CMPSCI 601: **Turing Machines** Lecture 4

$$M = (Q, \Sigma, \delta, s)$$

Q: finite set of states; $s \in Q$

Σ: finite set of symbols; $\triangleright, \sqcup \in \Sigma$

δ: $Q \times \Sigma \ \rightarrow \ (Q \cup \{h\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}$

s	$\boxed{\triangleright}$	1	1	0	1	$\sqcup$	$\sqcup$	$\cdots$

Example TM

mvRt.tm	s	q	q_0	q_1
0	$s, 0, \rightarrow$	$q_0, \sqcup, \rightarrow$		
1	$s, 1, \rightarrow$	$q_1, \sqcup, \rightarrow$		
$\sqcup$	$q, \sqcup, \leftarrow$		$s, 0, \leftarrow$	$s, 1, \leftarrow$
$\triangleright$	$s, \triangleright, \rightarrow$	$h, \triangleright, -$		
comment	find $\sqcup$	memorize & erase	change $\sqcup$ to 0	change $\sqcup$ to 1

s	$\boxed{\triangleright}$	1	1	0	1	$\sqcup$	$\sqcup$	$\cdots$
s	$\triangleright$	$\boxed{1}$	1	0	1	$\sqcup$	$\sqcup$	$\cdots$
s	$\triangleright$	1	$\boxed{1}$	0	1	$\sqcup$	$\sqcup$	$\cdots$
s	$\triangleright$	1	1	$\boxed{0}$	1	$\sqcup$	$\sqcup$	$\cdots$
s	$\triangleright$	1	1	0	$\boxed{1}$	$\sqcup$	$\sqcup$	$\cdots$
s	$\triangleright$	1	1	0	1	$\boxed{\sqcup}$	$\sqcup$	$\cdots$
q	$\triangleright$	1	1	0	$\boxed{1}$	$\sqcup$	$\sqcup$	$\cdots$
q_1	$\triangleright$	1	1	0	$\sqcup$	$\boxed{\sqcup}$	$\sqcup$	$\cdots$
s	$\triangleright$	1	1	0	$\boxed{\sqcup}$	1	$\sqcup$	$\cdots$

mvRt.tm	s	q	q_0	q_1
0	$s, 0, \rightarrow$	$q_0, \sqcup, \rightarrow$		
1	$s, 1, \rightarrow$	$q_1, \sqcup, \rightarrow$		
$\sqcup$	$q, \sqcup, \leftarrow$		$s, 0, \leftarrow$	$s, 1, \leftarrow$
$\triangleright$	$s, \triangleright, \rightarrow$	$h, \triangleright, -$		

s	$\boxed{\triangleright}$	1	1	0	1	$\sqcup$	$\sqcup$	$\cdots$
$\vdots$		$\vdots$		$\vdots$		$\vdots$		
q_1	$\triangleright$	1	1	0	$\sqcup$	$\boxed{\sqcup}$	$\sqcup$	$\cdots$
s	$\triangleright$	1	1	0	$\boxed{\sqcup}$	1	$\sqcup$	$\cdots$
q	$\triangleright$	1	1	$\boxed{0}$	$\sqcup$	1	$\sqcup$	$\cdots$
q_0	$\triangleright$	1	1	$\sqcup$	$\boxed{\sqcup}$	1	$\sqcup$	$\cdots$
s	$\triangleright$	1	1	$\boxed{\sqcup}$	0	1	$\sqcup$	$\cdots$
$\vdots$		$\vdots$		$\vdots$		$\vdots$		
q	$\boxed{\triangleright}$	$\sqcup$	1	1	0	1	$\sqcup$	$\cdots$
h	$\boxed{\triangleright}$	$\sqcup$	1	1	0	1	$\sqcup$	$\cdots$

Hilbert's Program [1901]: Give a complete axiomization of all of mathematics!

Such a complete axiomization would have provided a mechanical procedure to churn out exactly all true statements in mathematics.

This led to active interest in 1930's in the question: **"What is a mechanical procedure?"**

Church: Lambda calculus

Gödel: Recursive function

Kleene: Formal system

Markov: Markov algorithm

Post: Post machine

Turing: Turing machine

Fact: The above models are all define exactly the same class of "computable" functions.

Church-Turing Thesis: The intuitive idea of "effectively computable" is captured by the precise definition of "computable" in any of the above models.

"Why is a Turing machine as powerful as any other computational model?"

Intuitive answer: Imagine any computational device. It has:

- Finitely many states
- Ability to scan limited amount per step: one page at a time
- Ability to print limited amount per step: one page at a time
- Next state *determined* by current state and page currently being read (but what about randomization?)

Note: Without the potentially infinite supply of tape cells, paper, extra disks, extra tapes, etc. we have just a (potentially huge) **finite state machine**.

The PC on your desk, with 20 GB of hard disk is a finite state machine with over $2^{160,000,000,000}$ states!

This is better modeled as a TM with a bounded number of states, and an "infinite tape", actually meaning a finite memory that expands whenever necessary .

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

$\Sigma_0 \;\equiv\; \Sigma - \{\triangleright, \sqcup\}$

Usually, $\Sigma_0 = \{0, 1\}$

Definition 4.1 Let $f : \Sigma_0^\star \to \Sigma_0^\star$ be a total or partial function. We say that f is **recursive** iff $\exists$ TM M, $f = M(\cdot)$, i.e.,

$$(\forall w \in \Sigma_0^\star) \quad f(w) = M(w) \; . \qquad \spadesuit$$

Remark 4.2 *There is an easy to compute 1:1 and onto map between* $\{0,1\}^\star$ *and* **N**. *Thus we can think of the contents of a TM tape as a natural number and talk about* $f : \mathbf{N} \rightarrow \mathbf{N}$ *being* **recursive**. *(We may visit this issue in HW#2.)*

Partial function $f : \mathbf{N} \rightarrow \mathbf{N}$ is a total function $f : D \rightarrow \mathbf{N}$ where $D \subseteq \mathbf{N}$. A partial function that is not total is called **strictly partial**. If $n \in \mathbf{N} - D$, $f(n) = \nearrow$.

