Initial functions:

$\zeta() = 0$

$\sigma(x) = x + 1$

$\pi_i^n(x_1, \ldots, x_n) = x_i, \quad n = 1, 2, \ldots, \quad 1 \leq i \leq n$

Composition: $g_i : \mathbf{N}^k \to \mathbf{N}, 1 \leq i \leq m; \; ; h : \mathbf{N}^m \to \mathbf{N}$:

$$\mathcal{C}(h; g_1, \ldots, g_m)(x_1, \ldots, x_k) \; = \; h(g_1(\overline{x}), \ldots, g_m(\overline{x}))$$

Primitive Recursion: $g : \mathbf{N}^k \to \mathbf{N}; \; h : \mathbf{N}^{k+2} \to \mathbf{N}$: $f(n, y_1, \ldots, y_k) = \mathcal{P}(g, h)(n, y_1, \ldots y_k)$, given by:

$$\begin{aligned} f(0, y_1, \ldots y_k) \; &= \; g(y_1, \ldots, y_k) \\ f(n+1, y_1, \ldots y_k) \; &= \; h(f(n, y_1, \ldots, y_k), n, y_1, \ldots, y_k) \end{aligned}$$

Def: The **primitive recursive functions**, **PrimRecFcns**, is the smallest class of functions containing the Initial functions and closed under Composition and Primitive Recursion.

Exercises (HW#3):

1. A function is primitive recursive iff it is computable in Bloop.
2. Every primitive recursive function is total recursive.
3. There is a total recursive function that is not primitive recursive.

Prop: The following functions are Primitive Recursive:

1. $M_1(x) \;=\;$ **if** $(x > 0)$ **then** $(x - 1)$ **else** 0
2. $x \ominus y \;=\;$ **if** $(y \leq x)$ **then** $(x - y)$ **else** 0
3. $+$
4. $*$
5. $\exp(x, y) \;= y^x$
6. $\exp^*(x) \;=\; 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \Big\}x$
7. $=, \leq, <, >, \neq$.
8. P, L, R **exercise**

As we will start to see now (maybe with HW#3), you can do almost anything with primitive recursive functions:

Primitive Recursive COMP Theorem: [Kleene]
Let COMP(n, x, c, y) mean $M_n(x) = y$, and that c is M_n's complete computation on input x.
Then COMP is a Primitive Recursive predicate.

Proof: We will encode TM computations:

$$c = \mathrm{Seq}(\mathrm{ID}_0, \mathrm{ID}_1, \ldots, \mathrm{ID}_t)$$

Where each ID_i is a sequence number of tape-cell contents:

$$\mathrm{ID}_i = \mathrm{Seq}(\triangleright, a_1, \ldots, a_{i-1}, [\sigma, a_i], a_{i+1}, \ldots, a_r)$$

$$\mathrm{COMP}(n, x, c, y) \;\equiv$$

$$\mathrm{START}(\mathrm{Item}(c, 0), x) \;\wedge\; \mathrm{END}(\mathrm{Item}(c, \mathrm{Length}(c) - 1), y) \;\wedge$$
$$(\forall i < \mathrm{Length}(c))\mathrm{NEXT}(n, \mathrm{Item}(c, i), \mathrm{Item}(c, i + 1))$$

♠

Theorem 9.1 *The following problems are decidable in polynomial time.*

$$\begin{aligned}
\text{EmptyNFA} &= \{N \mid N \textit{ is an NFA};\ \mathcal{L}(N) = \emptyset\} \\
\Sigma^\star\text{DFA} &= \{D \mid D \textit{ is a DFA};\ \mathcal{L}(D) = \Sigma^\star\} \\
\text{MemberNFA} &= \{\langle N, w\rangle \mid N \textit{ is an NFA};\ w \in \mathcal{L}(N)\} \\
\text{EqualDFA} &= \{\langle D_1, D_2\rangle \mid D_1, D_2 \textit{ DFAs};\ \mathcal{L}(D_1) = \mathcal{L}(D_2)\} \\
\text{EmptyCFL} &= \{G \mid G \textit{ is a CFG};\ \mathcal{L}(G) = \emptyset\} \\
\text{MemberCFL} &= \{\langle G, w\rangle \mid G \textit{ is a CFG};\ w \in \mathcal{L}(G)\}
\end{aligned}$$

$$\text{MemberCFL} \quad = \quad \{\langle G, w\rangle \mid G \text{ is a CFG; } w \in \mathcal{L}(G)\}$$

CYK Dynamic Programming Algorithm:

1. Assume G in **Chomsky Normal Form:** $N \to AB$, $N \to a$.
2. **Input:** $w = w_1 w_2 \ldots w_n$; G with nonterminals $S, A, B, \ldots$
3. $N_{ij} \equiv \begin{cases} 1 & \text{if } N \stackrel{\star}{\to} w_i \cdots w_j \\ 0 & \text{otherwise} \end{cases}$
4. **return**(S_{1n})

$$N_{i,i} \;=\; \textbf{if } (\text{“}N \to w_i\text{”} \in R) \textbf{ then } 1 \textbf{ else } 0$$

$$N_{i,j} \;=\; \bigvee_{\text{“}N \to AB\text{”} \in R} (\exists k)(i \le k < j \;\wedge\; A_{i,k} \;\wedge\; B_{k+1,j})$$

Theorem 9.2 *The following problem is co-***r.e.***-complete:*

$$\Sigma^\star\text{CFL} \quad = \quad \{G \mid G \text{ is a CFG; } \mathcal{L}(G) = \Sigma_G^\star\}$$

Proof: [J. Hartmanis, Neil's advisor]

$\overline{\Sigma^\star\text{CFL}} \in$ **r.e.**:

Input: G

Define: $\Sigma_G^\star = \{w_0, w_1, w_2, \ldots\}$

1. **for** $i := 0$ to ∞ {
2. **if** $w_i \notin \mathcal{L}(G)$, **then return**(1)}

Clearly this returns 1 iff $G \in \overline{\Sigma^\star\text{CFL}}$.

Proposition 9.3 *EMPTY is co-r.e. complete, where,*

$$\textit{EMPTY} \quad = \quad \{n \mid W_n = \emptyset\}$$

Proof: Follows from HW#2 where we showed NON-EMPTY to be r.e.-complete. ♠

Claim 9.4 *EMPTY* $\leq \Sigma^\star$CFL.

Corollary 9.5 $\Sigma^\star$CFL *is co-r.e. complete and thus not recursive.*

How can we prove the Claim?

We need to define: $g : \mathbf{N} \rightarrow \{0,1\}^\star$,

$$n \in \text{EMPTY} \quad \Leftrightarrow \quad g(n) \in \Sigma^\star\text{CFL}$$

$$(\forall x) M_n(x) \neq 1 \quad \Leftrightarrow \quad \mathcal{L}(g(n)) = \Sigma_n^\star$$

$$M_n \text{ has no accepting computations} \quad \Leftrightarrow \quad \mathcal{L}(g(n)) = \Sigma_n^\star$$

Instantaneous Description (ID)

of a computation of M_n:

M_n has alphabet $\{0, 1\}$, states $\{\hat{0}, \hat{1}, \ldots, \hat{q}\}$ where $\hat{0}$ is the halting state and $\hat{1}$ is the start state.

$$\mathrm{ID}_0 \quad = \quad \hat{1} \triangleright w_1 \, w_2 \, \cdots \, w_r \sqcup$$

Suppose M_n in state $\hat{1}$ looking at a "$\triangleright$" writes a "$\triangleright$" changes to state $\hat{3}$, and moves to the right.

$$\mathrm{ID}_1 \quad = \quad \triangleright \hat{3} \, w_1 \, w_2 \, \cdots \, w_r \sqcup$$

YesComp(n) =

$$\left\{ \text{ID}_0 \# \text{ID}_1^R \# \text{ID}_2 \# \text{ID}_3^R \# \cdots \# \text{ID}_t \;\middle|\; \text{ID}_0 \cdots \text{ID}_t \begin{array}{c} \text{accepting} \\ \text{comp of } M_n \end{array} \right\}$$

Lemma 9.6 *For each n, $\overline{\text{YesComp}(n)}$ is a CFL.*

Furthermore, there is a function $g \in F(\mathbf{L})$, for all n,

$$\mathcal{L}(g(n)) \quad = \quad \overline{\text{YesComp}(n)}$$

$\Sigma_n \;=\; \{0, 1, \triangleright, \sqcup, \#, \hat{0}, \hat{1}, \ldots, \hat{q_n}\}$ where M_n has q_n states.

