A Note on SpanP Functions

Meena Mahajan* Thomas Thierauf† Vinodchandran N.V.‡

Keywords. Computational Complexity.

1 Introduction

Valiant [10] introduced the class #P to count the number of solutions of NP sets. Recently, Fenner, Fortnow, and Kurtz [3] considered the class GapP, the closure of #P under subtraction, and showed many interesting properties of this class.

Köbler, Schöning, and Torán introduced the class SpanP that counts the number of distinct outputs produced by a nondeterministic Turing machine. With this concept, they could distinguish between whether two given graphs are isomorphic or not.

We introduce the class GapSpanP, the closure of SpanP under subtraction. We show that this class of functions coincides with the class $\mathrm{GapP}^{\mathrm{NP}}$.

2 Preliminaries

We follow the standard definitions and notations in computational complexity theory (see, e.g., [1] or [4]). We fix an alphabet to $\Sigma = \{0, 1\}$. P (NP, PP) is the class of sets that are accepted by some deterministic (nondeterministic, probabilistic) polynomial-time bounded Turing machine. FP denotes the class of all polynomial-time computable functions from Σ^* to $\mathbf{N}$.

A *counting machine* M is a nondeterministic Turing machine with two halting states: accepting and rejecting, and every computation path must end in one of these states. $\overline{M}$ denotes the machine obtained by interchanging the accepting and rejecting states of M.

$acc_M(x)$ denotes the number of accepting paths of a CM M on input x. $gap_M(x)$ denotes the value $acc_M(x) - acc_{\overline{M}}(x)$, i.e., the difference between the number of accepting paths and the

*The Institute of Mathematical Sciences, Madras 600 113, India. Part of the work done when the author was at the Department of Computer Science and Engineering, Indian Institute of Technology, Madras 600 036, India.

†Abteilung für Theoretische Informatik, Universität Ulm, 89069 Ulm, Germany. Part of the work done while visiting the Department of Computer Science at the University of Rochester, Rochester, NY. Supported in part by DFG Postdoctorial Stipend Th 472/1-1 and NSF grant CCR-8957604.

‡The Institute of Mathematical Sciences, Madras 600 113, India. Part of the work done when the author was at the Department of Computer Science and Engineering, Indian Institute of Technology, Madras 600 036, India.

number of rejecting paths. #P [10] and *GapP* [3] are the classes of functions f for which there exists a polynomial-time bounded counting machine M such that $f = acc_M$ and $f = gap_M$, respectively.

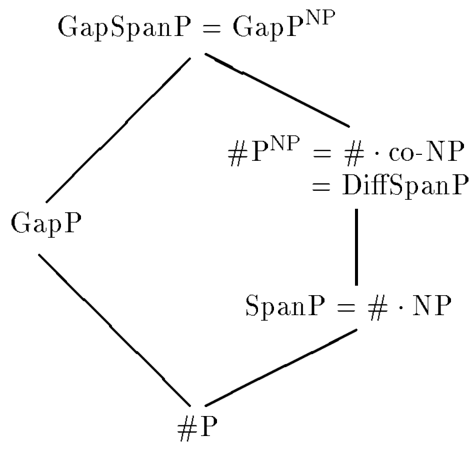
Another extension of #P is as follows. For a class $\mathcal{C}$ of sets, $\# \cdot \mathcal{C}$ is the class of functions f for which there exist a set $A \in \mathcal{C}$ and a polynomial p such that $f(x) = \|\{\, y \in \Sigma^{p(|x|)} \mid \langle x, y \rangle \in A \,\}\|$. Clearly, $\#\mathrm{P} = \# \cdot \mathrm{P}$.

For a nondeterministic Turing transducer M and an $x \in \Sigma^*$, let $out_M(x)$ denote the set of outputs produced by M along accepting paths on input x. $span_M(x)$ is defined as the number of different outputs of M on input x, i.e., $span_M(x) = \|out_M(x)\|$. *SpanP* [6] is the class of functions f for which there exists a polynomial-time bounded transducer M such that $f = span_M$.

