# Busy Beaver Sets: <br> Characterizations and Applications* 

Robert P. Daley<br>Computer Science Department, University of Pittsburgh, Pittsburgh, Pennsylvania 15260

## 1. Introduction

Busy beaver sets have proved themselves useful as examples for certain interesting properties in recursive function theory and computational complexity which are not easily demonstrated and as a means of simplifying constructions in the degrees of unsolvability. In many cases the power of the busy beaver sets seems to reside with a few very nice properties possessed by them. This article is based on some recent investigations into the nature of these properties (see (Daley and Reynolds, 1980; Daley, 1981b), and we will show here that two of these properties each provide characterizations for busy beaver sets. We will also present here an improvement obtained with William Reynolds for the construction of a solution to Post's problem. The work here on retraceability complements and elaborates upon some early work of Dekker (1954) and Yates (1962). In particular, a necessary and sufficient condition is given on the kind of retracing function which a retraceable set must have in order for its complement to be recursively enumerable.

Although many variations on the definition of busy beaver set have been used, the basic definition (see Daley, 1978) of a busy beaver set $B$ is given by

$$
\begin{aligned}
b(n) & =\max \left\{\Phi_{i}() \mid \mu(i) \leqslant n \quad \text { and } \quad \phi_{i}() \downarrow\right\} \\
B & =\{b(n)\}
\end{aligned}
$$

where $\left\{\phi_{i}\right\}$ is an acceptable Gödel numbering, $\left\{\Phi_{i}\right\}$ is a computational complexity measure for $\left\{\phi_{i}\right\}$ (see Blum, 1967a), and $\mu$ is a program size measure (see Blum, 1967b), and where we use $f()$ to denote a function of 0 arguments. We fix an acceptable Gödel numbering $\left\{\phi_{i}\right\}$ and assume without loss of generality that the $S-1-1$ function $S$ for $\left\{\phi_{i}\right\}$, which satisfies

[^0]$\phi_{S(i, j)}(x)=\phi_{i}(j, x)$, also satisfies $S(i, j) \geqslant \max \{i, j\}$. This stipulation will permit us to decide whether a given program takes a certain form.

As shown in Daley (1978), all such busy beaver sets are complete with respect to the Turing degrees. However, a different type of busy beaver set was used in Daley (1979) to provide an alternative construction for a proof of Sacks' Density Theorem. To motivate an enlarged notion of busy beaver set, which we present below, we recast the definition of the busy beaver set from Daley (1979) as follows:

$$
\begin{aligned}
b_{e}(n) & =\max \left\{\Phi_{S(e, p)}() \mid p \leqslant n \text { and } \phi_{S(e, p)}() \downarrow\right\}, \\
B_{e} & =\left\{b_{e}(n) \mid n \geqslant 1\right\}
\end{aligned}
$$

where $\Phi_{i}(x)>x$. Since $\phi_{S(e, p)}()=\phi_{e}(p)$, it is clear that $b_{e}(n)$ represents the maximum runtime (when defined) of program $e$ on all inputs $\leqslant n$. The set $B_{e}$ satisfies the properties that $A_{e}=\bar{B}_{e}$ is recursively enumerable and $\operatorname{deg} A_{e}=$ $\operatorname{deg} B_{e}=\operatorname{deg} W_{e}$. Thus it is sometimes useful to consider maximum runtimes over a proper subset of all possible computations. The definition of $b_{e}(n)$ can be reformulated so as to appear closer to the original definition if we weaken the notion of a program size measure to what we might call a program "type" measure. A program size measure $\mu$ must satisfy (see Blum, 1967b), the following two conditions:
(a) $\mu$ is total recursive;
(b) there exists a total recursive function $\kappa$ such that $\kappa(n)=$ cardinality $\{i \mid \mu(i)=n\}$; i.e., the finitely many programs of size $n$ can be effectively determined from $n$.
To obtain our notion of program "type" measure we weaken the second condition to permit an unbounded number of programs of a given size (with some restrictions), and also disregard programs of type " 0 " in order to ignore computations which are not of interest. For example, the definition of $b_{e}(n)$ can be reformulated as

$$
b_{e}(n)=\max \left\{\Phi_{i}() \mid 0<\mu(i) \leqslant n \text { and } \phi_{i}() \downarrow\right\},
$$

where

$$
\begin{aligned}
\mu(i) & =n, & & \text { if } i=\dot{S}(e, n), \\
& =0, & & \text { otherwise. }
\end{aligned}
$$

Also in Theorem 7 below we will consider an even more liberal notion of program type defined by

$$
\begin{aligned}
\mu(i) & =n, & & \text { if } \quad(\exists m)[i=S(S(e, n), m)], \\
& =0, & & \text { otherwise. }
\end{aligned}
$$

Observe here that if $i=S(S(e, n), m)$ then $m \leqslant i$ and $n \leqslant i$ by our assumption about $S$, and $\mu$ is total recursive. Also, we here have allowed infinitely many programs of type $n$. Finally, in all the cases of $\mu$ considered above for any computational complexity measure $\left\{\Phi_{i}\right\}$ for $\left\{\phi_{i}\right\}$ satisfying $\Phi_{i}(x)>$ $\max \{\mu(i), x\}$, we have that the function $g$ defined by

$$
\begin{aligned}
g(s) & =\min \left\{n>0 \mid(\exists i)\left[\mu(i)=n \text { and } \Phi_{i}()=s\right]\right\}, \\
& =0, \quad \text { if no such } i \text { exists, }
\end{aligned}
$$

is total recursive.
We now formally define our enlargement of the notion of busy beaver sets. We call $\Omega=\left\langle\mu,\left\{\Phi_{i}\right\}\right\rangle$ a measured system if $\mu$ is a total recursive function and $\left\{\Phi_{i}\right\}$ is a computational complexity measure for $\left\{\phi_{i}\right\}$ satisfying $\Phi_{i}(x)>\max \{\mu(i), x\}$ and for which the function $g_{\Omega}$ defined by

$$
\begin{aligned}
g_{\Omega}(s) & =\min \left\{n>0 \mid(\exists i)\left[\mu(i)=n \text { and } \Phi_{i}()=s\right]\right\}, \\
& =0, \quad \text { if no such } i \text { exists, }{ }^{1}
\end{aligned}
$$

is total recursive. For each measured system $\Omega$ we define

$$
\begin{aligned}
b_{\Omega}(n) & =\max \left\{\Phi_{i}() \mid 0<\mu(i) \leqslant n \text { and } \phi_{i}() \downarrow\right\} \\
B_{\Omega} & =\left\{b_{\Omega}(n)\right\}, \\
A_{\Omega} & =\bar{B}_{\Omega}, \quad \text { the complement of } B_{\Omega}
\end{aligned}
$$

Next, we define the total recursive function

$$
\begin{aligned}
b_{\Omega}(n, s) & =\max \left\{\Phi_{i}() \mid 0<\mu(i) \leqslant n \text { and } \Phi_{i}() \leqslant s\right\}, \\
& =0, \quad \text { if no such } i \text { exists. }{ }^{1}
\end{aligned}
$$

