Limited nondeterminism and verification

Jeffrey Finkelstein
Computer Science Department, Boston University
October 29, 2015

1 Completeness respecting verification

Completeness under many-one reductions in nondeterministic complexity classes fails in differentiating problems with different verification complexity. Consider the satisfiability problem for Boolean circuits, for Boolean formulas, and for 3-CNF Boolean formulas. Each problem can be solved by nondeterministically choosing an input $x$ of $n$ bits, where $n$ is the number of variables in the circuit or formula, then verifying whether the input $x$ satisfies the circuit or formula. However, each verification process has vastly different computational complexity: evaluating a circuit is a $P$-complete problem, evaluating a Boolean formula is an $NC^1$-complete problem, and evaluating a 3-CNF Boolean formula is an $AC^0$ problem. Contrast this with the fact that each of the original decision problems is well-known to be $NP$-complete (and even complete for $NP$ under logarithmic space many-one reductions).

1.1 Reductions respecting verification complexity

Let $L_1$ and $L_2$ be decision problems with corresponding witness relations $R_1$ and $R_2$. There is a many-one reduction from $L_1$ to $L_2$, denoted $L_1 \leq_m L_2$, if there is a computable function $f$ such that $x \in L_1$ if and only if $f(x) \in L_2$. There is a weak Levin reduction from $L_1$ to $L_2$, denoted $L_1 \leq_w L_2$, if $L_1 \leq_m L_2$ and $R_1 \leq_m R_2$. There is a Levin reduction from $L_1$ to $L_2$, denoted $L_1 \leq_L L_2$, if there are computable functions $f$, $g$, and $h$ such that

1. if $(x, w) \in R_1$, then $(f(x), g(x, w)) \in R_2$,
2. if $(f(x), w') \in R_2$, then $(x, h(x, w')) \in R_1$. 

Copyright 2015 Jeffrey Finkelstein (jeffreyf@bu.edu).
This document is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License, which is available at https://creativecommons.org/licenses/by-sa/4.0/
The LaTeX markup that generated this document can be downloaded from its website at https://github.com/jfinkels/limndver The markup is distributed under the same license.
In a Levin reduction, the goal is to recover a witness for an instance of $L_1$ given a witness to an instance of $L_2$. In the weak Levin reduction, there is no requirement that the original witness is recoverable.

**Lemma 1.1.** $L_1 \leq_L L_2$ implies $L_1 \leq_m L_2$.

**Proof.** The function $f$ is the necessary many-one reduction. First suppose $x \in L_1$.

\[
x \in L_1 \implies \exists w: (x, w) \in R_1
\]

\[
\implies \exists w: (f(x), g(x, w)) \in R_2
\]

\[
\implies \exists w': (f(x), w') \in R_2
\]

\[
\implies f(x) \in R_2
\]

Next suppose $f(x) \in L_2$.

\[
f(x) \in L_2 \implies \exists w': (f(x), w') \in R_2
\]

\[
\implies \exists w': (x, h(x, w')) \in R_1
\]

\[
\implies \exists w: (x, w) \in R_1
\]

\[
\implies x \in L_1
\]

Thus $x \in L_1$ if and only if $f(x) \in L_2$. \(\square\)

The weak Levin reduction is so named because it is a compromise between the Levin reduction and the many-one reduction.

**Proposition 1.2.** $L_1 \leq_L L_2$ implies $L_1 \leq_w L_2$ implies $L_1 \leq_m L_2$.

**Proof.** Suppose $L_1 \leq_L L_2$. By the previous lemma, $L_1 \leq_m L_2$. The function $(x, w) \mapsto (f(x), g(x, w))$ proves that $R_1 \leq_m R_2$. \(\square\)

Since the weak Levin reduction is a more fine-grained comparison than the many-one reduction, a weak Levin reduction from a harder problem to an easier problem has more severe implications than a many-one reduction between the same problems. There is a $(C_1, C_2)$ weak Levin reduction from $L_1$ to $L_2$, denoted $L_1 \leq_{C_1, C_2} L_2$, if $L_1 \leq_{C_1} L_2$ and $R_1 \leq_{C_2} R_2$. A $(C, C)$ weak Levin reduction is simply written as a $C$ weak Levin reduction and denoted $\leq_{C}^{C}$.

**Proposition 1.3.** For any integer $d$ greater than 1,

1. $\beta_2 \text{CSAT} \leq_m^{L} \beta_2 \text{NC}^d \text{CSAT}$ if and only if $\beta_2 \text{NC}^d = \beta_2 \text{P}$.

2. $\beta_2 \text{CSAT} \leq_m^{L} \beta_2 \text{NC}^d \text{CSAT}$ if and only if $\text{NC}^d = \text{P}$.

