$$M(w) \equiv \begin{cases} y & \text{if } M \text{ on input "}\triangleright w \sqcup\text{" eventually} \\ & \text{halts with output "}\triangleright y \sqcup\text{"} \\ \nearrow & \text{otherwise} \end{cases}$$

$\Sigma_0 \equiv \Sigma - \{\triangleright, \sqcup\}$; Usually, $\Sigma_0 = \{0, 1\}$; $w, y \in \Sigma_0^\star$

Definition 2.1 Let $f : \Sigma_0^\star \to \Sigma_0^\star$ be a total or partial function. We say that f is a **partial, recursive function** iff $\exists$ TM $M(f = M(\cdot))$, i.e., $\forall w \in \Sigma_0^\star(f(w) = M(w))$. □

Remark 2.2 *There is an easy to compute 1:1 and onto map between $\{0, 1\}^\star$ and* **N** *[Exercise]. Thus we can think of the contents of a TM tape as a natural number and talk about $f : \mathbf{N} \to \mathbf{N}$ being a* **recursive** *function.*

If the partial, recursive function f is total, i.e., $f : \mathbf{N} \to \mathbf{N}$ then we say that f is a **total, recursive function**. A partial function that is not total is called **strictly partial**.

Proposition 2.3 *The following functions are recursive. They are all total except for p_{even}.*

$$
\begin{aligned}
copy(w) &= ww \\
\sigma(n) &= n+1 \\
plus(n,m) &= n+m \\
mult(n,m) &= n \times m \\
exp(n,m) &= n^m \qquad \textit{(we let } exp(0,0) = 1\textit{)} \\
\chi_{even}(n) &= \begin{cases} 1 & \textit{if } n \textit{ is even} \\ 0 & \textit{otherwise} \end{cases} \\
p_{even}(n) &= \begin{cases} 1 & \textit{if } n \textit{ is even} \\ \nearrow & \textit{otherwise} \end{cases}
\end{aligned}
$$

Proof: Exercise: please convince yourself that you can build TMs to compute all of these functions! $\square$

Recursive Sets $=$ Decidable Sets $=$ Computable Sets

Definition 2.4 Let $S \subseteq \Sigma_0^\star$ or $S \subseteq \mathbf{N}$.

S is a **recursive set** iff the function χ_S is a (total) recursive function,

$$\chi_S(x) \;=\; \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases}$$

Examples: The following sets are recursive:

- $\{2n \;\big|\; n \in \mathbf{N}\}$, $\{2n+1 \;\big|\; n \in \mathbf{N}\}$
- $\{p \in \mathbf{N} \;\big|\; n \text{ is prime}\}$
- $\mathbf{N}$, $\emptyset$
- $0^\star$, in fact, every regular set is recursive.
- $\{0^n 1^n \;\big|\; n \in \mathbf{N}\}$, in fact, every CFL is recursive.
- $\{0^{n^2} \;\big|\; n \in \mathbf{N}\}$

S is a **recursive set** iff the function χ_S is a (total) recursive function,

$$\chi_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases} \quad \text{the characteristic function of } S$$

Definition 2.5 S is a **recursively enumerable set** (S is **r.e.**) iff the function p_S is a (partial) recursive function,

$$p_S(x) = \begin{cases} 1 & \text{if } x \in S \\ \nearrow & \text{otherwise} \end{cases} \quad \text{the polite characteristic function of } S$$

Proposition 2.6 *If S is recursive then S is r.e.*

Proof: Suppose S is recursive and let M be the TM computing χ_S. Build M' computing the following function:

$$M'(x) = \begin{cases} 1 & \text{if } M(x) = 1 \\ \nearrow & \text{otherwise} \end{cases} = p_S(x)$$

$\square$

C a class of sets, define co-**C** the class of sets whose complements are in **C**, co-**C** $= \{S \mid \overline{S} \in \mathbf{C}\}$.

