A. My regular expression is

RR^* `Initialize` R^* `\(` `&`

(without any white space), where R is the regular expression

$(\texttt{a} \cup \cdots \cup \texttt{z} \cup \texttt{A} \cup \cdots \cup \texttt{Z} \cup \texttt{0} \cup \cdots \cup \texttt{9})$,

which matches a single alpha-numeric character. [By the way, the three examples come verbatim from CS 311: Graphics. Also, I meant to stipulate that the first function name character should be a lower-case letter and the first function name character after the `Initialize` (if any) should be an upper-case letter. I forgot. But you might try to solve that harder version of the problem.]

B.A. Let $w = 0^p 110^p 1$. [Notice that $w = 0^p 1^p 0^p$ does not work!]

B.B. Let $w = 0^p 1^p$.

B.C. Let w be the string $1^p \wedge 1 = 1^p$.

C. In the following context-free grammar, the start variable Fun generates function calls. The variable Non generates non-empty argument lists. The variable Arg generates arguments. The variable Id generates identifiers. The variable Let generates letters.

$$
\begin{aligned}
\text{Fun} &\to \text{Id()} \mid \text{Id(Non)} \\
\text{Non} &\to \text{Arg} \mid \text{Arg,Non} \\
\text{Arg} &\to \text{Id} \mid \text{Fun} \\
\text{Id} &\to \text{Let} \mid \text{LetId} \\
\text{Let} &\to \texttt{a} \mid \texttt{b} \mid \ldots \mid \texttt{z} \mid \texttt{A} \mid \texttt{B} \mid \ldots \mid \texttt{Z}
\end{aligned}
$$

[By the way, it would be more realistic to include upper-case letters and digits, but not to let identifiers begin with digits. Think of how to do that.]

D.A. TRUE. [There are only finitely many strings of length less than or equal to b, so $A_{\leq b}$ is finite. As has been mentioned in class and proved on earlier exams, any finite language is regular.]

D.B. TRUE. [This is one reason why the alphabet Σ tends to be boring.]

D.C. FALSE. [We mentioned this in class. Here's a quick proof. The languages $\{a^m b^m c^k\}$ and $\{a^k b^m c^m\}$ are context-free, but their intersection $\{a^m b^m c^m\}$ is not. So the class of context-free languages is not closed under intersection. But it's closed under union. So it can't be closed under complementation.]

D.D. TRUE. [One can build a DFA for the reversal of A. It processes a string, one column at a time, keeping track of whether it is carrying 0, 1, or 2.]

D.E. TRUE. [We used this fact in one of our proof sketches in class. The PDA N can be modified to produce P as follows. Wherever N has a transition $a, t \to u$ that both pops and pushes, replace it with two transitions $a, t \to \epsilon$ and $\epsilon, \epsilon \to u$ passing through a new state. Wherever N has a transition $a, \epsilon \to \epsilon$ that neither pops nor pushes, replace it with two transitions $a, \epsilon \to t$ and $\epsilon, t \to \epsilon$ passing through a new state.]