That $b_{\Omega}(n, s)$ is recursive follows from the relation $b_{\Omega}(n, s)=\max \{t \leqslant s \mid 0<$ $\left.g_{\Omega}(t) \leqslant n\right\}$. We observe the following simple relationships.

$$
\begin{gather*}
b_{\Omega}(n, s) \leqslant b_{\Omega}(n, s+1) .  \tag{1}\\
b_{\Omega}(n, s) \leqslant b_{\Omega}(n+1, s) .  \tag{2}\\
g_{\Omega}\left(b_{\Omega}(n, s)\right) \leqslant n .  \tag{3}\\
b_{\Omega}\left(g_{\Omega}(s), s\right)=s . \tag{4}
\end{gather*}
$$

[^1]\[

$$
\begin{gather*}
s \geqslant b_{\Omega}(n) \Rightarrow b_{\Omega}(n, s)=b_{\Omega}(n) .  \tag{5}\\
b_{\Omega}(n) \leqslant b_{\Omega}(n+1) .  \tag{6}\\
g_{\Omega}\left(b_{\Omega}(n)\right) \leqslant n .  \tag{7}\\
b_{\Omega}\left(g_{\Omega}(s)\right) \geqslant s .  \tag{8}\\
s>b_{\Omega}(n) \Rightarrow g_{\Omega}(s)>n .  \tag{9}\\
b_{\Omega}(n)=\max \left\{s \mid 0<g_{\Omega}(s) \leqslant n\right\} . \tag{10}
\end{gather*}
$$
\]

Lemma 1. $A_{\Omega}$ is recursively enumerable.
Proof. Let $\phi(x)=\min \left\{y \mid y>x\right.$ and $\left.0<g_{\Omega}(y) \leqslant g_{\Omega}(x)\right\}$. Then by (10),

$$
\begin{aligned}
x \in B_{\Omega} & \Leftrightarrow x=\max \left\{y \mid 0<g_{\Omega}(y) \leqslant g_{\Omega}(x)\right\} \\
& \Leftrightarrow(\forall y>x)\left[g_{\Omega}(y)>g_{\Omega}(x)\right] \\
& \Leftrightarrow \phi(x) \uparrow .
\end{aligned}
$$

Therefore, $A_{\Omega}=$ dom $\phi$. We observe here for future use that $\phi$ clearly satisfies

$$
\begin{aligned}
& \phi(x)=\min \left\{y \mid y>x \text { and } 0<g_{\Omega}(y) \leqslant g_{\Omega}(x)\right. \\
& \left.\quad \text { and }(\forall z)\left[x<z<y \Rightarrow g_{\Omega}(z)>g_{\Omega}(y)\right]\right\} .
\end{aligned}
$$

## 2. Retraceability

Given a set $X$ the principal function of $X$ is defined by $\pi_{X}(n)=n$th member in increasing order of $X$. A set $X$ is called retraceable if and only if there exists a partial recursive function $\psi$ such that $\psi\left(\pi_{X}(1)\right)=\pi_{X}(1)$ and $\psi\left(\pi_{X}(n)\right)=\pi_{X}(n-1)$ for all $n>1$. The function $\psi$ is called a retracing function for $X$.

Theorem 2. For every measured system $\Omega$ the busy beaver set $B_{\Omega}$ is retraceable.

Proof. Let $\Omega=\left\langle\mu,\left\{\Phi_{i}\right\}\right\rangle$ and define

$$
\begin{aligned}
\psi_{\Omega}(x) & =\max \left\{y \mid y<x \text { and } 0<g_{\Omega}(y)<g_{\Omega}(x)\right\} \\
& =x, \quad \text { if no such } y \text { exists. }
\end{aligned}
$$

Since $g_{\Omega}$ is total recursive, so is $\psi_{\Omega}$. Let $b_{n}=\pi_{B_{\Omega}}(n)$. Observe that it is possible that $b_{\Omega}$ may not be a one-to-one function so that $b_{n} \neq b_{\Omega}(n)$ is
possible. Since $b_{1}=b_{\Omega}(1)$, it is clear that $\psi_{\Omega}\left(b_{1}\right)=b_{1}$. From the definition of $b_{\Omega}(n, s), g_{\Omega}$ and $\psi_{\Omega}$ it follows easily that

$$
\begin{equation*}
\text { if } g_{\Omega}(x)>1 \text { then } \psi_{\Omega}(x)=b_{\Omega}\left(g_{\Omega}(x)-1, x\right) \tag{11}
\end{equation*}
$$

Therefore, since $b_{n}=b_{\Omega}\left(g_{\Omega}\left(b_{n}\right)\right)$ we see that for $n>1, \psi_{\Omega}\left(b_{n}\right)=b_{n-1}$.
For any retracing function $\psi$ we define its rank function $\rho_{\psi}$ by

$$
\rho_{\psi}(x)=\min \left\{m \mid \psi^{m}(x)=\psi^{m-1}(x)\right\},
$$

where $\psi^{0}(x)=x$ and $\psi^{k+1}=\psi\left(\psi^{k}(x)\right)$.
We point out some additional properties of the retracing function $\psi_{\Omega}$. First, as was observed in the proof above $\psi_{\Omega}$ is total recursive. Second, from its definition it is clear that $\psi_{\Omega}(x) \leqslant x$. Consequently, $p_{\psi}$ is total recursive. Third, if $x$ and $y$ are such that $\psi_{\Omega}(x)<y<x$, then $g_{\Omega}(y) \geqslant g_{\Omega}(x)$ and we have by (11), $\psi_{\Omega}(x)=b_{\Omega}\left(g_{\Omega}(x)-1, x\right)=b_{\Omega}\left(g_{\Omega}(x)-1, y\right)$. Therefore, it follows that $y>\psi_{\Omega}(y) \geqslant \psi_{\Omega}(x)$, and there must exist some $m>1$ such that $\psi_{\Omega}^{m}(y)=\psi_{\Omega}(x)$. It then follows that $\rho_{\psi_{\Omega}}(y) \geqslant \rho_{\psi_{\Omega}}(x)$ and so if $\psi_{\Omega}(x)<x$ then $\psi_{\Omega}(x)=\max \left\{y \mid y<x\right.$ and $\left.\rho_{\phi_{\Omega}}(y)<\rho_{\psi_{\Omega}}(x)\right\}$.