Define:

$$r_S(x) = \begin{cases} \nearrow & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases}$$ the rude characteristic function of S

Observation 2.7 *S is co-*r.e. *iff r_S is a partial recursive function.*

Theorem 2.8 *S is recursive iff S and $\overline{S}$ are both r.e.*

Thus, Recursive $=$ r.e. $\cap$ co-r.e.

Proof: If $S \in$ Recursive then χ_S is a recursive function.

Thus so is $\chi_{\overline{S}}(x) = 1 - \chi_S(x)$

Thus, S and $\overline{S}$ are both recursive and thus both r.e.

Suppose $S \in$ r.e. $\cap$ co-r.e.

$p_S = M(\cdot); \quad r_S = M'(\cdot).$

Run M and M' in parallel – sometimes called **dovetailing**:

Define $T = M \| M'$ on input x:

1. **for** $n := 1$ to ∞ {
2. run $M(x)$ for n steps.
3. **if** $M(x) = 1$ in n steps **then return**(1)
4. run $M'(x)$ for n steps.
5. **if** $M'(x) = 0$ in n steps **then return**(0)}

Thus, $T(\cdot) = \chi_S$ and thus $S \in$ Recursive. $\square$

Definition 2.9 The **busy beaver function** $\text{bb}(n)$ is the maximum number of one's that an n state TM with alphabet $\Sigma = \{0, 1\}$ can leave on its tape and halt when started on the all 0 tape. $\square$

Note: In this example, to be consistent with literature, e.g., Hopcroft article[1], we are taking our tape alphabet $\Sigma = \{0, 1\}$.

Note that $\text{bb}(n)$ is **well defined**:

There are only finitely many n-state TMs, with $\Sigma = \{0, 1\}$.

$M_0 \quad M_1 \quad M_2 \quad M_3 \quad \ldots \quad M_{289} \quad M_{290} \quad \ldots \quad M_{3042} \quad M_{3043} \quad \ldots \quad M_{t(n)}$

Some finite subset of these eventually halt on input 0.

$M_1 \qquad M_3 \qquad M_{289} \qquad M_{3043}$

One of these prints the max # of 1's = $\text{bb}(n)$.

$1 \qquad 17 \qquad 22,961 \qquad 407$

[1] John Hopcroft, "Turing Machines" *Scientific American*, May 1984.

	q_1	q_2	q_3
0	$q_2, 1, \rightarrow$	$q_3, 0, \rightarrow$	$q_3, 1, \leftarrow$
1	$h, 1, -$	$q_2, 1, \rightarrow$	$q_1, 1, \leftarrow$

$\mathbf{bb}(3) \geq 6$

q_1	0	$\boxed{0}$	0	0	0	0	0
q_2	0	1	$\boxed{0}$	0	0	0	0
q_3	0	1	0	$\boxed{0}$	0	0	0
q_3	0	1	$\boxed{0}$	1	0	0	0
q_3	0	$\boxed{1}$	1	1	0	0	0
q_1	$\boxed{0}$	1	1	1	0	0	0
q_2	1	$\boxed{1}$	1	1	0	0	0
q_2	1	1	$\boxed{1}$	1	0	0	0
q_2	1	1	1	$\boxed{1}$	0	0	0
q_2	1	1	1	1	$\boxed{0}$	0	0
q_3	1	1	1	1	0	$\boxed{0}$	0
q_3	1	1	1	1	$\boxed{0}$	1	0
q_3	1	1	1	$\boxed{1}$	1	1	0
q_1	1	1	$\boxed{1}$	1	1	1	0
h	1	1	$\boxed{1}$	1	1	1	0

How quickly does $\mathrm{bb}(n)$ grow as n gets large?

