Separation of \textbf{PSPACE} and \textbf{EXP}

Reiner Czerwinski  
TU Berlin (Alumnus)  
reiner.czerwinski@posteo.de

March 11, 2024

Abstract

This article shows that \textbf{PSPACE} is not equal to \textbf{EXP}. A novel proof technique has been used to separate these two classes. This paper is about the question, whether an arbitrary Turing machine accepts an input, if the running time is limited, and the limit converges towards infinity. Thus, methods of the recursion theory can be applied to problems of computational complexity theory without violating the relativization barrier.

1 Introduction

To show that \textbf{PSPACE} \neq \textbf{EXP}, we take an \textbf{EXP}-complete problem $\mathcal{E}$ and prove that there is no p-reduction on a \textbf{PSPACE}-complete problem $\mathcal{P}$. Thereby, the following definition holds:

**Definition 1.**

\[
\begin{align*}
\mathcal{E} &:= \{(M, x, k) \mid \text{TM } M \text{ accepts } x \text{ within } k \text{ steps}\} \quad k \text{ binary coded} \\
\mathcal{P} &:= \{(M, x, 1^k) \mid \text{TM } M \text{ accepts } x \text{ within } k \text{ tape space}\}
\end{align*}
\]

The TMs are the one-tape Turing machines. TMs with an empty tape are used as an input in this paper. Due to the s-m-n theorem, this is sufficient.

**Definition 2.**

\[
\begin{align*}
\mathcal{E}_0 &:= \{(M, k) \mid \text{TM } M \text{ accepts } \epsilon \text{ within } k \text{ steps}\} \quad k \text{ binary coded} \\
\mathcal{P}_0 &:= \{(M, 1^k) \mid \text{TM } M \text{ accepts } \epsilon \text{ within } k \text{ tape space}\}
\end{align*}
\]

It is $\mathcal{E}_0$ \textbf{EXP}-complete and $\mathcal{P}_0$ \textbf{PSPACE}-complete.
Corollary 1. $\mathcal{E} \leq_{1}^{p} \mathcal{E}_{0}$ and $\mathcal{P} \leq_{1}^{p} \mathcal{P}_{0}$

Proof. Let $M$ be an arbitrary TM. According to the s-m-n theorem, a TM $M_x$ can be constructed which writes $x$ to an empty tape and then executes $M$. Then $(M, x, k) \in \mathcal{E} \iff (M_x, k + |x|) \in \mathcal{E}_0$ and $(M, x, k) \in \mathcal{P} \iff (M_x, k) \in \mathcal{P}_0$.

To show that there is no p-reduction from $\mathcal{E}_0$ to $\mathcal{P}_0$, in Section 5, $k \to \infty$ is applied to $\mathcal{E}_0$.

Lemma 1. The following statements are equivalent:

(i) $\mathcal{E}_0 \not\leq_{m} \mathcal{P}_0$

(ii) $\text{PSPACE}^\mathcal{P} \neq \text{EXP}^\mathcal{P}$

(iii) $\text{PSPACE} \neq \text{EXP}$

Proof. $\mathcal{E}$ is $\text{EXP}$-complete, so

$$\forall A \in \text{EXP} \Rightarrow A \leq_{m}^{p} \mathcal{E}$$

(i) $\iff$ (iii): if $\mathcal{E}_0 \leq_{m}^{p} \mathcal{P}_0$, then $A \in \text{EXP} \Rightarrow A \leq_{m}^{p} \mathcal{P}_0$ and $A \in \text{PSPACE}$, because $\mathcal{P}$ is $\text{PSPACE}$-complete

(iii) $\iff$ (i): immediately

(ii) $\iff$ (iii): immediately, because $\text{EXP} = \text{EXP}^\mathcal{P}$ and $\text{PSPACE} = \text{PSPACE}^\mathcal{P}$

Remark. This method does not violate the relativization barrier. If there is an oracle $A$ with $\text{EXP}^A = \text{PSPACE}^A$, then, of course, there is a reduction from $\mathcal{E}_0$ to $\mathcal{P}_0$ relative to $A$. We discuss this at the end of the paper in Section 6.

To this end, it is shown in Section 2 that it is undecidable whether the tape space of a TM grows polynomially or logarithmically within the runtime.

Since it is hard to show directly that $\mathcal{E}_0$ is not polynomially reducible to $\mathcal{P}_0$, the length increasing reduction has been introduced in Section 4. Theorem 2 shows in Section 5 that it is sufficient to demonstrate that there is no length increasing reduction from $\mathcal{E}_0$ to $\mathcal{P}_0$.  