A retracing function $\psi$ which satisfies these three properties presents in a certain sense an enigma to any one wishing to gain information from $\psi$ about the recursively enumerable set $W_{i}$ whose complement is retraced by $\psi$. In other words information about $W_{i}$ can be gained from any retracing function at any place where those conditions are violated. For example, if $\psi(x) \uparrow$ then $x \in W_{i}$, so that by computing $\psi(x)$ and $\phi_{i}(x)$ concurrently until one halts one can discover that $x \in W_{i}$ whenever $\psi(x) \uparrow$. Also, if $\psi(x)>x$ then clearly $x \in W_{i}$. Finally, suppose that $x$ and $y$ are such that $\psi(x)<y<x$ and $\rho_{\psi}(y) \leqslant \rho_{\psi}(\psi(x))$. It follows that there must exist $m$ and $z$ such that $z=\psi_{\Omega}^{m}(y)$ and either $z=\psi(z)>\psi(x)$ or $z>\psi(x)>\psi(z)$. Therefore, either both $z \in W_{i}$ and $y \in W_{i}$ or, both $x \in W_{i}$ and $\psi(x) \in W_{i}$. Thus by computing $\phi_{i}(y)$ concurrently with $\phi_{i}(z)$ for all $z$ such that $\psi(y)<z<y$ (which includes $\psi(x))$ either we will discover during input $y$ that $y \in W_{i}$ (if $\phi_{i}(y)$ halts first), or later during input $x$ we will already know that $x \in W_{i}$ (if $\phi_{i}(\psi(x)$ ) halts first).

In view of these remarks we call a retracing function $\psi$ enigmatic if and only if
(1) $\psi$ is total recursive,
(2) $\psi(x) \leqslant x$,

$$
\begin{equation*}
\psi(x)<x \Rightarrow \psi(x)=\max \left\{y \mid y<x \text { and } \rho_{\psi}(y)<\rho_{\psi}(x)\right\} . \tag{3}
\end{equation*}
$$

Corollary 3. For any measured system $\Omega, B_{\Omega}$ is retraced by an enigmatic retracing function.

Proposition 4. Let $\psi$ be an enigmatic retracing function for the set $X$. Then
(a) $\psi(x)<y<x \Rightarrow \psi(x) \leqslant \psi(y)$.
(b) If $X$ is infinite then
(i) $x \in X \Leftrightarrow x=\max \left\{y \mid \rho_{\psi}(y)=\rho_{\psi}(x)\right\}$,
(ii) $\psi$ is finite-to-one.
(c) $\psi$ retraces at most one infinite set.
(d) $\bar{X}$ is recursively enumerable.

Proof. (a) Suppose $\psi(x)<y<x$. Then by condition (3) of the definition of an enigmatic retracing function $\rho_{\psi}(y) \geqslant \rho_{\psi}(x)>\rho_{\psi}(\psi(x))$, so that again by this condition $\psi(y) \geqslant \psi(x)$.
(b) Suppose $X$ is infinite and let $x_{n}=\pi_{X}(n)$.
(i) If $x_{n}<y<x_{n+1}$ then since $\psi\left(x_{n+1}\right)=x_{n}$ we see that $\rho_{\psi}(y)>$ $\rho_{\psi}\left(x_{n}\right)=n$. Thus, if $x_{n}<y$ then $\rho_{\psi}(y)>n$ and $x_{n}=\max \left\{y \mid \rho_{\psi j}(y) \leqslant n\right\}=$ $\max \left\{y \mid \rho_{\psi}(y)=n\right\}$.
(ii) From (i) above we see that if $y>x_{n}$ then $\psi(y) \geqslant x_{n}$. Therefore, $\psi$ is finite-to-one.
(c) This follows directly from part (b)(i).
(d) If $X$ is finite then $X$ is recursive so that $\bar{X}$ is recursively enumerable. Suppose $X$ is infinite and define $\phi_{i}(x)=\min \{y \mid y>x$ and $\left.\rho_{\psi}(y) \leqslant p_{\psi}(x)\right\}$. By part (b)(i) we have $\phi_{i}(x) \downarrow \Leftrightarrow x \neq \max \left\{y \mid p_{\psi}(y) \leqslant\right.$ $\left.\rho_{\psi}(x)\right\} \Leftrightarrow x \notin X$. Therefore, $\bar{X}=\operatorname{dom} \phi_{i}$ and $\bar{X}$ is recursively enumerable.

We point out a curious property of finite sets. Let $X=\left\{x_{1}, \ldots, x_{n}\right\}$. Define $\psi_{1}$ and $\psi_{2}$ by

$$
\begin{aligned}
\psi_{1}(x) & =x, & & \text { if } x=1 \text { or } x=x_{1} \\
& =x_{m-1}, & & \text { if } x=x_{m} \text { for } 1 \leqslant m<n \\
& =x-1, & & \text { otherwise }
\end{aligned}
$$

$$
\begin{aligned}
\psi_{2}(x) & =x, & & \text { if } x=1 \text { or } x=x_{1}, \\
& =\max \{y \mid y<x \text { and } y \in X\}, & & \text { otherwise. }
\end{aligned}
$$

Then $\psi_{1}$ and $\psi_{2}$ both are enigmatic retracing functions for $X$, and $\psi_{1}$ is finite-to-one but does not satisfy condition (b)(i) of Proposition 4, and $\psi_{1}$ satisfies (b)(i) but is not finite-to-one. In fact it is easy to see that no enigmatic retracing function for a finite set can satisfy both of these conditions.

Theorem 5. Every retraceable set whose complement is recursively enumerable has an enigmatic retracing function.

Proof. Let $\psi$ be a retracing function for the complement of the recursively enumerable set $W_{i}$. We will assume that $\bar{W}_{i}$ is infinite since by the above remarks such a enigmatic retracing function is easily constructed if $\bar{W}_{i}$ is finite. We will construct an enigmatic retracing function $\hat{\psi}$ from $\psi$ and $\phi_{i}$ by using a screening process for each input similar to that described previously. This process will involve concurrent computations and we will assume that no two of the concurrent computations halt simultaneously. We use $f(x) \rightarrow g(y)$ to denote that $f(x)$ halts before $g(y)$ when $f(x)$ and $g(y)$ are computed concurrently. We will also assume that the computation of $\hat{\psi}(x)$ halts as soon as $\hat{\psi}(x)$ is assigned a value. Define $\hat{\psi}(1)=1$ and $\hat{\psi}(x)$ for $x>1$ as follows:
$\hat{\psi}(x)$ : Compute $\psi(x)$ concurrently with $\phi_{i}(x)$.
(A) If $\phi_{i}(x) \rightarrow \psi(x)$ then set $\hat{\psi}(x)=x-1$.
(B) If $\psi(x) \rightarrow \phi_{i}(x)$ then
(1) If $\psi(x)>x$ then set $\hat{\psi}(x)=x-1$.
(2) If $\psi(x)=x$ then find $y=\max \{z<x \mid \hat{\psi}(z)=z\}$.
(a) If $y$ does not exist then set $\hat{\psi}(x)=\psi(x)$.
(b) If $y$ exists then compute $\phi_{i}(x)$ concurrently with $\phi_{i}(y)$.
(i) If $\phi_{i}(x) \rightarrow \phi_{i}(y)$ then set $\hat{\psi}(x)=x-1$.
(ii) If $\phi_{i}(y) \rightarrow \phi_{i}(x)$ then set $\hat{\psi}(x)=\psi(x)$.
(3) If $\psi(x)<x$ then
(a) If $\hat{\psi}(z)=z$ for some $\psi(x)<z<x$ then set $\hat{\psi}(x)=x-1$.
(b) If $\hat{\psi}(z)<z$ for all $\psi(x)<z<x$ then compute $\phi_{i}(x)$ and $\phi_{i}(\psi(x))$ concurrently with $\phi_{i}(z)$ for all $\psi(x)<z<x$.
(i) If $\phi_{i}(x) \rightarrow \phi_{i}(z)$ or $\phi_{i}(\psi(x)) \rightarrow \phi_{i}(z)$ for some such $z$ then set $\hat{\psi}(x)=x-1$.
(ii) If $\phi_{i}(z) \rightarrow \phi_{i}(\psi(x))$ and $\phi_{i}(z) \rightarrow \phi_{i}(x)$ for all such $z$ then set $\hat{\psi}(x)=\psi(x)$.