$$
\begin{array}{lrl}
\text{Is } \mathrm{bb}(n) \in & O(n^2) & ? \\
& O(n^3) & ? \\
& O(2^n) & ? \\
& O(n!) & ? \\
& O(2^{2^n}) & ? \\
& O(\text{hyperexp}(n)) & ? \\
& O(\text{hyperexp}(\text{hyperexp}(n))) & ?
\end{array}
$$

$$
\text{hyperexp}(n) \quad = \quad 2^{2^{\cdot^{\cdot^{2}}}} \Big\} n
$$

Some Values of $\mathrm{bb}(n)$

States	**Max # of 1's**	**Lower Bound for** $\mathrm{bb}(n)$
3	$\mathrm{bb}(3)$	6
4	$\mathrm{bb}(4)$	12
5	$\mathrm{bb}(5)$	17
6	$\mathrm{bb}(6)$	35
7	$\mathrm{bb}(7)$	$22,961$
8	$\mathrm{bb}(8)$	$3^{92} \sim 7.9 \times 10^{43}$

Theorem 2.10 *Let $f : \mathbf{N} \rightarrow \mathbf{N}$ be a total, recursive function.*

$$\lim_{n\to\infty}\left(\frac{f(n)}{\text{bb}(n)}\right) \quad = \quad 0$$

That is, bb(n) is eventually bigger than any total, computable function!

Proof:

Let $g(n) = n \cdot (1 + \max_{i\leq n} f(i))$. Note: g is total, recursive.

Note: $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = 0$

We'll show that, bb$(n) \geq \quad g(n)$, for all sufficiently large n.

$g(n)$ is computed by some k-state TM.

For any n, let TM $\quad C_n = \underbrace{\boxed{\text{print} \quad n}}_{\lceil \log n \rceil}\underbrace{\boxed{\text{compute } g}}_{k}\underbrace{\boxed{\begin{array}{c}\text{binary}\\ \text{to unary}\end{array}}}_{17}$

C_n has $\lceil \log n \rceil + k + 17$ states.

C_n prints $g(n)$ 1's.

For $n \geq \lceil \log n \rceil + k + 17$ states,

$$\textbf{bb}(n) \;\geq\; \textbf{bb}(\lceil \log n \rceil + k + 17) \;\geq\; g(n)$$

$\square$

Fact 2.11 (Cantor) *The* **pairing function** $P(i,j) = \frac{(i+j)(i+j+1)}{2} + i$ *is a 1:1 correspondence between* $\mathbf{N} \times \mathbf{N}$ *and* $\mathbf{N}$: $\quad P : \mathbf{N} \times \mathbf{N} \underset{onto}{\overset{1:1}{\rightarrow}} \mathbf{N}$.

$$
\begin{aligned}
P(L(w), R(w)) &= w \\
L(P(i,j)) &= i \\
R(P(i,j)) &= j
\end{aligned}
$$

We can use the pairing function to think of a natural number as a pair of natural numbers.

Thus, the input to a Turing machine is a single binary string which may be thought of as a natural number, a pair of natural numbers, a triple of natural numbers, etc.

Turing machines can be encoded as **character strings** which can be encoded as **binary strings** which can be encoded as **natural numbers**.

TM$_n$	1	2	3	4
0	$1, 0, \rightarrow$	$3, \sqcup, \rightarrow$	$0, 0, -$	$0, 0, -$
1	$1, 1, \rightarrow$	$4, \sqcup, \rightarrow$	$0, 1, -$	$0, 1, -$
$\sqcup$	$2, \sqcup, \leftarrow$	$0, \sqcup, -$	$1, 0, \leftarrow$	$1, 1, \leftarrow$
$\triangleright$	$1, \triangleright, \rightarrow$	$0, \triangleright, -$	$0, \triangleright, -$	$0, \triangleright, -$

ASCII: $1, 0, \rightarrow; 1, 1, \rightarrow; 2, \sqcup, \leftarrow; 1, \triangleright, \rightarrow; ; \;\cdots\; 0, \triangleright, -$

$\{0,1\}^\star :$ w

$\mathbf{N} :$ n

There is a simple, countable listing of all TM's: $M_0, M_1, M_2, \cdots$

Encoding (transition function to binary string) and

Decoding (binary string to transition function) are trivial to compute.