2
2 Time vs. Space Complexity

If $M$ is a Turing machine (TM) then

$$s_M(x) := \text{tape space used by } M \text{ for calculation on input } x$$

$$t_M(x) := \text{time used by } M \text{ for calculation on input } x$$

It is well known [5] that

$$s_M \leq t_M \quad (1)$$

$$t_M \leq s_M \times 2^{O(s_M)} \quad (2)$$

**Lemma 2.** Let $M'$ be an arbitrary two-tape Turing machine that does not terminate. Then it is undecidable whether the used space grows polynomially with time or logarithmically.

**Proof.** A two-tape TM $M'$ is constructed from a one-tape TM $M$. The calculation of $M$ is carried out on the first tape. 1 is written on the second tape at each step and then the second tape is moved to the left. As soon as $M$ terminates, the word on the second tape is interpreted as a binary number. $M'$ increments this in an infinite loop. In this case, $M'$ is then a binary counter.

If $M$ does not terminate, the tape space of $M'$ grows linear with time.

If $M$ terminates, the tape space of $M'$ grows asymptotically, no faster than the logarithm of time.

Due to the halting problem, it is undecidable how fast the tape space of $M'$ grows.

**Corollary 2.** For an arbitrary TM $M$, it is undecidable, if a polynomial $p$ exists with it as follows:

$$\forall x \ p(s_M(x)) \geq t_m(x)$$

**Proof.** Let $U$ be a universal Turing machine that executes a two-tape TM $M$ with input $x$ and stops after $t$ steps. $U$ has the memory requirement of $s_U((M, x, t)) = s_M(x) + \log(t)$ and the runtime $t_U((M, x, t)) = t$. By Lemma 2, it is undecidable whether there exists a phenom $p$ with $\forall t : p(s_U((M, x, t))) > t_M((M, x, t))$.

From $U$, a one-tape TM $U'$ can be constructed with $L(U') = L(U)$ and $t_{U'} \leq t_U^2$ according to the Hennie-Stearns Theorem[3]. For the one-tape TM $U'$, it is also undecidable whether a polynomial exists with $p(s_{U'}) \geq t_{U'}$. 

\[\Box\]
Here is a corollary, that we will also need in later papers:

**Corollary 3.** Let $M$ be a TM with input $x$, both arbitrary but fixed. In order to check whether $x$ is accepted by $M$ within $t$ steps, one needs $\Omega(t)$ time in the worst case scenario.

**Proof.** Immediately, because $\mathcal{E}$ is EXP-complete. 

---

### 3 Tape Space of $\mathcal{E}_0$ and the Arithmetical Hierachy

Let $K_0 := \{M \mid \text{TM } M \text{ terminates on empty input} \}$ and 
$\overline{K_0} := \{M \mid \text{TM } M \not\in K_0 \}$. It is well known, that $K_0$ is $\Sigma_1$-complete and 
$\overline{K_0}$ is $\Pi_1$-complete. 

If $M$ is TM then 

$\tilde{S}_M(x) :=$ tape space used by $M$ for calculation on empty input within $t$ steps

Let 

$\bar{P} := \{M \mid \exists \text{ polynom } p \forall t \tilde{S}_M(t) < p(\log(t))\}$

**Lemma 3.** $K_0 \leq_1 \bar{P}$

**Proof.** The function $h$ transforms $M$ to a multi-taped TM with one more 
tape then $M$

```python
def h(M):
    while M not terminates:
        calculate one step on M
        write '1' to the last tape and move it one left
        when $M$ has terminated,
        apply a binary counter on the last tape
    if M does not terminate, the tape space of $h(M)$ grows linear with time, so

\[ M \in K_0 \iff h(M) \in \bar{P} \]
```

**Lemma 4.** $\overline{K_0} \leq_1 \bar{P}$

**Proof.** The function $h$ transforms $M$ to a multi-taped TM with one more 
tape then $M$
```python
def l(M):
    while M not terminates:
        calculate one step on M
        apply a binary counter beginning with 0
        until the length of the counter equals
            length of the used tape space of M
    when M has terminated:
        while true:
            write '1' on the last tape and move it one left

if M does not terminate, the tape space of l(M) grows with the logarithm
    of time, so

M \not\in K_0 \iff h(M) \in \tilde{(P)}
```

**Corollary 4.** $K_0 <_T \tilde{P}$

*Proof.* $K_0 \leq_T \tilde{P}$, because $K_0 \leq_1 \tilde{P}$. But $\overline{K_0} \leq_1 \tilde{P}$, so $K_0 \not\leq_T \tilde{P}$. 

