1 March 2, 1995

Shiyu Zhou

Let $n$ be a positive integer and $t(\cdot)$ an integer function. A straightline program $P$ of type $(n, t(n))$ is a sequence $[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n, g_1, \ldots, g_{t(n)}]$ such that each $g_i$ is of the form $h_1 \circ h_2$ where $h_1, h_2$ are elements prior to $g_i$ in the sequence and $\circ \in \{\wedge, \vee\}$. We may think of each of the elements in $P$ as a Boolean function with domain $\{0,1\}^n$ that operates in the natural way: for $x \in \{0,1\}^n$, $x_i(x)$ (resp. $\bar{x}_i(x)$) evaluates to 1 only if the $i$-th bit of $x$ is 1 (resp. 0), and $g_i(x)$, where $g_i = h_1 \circ h_2$, evaluates to 1 only if $h_1(x) \circ h_2(x) = 1$. Let $U_f = \{x \in \{0,1\}^n \mid f(x) = 0\}$. We will often identify each $h \in P$ with the set of elements $x \in U_f$ such that $h(x)$ evaluates to 1. That is to say, for any $x \in U_f$, $h(x) = 1$ if and only if $x \in h$. Consequently, we identify the Boolean operators $\wedge$ and $\vee$ with the set operators $\cap$ and $\cup$ respectively.

The function computed by such a straightline program is defined to be $g_{t(n)}$. The following fact is easily seen

**Proposition 1.1** A Boolean function $f$ on $n$ input variables is computable by a circuit (with all gates of fan-in 2) of size $t(n)$ if and only if $f$ has a straightline program of size $t(n)$.

Suppose straightline program $P$ computes the Boolean function $f$. A usual convenient way to view such a straightline program $P$, as what we will do in the following, is to think of it as a matrix with columns indexed by the sequence $P$, rows indexed by the set $U_f$ and with $h(x)$ being the value of the entry corresponding to row $x \in U_f$ and column $h \in P$. Thus each $h \in P$ can be thought of as the set of row indices corresponding to the 1's in column $h$ in the matrix.

**Remark:** We may index the rows of the matrix with a subset of $U_f$ if necessary, but we will use $U_f$ in the following discussion.

Let $f$ be a Boolean function on $n$ input variables. We define $\Omega = \{\omega : 2^U_f \rightarrow \{0,1\}\}$. Let $P$ be a straightline program of type $(n, t)$ for some $t$ that computes $f$ and suppose $\circ \in \{\wedge, \vee\}$. We say that $\omega \in \Omega$ is consistent with $P$ if for all $g_i = h_1 \circ h_2$, we have that $\omega(h_1 \circ h_2) = \omega(h_1) \circ \omega(h_2)$; we say that $\omega$ defines $v \in \{0,1\}^n$ if for $1 \leq i \leq n$, $\omega(x_i) = v_i$ and $\omega(\bar{x}_i) = \bar{v}_i$; and we say that $\omega$ is rejecting if $\omega(\emptyset) = 0$, i.e. $\omega(g_{t}) = 0$. Let $\Omega_f = \{\omega \in \Omega : \omega$ is rejecting and defines some $v \notin U_f\}$.

**Lemma 1.1** If $P$ rejects precisely $U_f$, then there is no $\omega \in \Omega$ that is consistent with $P$, rejecting and defines some $v \notin U_f$.

**Proof:** Suppose that $\omega \in \Omega$ defines a $v = v_1v_2 \ldots v_n \notin U_f$ and is consistent with $P$. We claim that for any $h \in P$, $\omega(h) = h(v)$. This will suffice the proof since then we would have $\omega(g_t) = g_t(v) = 1$ which implies that $\omega$ is not rejecting.
To see the claim, we first observe that by the assumption that \( \omega \) defines \( v \), \( \omega(x_i) = v_i = x_i(v) \) and \( \omega(\overline{x}_i) = \overline{v}_i = \overline{x}_i(v) \) for all \( i \). Inductively, if \( g_i = h_1 \circ h_2 \in P \), then
\[
\omega(g_i) = \omega(h_1 \circ h_2) \\
= \omega(h_1) \circ \omega(h_2) \\
= h_1(v) \circ h_2(v) \\
= g_i(v),
\]
where the second equality follows from the fact that \( \omega \) is consistent with \( P \) and the third equality is by induction. \( \square \)

**Corollary 1.1** If \( P \) rejects precisely \( U_f \), then no \( \omega \in \Omega_f \) can be consistent with \( P \).