Theorem 2.12 *There is a* **Universal Turing Machine** *U such that,*

$$U(P(n,m)) \quad = \quad M_n(m)$$

Proof: Given $P(n,m)$, compute n and m. n is a binary string encoding the state table of TM M_n. We can simulate M_n on input m by keeping track of its state, its tape, and looking at its state table, n, at each simulated step.

$$\boxed{\quad n \quad} \qquad \boxed{\triangleright}\boxed{\quad m \quad}\boxed{\sqcup}$$

$\square$

Theorem 2.12 is perhaps the most fundamental fact about computation.

$$\text{HALT} \quad = \quad \{P(n,m) \mid \text{TM } M_n(m) \text{ eventually halts}\}$$

Corollary 2.13 (Unsolvability of the Halting Problem) HALT *is not recursive.*

Proof: Suppose HALT were recursive. Then bb(n) would be a total recursive function: Cycle through all n-state TMs, M_i, and if $P(i,0) \in$ HALT, then count the number of 1's in $M_i(0)$. Return the maximum of these.

$\Rightarrow\Leftarrow$ $\square$

Listing All r.e. Sets

$$W_i \quad = \quad \{n \mid M_i(n) = 1\} \quad = \quad \mathcal{L}(M_i)$$

The set of all r.e. sets $= W_0, W_1, W_2, \cdots$

n	0	1	2	3	4	5	6	7	8	$\cdots$	W_n
0	$\boxed{0}$	0	0	0	0	0	0	0	0	$\cdots$	W_0
1	1	$\boxed{1}$	1	1	1	1	1	1	1	$\cdots$	W_1
2	1	0	$\boxed{1}$	0	1	0	1	0	1	$\cdots$	W_2
3	0	1	0	$\boxed{1}$	0	1	0	1	0	$\cdots$	W_3
4	1	0	0	0	$\boxed{0}$	0	0	0	0	$\cdots$	W_4
5	0	1	1	0	1	$\boxed{0}$	0	0	1	$\cdots$	W_5
6	1	0	0	1	0	0	$\boxed{1}$	0	0	$\cdots$	W_6
7	1	1	0	0	0	0	0	$\boxed{0}$	0	$\cdots$	W_7
8	0	1	0	0	0	0	0	0	$\boxed{0}$	$\cdots$	W_8
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
	0	1	1	1	0	0	1	0	0	$\cdots$	K
	1	0	0	0	1	1	0	1	1	$\cdots$	$\overline{K}$

$$\begin{aligned} K &= \{n \mid M_n(n) = 1\} \\ &= \{n \mid U(P(n,n)) = 1\} \\ &= \{n \mid n \in W_n\} \end{aligned}$$

