circuits.

Thanks!