$n \in \text{EMPTY} \quad \Leftrightarrow \quad \overline{\text{YesComp}(n)} = \Sigma_n^\star \quad \Leftrightarrow \quad g(n) \in \Sigma^\star\text{CFL}$

Proof:

$$\overline{\text{YesComp}(n)} = U(n) \cup A(n) \cup D(n) \cup Z(n)$$

$$U(n) = \{w \in \Sigma^\star \mid w \text{ not in form } \text{ID}_0\# \cdots \#\text{ID}_t\}$$

$$A(n) = \{w \in \Sigma^\star \mid w \text{ doesn't start with initial ID of } M_n\}$$

$$D(n) = \{w \in \Sigma^\star \mid (\exists i)(\text{ID}_{i+1} \text{ doesn't follow from } \text{ID}_i\}$$

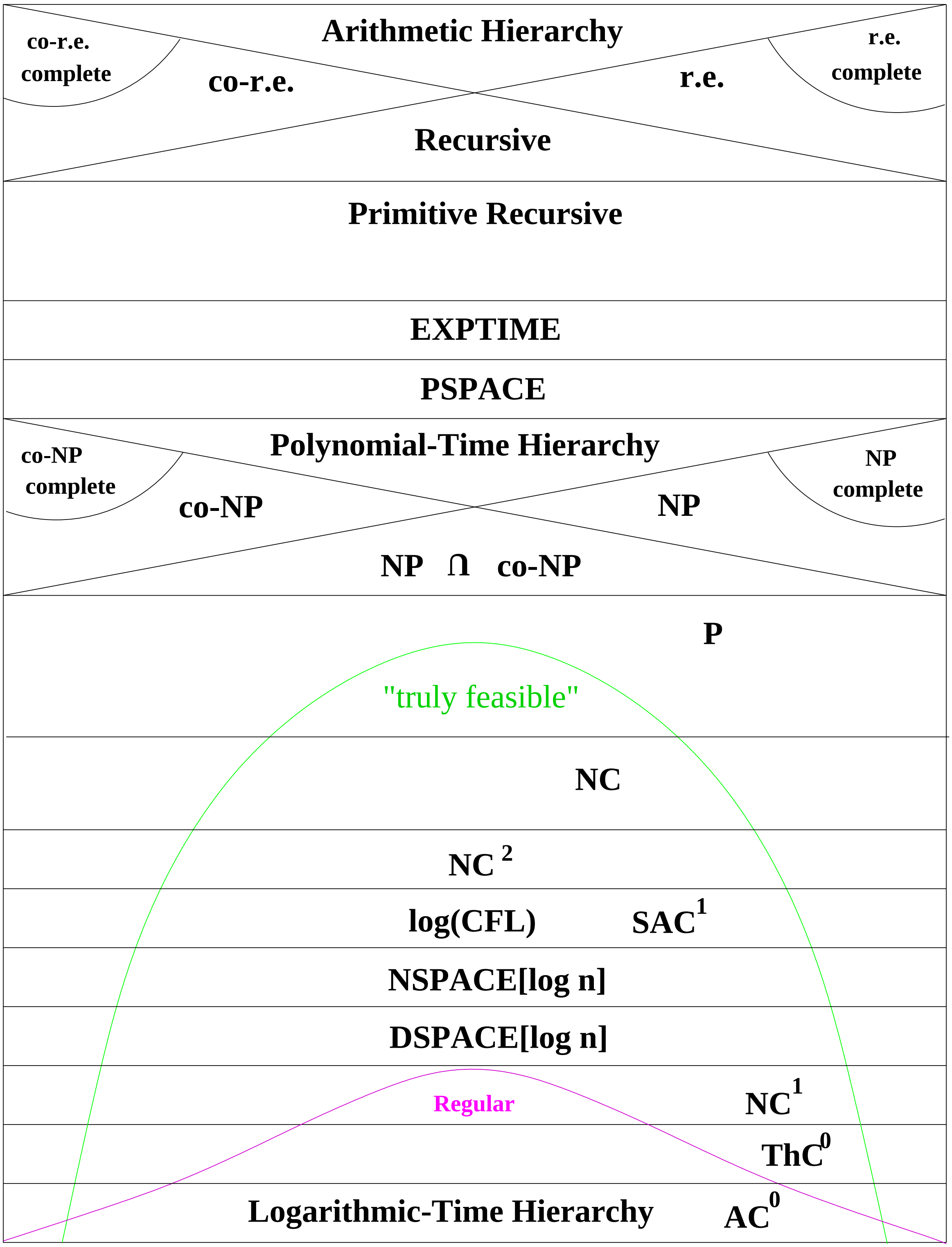
$$Z(n) = \{w \in \Sigma^\star \mid w \text{ doesn't end with } \hat{0} \triangleright 1 \sqcup\}$$

♠

Thus, g : EMPTY $\leq \Sigma^{\star}$CFL

$$
\begin{aligned}
n \in \text{EMPTY} \;&\Leftrightarrow\; \overline{\text{YesComp}(n)} = \Sigma_n^{\star} \\
&\Leftrightarrow\; g(n) \in \Sigma^{\star}\text{CFL}
\end{aligned}
$$

♠


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