Here are two brief comments about this proposition. First, this proposition extends to other levels of the beta hierarchy as well. Second, the restriction that the reductions are computable in logarithmic space instead of polynomial time is not onerous: under $\leq_m^{L}$ reductions, $\beta_2 \text{CSAT}$ is complete for $\beta_2 \text{P}$ and $\beta_2 \text{NC}^d \text{CSAT}$ is complete for $\beta_2 \text{NC}^d$ (TODO cite Sam H. here).
1.2 Complexity classes specifying verification complexity

The complexity class $\text{GC}(f(n), C)$ is the class of all languages $L$ for which there is a witness relation $R$ such that $x \in L$ if and only if there is a $w$ of length at most $O(f(n))$ such that $(x, w) \in R$ \[1\]. This notation generalizes several other classes:

$$\text{NP} = \bigcup_{k \in \mathbb{N}} \text{GC}(n^k, \text{P})$$
$$\beta \text{P} = \bigcup_{k \in \mathbb{N}} \text{GC} (\log^k n, \text{P})$$
$$\beta \text{NC} = \bigcup_{k \in \mathbb{N}} \text{GC} (\log^k n, \text{NC})$$

Inclusions of complexity classes are preserved when adding nondeterminism.

**Lemma 1.4.** For complexity classes $\mathcal{C}_1$ and $\mathcal{C}_2$, if $\mathcal{C}_1 \subseteq \mathcal{C}_2$, then $\text{GC}(f(n), \mathcal{C}_1) \subseteq \text{GC}(f(n), \mathcal{C}_2)$.

The set of NP-complete problems spans several of these “guess-and-check” classes.

**Lemma 1.5.**
1. 3-SAT is NP-complete and in $\text{GC}(\text{poly}, \text{AC}^0)$.
2. FSAT is NP-complete and in $\text{GC}(\text{poly}, \text{NC}^1)$.
3. CSAT is NP-complete and in $\text{GC}(\text{poly}, \text{P})$.

Unfortunately, even specifying complexity classes in terms of verification complexity is not satisfactory, since $\text{NP} = \text{GC}(\text{poly}, \text{P}) = \text{GC}(\text{poly}, \text{NC}^1)$ \[2\] [Theorem 2.2]. This puts FSAT and CSAT in the same class still. We need a more fine-grained approach to comparing the complexity of problems based on their verification complexity.

1.3 Completeness under weak Levin reductions

We want to separate the satisfiability problem for circuits, formulas, and 3-CNF formulas.

**Theorem 1.6.**
1. CSAT is complete for $\text{GC}(\text{poly}, \text{P})$ under $\leq_w^P$ reductions.
2. FSAT is complete for $\text{GC}(\text{poly}, \text{NC}^1)$ under $\leq_w^{\text{P,AC}^0}$ reductions.

The following corollary essentially shows that NP is not closed under $\leq_w^L$ reductions unless NC = P.

**Corollary 1.7.**
$3\text{-SAT} \equiv_m^L \text{FSAT} \equiv_m^L \text{CSAT}$ and $3\text{-SAT} \leq_w^{\text{AC}^0} \text{FSAT} \leq_w^{\text{AC}^0} \text{CSAT}$

but
1. $\text{FSAT} \not\leq_p \text{AC}^0$ $3$-$\text{SAT}$ unless $\text{AC}^0 = \text{NC}^1$,

2. $\text{CSAT} \not\leq_p \text{L}$ $3$-$\text{SAT}$ unless $\text{NC} = \text{P}$,

3. $\text{CSAT} \not\leq_p \text{FSAT}$ unless $\text{NC} = \text{P}$.

In general, how can we construct a problem that is complete for a guess-and-check class under weak Levin reductions? There are two strategies: start from a problem complete for $\text{GC}(f(n), C)$ under many-one reductions and ensure that its witness relation is complete for $C$ under many-one reductions, or start from a $C$-complete witness relation and consider its projection onto its first component. A third strategy is to choose a complete problem for $C$ and define our complexity classes as the closure of that problem under weak Levin reductions; this is what happens in the next subsection.