It is not difficult to prove by induction that $\hat{\psi}$ is total recursive. This is done by showing that at each concurrent branch point in the procedure for $\hat{\psi}$ at least one of the branches terminates: either $\phi_{i}(x) \downarrow$ or $\psi(x) \downarrow$; if $\psi(x)=x$ and $\psi(y)=y$ then either $\phi_{i}(x) \downarrow$ or $\phi_{i}(y) \downarrow$; if $\psi(x)<x$ then either both $\phi_{i}(x) \downarrow$ and $\phi_{i}(\psi(x)) \downarrow$, or $\phi_{i}(z) \downarrow$ for all $\psi(x)<z<x$ (in case $\left.x \notin W_{i}\right)$. Observe that if $\hat{\psi}(x) \neq \psi(x)$ then $\hat{\psi}(x)=x-1$ and $x \in W_{i}$. From this observation and step $(B)(1)$ it is clear that $\hat{\psi}(x) \leqslant x$. Suppose now that $\hat{\psi}(x)<y<x$. Since $\hat{\psi}(x) \neq x-1$ we have $\hat{\psi}(x)=\psi(x)$. By step (B3)(a) we see that $\hat{\psi}(z)<z$ for all $\hat{\psi}(x)<z<x$ (otherwise $\hat{\psi}(x)=x-1 \neq \psi(x)$ ). Similarly, from step (B3b)(i) we see that $\phi_{i}(z) \rightarrow \phi_{i}(\hat{\psi}(x))$ for all $\psi(x)<z<x$. Now, if $\psi(y)<\hat{\psi}(x)<y$ then since $\phi_{i}(y) \rightarrow \phi_{i}(\hat{\psi}(x))$, we see
that $\hat{\psi}(y)=y-1 \geqslant \hat{\psi}(x)$. So in any case $\hat{\psi}(x) \leqslant \hat{\psi}(y)<y$. Repeating this argument it follows for some $m>1$ that $\hat{\psi}^{m}(y)=\hat{\psi}(x)$ and so $\rho_{\hat{\psi}}(y) \geqslant \rho_{\hat{\psi}}(x)$. Thus, $\hat{\psi}(x)=\max \left\{y \mid y<x\right.$ and $\left.\rho_{\hat{\psi}}(y)<\rho_{\hat{\psi}}(x)\right\}$, and we conclude that $\hat{\psi}$ is enigmatic.

In view of Theorem 5 and part (d) of Proposition 4 we see that retraceable sets with recursively enumerable complements are characterized in terms of enigmatic retracing functions.

Theorem 6. A retraceable set has a recursively enumerable complement if and only if it is retraced by some enigmatic retracing function.

We now show that retraceability is a characterizing property for $B_{\Omega}$.
Theorem 7. For any recursively enumerable set $A$ with retraceable complement there is a measured system $\Omega$ such that $A=A_{\Omega}$.

Proof. Without loss of generality we can assume that $\bar{A}$ is infinite. Let $\psi$ be an enigmatic retracing function for $\bar{A}$, so that by Proposition $4, x \in \bar{A} \Leftrightarrow$ $x=\max \left\{y \mid \rho_{\psi}(y) \leqslant \rho_{\psi}(x)\right\}$. Let

$$
\begin{aligned}
\phi_{e}(n, 1) & =\min \left\{x \mid \rho_{\psi}(x)=n\right\}, \\
\phi_{e}(n, m+1) & =\min \left\{x \mid x>\phi_{e}(n, m) \text { and } \rho_{\psi}(x)=n\right\} .
\end{aligned}
$$

Let $\sigma(n, m)=S(S(e, n), m)$ so that $\phi_{\sigma(n, m)}()=\phi_{e}(n, m)$. Since $S(i, j)>$ $\max \{i, j\}$ we have $\sigma(n, m)>\max \{n, m\}$. We define $\mu$ by

$$
\begin{aligned}
\mu(i) & =n, & & \text { if }(\exists m<i)(\exists n<i)[i=\sigma(n, m)], \\
& =0, & & \text { otherwise. }
\end{aligned}
$$

Clearly, $\mu$ is total recursive. Let $\left\{\Phi_{i}\right\}$ be any computational complexity measure for $\left\{\phi_{i}\right\}$. We modify $\left\{\Phi_{i}\right\}$ as follows:

$$
\begin{aligned}
\hat{\Phi}_{i} & =\phi_{i}, & & \text { if } \quad \mu(i)>0 \\
& =\Phi_{i}, & & \text { if } \quad \mu(i)=0 .
\end{aligned}
$$

Observe that $\hat{\Phi}_{\sigma(n, m)}()=\phi_{\sigma(n, m)}() \leqslant y$ if and only if there are at least $m$ integers $z \leqslant y$ such that $\rho_{\psi}(z)=n$. Therefore, $\left\{\hat{\Phi}_{i}\right\}$ is a computational complexity measure for $\left\{\phi_{i}\right\}$, and since $g_{\Omega}=\rho_{\psi}$ is clearly total recursive, $\Omega=\left\langle\mu,\left\{\hat{\Phi}_{i}\right\}\right\rangle$ is a measured system. Then

$$
\begin{aligned}
b_{\Omega}(n) & =\max \left\{\hat{\Phi}_{i}() \mid 0<\mu(i) \leqslant n \text { and } \phi_{i}() \downarrow\right\} \\
& =\max \left\{\phi_{\sigma(j, m)}() \mid j \leqslant n \text { and } m>0 \text { and } \phi_{\sigma(j . m)}() \downarrow\right\} \\
& =\max \left\{y \mid \rho_{\psi}(y) \leqslant n\right\} .
\end{aligned}
$$

Thus, $\bar{A}=B_{\Omega}$ and so $A=A_{\Omega}$. We observe further that $\psi_{\Omega}=\psi$.