Definition 4.3 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,

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

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

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

♠

Proposition 4.4 *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' simulating M but diverging if $M(x) = 0$. Thus M' computes p_S. ♠

Proposition 4.5 *The following functions are recursive. They are all total except for* p_{even}.

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

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

If $\mathcal{C}$ is any class of sets, define co-$\mathcal{C}$ to be the class of sets whose complements are in $\mathcal{C}$,

$$\text{co-}\mathcal{C} \quad = \quad \{S \mid \overline{S} \in \mathcal{C}\}$$

Theorem 4.6 *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.

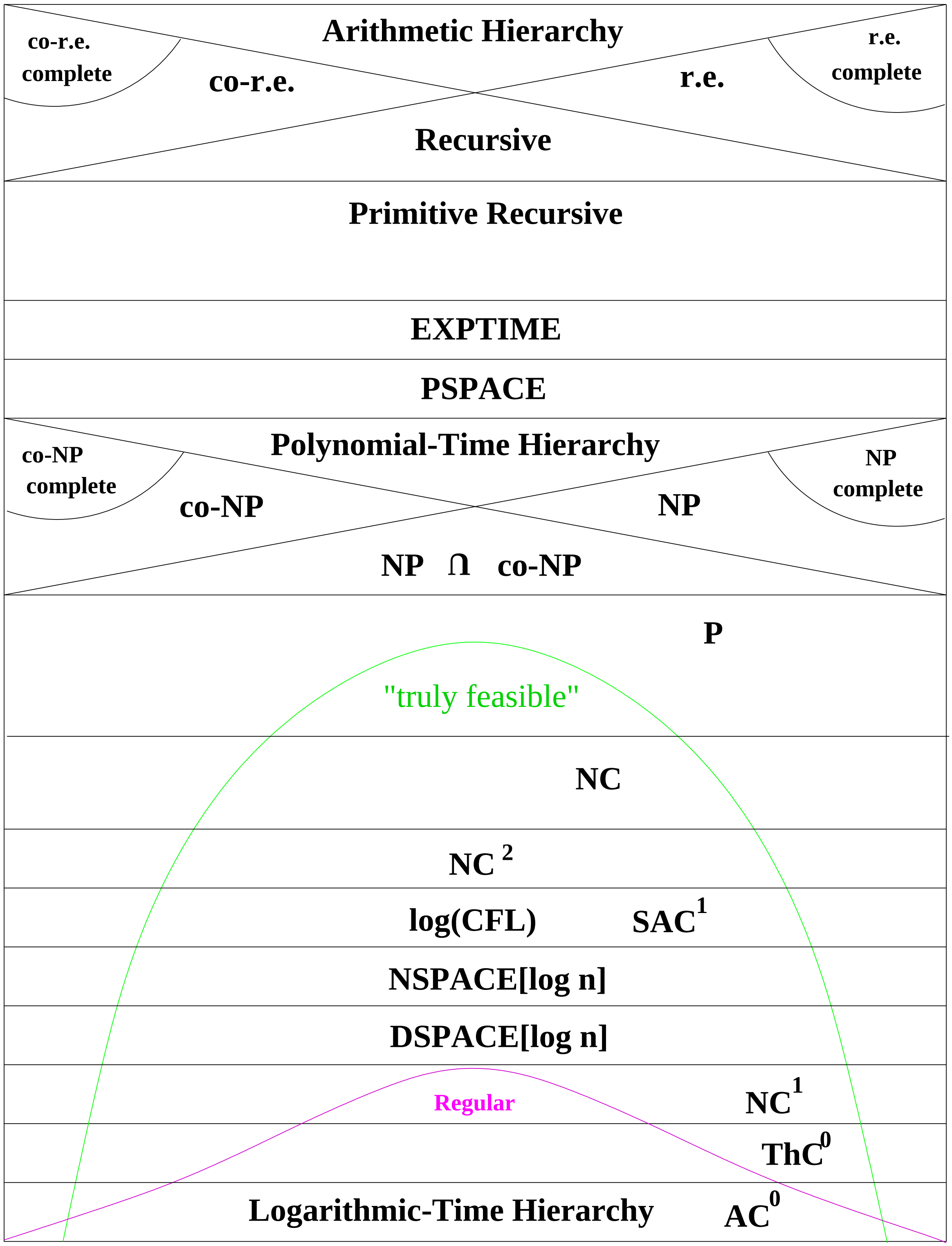
Suppose $S \in$ **r.e.** $\cap$ co-**r.e.**

$p_S = M(\cdot); \quad p_{\overline{S}} = M'(\cdot)$

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** $M'(x) = 1$ in n steps **then return**(0)$\}$

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


Arithmetic Hierarchy
co-r.e. complete
co-r.e.
r.e.
r.e. complete
Recursive
Primitive Recursive
EXPTIME
PSPACE
Polynomial-Time Hierarchy
co-NP complete
co-NP
NP
NP complete
NP ∩ co-NP
P
"truly feasible"
NC
NC 2
log(CFL)
SAC1
NSPACE[log n]
DSPACE[log n]
Regular
NC1
ThC0
Logarithmic-Time Hierarchy
AC0