Here, we introduce the gap analog of the class SpanP. For a nondeterministic transducer M, let $gapspan_M$ denote the function $span_M - span_{\overline{M}}$. In other words, for any $x \in \Sigma^*$, $gapspan_M(x)$ is the difference between the number of distinct witnesses M can produce for x and against x. *GapSpanP* is the class of functions f for which there exists a polynomial-time bounded nondeterministic transducer M such that $f = gapspan_M$.

As a variant of this definition, we consider the number of elements in the set theoretic difference of the span of M and $\overline{M}$. That is, for a nondeterministic transducer M, let $diffspan_M$ denote the function $\|out_M(x) - out_{\overline{M}}(x)\|$. In other words, we are counting the number of different outputs on accepting paths of M on input x that are *not* outputs on rejecting paths of M on input x. *DiffSpanP* is the class of functions f for which there is a polynomial-time bounded transducer M such that $f = diffspan_M$. Relativized versions of these classes are defined in the obvious way.

In this note, we characterize GapSpanP and DiffSpanP in terms of the class GapP and #P, respectively. Namely, we show that $\mathrm{GapSpanP} = \mathrm{GapP}^{\mathrm{NP}}$ and $\mathrm{DiffSpanP} = \#\mathrm{P}^{\mathrm{NP}}$. The following figure summarizes the relationships among these classes.



3 GapSpanP

Clearly, SpanP $\subseteq$ GapSpanP. This holds since any given transducer M can be modified to obtain a transducer M' that simulates M and if M reaches a rejecting state, M' branches once more and halts. One path will be accepting and the other rejecting. On both, M' outputs a special symbol, say #. Then $span_M = gapspan_{M'}$.

GapP is the closure of #P under subtraction.

Theorem 3.1. [3] GapP = #P − #P = #P − FP = FP − #P.

We show in the following Theorem that GapSpanP is the closure of SpanP under subtraction. Below, we will also establish the other equalities in Theorem 3.1 for GapSpanP.

Theorem 3.2. GapSpanP = SpanP − SpanP.

Proof. From the definition of GapSpanP it is clear that GapSpanP $\subseteq$ SpanP − SpanP. To show the other containment, let f and g be SpanP functions. Let $f = span_{M_0}$ and $g = span_{M_1}$. Without loss of generality, we can assume that both, M_0 and M_1 have rejecting computations on all inputs. For $i = 0, 1$, let M_i' be the transducer obtained from M_i that outputs a special symbol, say #, on any rejecting computation of M_i.

Consider the following transducer N on input x. First, N branches once. Then N simulates M_0' on input x on one branch, $\overline{M_1'}$ on input x on the other branch, and finally makes the same output as these machines on each branch.

Then we have $gapspan_N = f - g$. ■

Many arithmetic properties of GapP carry over to GapSpanP, because SpanP, like #P, is closed under addition and multiplication. In particular, GapSpanP is closed under addition, multiplication and subtraction.

For our other characterizations of GapSpanP, we need the following result from [5], [6] and [9] (see also [8]).

Lemma 3.3. [5, 6, 9] SpanP $= \# \cdot \mathrm{NP} \subseteq \#\mathrm{P}^{\mathrm{NP}} = \# \cdot \text{co-NP} \subseteq \mathrm{FP} - \mathrm{SpanP}$.

Theorem 3.4. GapSpanP = SpanP − FP = FP − SpanP = $\mathrm{GapP}^{\mathrm{NP}}$.

Proof. It follows from Theorem 3.2 that SpanP − FP $\subseteq$ GapSpanP and FP − SpanP $\subseteq$ GapSpanP. On the other hand, we have

$$\begin{aligned}
\mathrm{GapSpanP} &\subseteq \#\mathrm{P}^{\mathrm{NP}} - \#\mathrm{P}^{\mathrm{NP}} && \text{(by Theorem 3.2 and Lemma 3.3)}\\
&= \mathrm{GapP}^{\mathrm{NP}} && \text{(by Theorem 3.1, relativized)}\\
&= \#\mathrm{P}^{\mathrm{NP}} - \mathrm{FP} && \text{(by Theorem 3.1, relativized)}\\
&\subseteq (\mathrm{FP} - \mathrm{SpanP}) - \mathrm{FP} && \text{(by Lemma 3.3)}\\
&= \mathrm{FP} - (\mathrm{SpanP} + \mathrm{FP})\\
&= \mathrm{FP} - \mathrm{SpanP},
\end{aligned}$$

