Problem 1.

Define a context-free grammar for the language L $= \{0^n1^m0^m1^n : n, m \in \mathbb{N}\}$.

Solution

Assumes $0 \in \mathbb{N}$.
G $= (\{S, X\}, \{0, 1\}, P, S)$ with productions P given by

$$S \to 0S1$$
$$S \to \epsilon$$
$$S \to X$$
$$X \to 1X0$$
$$X \to \epsilon$$

Here, S is used to create the string 0^n1^n. Once these have been created, S transfers control to X who creates 1^m0^m exactly in the middle of the string created by S. When put together we end up with $0^n1^m0^m1^n$. Note that 0^n1^n can be created by sending $S \to \epsilon$ and the string 1^m0^m can be created by immediately sending $S \to X$.

Problem 2.

Define a context-free grammar for the language L = $\{a^n b^m : n \leq 3m\}$.

Solution

Assumes $n, m \in \mathbb{N}$ and $0 \in \mathbb{N}$.
G = $(\{S\}, \{a, b\}, P, S)$ with productions P given by

$$S \to Sb$$
$$S \to aSb$$
$$S \to aaSb$$
$$S \to aaaSb$$
$$S \to \epsilon$$

My approach here is pretty simple, first note that we can have any number of a's that we want so long as we don't have more than 3 for every b at the end of the string. Thus we can say that for every b, there are no more than $3a$'s present in the string. This is exactly what my grammar shows, along with the requirement that all a's precede all b's. Note that the empty string is of course allowed, with $n = m = 0$.

Problem 3.

Define a CFG that generates the following language over $\{t, f, \wedge, \vee, \neg, (,), =\}$:
L = $\{w = x : w$ is a logical expression over $\{t, f\}, x \in \{t, f\}$ and x is the truth value of $w\}$

Solution

TODO