### 1.4 Classes respecting verification complexity

We wish to determine complexity classes that are closed under weak Levin reductions. Let $\text{cl}(L, \leq)$ denote the class of all languages $\leq$-reducible to $L$, and let $\text{cl}(C, \leq) = \bigcup_{L \in C} \text{cl}(L, \leq)$. (If $L$ is complete for $C$ under $\leq$ reductions, then $\text{cl}(C, \leq) = \text{cl}(L, \leq)$.) For example,

$$
\begin{align*}
\text{NP} &= \text{cl}(\text{CSAT}, \leq^L_m) \\
\text{NNC}^d &= \text{cl}(\text{NC}^d \text{CSAT}, \leq^L_m) \\
\text{NNC}^1 &= \text{cl}(\text{FSAT}, \leq^\text{AC}^0_m) \\
\beta_2 \text{P} &= \text{cl}(\beta_2 \text{CSAT}, \leq^L_m) \\
\beta_2 \text{NC}^d &= \text{cl}(\beta_2 \text{NC}^d \text{CSAT}, \leq^L_m) \\
\beta_2 \text{NC}^1 &= \text{cl}(\beta_2 \text{FSAT}, \leq^\text{AC}^0_m) \\
\text{P} &= \text{cl}(\text{CVP}, \leq^L_m) \\
\text{NC}^d &= \text{cl}(\text{NC}^d \text{CVP}, \leq^L_m) \\
\text{NC}^1 &= \text{cl}(\text{FVP}, \leq^\text{AC}^0_m).
\end{align*}
$$

For each of the nondeterministic complexity classes $C$, we define $C_w$ to be the closure under the appropriate resource-bounded weak Levin reduction instead of many-one reduction. (The notion does not make sense for the deterministic complexity classes, since it requires that a language have a witness relation.)

The closure under the weak Levin reduction is a subset of the closure under the many-one reduction, by Proposition 1.2. Are the closures under these two types of reductions equal? The following theorem provides evidence that they are not.

**Theorem 1.8.** For each positive integer $d$ greater than one, $\text{NNC}^1 = \text{NNC}^d = \text{NP}$, but
1. if \( \text{NNC}_w^1 = \text{NNC}_w^d \), then \( \text{NC}_1^1 = \text{NC}_d^d \).
2. if \( \text{NNC}_w^d = \text{NP}_w \), then \( \text{NC}_d^d = \text{P} \).
3. if \( \text{NNC}_w^1 = \text{NP}_w \), then \( \text{NC}_1^1 = \text{P} \).

**Proof.** \( \text{NNC}_1^1 = \text{NP} \) by [2, Theorem 2.2]. If \( \text{NNC}_w^1 = \text{NNC}_w^d \), then \( \text{NC}_d^d \leq \text{AC}^0 \) \( \text{FVP} \), so \( \text{NC}_d^d \) is in \( \text{NC}_1^1 \). If \( \text{NNC}_w^d = \text{NP}_w \), then \( \text{CVP} \leq \text{AC}^0 \) \( \text{FVP} \), so \( \text{CVP} \) is in \( \text{NC}_1^1 \). \( \Box \)

We wish to apply this strategy more generally. This is interesting because complexity classes traditionally don’t have this “downward collapse” property. (This generalizes not only the previous theorem but also Proposition 1.3.)

**Theorem 1.9.** Suppose
- \( \mathcal{C} \) and \( \mathcal{D} \) are complexity classes,
- \( \mathcal{F} \) is a class of functions,
- \( \leq_{\mathcal{F}} \) reductions are transitive,
- \( f \) is a function,
- \( \mathcal{C} \subseteq \mathcal{D} \),
- \( \mathcal{C} \) is closed under \( \leq_{\mathcal{F}} \) reductions.
- there are languages \( K_\mathcal{C} \) and \( K_\mathcal{D} \) that are \( \leq_{\mathcal{F}} \)-complete for \( \text{GC}(f(n), \mathcal{C}) \) and \( \text{GC}(f(n), \mathcal{D}) \), respectively,
- the witness relations \( R_\mathcal{C} \) and \( R_\mathcal{D} \) for \( K_\mathcal{C} \) and \( K_\mathcal{D} \), respectively, are \( \leq_{\mathcal{F}} \)-complete for \( \mathcal{C} \) and \( \mathcal{D} \), respectively.

If \( \text{GC}(f(n), \mathcal{C})_w = \text{GC}(f(n), \mathcal{D})_w \), then \( \mathcal{C} = \mathcal{D} \).

**Proof.** Since \( \mathcal{C} \) is closed under \( \leq_{\mathcal{F}} \) reductions and such reductions are transitive, it suffices to show \( R_\mathcal{D} \leq_{\mathcal{F}} R_\mathcal{C} \). By definition,

\[
\text{GC}(f(n), \mathcal{C})_w = \text{cl}(\text{GC}(f(n), \mathcal{C}), \leq_{\mathcal{F}}) = \text{cl}(K_\mathcal{C}, \leq_{\mathcal{F}})
\]

and similarly \( \text{GC}(f(n), \mathcal{D})_w = \text{cl}(K_\mathcal{D}, \leq_{\mathcal{F}}) \). By hypothesis, \( \text{GC}(f(n), \mathcal{C})_w = \text{GC}(f(n), \mathcal{D})_w \), so \( \text{cl}(K_\mathcal{C}, \leq_{\mathcal{F}}) = \text{cl}(K_\mathcal{D}, \leq_{\mathcal{F}}) \). Thus \( K_\mathcal{D} \leq_{\mathcal{F}} K_\mathcal{C} \) and hence \( R_\mathcal{D} \leq_{\mathcal{F}} R_\mathcal{C} \). \( \Box \)

How do these complexity classes compare with our traditional classes?