Suppose \( g, h \subseteq U_f \) and \( \circ \in \{\land, \lor\} \). A triple \((g, h, \circ)\) covers \( \omega \in \Omega_f \) if and only if \( \omega(g) \circ \omega(h) \neq \omega(g \circ h) \).

Define \( \rho_\Omega(f) \) to be the minimum number of triples needed to cover \( \Omega_f \) and \( s(f) \) to be the size of the smallest straightline program that computes \( f \). Then we have

**Theorem 1.1** For any Boolean function \( f \), \( s(f) \geq \rho_\Omega(f) \).

**Proof:** Let \( P \) be a straightline program computing \( f \). Then by Corollary 1.1, no function in \( \Omega_f \) is consistent with \( P \). Thus \( P \) gives an obvious cover of \( \Omega_f \). \( \square \)

## 2 March 20, 1995

For a set \( U \), a function \( \omega : 2^U \rightarrow \{0, 1\} \) is said to be monotone if \( g \subseteq h \subseteq U \) implies that \( \omega(g) \leq \omega(h) \). For the following discussion, we fix \( \Omega \) to be the set of all monotone functions from \( 2^{U_f} \) to \( \{0, 1\} \), which is called the \textsc{Filters}. Define \( \hat{\rho}_\Omega(f) \) to be the minimum number of triples of the form \((g, h, \land)\) needed to cover \( \Omega_f \), where \( g, h \subseteq U_f \). For notational convenience, we abbreviate a triple \((g, h, \land)\) as \((g, h)\).

**Theorem 2.1** For any Boolean function \( f \), \( s(f) \geq \hat{\rho}_\Omega(f) \).

**Proof:** Let \( P \) be any straightline program of type \((n, t)\) that computes \( f \). It suffices to show that any \( \omega \in \Omega_f \) is covered by some \( \land \) gate in \( P \).

Let \( \omega \in \Omega_f \) and let \( v \notin U_f \) be defined by \( \omega \). Suppose otherwise (for the sake of contradiction) that \( \omega \) is not covered by any \( \land \) gate in \( P \), then we will show that for any \( h \in P \), \( \omega(h) \geq h(v) \). This would imply in particular that \( \omega(g_i) \geq g_i(v) = f(v) = 1 \) which is a contradiction since \( \omega \) is rejecting by definition.

In the case where \( h = x_i \) or \( \overline{x}_i \), we have \( \omega(h) = h(v) \) since \( \omega \) defines \( v \). If \( h \in P \) is an \( \land \) gate, say \( h = h_1 \land h_2 \), then since \( \omega \) is not covered by \( h \) by assumption, we have \( \omega(h) = \omega(h_1) \land \omega(h_2) \) which is inductively at least \( h_1(v) \land h_2(v) = h(v) \). Finally, suppose \( h \) is an \( \lor \) gate, say \( h = h_1 \lor h_2 \). If \( \omega \) is not covered by \( h \), then by the same inductive argument as above we are done. Suppose \( \omega \) is covered by \( h \), i.e. \( \omega(h) = \omega(h_1 \lor h_2) \neq \omega(h_1) \lor \omega(h_2) \).
Then since \( \omega \) is monotone, it must be the case that \( \omega(h_1) = \omega(h_2) = 0 \) and \( \omega(h) = 1 \), which implies \( \omega(h) \geq h(v) \).

\[ \square \]

Next we show that \( \hat{\rho}_0(f) \) is in fact a relatively tight bound of the minimum size of a straightline program (and thus a circuit) that computes \( f \).

**Theorem 2.2** For any Boolean function \( f \), \( s(f) \leq O((\hat{\rho}_0(f))^2) \).

**Proof:** Given an (unordered) collection \( C = \{(g_1, h_1), \ldots, (g_l, h_l)\} \) (we assume w.l.o.g. that \( l \geq n \)) of triples that covers \( \Omega_f \), we will build a circuit (and thus a straightline program) of size \( O(l^2) \) that computes \( f \).

First we observe the following fact.