The construction given in the proof of Theorem 7 illustrates the true nature of the equivalence between busy beaver sets and retraceable sets with recursively enumerable complements (or via Theorem 6 sets retraced by enigmatic retracing functions). The equivalence rests in the correspondence between runtimes of programs of size $n$ and integers of retracing rank $n$. One of the referees of this article has pointed out that a shorter proof of the above equivalence can be obtained by combining the results of Yates (1962) that a set $A$ is retraceable if and only if there exists a total recursive function $f$ such that

$$
A=\{x \mid(\exists y<x)[f(y) \leqslant f(x)]\},
$$

with the property that $A_{\Omega}=\left\{x|(\exists y<x)| 0<g_{\Omega}(y)<g_{\Omega}(x)\right\}$ and with the fact that for every total recursive function $f$ there exists a measured system $\Omega$ such that $f=g_{\Omega}$.

## 3. Replete Sets

In Daley (1981a) it was observed that one property of the busy beaver set $B^{\circ}$ which was crucial to the constructions there was that $s \in B^{\circ} \Leftrightarrow A_{s}^{\circ}=A^{\circ} \mid s$, where $X \mid s=X \cap\{1, \ldots, s\}$ for any set $X$. For a recursively enumerable set $A$ the set of integers $s$ such that $A_{s}=A \mid s$ is called by Dekker (1954) the nondeficiency stages in the enumeration of $A$ (and later by Soare (1976) the true stages in the enumeration of $A$ ). Since we are assuming that $\Phi_{i}(x)>x$, it is clear that the set of non-deficiency stages in the enumeration of $A$ must form a subset of $\bar{A}$. We see that $B^{\circ}$ is precisely the set of non-deficiency stages for $A^{\circ}$. We therefore call a recursively enumerable set $W_{i}$ replete if there is some computational complexity measure $\left\{\Phi_{i}\right\}$ for $\left\{\phi_{i}\right\}$ such that $W_{i, s}=$ $W_{i} \mid s \Leftrightarrow s \in \bar{W}_{i}$.

Theorem 8. For every measured system $\Omega$ the set $A_{\Omega}$ is replete.
Proof. Let $\phi_{e}(x)=\min \left\{y \mid y>x\right.$ and $\left.0<g_{\Omega}(y) \leqslant g_{\Omega}(x)\right\}$. Given any computational complexity measure $\left\{\Phi_{i}\right\}$ for $\left\{\phi_{i}\right\}$ we define $\left\{\hat{\Phi}_{i}\right\}$ by

$$
\begin{aligned}
\hat{\Phi}_{i} & =\Phi_{i}, & & \text { if } i \neq e \\
& =\phi_{e}, & & \text { if } i=e
\end{aligned}
$$

Then it is easy to see that $\left\{\hat{\Phi}_{i}\right\}$ is a computational complexity measure for $\left\{\phi_{i}\right\}$ and that $A_{\Omega}=\operatorname{dom} \phi_{e}$. If $s \in W_{i}$ then $\hat{\Phi}_{e}(s)>s$ so $W_{e} \mid s \neq W_{e, s}$. Suppose $s \notin W_{e}$ and $x<s$ and $x \in W_{e}$. Then $\hat{\Phi}_{e}(x)=\min \{y \mid y>x$ and $\left.0<g_{\Omega}(y) \leqslant g_{\Omega}(x)\right\}$. If $g_{\Omega}(x) \leqslant g_{\Omega}(s)$ then we clearly have $\hat{\Phi}_{e}(x) \leqslant s$ (since
$g_{\Omega}\left(\hat{\phi}_{e}(x)\right) \leqslant g_{\Omega}(x)$ and $s=\max \left\{y \mid g_{\Omega}(y) \leqslant g_{\Omega}(s)\right\}$. If $g_{\Omega}(x)>g_{\Omega}(s)$ then $\phi_{e}(x) \leqslant s$ and so $\hat{\phi}_{e}(x) \leqslant s$. Therefore $x \in W_{e, s}$ and $A_{\Omega}$ is replete.

Theorem 9. Every replete recursively enumerable set has a retraceable complement.

Proof. Let $W_{e}$ be a replete recursively enumerable set so that for some computational complexity measure $\left\{\Phi_{i}\right\}$ for $\left\{\phi_{i}\right\}, W_{e, s}=W_{e} \mid s \Leftrightarrow s \in \bar{W}_{e}$. Define

$$
\begin{aligned}
\psi(x) & =\max \left\{y \mid y<x \text { and } \Phi_{i}(y)>x\right\}, \\
& =x, \quad \text { if no such } y \text { exists. }
\end{aligned}
$$

Let $\bar{W}_{e}=\left\{\bar{w}_{1}, \bar{w}_{2}, \ldots\right\}$. Since $W_{e}$ is replete, if $x \in W_{e}$ and $x<\bar{w}_{n}$ then $\Phi_{e}(x) \leqslant \bar{w}_{n}$, so that $\psi\left(\bar{w}_{1}\right)=\bar{w}_{1}$ and $\psi\left(\bar{w}_{n+1}\right)=\bar{w}_{n}$. Therefore $\psi$ retraces $\bar{W}_{e}$. We point out that $\psi$ is also enigmatic.

Summarizing the main characterization results of this article we have the following.

Theorem 10. Let a be a recursively enumerable set. Then the following statements are equivalent:
(a) $A$ is replete;
(b) $\bar{A}$ is retraceable;
(c) $\bar{A}$ is retraced by an enigmatic retracing function;
(d) $A=A_{\Omega}$ for some measured system $\Omega$.

## 4. Post's Problem

In this section we show how the busy beaver construction can be applied to construct recursively enumerable sets which are non-recursive and noncomplete. We will construct a particular measured system $\Omega=\left\langle\mu,\left\{\Phi_{i}\right\}\right\rangle$, where $\mu$ will be a program size measure and $\left\{\Phi_{i}\right\}$ is a modification of the space measure for Turing machines which we briefly describe. Although the description is in terms of multi-tape Turing machines, this measure can be defined for any acceptable Gödel numbering by using the intertranslatability between the programs of acceptable Gödel numberings. The multi-tape oracle Turing machine will have a read-only input tape, a write-only output tape, a read-only oracle tape on which is written the characteristic function of the oracle set, and some number of work tapes. Then, $\Phi_{i}^{X}(x)$ will be the maximum number of tape squares on any of these tapes used by Turing machine $i$ with oracle set $X$ on input $x$.