and analogously, $\mathrm{GapSpanP} \subseteq \mathrm{FP} - \#\mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{FP} - (\mathrm{FP} - \mathrm{SpanP}) = \mathrm{SpanP} - \mathrm{FP}$. ■

Note that all the inclusions in the above proof are in fact equations. [1]

For any relativizable class of sets $\mathcal{C}$, let $Low(\mathcal{C})$ denote the class of sets that are low for $\mathcal{C}$, that is, $Low(\mathcal{C}) = \left\{ L \mid \mathcal{C}^L = \mathcal{C} \right\}$.

SPP [3] is the class of sets L for which there is a GapP function f such that for all x,

$$\begin{aligned} x \in L &\implies f(x) = 1, \\ x \notin L &\implies f(x) = 0. \end{aligned}$$

Fenner, Fortnow, and Kurtz [3] have shown that SPP is precisely the class of sets that are low for GapP, i.e., SPP = Low(GapP). By Theorem 3.4, GapSpanP and GapP are different, unless NP is low for GapP.

Corollary 3.5. GapSpanP = GapP $\Leftrightarrow$ NP $\subseteq$ SPP

This nicely contrasts with the result in [6] that SpanP = #P $\Leftrightarrow$ NP $\subseteq$ UP, where UP is the subset of NP where the nondeterministic machines accepting a set have to be unambiguous. Note also that there is an oracle relative to which NP is not a subset of SPP (for example, an oracle such that NP is not contained in $\oplus$P [9]). Therefore, such an oracle also separates GapP and GapSpanP.

Next, we define the span analog of SPP.

Definition 3.6. A set L is in SpanSPP if there exists a machine M such that for all x,

$$\begin{aligned} x \in L &\implies gapspan_M(x) = 1, \\ x \notin L &\implies gapspan_M(x) = 0. \end{aligned}$$

It follows from Theorem 3.4 that SpanSPP = $\mathrm{SPP}^{\mathrm{NP}}$ (in a different setting, this has been observed independently in [11]). An obvious question now is whether SpanSPP is the class of sets that are low for GapSpanP. We can show only one inclusion, namely that any set that is low for GapSpanP must be in SpanSPP. Note that not all SpanSPP sets are low for GapSpanP, unless NP is low for $\mathrm{PP}^{\mathrm{NP}}$. Also, it is not even known whether SPP is low for GapSpanP.

Proposition 3.7. Low(GapSpanP) $\subseteq$ SpanSPP.

Proof. Let L be low for GapSpanP. First, we show that the characteristic function of L is in $\mathrm{GapSpanP}^L$. Let M be a transducer that, on input x, queries its oracle on x. If x is in the

[1] The authors have been informed by Klaus Wagner that, as reported in [11], the equations $\mathrm{SpanP} - \mathrm{SpanP} = \mathrm{SpanP} - \mathrm{FP} = \mathrm{FP} - \mathrm{SpanP} = \#\mathrm{P}^{\mathrm{NP}} - \#\mathrm{P}^{\mathrm{NP}} = \#\mathrm{P}^{\mathrm{NP}} - \mathrm{FP} = \mathrm{FP} - \#\mathrm{P}^{\mathrm{NP}}$ have independently been observed by the participants of a workshop in Georgenthal [2].

oracle set, M accepts and outputs x. Otherwise, M branches, with one branch accepting and one branch rejecting, and outputs x on both branches. Thus, we have

$$gapspan_{M^L}(x) = \begin{cases} 1 & \text{if } x \in L, \\ 0 & \text{otherwise.} \end{cases}$$

Since L is low for GapSpanP, there is a machine N such that $gapspan_N = gapspan_{M^L}$, and therefore, $L \in$ SpanSPP. ■

It follows from (relativized versions of) Theorem 3.2 and Theorem 3.4 that any set that is low for SpanP or $\#\mathrm{P}^{\mathrm{NP}}$ must be low for GapSpanP.

