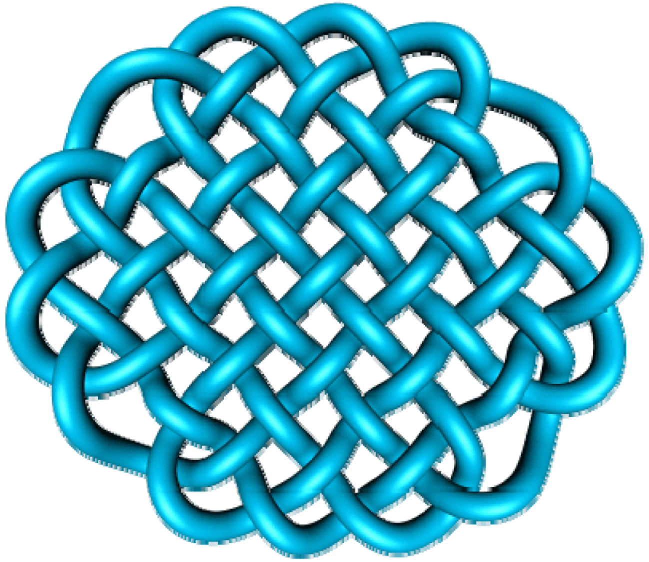
ESF
EIROPAS SOCIĀLAIS FONDS
EIROPAS SAVIENĪBA
IEGULDĪJUMS TAVĀ NĀKOTNĒ



Quantum computation with devices whose contents are never read

Abuzer Yakaryilmaz, Rūsiņš Freivalds, A.C. Cem Say, and Ruben Agadzanyan

European Social Fund project "Datorzinātnes pielietojumi un tās saiknes ar kvantu fiziku"
Nr.2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044

We demonstrate a case where the usage of "write-only memory" (WOM), a computational component that is used exclusively for being written to, and never being read, (which is little more than a joke in the classical setup,) improves the power of a quantum computer significantly.

For any standard machine model, say, M, we use the name M-WOM to denote M augmented with a WOM component. A TM-WOM has an additional write-only tape associated with a finite alphabet Υ. In each step of the computation, either a symbol from the alphabet, $\upsilon \in \Upsilon$, is printed on the current tape square, and the head moves one square to the right, or the empty string, ε, is "printed," and so the head remains at the same position. The computational power of the PTM-WOM is easily seen to be the same as that of the PTM; since the machine does not use the contents of the WOM in any way when it decides what to do in the next move, every write-only action can just as well be replaced with a write-nothing action.

However, this is not the case for the QTM-WOM, as will be shown in the next section. We will focus on quantum finite automata with WOM (QFA-WOM's), which are just QTM-WOM's which do not use their work tapes and move the input tape head to the right in every step. The configuration of a QFA-WOM is a pair (q, w), where q is an internal state, and w is the string written in the WOM.

Fig. 1. Transitions of the QFA-WOM

In the table below, the amplitude of the transition that takes place when the machine scans tape symbol σ while it is in state q, causing it to set the halting register to symbol ω, switch to state q' and add υ to the string in the WOM can be read in the row labeled by q, at the column labeled by (ω, q', υ). Empty boxes indicate zero amplitude. The columns corresponding to the "missing" elements of $\Omega \times Q \times (\Upsilon \cup \{\varepsilon\})$ contain all zeros, and have been omitted. In order for the machine to be well-formed, the rows of this table corresponding to the same tape symbol must be orthonormal to each other.

		n								a	r			
		q_1			q_2	q_3	q_4			q_1	q_1	q_2	q_3	q_4
		ε	0	1	ε	ε	ε	0	1	ε	ε	ε	ε	ε
	q_1	$\frac{1}{\sqrt{3}}$				$\frac{1}{\sqrt{3}}$					$\frac{1}{\sqrt{3}}$			
	q_2											1		
	q_3												1	
	q_4													1
0	q_1		1											
0	q_2				1									
0	q_3					1								
0	q_4							1						
1	q_1			1										
1	q_2				1									
1	q_3					1								
1	q_4								1					
2	q_1				1									
2	q_2										1			
2	q_3						1							
2	q_4											1		
	q_1											1		
	q_2									$3mu\ \frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$			
	q_3												1	
	q_4									$\frac{1}{\sqrt{2}}$	$\frac{-1}{\sqrt{2}}$			