We first observe that if $X$ and $Y$ are infinite sets, then $\operatorname{deg} X \leqslant \operatorname{deg} Y \Leftrightarrow(\exists i)\left[v_{X}=\phi_{i}^{Y}\right]$, where $v_{X}$ is the next element function for the set $X$ defined by

$$
v_{X}=\min \{y \mid y>x \text { and } y \in X\} .
$$

We define a program size measure $\mu$ for $\left\{\phi_{i}\right\}$ by

$$
\mu(i)=n \Leftrightarrow f(n-1)<i \leqslant f(n),
$$

where the total recursive function $f$ is defined by

$$
\begin{aligned}
f(1) & =1 \\
f(n+1) & =\max \{S(e, i, j) \mid e, i, j \leqslant f(n)\},
\end{aligned}
$$

and where $S$ is the $S-2-1$ function given for $\left\{\phi_{i}\right\}$ such that $\phi_{S(e, i, j)}(x)=$ $\phi_{e}(i, j, x)$. Since $\mu$ is clearly a program size measure, it is clear that $\Omega$ is a measured system and $\operatorname{deg} A_{\Omega}=\operatorname{deg} B_{\Omega}=O^{\prime}$.

By the definition of $\mu$ if $n \geqslant \max \{\mu(i), \mu(j), \mu(e)\}$ then $\mu(S(e, i, j)) \leqslant n+1$. Let $e$ be a program such that $\phi_{e}(i, j)=\phi_{i}\left(\Phi_{j}()\right)$ and $\Phi_{S(e, i, j)}()>\phi_{S(e, i, j)}()$. Let $n \geqslant \mu(e)$ and let $j$ be such that $\Phi_{j}()=b_{\Omega}(n)$ and $\mu(j)=n$. If $\mu(i) \leqslant n$ and $\phi_{i}\left(b_{\Omega}(n)\right) \downarrow$ then $\mu(S(e, i, j)) \leqslant n+1 \quad$ so that $\quad \phi_{i}\left(b_{\Omega}(n)\right)=\phi_{i}\left(\Phi_{j}()\right)=$ $\phi_{S(e, i, j)}()<\Phi_{S(e, i, j)}() \leqslant b_{\Omega}(n+1)$. Thus we have

$$
\begin{align*}
(\forall i \geqslant e)(\forall n & \geqslant \mu(i))\left[\phi_{i}\left(b_{\Omega}(n)\right) \downarrow\right. \\
& \left.\Rightarrow \phi_{i}\left(b_{\Omega}(n)\right)<b_{\Omega}(n+1)\right] . \tag{12}
\end{align*}
$$

Recall from Lemma 1 that $A_{\Omega}=\operatorname{dom} \phi$, where

$$
\begin{aligned}
\phi(x)= & \min \left\{y \mid y>x \text { and } 0<g_{\Omega}(y) \leqslant g_{\Omega}(x)\right. \\
& \text { and } \left.(\forall z)\left[x<z<y \Rightarrow g_{\Omega}(z)>g_{\Omega}(y)\right]\right\} .
\end{aligned}
$$

Let $E$ denote the set of even integers. We define a recursively enumerable set $A_{E}$, which will provide a solution to Post's Problem, as follows: $A_{E}=\operatorname{dom} \phi_{E}$, where

$$
\begin{aligned}
\phi_{E}(x)= & \min \left\{y \mid y>x \text { and } 0<g_{\Omega}(y) \leqslant g_{\Omega}(x) \text { and } g_{\Omega}(y)\right. \text { is even } \\
& \text { and } \left.(\forall z)\left[x<z<y \Rightarrow g_{\Omega}(z)>g_{\Omega}(y)\right]\right\} .
\end{aligned}
$$

If $A_{E, s}$ denotes the elements enumerated by stage $s$ then we see that $A_{E, s}$ is recursive and that

$$
A_{E, s}=\left\{x \mid \phi_{E}(x) \leqslant s\right\} .
$$

We define

$$
\begin{aligned}
B_{E} & =\bar{A}_{E} \\
B_{E, s} & =\bar{A}_{E, s} \mid s
\end{aligned}
$$

Lemma 11. (a) $A_{E}$ is recursively enumerable.
(b) $B_{\Omega} \subseteq B_{E}$.
(c) $A_{E}$ is replete.
(d) $\quad(\forall n)\left[n\right.$ even $\left.\Rightarrow B_{E} \cap\left(b_{\Omega}(n-1), b_{\Omega}(n)\right)=\varnothing\right]$.

Proof. (a) This is immediate, since $A_{E}=\operatorname{dom} \phi_{E}$, and $\phi_{E}$ is clearly a partial recursive function.
(b) This follows from the definition of $\phi_{E}$ and (9).
(c) From the enumeration procedure for $A_{E}$ it is clear that $s \in B_{E, s}$ for all $s$ so that if $B_{E, s}=B_{E} \mid s$ then $s \in B_{E}$. Let $x \leqslant s$ be such that $x \in B_{E, s}$ and $x \in A_{E}$. Let $y=\phi_{E}(x)$ so that we have that $y>s$ and $g_{\Omega}(y)$ is even and $g_{\Omega}(y) \leqslant g_{\Omega}(x) \quad$ and also $(\forall z)\left[x<z<y \Rightarrow g_{\Omega}(z)>g_{\Omega}(y)\right]$. Then $g_{\Omega}(y) \leqslant g_{\Omega}(s)$ and $(\forall z)\left[s<z<g_{\Omega}(y) \Rightarrow g_{\Omega}(z)>g_{\Omega}(y)\right]$ also, so that $\phi_{E}(s)=y \quad$ and $s \in A_{E}$. Therefore, if $B_{E, s} \neq B_{E} \mid s$ then $s \notin B_{E}$, i.e., $s \in B_{E} \Rightarrow B_{E, s}=B_{E} \mid s$.
(d) Suppose that $n$ is even. Let $b_{\Omega}(n-1)<x<b_{\Omega}(n)$ and let $t$ be the least integer such that $b_{\Omega}(n-1, t)<x<b_{\Omega}(n, t)$. Then it is easy to see that $\phi_{E}(x)=t$ so $x \in A_{E}$.

Lemma 12. There exists a computational complexity measure $\left\{\boldsymbol{\Phi}_{i}\right\}$ such that
(a) $\Phi_{i}^{X}(x) \geqslant \max \left\{i, \phi_{i}^{X}(x)\right\}$.
(b) $s \geqslant \Phi_{i}^{X}(x) \Rightarrow \Phi_{i}^{X \mid s}(x)=\Phi_{i}^{X}(x)$.
(c) $\Phi_{i}^{X \mid s}(x) \leqslant s \Rightarrow \Phi_{i}^{X \mid s}(x)=\Phi_{i}^{X}(x)$.
(d) There exists a program transformation $\lambda$ such that

$$
\begin{aligned}
\phi_{\lambda(i, j)}() & =\phi_{i}\left(\Phi_{j}()\right) \\
\Phi_{\lambda(i, j)}() & =\Phi_{i}\left(\Phi_{j}()\right)
\end{aligned}
$$

(e) There exists a program transformation $\tau$ such that

$$
\begin{aligned}
\phi_{\tau(i)}(x) & =\min \left\{s \mid s>x \text { and } \Phi_{i}^{B} E, s(x) \leqslant s\right\}, \\
\Phi_{\tau(i)}(x) & =\phi_{\tau(i)}(x) .
\end{aligned}
$$

Proof. Although we use a particular model for the recursive functions to describe the complexity measure, this measure can be applied to any Gödel numbering because of the effective intertranslatability between programs of acceptable Gödel numberings.