**Observation 2.1** \( f(z) = 0 \) if and only if \( \exists \omega \in \Omega \) such that \( \omega \) defines \( z \), is consistent with \( C \), and is rejecting.

**Proof:** We first show the only if direction. Let \( f(z) = 0 \). Define \( \omega_z \) to be such that \( \omega_z(g) = g(z) \) for all \( g \subseteq U_f \). Then \( \omega_z \) is monotone since \( g(z) = 1 \) iff \( z \in g \). It is easy to check that \( \omega_z \) defines \( z \) since \( z \in U_f \). Also for any \( g, h \in U_f \), \( \omega_z(g \wedge h) = g(z) \wedge h(z) = \omega_z(g) \wedge \omega_z(h) \). This implies that \( \omega_z \) is consistent with \( C \). Since \( \omega_z(\phi) = \phi(z) = 0 \) (\( z \notin \phi \)), \( \omega_z \) is rejecting.

To see the if direction, suppose otherwise that \( f(z) = 1 \). Then if \( \omega \) is monotone, rejecting and defines \( z \) which is not in \( U_f \), then by the definition of a cover of \( \Omega_f \), \( \omega \) cannot be consistent with \( C \).

Thus, our goal will be to build a circuit based on \( C \) such that, given as input \( z \), it checks whether there exists an \( \omega \) satisfying the properties in the above observation and outputs 1 if and only if such an \( \omega \) does not exist.

Let \( S = \{\phi, x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n, g_1, h_1, g_1 \wedge h_1, \ldots, g_l, h_l, g_l \wedge h_l\} \). Since \( C \) is given, for each \( S, T \in S \), we know the information whether \( S \subseteq T \) or not. The circuit we build executes the following routine:

On input \( z = z_1z_2\ldots z_n \)

For \( i = 1, 2, \ldots, n \) Do
  If \( z_i = 1 \)
    Then \( \omega(x_i) \leftarrow 1 \);
  Else \( \omega(\bar{x}_i) \leftarrow 1 \);

Loop
  If \( \exists S, T \in S \) such that \( T \subseteq S \), \( \omega(T) = 1 \) and \( \omega(S) \) is not set
    Then \( \omega(S) \leftarrow 1 \);
  If \( \exists S \in S \) and \( i \) such that \( \omega(g_i) = \omega(h_i) = 1 \), \( S = g_i \wedge h_i \), and \( \omega(S) \) is not set
    Then \( \omega(S) \leftarrow 1 \);

Until no progress

The circuit outputs 1 if and only if \( \omega(\phi) = 1 \).
Since $C$ is given and thus $S$ is known, it is then not difficult to see that a circuit of size $O(l^2)$ that computes the routine can be derived. By the above observation, to prove the theorem, now it suffices to show that for any $S \in S$, our circuit sets $\omega(S) = 1$ if and only if for every monotone $\omega'$ such that $\omega'$ defines $z$ and is consistent with $C$, it is the case that $\omega'(S) = 1$.

To see this, we define $\omega_0$ as follows: for any $g \subseteq U_f$,

$$\omega_0(g) = \begin{cases} 1 & \text{if } \exists h \in S \text{ such that } h \subseteq g \text{ and } \omega(h) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

It is then easy to check that the following facts hold:

**Fact 1**: For any $S \in S$, $\omega_0(S) = 1$ iff $\omega(S) = 1$.

**Fact 2**: $\omega_0$ is monotone, consistent with $C$ and defines $z$.

From these facts, by induction, it is easy to show that for every monotone $\omega'$, if $\omega'$ defines $z$ and is consistent with $C$, then $\omega'(S) \geq \omega_0(S)$ for all $S \in S$. This completes the proof. ☐

Applying a similar argument, we can show the following

**Theorem 2.3** For any Boolean function $f$, the minimum size of a nondeterministic circuit that computes $f$ is $(\hat{\rho}_{\text{ULTRAFILTERS}}(f))^{O(1)}$.

where $\text{ULTRAFILTERS} = \{\omega \in \text{FILTERS} : \forall S \in U_f, \omega(S) \neq \omega(\bar{S})\}$.

It follows that to show $P \neq NP$, it suffices to show that $\hat{\rho}_{\text{ULTRAFILTERS}}(f) < \hat{\rho}_{\text{FILTERS}}(f)$ for some $f \in \{f_n : n \geq 1\} \in NP$. 