If one changes the model in Theorem 1 so that the WOM is now an output tape, the machine becomes a quantum finite state transducer computing the function

$$f(x) = \begin{cases} w, & \text{if } x = w2w, \text{ where } w \in \{0,1\}^* \\ \text{undefined}, & \text{otherwise} \end{cases},$$

with bounded error.

Definition. Language A is probabilistically m-reducible to language B with probability $p > \frac{1}{2}$, denoted by $A_{prob(m),p}B$, if there is a PTM which outputs $y_1, \ldots, y_k$ with probabilities $p_1, \ldots, p_k$, respectively, for a given input x, satisfying the following conditions:
$\Sigma_{y_i \in B}\, p_i \geq p$ when $x \in A$, and
$\Sigma_{y_i \notin B}\, p_i \geq p$ when $x \notin A$.

Theorem. There exist recursively enumerable languages A and B such that

1. $A \leq_m B$,

2. $A \leq_{prob(m),\frac{2}{3}} B$.

$B \subset$	$0 \quad 1 \quad 2$	$3 \quad 4 \quad 5$	$6 \quad 7 \quad 8$	$\cdots$	$3x \quad 3x+1 \quad 3x+2$	$\cdots$
	$\nwarrow \uparrow \nearrow$	$\nwarrow \uparrow \nearrow$	$\nwarrow \uparrow \nearrow$	$\cdots$	$\nwarrow \quad \uparrow \quad \nearrow$	$\cdots$
$A \subset$	0	1	2	$\cdots$	x	$\cdots$

Let $\varphi_0, \varphi_1, \ldots$ be an enumeration of deterministic TM's with output tapes. In the following, φ_i will be named marker-i. φ_i has higher priority than φ_j if $i < j$.
The algorithm based on Friedberg and Muchnik's priority method effectively constructs the languages A and B.

Fig. 2.

FOR $n = 1, 2, \ldots$
    ## STAGE n
    MARK the first free number, say y, with marker-$(n-1)$, which becomes *active*
    MARK y at $line_A$ and $3y$, $3y+1$, and $3y+2$ at $line_B$ with "$-$"
    LOOP
        ## markers with higher priority will be simulated earlier
        SIMULATE each active marker (φ_i) for n steps with the associated number (x) as input
        IF $\varphi_i(x)$ returns a value, say t
            CALL UPDATE SIGNS(x,t)
            MAKE marker-i *inactive*
            FOR $j = i+1, \ldots, n-1$
                MOVE marker-j to the first free number on $line_A$
                MAKE marker-j *active*
            END
            GOTO NEXT STAGE
        END
    END
END

.

Fig. 3.

If $t \notin \{3x, 3x+1, 3x+2\}$, then we have two cases:

If t has no sign, then mark t and its relatives at $line_B$ and $\lfloor \frac{t}{3} \rfloor$ at $line_A$ with "$-$". Mark x at $line_A$ and at least two of $\{3x, 3x+1, 3x+2\}$ at $line_B$ with "$+$".

If t has a sign, say S: If S is "$+$", there is no need for marking since x is already marked with "$-$". If S is "$-$", then mark x at $line_A$ and at least two of $\{3x, 3x+1, 3x+2\}$ at $line_B$ with "$+$".

If $t \in \{3x, 3x+1, 3x+2\}$:

Mark t at $line_B$ with "$+$".