Theorem 2.14 *K is r.e., but $\overline{K}$ is not r.e.*

Proof: $$\overline{K} \quad = \quad \{n \mid n \notin W_n\}$$

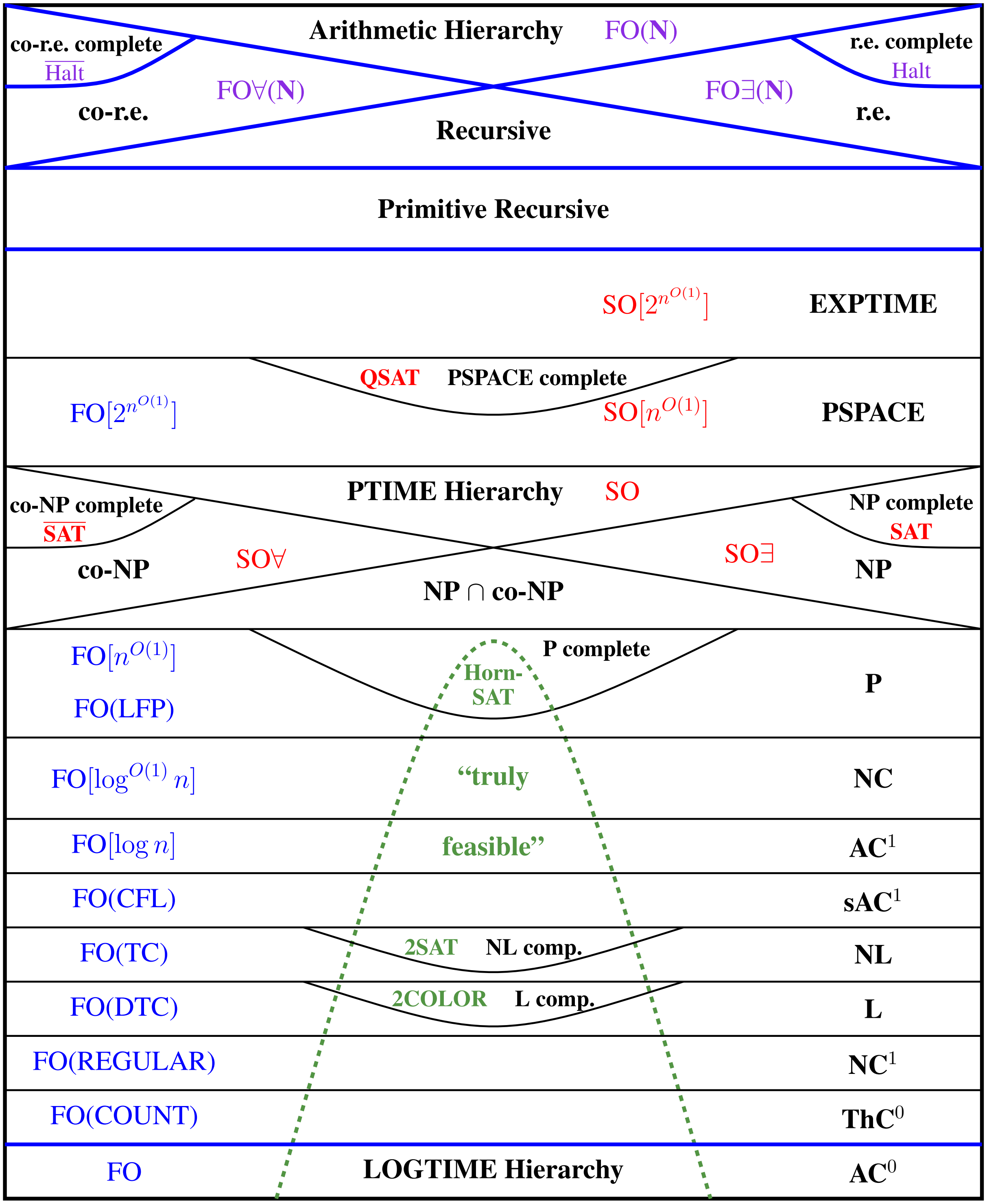
Suppose $\overline{K}$ were r.e. Then for some c,

$$\overline{K} \quad = \quad W_c \quad = \quad \{n \mid M_c(n) = 1\}$$

$$c \in K \;\Leftrightarrow\; M_c(c) = 1 \;\Leftrightarrow\; c \in W_c \;\Leftrightarrow\; c \in \overline{K}$$

$$\Rightarrow\Leftarrow$$ $\square$

Corollary 2.15 $K \in$ r.e. $-$ Recursive


Arithmetic Hierarchy
FO(N)
co-r.e. complete
$\overline{\text{Halt}}$
r.e. complete
Halt
FO∀(N)
FO∃(N)
co-r.e.
r.e.
Recursive
Primitive Recursive
$SO[2^{n^{O(1)}}]$
EXPTIME
QSAT
PSPACE complete
$FO[2^{n^{O(1)}}]$
$SO[n^{O(1)}]$
PSPACE
PTIME Hierarchy
SO
co-NP complete
$\overline{\text{SAT}}$
NP complete
SAT
co-NP
SO∀
SO∃
NP
NP ∩ co-NP
$FO[n^{O(1)}]$
P complete
Horn-
SAT
P
FO(LFP)
$FO[\log^{O(1)} n]$
"truly
NC
$FO[\log n]$
feasible"
AC^1
FO(CFL)
sAC^1
FO(TC)
2SAT
NL comp.
NL
FO(DTC)
2COLOR
L comp.
L
FO(REGULAR)
NC^1
FO(COUNT)
ThC^0
FO
LOGTIME Hierarchy
AC^0