Corollary 3.8. $\mathit{Low}(\mathrm{SpanP}) \cup \mathit{Low}(\#\mathrm{P}^{\mathrm{NP}}) \subseteq \mathit{Low}(\mathrm{GapSpanP})$.

On the other hand, SpanSPP is clearly a weak class that does not, for example, contain PP unless the Counting Hierarchy [12] collapses.

Proposition 3.9. If PP is in SpanSPP, then the Counting Hierarchy collapses to SpanSPP.

Proof. If PP is in SpanSPP, then $\mathrm{PP}^{\mathrm{PP}} \subseteq \mathrm{PP}^{\mathrm{SPP}^{\mathrm{NP}}} \subseteq \mathrm{PP}^{\mathrm{NP}}$. The latter inclusion holds because SPP is low for GapP and the proof relativizes. Since $\mathrm{PP}^{\mathrm{NP}} \subseteq \mathrm{P}^{\mathrm{PP}}$ by Toda's theorem [7], we get $\mathrm{PP}^{\mathrm{PP}} \subseteq \mathrm{P}^{\mathrm{SPP}^{\mathrm{NP}}} = \mathrm{SPP}^{\mathrm{NP}}$. Consequently, by an inductive argument, the Counting Hierarchy collapses to $\mathrm{SPP}^{\mathrm{NP}}$. ■

Finally, we show an interesting characterization of $\#\mathrm{P}^{\mathrm{NP}}$. In [6], it is shown that the class $\#\mathrm{P}^{\mathrm{NP}}$ can be characterized as the class of all functions f, such that there exist two transducers M and M' such that $f(x)$ is the number of witnesses that M produces but M' does not produce on input x. This is denoted as $f = span_{M-M'}$. We will use this result in the following Theorem.

Theorem 3.10. $\#\mathrm{P}^{\mathrm{NP}} = \mathrm{DiffSpanP}$

Proof. From the definition, we have DiffSpanP $\subseteq \#\mathrm{P}^{\mathrm{NP}}$. For the reverse inclusion, let $f \in \#\mathrm{P}^{\mathrm{NP}}$. Then there exist two transducers M_0 and M_1 such that $f(x) = span_{M_0-M_1}(x)$ [6]. Now, let transducer N be defined as in the proof of Theorem 3.2. Then we have $diffspan_N(x) = span_{M_0-M_1}(x)$. ■

References

[1] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity – I.* Springer Verlag, Berlin Heidelberg, 1988.

[2] H. J. Burtschick. Notes of a seminar on complexity theory. Manuscript, Georgenthal 1991.

[3] S. A. Fenner, L. J. Fortnow, and S. A. Kurtz. Gap-definable counting classes. In *Proceedings of the Sixth Annual Conference on Structure in Complexity Theory*, pages 30–42, 1991.

[4] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[5] J. Köbler. *Strukturelle Komplexität von Anzahlproblemen.* PhD thesis, Institut für Informatik der Universität Stuttgart, 1989.

[6] J. Köbler, U. Schöning, and J. Torán. On countimg and approximation. *Acta Informatica*, 26:363–379, 1989.

[7] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing* 20:865-877, 1991.

[8] S. Toda and M. Ogiwara. Counting classes are at least as hard as the polynomial-time hierarchy. In *Proceedings of the Sixth Annual Conference on Structure in Complexity theory*, pages 2–12, 1991.

[9] J. Torán. Structural properties of the counting hierarchies. PhD thesis, Barcelona, 1988.

[10] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[11] H. Vollmer and K.W. Wagner. The complexity of finding middle elements. Manuscript, 1993. To appear in *International Journal of Foundation of Computer Science.*

[12] K.W. Wagner. The complexity of combinatorial problems with succinct input representations. *Acta Informatica* 23:325–356, 1986.