TODO Problem: see the next theorem; this applies to \( \text{NNC}_w \) also, which we already proved not equal to \( \text{NNC} \).

**Theorem 1.10.** \( \text{NP}_w = \text{NP} \).
Proof. The inclusion \( \text{NP}_w \subseteq \text{NP} \) follows immediately from the definitions. Let \( L \) be a language in \( \text{NP} \) with witness relation \( R \) in \( \text{P} \). Since \( \text{CSAT} \) is \( \text{NP} \)-complete, \( L \leq_{\text{w}} \text{CSAT} \), and since \( \text{CVP} \) is \( \text{P} \)-complete, \( R \leq_{\text{w}} \text{CVP} \). Thus \( L \leq_{\text{w}} \text{CSAT} \), hence \( L \in \text{NP}_w \) by the definition of the complexity class. \(\)

**Theorem 1.11.** For each positive integer \( d \) greater than one, if \( \text{NNC}^d_w = \text{NNC}^d \), then \( \text{NC}^d = \text{P} \).

Proof. Since \( \text{NP} = \text{NNC}^d \) by [2, Theorem 2.2] and \( \text{NNC}^d = \text{NNC}^d_w \) by hypothesis, we have \( \text{NP} = \text{NNC}^d_w \). Thus there is a logarithmic space weak Levin reduction from \( \text{CSAT} \) to \( \text{NC}^d \text{SAT} \). This implies \( \text{NC}^d = \text{P} \). \(\)

## 2 Limited nondeterminism

Now consider the tournament dominating set problem: given a tournament, decide whether it has a dominating set of size at most \( k \). Each tournament has a dominating set of size at most \( \log n \), so the parameter \( k \) must be bounded above by \( \log n \). Thus the problem can be solved by the following algorithm: given a tournament with vertex set \( V \) and edge set \( E \), nondeterministically choose a subset \( S \) of \( V \) of cardinality \( k \) and decide whether for each \( v \) in \( V \) there is a \( u \) in \( S \) such that \( (u, v) \in E \). This algorithm chooses \( O(\log^2 n) \) nondeterministic bits and verifies whether the chosen set is a dominating set by an \( \text{AC}^0 \) circuit computing

\[
\bigwedge_{v \in V} \bigvee_{u \in S} (u, v) \in E.
\]

Thus the problem is in \( \beta_2 \text{AC}^0 \). A similar algorithm applies when deciding whether a graph has a clique of size at most \( O(\log n) \): nondeterministically choose a candidate clique \( S \) then decide whether

\[
\bigwedge_{u \in S} \bigwedge_{v \in S} (u, v) \in E,
\]

again a \( \beta_2 \text{AC}^0 \) algorithm.

Since the tournament dominating set problem and the logarithmically-bounded clique problem are both \( \text{LOGSNP} \)-complete, and each is in \( \beta_2 \text{AC}^0 \), and \( \beta_2 \text{AC}^0 \subseteq \beta_2 \text{NC}^2 \), and the problems are actually complete for \( \text{LOGSNP} \) under \( L \) (\( \text{NC}^1 \)?) man-one reductions, and logarithmic space functions compose, and \( L \in \text{NC}^2 \), we have \( \text{LOGSNP} \subseteq \beta_2 \text{NC}^2 \). TODO Is \( \text{LOGNP} \) in there more generally? TODO Show how the consequences of \( P = \text{LOGSNP} \) compare with the consequences of \( P = \beta_2 \text{NC}^2 \). TODO Show a \( \text{NCPCP} \) algorithm for tournament dominating set. TODO Adapt \( \text{FO}[\text{polylog}] \) definition of \( \text{NC} \) to \( \beta_2 \text{NC} \).

TODO For \( \text{LOGNP} \) and \( \text{LOGSNP} \), require the formula \( \phi \) to be not just a quantifier first-order formula, but to be a \( \text{FO}(n^{O(1)}) \) formula in order to get \( P \)-hard verification?
3 Combining limited nondeterminism and reductions respecting verification

Here, we wish to show that for classes like LOGSNP and classes like APX, these reductions can be used when showing completeness to further classify the complexity of these problems, while still demonstrating the comparison due to many-one reductions.

References