The main idea of the algorithm is that each marker can be moved only a finite number of times, and so any marker (φ_i) remains ultimately at a number (x) on $line_A$. Thus, it is easy to make sure that the signs of x and $\varphi_i(x) = t$ contradict in order to get

$$x \notin A \Leftrightarrow \varphi_i(x) = t \in B,$$

while employing the additional numbers in the triples on $line_B$ to ensure that $\mathcal{P}$ works correctly. Note that some markers may never halt, but this is not a problem, since such markers are not proper reductions by definition.

Definition Language A is probabilistic (respectively, quantum) Turing reducible with k queries to language B with probability $p > \frac{1}{2}$, denoted $A \leq_{prob(T\text{-}k),p} B$ (respectively, $A \leq_{quan(T\text{-}k),p} B$), if there exists a PTM (respectively, QTM), which is restricted to query the oracle for B at most k times, that recognizes A (that is, responds correctly to all questions of membership in A) with probability at least p.

Theorem. There exist recursively enumerable languages A and B such that

1. $A_{prob(T\text{-}1),\frac{2}{3}}B$,
2. $A_{quan(T\text{-}1),1}B$.

$B \subset$	0	1	2	3	4	5	$\cdots$	$2x$	$2x+1$	$\cdots$
	$\nwarrow$	$\nearrow$	$\nwarrow$	$\nearrow$	$\nwarrow$	$\nearrow$	$\cdots$	$\nwarrow$	$\nearrow$	$\cdots$
$A \subset$	0		1		2		$\cdots$	x		$\cdots$

We again put "$+$" and "$-$" signs on the numbers to indicate their membership status.

$B \subset$	$0 \quad 1$	$2 \quad 3$	$4 \quad 5$	$\cdots$	$2x \qquad 2x+1$	$\cdots$
	$\nwarrow \nearrow$	$\nwarrow \nearrow$	$\nwarrow \nearrow$	$\cdots$	$\nwarrow \qquad \nearrow$	$\cdots$
$A \subset$	0	1	2	$\cdots$	x	$\cdots$

We again put " $+$ " and " $-$ " signs on the numbers to indicate their membership status.

1. path_1 and path_2 respectively prepare $2x$ and $2x+1$ on the oracle tape for their single query.
2. If the answer of the oracle is negative, the amplitude of that path is multiplied with -1.
3. Both paths enter the twin configurations, and then make the following Hadamard transformation:

$$\mathsf{path}_1 \to \frac{1}{\sqrt{2}}Reject + \frac{1}{\sqrt{2}}Accept$$
$$\mathsf{path}_2 \to \frac{1}{\sqrt{2}}Reject - \frac{1}{\sqrt{2}}Accept.$$

Table 1.

$2x$	$2x+1$	x
$-$	$-$	$-$
$-$	$+$	$+$
$+$	$-$	$+$
$+$	$+$	$-$

```
FOR n = 1, 2, ...
    ## STAGE n
    MARK the first free number, say y, with marker-(n − 1), which becomes active
    MARK y at line_A and 2y and 2y + 1 at line_B with "−"
    LOOP
        ## The markers with higher priority are simulated earlier
        SIMULATE the first n levels of the probabilistic computation tree of each
        active marker (φ_i) with the associated number (x) as input
        ## The oracle for B is assumed to respond with "no" to queries about num-
            bers at line_B which are not signed yet
        LET T = {t_1, t_2, ..., t_m} be the set of numbers for which φ_i queries B's oracle
        in the various branches of its simulation
        FIND all subsets of T, 𝒯 = {T' | T' = {t'_1, t'_2, ..., t'_l}, l ≤ m}, such that all
        branches of φ_i(x) that query the oracle about the numbers in T' halt with the
        same decision, say D, and the total probability of those branches exceeds 1/3
        LET 𝒯' be the biggest subset of 𝒯 whose elements are associated with "no",
        and contain both 2x and 2x + 1
        IF 𝒯' = 𝒯 and 𝒯' ≠ ∅
            PUT a temporary sign "∗" on 2x at line_B
            RE-SIMULATE φ_i for the first n levels on this new line_B, and RE-FIND
            𝒯 based on this new simulation
            ## The oracle for B is assumed to respond with "yes" to queries about
                numbers with sign "∗"
            SET 𝒯' to ∅
        IF there is a T' ∈ 𝒯 \ 𝒯' (pick one arbitrarily if there exists more than one
        such set)
            CALL UPDATE SIGNS(x, D, T')
            MAKE marker-i inactive
            FOR j = i + 1, ..., n − 1
                MOVE marker-j to the first free number on line_A
                MAKE marker-j active
            END
            GOTO NEXT STAGE
        END
        REPLACE any "∗" with "−"
    END
END
```