We consider oracle Turing machines depicted in Fig. 1 which have a read-only input tape, a write-only output tape, a read-only oracle tape on which is written the characteristic function of the oracle set, and some number of work tapes. We first stipulate that $\mu(i)$ can be computed within $i$ work tape squares. Let $T O_{i}^{X}(x)$ denote the number of oracle tape squares used by program $i$ with oracle set $X$ on input $x$, and $T W_{i}^{X}(x)$ the maximum number of work tape squares on any of its work tapes used by program $i$ with oracle set $X$ on input $x$. Then we define

$$
\Phi_{i}^{X}(x)=\max \left\{i, x, \phi_{i}^{X}(x), T O_{i}^{X}(x), T W_{i}^{X}(x)\right\}
$$

Lemma 12(a) clearly follows from this definition. Lemma 12 (b) and Lemma 12(c) both follow from the fact that since $\Phi_{i}^{X}(x) \geqslant T O_{i}^{X}(x)$ no information about the oracle set $X$ involving members greater than $\Phi_{i}^{X}(x)$ can affect the computation. Lemma $12(\mathrm{~d})$ holds since the program $\lambda(i, j)$ can


Figure 1
(using exactly $\Phi_{j}$ () work tape squares) write $\Phi_{j}$ () on one work tape and then mimic the operation of program $i$ using this work tape as its input tape.

Finally, we indicate the proof of Lemma 12(e). Supposing that $B_{E, s}$ can be computed within $s$ work tape squares, then $\Phi_{i}^{B_{E, s}}(x) \leqslant s$ can also be tested within $s$ work tape squares by writing $B_{E, s}$ on one work tape and then mimicing program $i$ on input $x$ with this work tape as its oracle tape. Then program $\tau(i)$ on input $x$ tries successively larger integers $s>x$ until it finds one such that $\Phi_{i}^{B_{E, s}}(x) \leqslant s$, which by the above will take at most (and, since $\Phi_{i}^{X}(x) \geqslant \dot{\phi}_{i}^{X}(x)$, at least) $s$ work tape squares.

To see that $B_{E, s}$ can be computed within $s$ work tape squares, we first observe that since $\mu(i)$ can be computed within $i$ tape squares for each $t \leqslant s$, $g_{\Omega}(t)$ can be computed within $s$ tape squares, because $g_{\Omega}(t)$ is the size $(\leqslant t)$ of a program which uses $t$ work tape squares in its computation. Then to compute $B_{E, s}$, for successively larger values of $t$ such that $t \leqslant s$ we add (by setting bits on our special work tape for $\left.B_{E, s}\right)$ to $B_{E, s}$ all integers $x$ such that $b_{\Omega}\left(g_{\Omega}(t)-1, t\right)<x<t$ provided that $g_{\Omega}(t)$ is even. We can maintain the current value of $b_{\Omega}(n, t)$ by using a work tape and updating the values as new terminating computations are discovered.

We point out that $\lambda(i, j)$ takes the form $S(e, i, j)$ for some appropriate program $e$ so that there exists an $n_{\lambda}$ such that

$$
\begin{equation*}
\mu(j) \geqslant n_{\lambda} \Rightarrow \mu(\lambda(i, j)) \leqslant 1+\max \{\mu(i), \mu(j)\} . \tag{13}
\end{equation*}
$$

Lemma 13. $(\forall i)\left(\forall^{\infty} n\right)\left[n\right.$ odd $\left.\Rightarrow \phi_{i}^{B_{E}}\left(b_{\Omega}(n-1)\right) \neq b_{\Omega}(n)\right]$.
Proof. Suppose $\phi_{i}$ is given. Let $n_{0} \geqslant \max \left\{n_{\lambda}, \mu(\tau(i))\right\}$. Suppose now that $n \geqslant n_{0}$ and $n$ is odd and $\phi_{i}^{B_{E}}\left(b_{\Omega}(n-1)\right) \downarrow$. Let $t \in B_{\Omega}$ be such that $t \geqslant \Phi_{i}^{B_{E}}\left(b_{\Omega}(n-1)\right)$ and $g_{\Omega}(t)$ is even. Then by our choice of $n_{0}$ we have from Lemma $12(\mathrm{~b})$ that $\Phi_{i}^{B_{E} \mid t}\left(b_{\Omega}(n-1)\right)=\Phi_{i}^{B_{E}}\left(b_{\Omega}(n-1)\right)$, and from Lemma
 which we conclude $\phi_{\tau(i)}\left(b_{\Omega}(n-1)\right) \downarrow$. Let $s=\phi_{\tau(i)}\left(b_{\Omega}(n-1)\right)$. Since $\mu(\tau(i)) \leqslant n$ and $\phi_{\tau(i)}\left(b_{\Omega}(n-1)\right) \downarrow$ from (12) we see that $s<b_{\Omega}(n)$. Let $j$ be such that $\mu(j)=n$ and $\Phi_{j}()=b_{\Omega}(n-1)$, so $\phi_{\tau(i)}\left(b_{\Omega}(n-1)\right)=\phi_{\lambda}(\tau(i), j)()$. From parts (d) and (e) of Lemma 12 we have $\Phi_{\lambda_{(\tau(i), j)}}()=$ $\Phi_{\tau(i)}\left(b_{\Omega}(n-1)\right)=\phi_{\tau(i)}\left(b_{\Omega}(n-1)\right)$, and consequently $\quad \Phi_{\lambda(\tau(i), j)}()=s$. Therefore since $\mu(\tau(i)) \leqslant n_{0}$ and $\mu(j)=n-1 \geqslant n_{\lambda}$ and $s>b_{\Omega}(n-1)$, we have by (13) that $\mu(\lambda(\tau(i), j))=n$. Thus $g_{\Omega}(s)=n$ and since $n$ is odd and $s>b_{\Omega}(n-1)$ there can be no $y>s$ such that $g_{\Omega}(y) \leqslant n$ and $g_{\Omega}(y)$ is even. Therefore, $s \in B_{E}$ and by Lemma 11(c), $B_{E, s}=B_{E} \mid s$ and $\Phi_{i}^{B_{E} \mid s}\left(b_{\Omega}(n-1)\right)=$ $\Phi_{i}^{B_{E, s}\left(b_{\Omega}(n-1)\right) \leqslant s \text {. Then from Lemma } 12(\mathrm{c}) \text { we have } \Phi_{i}^{B_{E}}\left(b_{\Omega}(n-1)\right)=}$ $\Phi_{i}^{B_{E}{ }^{s}}\left(b_{\Omega}(n-1)\right)$ so that by Lemma $12(\mathrm{a}), \quad \phi_{i}^{B_{E}}\left(b_{\Omega}(n-1)\right) \leqslant$ $\Phi_{i}^{B_{E}}\left(b_{\Omega}(n-1)\right) \leqslant s<b_{\Omega}(n)$. Finally, we conclude that $\phi_{i}^{B_{E}}\left(b_{\Omega}(n-1)\right) \neq$ $b_{\Omega}(n)$.