**Theorem 1.** $\tilde{P} \in \Sigma_2$

*Proof.* $M \in \tilde{P} \Rightarrow \exists n, k \forall t : \bar{S}_M(t) < k \cdot \log^a(t)$

## 4 Length increasing functions

Reduction with honest polynomial and length increasing functions was in-
vestigated in the 80s. There were far-reaching results of Homer[4], Ambos-
Spies[1] and Watanabe[7].

An honest polynomial function is a polynomial function $f : \Sigma^* \rightarrow \Sigma^*$
    which has the property that there exists a polynomial $p$ with $p(|f(x)|) > |x|$. A function $f$ is length increasing, when $|f(x)| > |x|$ and $f$ and $f^{-1}$ are
    polynomially computable.

Let $A, B \subset \Sigma^*$, then $A \leq_1^i B$, when there is a one-one reduction with a
    length increasing function. $A \equiv_1^i B$, when $A \leq_1^i B$ and $B \leq_1^i A$.

**Theorem 2 (Watanabe).** If $A$ and $B$ are $\text{EXP}$-complete then $A \leq_1^i B$ and $B \leq_1^i A$ hold.

*Proof.* See [7, Corollary 3.4.]
Lemma 5. Let \( A, B \subseteq \{ (M, x) \mid \text{TM } M \text{ accepts } k \} \) and \( A \leq^l_1 B \). Then there is a length increasing function \( g \) with

\[
(M, k) \in A \iff (M_g, k_g) := g(M, k) \in B
\]

where \( M_g \) depends only on \( M \). I.e. \( M_g = g_1(M) \).

Proof. If \( A \leq^l_1 B \), then there is a length increasing function \( g : A \rightarrow B \) with \( (M_g, k_g) = g(M, k) = (g_1(M, k), g_2(M, k)) \). By the S-m-n theorem, \( g_1(M, k) = g_{1,S(M)}(k) \) and \( g_1(M, k) = g_{1,S(M)}(k) \). By definition, \( g \) is invertible in polynomial time, i.e. \((M, k) = (g_1^{-1}(M_g, k_g), g_2^{-1}(M_g, k_g))\).

To decide whether \((M_g, k_g) \in B\), \( M_g \) must be calculated on \( k_g \):

\[
M_g(k_g) = g_1(M, k)(k_g) = g_{1,S(M)}(k)(k_g) = g_{1,S'(M)}(k)(g_{2,S(M)}(k))
\]

\( g \) is invertible in polynomial time. Thus, \( g_{S(M)} = (g_{1,S(M)}, g_{2,S(M)}) \) is also invertible. A TM can be constructed that executes the following algorithm:

\[
def M_g(k_g):
\hat{k} := g_{2,S(M)}^{-1}(k_g)
\text{if defined } (\hat{k}):\return g_{1,S(M)}(\hat{k})(k_g)
\]

\( \Box \)

5. \( \text{EXP} \neq \text{PSPACE} \)

Let us consider \( \mathcal{E}_0 \) and \( \mathcal{P}_0 \) from Definition 2 in Section 1. If \( \text{EXP} = \text{PSPACE} \), then \( \mathcal{P}_0 \) would be \( \text{EXP} \)-complete. According to Theorem 2, it would be \( \mathcal{E}_0 \leq^l_1 \mathcal{P}_0 \).

Theorem 3. There is no \( \leq^l_1 \)-reduction from \( \mathcal{E}_0 \) to \( \mathcal{P}_0 \).

Proof. Assuming \( \mathcal{E}_0 \leq^l_1 \mathcal{P}_0 \), there is a length increasing function \( g \) with \((M, k) \in \mathcal{E}_0 \iff g(M, k) := (M_g, 1^{k_g}) \in \mathcal{P}_0 \). \( g \) is length increasing implies

\[
|g(M, k)| = |M_g| + |1^{k_g}| > |(M, k)| = |M| + \log(k)
\]

Now let \( M \) be fixed and \( k \rightarrow \infty \). Due to Lemma 2, \( g \) can be chosen in such a way that \( M_g \) does not depend on \( k \).

As results of Lemma 5, there is \( |(M, k)| = \log(k) + O(1) \) and \( |g(M, k)| = k_g \in \Omega(\log(k)) \).
To test whether \((M, k) \in \mathcal{E}_0\), a universal TM \(U\) is used, which computes \(M\) on input \(\epsilon\) and stops after, at most, \(k\) steps. If \(M\) does not stop within \(k\) steps, \(U\) rejects the given tuple. Otherwise, \(U\) accepts the tuple, if \(M\) accepts \(\epsilon\).

