Initial functions:

$\zeta() = 0$

$\sigma(x) = x + 1$

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

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

$C(h; g_1, \ldots, g_m)(x_1, \ldots, x_k) = h(g_1(x), \ldots, g_m(x))$

**Primitive Recursion:**  
$g : \mathbb{N}^k \to \mathbb{N}; \quad h : \mathbb{N}^{k+2} \to \mathbb{N}$:

$f(n, y_1, \ldots, y_k) = \mathcal{P}(g, h)(n, y_1, \ldots y_k), \text{ given by:}$

$f(0, y_1, \ldots y_k) = g(y_1, \ldots, y_k)$

$f(n + 1, y_1, \ldots y_k) = h(f(n, y_1, \ldots, y_k), n, y_1, \ldots, y_k)$

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

\begin{center}
\text{exercise}
\end{center}
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 \( \text{COMP}(n, x, c, y) \) mean \( M_n(x) = y \), and that \( c \) is \( M_n \)'s complete computation on input \( x \).

Then \( \text{COMP} \) is a Primitive Recursive predicate.

**Proof:** We will encode TM computations:

\[
c = \text{Seq}(\text{ID}_0, \text{ID}_1, \ldots, \text{ID}_t)
\]

Where each \( \text{ID}_i \) is a sequence number of tape-cell contents:

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

\( \text{COMP}(n, x, c, y) \equiv \)

\[
\text{START}(\text{Item}(c, 0), x) \land \text{END}(\text{Item}(c, \text{Length}(c) - 1), y) \land \\
(\forall i < \text{Length}(c))\text{NEXT}(n, \text{Item}(c, i), \text{Item}(c, i + 1))
\]

\( \spadesuit \)
Theorem 9.1 The following problems are decidable in polynomial time.

\[\begin{align*}
\text{EmptyNFA} & = \{ N \mid N \text{ is an NFA; } \mathcal{L}(N) = \emptyset \} \\
\Sigma^*\text{DFA} & = \{ D \mid D \text{ is a DFA; } \mathcal{L}(D) = \Sigma^* \} \\
\text{MemberNFA} & = \{ \langle N, w \rangle \mid N \text{ is an NFA; } w \in \mathcal{L}(N) \} \\
\text{EqualDFA} & = \{ \langle D_1, D_2 \rangle \mid D_1, D_2 \text{ DFAs; } \mathcal{L}(D_1) = \mathcal{L}(D_2) \} \\
\text{EmptyCFL} & = \{ G \mid G \text{ is a CFG; } \mathcal{L}(G) = \emptyset \} \\
\text{MemberCFL} & = \{ \langle G, w \rangle \mid G \text{ is a CFG; } w \in \mathcal{L}(G) \} 
\end{align*}\]
MemberCFL = \{ \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**: \( N \rightarrow AB, N \rightarrow a \).

2. **Input**: \( w = w_1w_2 \ldots w_n \); \( G \) with nonterminals \( S, A, B, \ldots \)

3. \( N_{ij} \equiv \begin{cases} 1 & \text{if } N \xrightarrow{*} w_i \cdots w_j \\ 0 & \text{otherwise} \end{cases} \)

4. **return** \((S_{1n})\)

\[
N_{i,i} = \text{if } ("N \rightarrow w_i" \in R) \text{ then } 1 \text{ else } 0
\]

\[
N_{i,j} = \bigvee_{"N \rightarrow AB" \in R} (\exists k)(i \leq k < j \land A_{i,k} \land B_{k+1,j})
\]
Theorem 9.2 The following problem is co-r.e.-complete:

\[ \Sigma^*\text{CFL} = \{ G \mid G \text{ is a CFG; } \mathcal{L}(G) = \Sigma^*_G \} \]

Proof: [J. Hartmanis, Neil’s advisor]

\[ \Sigma^*\text{CFL} \in \text{r.e.:} \]

Input: \( G \)
Define: \( \Sigma^*_G = \{ w_0, w_1, w_2, \ldots \} \)
1. for \( i := 0 \) to \( \infty \) {
2. \[ \text{if } w_i \notin \mathcal{L}(G), \text{ then return}(1) \]}

Clearly this returns 1 iff \( G \in \Sigma^*\text{CFL} \).
Proposition 9.3  *EMPTY* is co-r.e. complete, where,

\[ *EMPTY* = \{ n \mid W_n = \emptyset \} \]

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

---

Claim 9.4  *EMPTY* \(\leq \Sigma^*\text{CFL} \).

Corollary 9.5  \(\Sigma^*\text{CFL} \) is co-r.e. complete and thus not recursive.

How can we prove the Claim?

We need to define: \( g : \mathbb{N} \rightarrow \{0, 1\}^* \),

\[
n \in *EMPTY* \iff g(n) \in \Sigma^*\text{CFL} \\
(\forall x) M_n(x) \neq 1 \iff \mathcal{L}(g(n)) = \Sigma^*_n \\
M_n \text{ has no accepting computations} \iff \mathcal{L}(g(n)) = \Sigma^*_n
\]
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.

$$ID_0 \quad = \quad \hat{1} \triangleright w_1 w_2 \cdots w_r \square$$

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

$$ID_1 \quad = \quad \triangleright \hat{3} w_1 w_2 \cdots w_r \square$$
YesComp(\(n\)) =
\[
\left\{ ID_0 \# ID_1^R \# ID_2 \# ID_3^R \# \cdots \# ID_t \mid ID_0 \cdots ID_t \text{ accepting comp of } M_n \right\}
\]

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

Furthermore, there is a function \(g \in F(\mathbf{L})\), for all \(n\),
\[
\mathcal{L}(g(n)) = \overline{\text{YesComp}(n)}
\]

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

\(n \in \text{EMPTY} \iff \overline{\text{YesComp}(n)} = \Sigma_n^* \iff g(n) \in \Sigma^*\text{CFL}\)
Proof:

\[ \text{YesComp}(n) = U(n) \cup A(n) \cup D(n) \cup Z(n) \]

\[ U(n) = \{ w \in \Sigma^* \mid w \text{ not in form ID}_0\# \cdots \#\text{ID}_i \} \]

\[ A(n) = \{ w \in \Sigma^* \mid w \text{ doesn’t start with initial ID of } M_n \} \]

\[ D(n) = \{ w \in \Sigma^* \mid (\exists i)(\text{ID}_{i+1} \text{ doesn’t follow from ID}_i) \} \]

\[ Z(n) = \{ w \in \Sigma^* \mid w \text{ doesn’t end with } \hat{0} \gg 1 \ll \} \]

\[ \clubsuit \]
Thus, $g : \text{EMPTY} \leq \Sigma^* \text{CFL}$

\[ n \in \text{EMPTY} \iff \overline{\text{YesComp}(n)} = \Sigma^*_n \]
\[ \iff g(n) \in \Sigma^* \text{CFL} \]