1. Mark x at $line_A$ with the sign that contradicts D.
   (Note that x could not have the sign " + " before this step.)
2. Mark all $t^{'} \in T^{'}$ having no sign with " − ", and so mark $\lfloor \frac{t^{'}}{2} \rfloor$ at $line_A$ and the relative of $t^{'}$ at $line_B$ with " − ".
3. Update the signs of $2x$ and/or $2x+1$ at $line_B$ if needed. All possible cases for this update are shown below.
   In case 2, since $2x+1 \notin T^{'}$, it is safe to change the sign of $2x+1$.

			before step 3			after step 3		
case	condition	D	x	$2x$	$2x+1$	x	$2x$	$2x+1$
1		*yes*	−	−	−	−	−	−
2		*yes*	−	∗	−	−	+	+
3.a	$2x+1 \notin T^{'}$	*no*	+	−	−	+	−	+
3.b	$2x+1 \in T^{'}$	*no*	+	−	−	+	+	−
4		*no*	+	∗	−	+	+	−

.

Theorem. There exist sets A and B such that $A \not\leq_{tt} B$ but $A \leq_{tt}^{2/3} B$.

$K = \{x|\varphi_x(x) \text{ is defined}\}$.
$\widetilde{K} = \{x|\,(\exists y)\,[\varphi_x(x) = y$ and
truth-table condition y is satisfied by$K\}$

It is known that $\widetilde{K} \not\leq_{tt} K$

Theorem. If $A \leq_{tt}^{prob} B$ with probability $p > \frac{2}{3}$, then $A \leq_{tt} B$.

Theorem. If $A \leq_{tt}^{prob} B$ with probability $p > \frac{1}{2}$, then $A \leq_{T} B$.

Definition. We say that a set A is frequentially reducible to the set B with frequency (m, n) if there is a totally defined algorithm M which for arbitrary input of n pairwise distinct natural numbers $x_1, x_2, \ldots, x_n$ outputs an m-tuple of natural numbers $y_1, y_2, \ldots, y_n$ such that for at least m numbers $i \in \{1, 2, \ldots, n\}$ the equality $x_i \in A \iff y_i \in B$ holds.

Theorem. For arbitrary natural number n there exist recursively enumerable sets A and B such that $A \not\leq_{prob}^{m/n} B$ but $A \leq_{freq}^{m/n} B$

$B \subset$	$0 \quad 1 \quad 2$	$3 \quad 4 \quad 5$	$6 \quad 7 \quad 8$	$\cdots$	$3x \quad 3x+1 \quad 3x+2$	$\cdots$
	$\nwarrow \uparrow \nearrow$	$\nwarrow \uparrow \nearrow$	$\nwarrow \uparrow \nearrow$	$\cdots$	$\nwarrow \quad \uparrow \quad \nearrow$	$\cdots$
$A \subset$	0	1	2	$\cdots$	x	$\cdots$

Theorem. For arbitrary natural number n there exist recursively enumerable sets A and B such that $A \not\leq_{freq}^{m/n} B$ but $A \leq_{prob}^{m/n} B$

We showed that write-only memory devices can increase the computational power of quantum computers, by demonstrating a language, which is known to be unrecognizable by both classical and quantum computers with certain restrictions, to be recognizable by a quantum computer employing a WOM under the same restrictions. As a separate contribution, we proved that quantum reductions among computational problems are more powerful than probabilistic reductions, which are in turn superior to deterministic reductions.

Thank you