Let \(M\) be fixed and \(k \to \infty\). It is assumed that \(M\) does not terminate. This is, of course, undecidable due to the halting problem\([6]\). According to Corollary 3, a TM needs \(\Omega(k)\) run time to decide whether \(M\) terminates within \(k\) steps or not.

Let \(S(k) = s_U((M, k))\) be the tape space used by \(M\) in \(k\) steps.

If there is a polynomial \(p\) with \(\lim_{k \to \infty} p(s_U((M, k))/t_U((M, k)) = \lim_{k \to \infty} p(S(k))/k > 0\), then \(k_g \in O(\log^n(S(k)))\) must hold with \(n \in \mathbb{N}\), so that \(g\) is polynomial.

If \(\lim_{k \to \infty} s_U((M, k))/\log(t_U((M, k))) = \lim_{k \to \infty} S(k)/\log(k) = C\) with \(0 \leq C < \infty\), then \(k_g \in \Omega(S(k))\) must hold, so that \(g\) is length increasing.

\(g\) must be able to decide whether the space grows polynomially or logarithmically with time. However, according to Corollary 2 this is undecidable.

Thus, \(\text{EXP} \neq \text{PSPACE}\) and \(\text{EXP}\)-complete problems have exponential space requirements for computation.

6 Notes on the Relativization Barrier

To separate \(\text{PSPACE}\) and \(\text{EXP}\), the relativization barrier applies as for \(\text{P}\) versus \(\text{NP}\) \([2]\). Therefore, here we have mentioned the reason why the proof in this paper does not violate this barrier such as diagonalization.

**Proposition 1.** There is a computable function \(f\) that holds \((M, x, k) \in \mathcal{E} \iff f((M, x, k)) \in \mathcal{P}\) with exponential computation time.

**Proof.** Let \(s_M(x, k)\) be the space consumption of \(M\) with input \(x\) after \(k\) steps. For fixed \(M\) and \(x\), \(S(k) := s_M(x, k)\) is a monotonically increasing function of \(k\). Thus, \(f\) easily determines and uses the space consumption. \(f\) is defined as:

\[
\text{def } f((M, x, k)) : s := s_M(x, k) \\
\text{if } (M \text{ accepts } x \text{ within } s \text{ space}) \text{ and not } (M \text{ accepts } x \text{ within } k \text{ steps}) : \\
\text{return } (M, x, 1^{(s-k)}) \\
\text{else}
\]

7
\begin{verbatim}
return (M, x, 1^s)
\end{verbatim}

Since \( \mathcal{P} \) is PSPACE-complete and \( \mathcal{E} \) is EXP-complete, it follows that PSPACE^{\text{EXP}} = EXP^{\text{EXP}}. There is a many-one reduction with a computable but not polynomial function from \( \mathcal{E} \) to \( \mathcal{P} \). This means \( \mathcal{E} \leq_m \mathcal{P} \), but \( \mathcal{E} \not\leq_p \mathcal{P} \) and so, PSPACE^{P} \neq EXP^{P}.

Acknowledgments

I am thankful to Klaus Ambos-Spies, who explained to me the principle of honest polynomial reduction in his paper[1] as well as through personal communication. Thanks to the anonymous helper for the tips, especially taking length increasing instead of honest polynomial functions. I thank Martin Grohe for pointing out the bugs in theorem 3. Thanks to Lance Fortnow for pointing out not to use Rice’s theorem in Section 2 and many other bugs.

References

[1] Klaus Ambos-Spies. Honest polynomial time reducibilities and the P = \( \text{NP} \) problem. J. Comput. Syst. Sci., 39(3):250–281, 1989.
[2] Theodore Baker, John Gill, and Robert Solovay. Relativizations of the p=\( \text{NP} \) question. SIAM Journal on computing, 4(4):431–442, 1975.
[3] F. C. Hennie and Richard Edwin Stearns. Two-tape simulation of multitape turing machines. J. ACM, 13(4):533–546, 1966.
[4] Steven Homer. Minimal degrees for honest polynomial reducibilities. In 25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984, pages 300–307. IEEE Computer Society, 1984.
[5] John Hopcroft, Wolfgang Paul, and Leslie Valiant. On time versus space and related problems. In 16th Annual Symposium on Foundations of Computer Science (sfcs 1975), pages 57–64. IEEE, 1975.
[6] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. J. of Math, 58(5):345–363, 1936.
[7] Osamu Watanabe. On one-one polynomial time equivalence relations. *Theor. Comput. Sci.*, 38:157–165, 1985.