

Theory of Computation, Fall 2021

Solutions

Q1 The grammar $G = (V, \Sigma, S, R)$ generates $\{ww : w \in \{a, b\}^*\}$ where

- $V = \{T, A, B, S_1, S, a, b\}$
- $\Sigma = \{a, b\}$
- R is the set of the following rules

$$\begin{aligned}
 S &\rightarrow TS_1 \\
 S_1 &\rightarrow AaS_1 \\
 S_1 &\rightarrow BbS_1 \\
 S_1 &\rightarrow e \\
 aA &\rightarrow Aa \\
 bA &\rightarrow Ab \\
 aB &\rightarrow Ba \\
 bB &\rightarrow Bb \\
 TA &\rightarrow aT \\
 TB &\rightarrow bT \\
 T &\rightarrow e
 \end{aligned}$$

Q2 Define $g : \mathcal{N} \times \mathcal{N} \rightarrow \mathcal{N}$ to be

$$g(m, n) = f(f(\dots f(n) \dots))$$

where the number of f 's in the definition is m . g can be write as

$$\begin{aligned}
 g(0, n) &= f(n) \\
 g(m+1, n) &= f(g(m, n)).
 \end{aligned}$$

Since f is primitive recursive, so is g . The function F can be seen as the composition of g with two identity functions. That is,

$$F(n) = g(id_{1,1}(n), id_{1,1}(n)).$$

Therefore, F is primitive recursive.

Q3 $factorial(n)$ can be written as

$$\begin{aligned}
 factorial(0) &= 1 \\
 factorial(n+1) &= (n+1) \cdot factorial(n).
 \end{aligned}$$

As a result, $factorial$ is primitive recursive.

Q4 Fix an arbitrary $k \geq 2$. For $i \in [1, k]$, define P_i as follows.

$$P_i(n_1, \dots, n_k) = \begin{cases} 1 & \text{if } n_i = \max\{n_1, \dots, n_k\} \\ 0 & \text{otherwise} \end{cases}$$

P_i is a primitive recursive predicate since P_i can be written as

$$P_i(n_1, \dots, n_k) = (n_i > n_1) \wedge (n_i > n_{i-1}) \wedge (n_i \geq n_{i+1}) \wedge \dots \wedge (n_i \geq n_k).$$

Now φ_k can be written as

$$\varphi_k(n_1, \dots, n_k) = P_1(n_1, \dots, n_k) \cdot n_1 + \dots + P_k(n_1, \dots, n_k) \cdot n_k.$$

As a result, φ_k is primitive recursive.