Theorem 14. The recursively enumerable set $A_{E}$ provides a solution to Post's Problem, i.e., $O<\operatorname{deg} A_{E}<O^{\prime}$.

Proof. Suppose $n$ is odd. Then by Lemma 13 we have $\phi_{i}^{B_{E}}\left(b_{\Omega}(n-1)\right) \neq$ $b_{\Omega}(n)$, but $v_{B}\left(b_{\Omega}(n-1)\right)=b_{\Omega}(n)$ so that $\phi_{i}^{B_{E}} \neq v_{B_{O}}$ and therefore $\operatorname{deg} A_{E}=\operatorname{deg} B_{E} \neq \operatorname{deg} B_{\Omega}=O^{\prime}$. If $n$ is even then by Lemma $11(\mathrm{~d}), B_{E} \cap$ $\left(b_{\Omega}(n-1), b_{\Omega}(n)\right)=\phi$ so that $v_{B_{E}}\left(b_{\Omega}(n-1)\right)=b_{\Omega}(n)$. By (12) we have $\phi_{i}\left(b_{\Omega}(n-1)\right)<b_{\Omega}(n)$ so that $\phi_{i} \neq v_{B_{E}}$ and $\operatorname{deg} A_{E}=\operatorname{deg} B_{E} \neq O$.

In the above constructions if we reversed the roles of even and odd we would have obtained a set $A_{o}$ such that $O<\operatorname{deg} A_{o}<O^{\prime}$, where

$$
\begin{aligned}
A_{o} & =\operatorname{dom} \phi_{o} \\
B_{O} & =\bar{A}_{o}
\end{aligned}
$$

and where

$$
\begin{aligned}
\phi_{o}(x)= & \min \left\{y \mid y>x \text { and } g_{\Omega}(y) \leqslant g_{\Omega}(x) \text { and } g_{\Omega}(y)\right. \text { is odd } \\
& \text { and } \left.(\forall z)\left[x<z<y \Rightarrow g_{\Omega}(z)>g_{\Omega}(y)\right]\right\} .
\end{aligned}
$$

We see also that $B_{o}$ satisfies

$$
\begin{gather*}
(\forall n)\left[n \text { odd } \Rightarrow B_{o} \cap\left(b_{\Omega}(n-1), b_{\Omega}(n)\right)=\varnothing\right] .  \tag{14}\\
(\forall i)\left(\forall{ }^{\infty} n\right)\left[n \text { even } \Rightarrow \phi_{i}^{B o}\left(b_{\Omega}(n-1)\right) \neq b_{\Omega}(n)\right] . \tag{15}
\end{gather*}
$$

Then combining Lemma 11 (d) with (15) we have $v_{B_{E}} \neq \phi_{i}^{B o}$ for any program $i$, and combining Lemma 13 with (14) we have $v_{B_{o}} \neq \phi_{i}^{B_{E}}$ for any program $i$. Therefore, we conclude that $\operatorname{deg} A_{E} \leqslant \operatorname{deg} A_{O}$ and $\operatorname{deg} A_{O} \leqslant \operatorname{deg} A_{E}$; that is, $\operatorname{deg} A_{E}$ and $\operatorname{deg} A_{O}$ are incomparable degrees of unsolvability.

Furthermore, these techniques can be generalized to provide some modularization of the construction of various degrees of unsolvability as follows. For each recursive set $R$ we define the partial recursive function

$$
\begin{aligned}
\phi_{R}(x)= & \min \left\{y \mid y>x \text { and } g_{\Omega}(y) \leqslant g_{\Omega}(x) \text { and } g_{\Omega}(y) \in R\right. \\
& \text { and } \left.(\forall z)\left[x<z<y \Rightarrow g_{\Omega}(z)>g_{\Omega}(y)\right]\right\},
\end{aligned}
$$

and the recursively enumerable set

$$
A_{R}=\operatorname{dom} \phi_{R}
$$

THEOREM 15. $\left\{\operatorname{deg} A_{R}\right\}$ is isomorphic to the boolean algebra of the recursive sets under inclusion modulo the finite sets.

Replacing the condition " $g_{\Omega}(y) \in R$ " by " $\Phi_{i}\left(g_{\Omega}(y)\right) \leqslant x$," where
$W=\operatorname{dom} \phi_{i}$ in the definition of $\phi_{R}$ above, yields a recursively enumerable set which we denote by $A_{W}$.

Theorem 16. $\left\{\operatorname{deg} A_{W}\right\}$ is isomorphic the lattice of the recursively enumerable sets modulo the finite sets.

The details of the proof of the analog of Lemma 12 for $A_{R}$ and $A_{W}$ above can be found in Daley and Reynolds (1980).

Received: August 20, 1981; revised: May 10, 1982

## References

Blum, M. (1967), A machine independent theory of the complexity of recursive functions, $J$. Assoc. Comput. Mach. 14, 322-336. (a)
Blum, M. (1967), On the size of machines, Inform. and Control 11, 257-265. (b)
Daley, R. (1978), On the simplicity of busy beaver sets, Z. Math. Logik Grundlag. Math. 24, 207-224.
Daley, R. (1981), Busy beaver sets and the degrees of unsolvability, J. Symbolic Logic 46, 460-474. (a)
Daley, R. (1981), Retraceability, repleteness, and busy beaver sets, Proceedings, Mathematical Foundations of Computer Science Symposium, Lecture Notes in Computer Science No. 118, Springer-Verlag, 252-261. (b)
Daley, R., and Reynolds, W. (1980), "Towards Modular Constructions in Degrees of Unsolvability," Technical Report 80-5, Department of Computer Science, University of Pittsburgh.
Daley, R. (1979), On the simplification of constructions in degrees of unsolvability, Proceedings, Mathematical Foundations of Computer Science Symposium, Lecture Notes in Computer Science No. 74, Springer-Verlag, 259-265.
Dekker, J. (1954), A theorem on hypersimple sets, Proc. Amer. Math. Soc. 5, 791-796.
Soare, R. (1976), The infinite injury priority method, J. Symbolic Logic 41, 513-530.
Yates, C. (1962), Recursively enumerable sets and retracing functions, Z. Math. Logik Grundlag. Math. 8, 331-345.


[^0]:    *This work was supported by NSF Grant 76-00102A. The results of this article were presented at the NSF Workshop on Recursion Theoretic Aspects of Computer Science and also at the 10th Symposium on the Mathematical Foundations of Computer Science.

[^1]:    ${ }^{1}$ We point out that in previous work the default values for $b_{\Omega}(n, s)$ and $g_{\Omega}(s)$ have been defined to be $s$. Robert DiPaola has suggested to us that 0 might be a more appropriate default value. In any case, only the finitely many values before the first runtime of interest would be affected.

