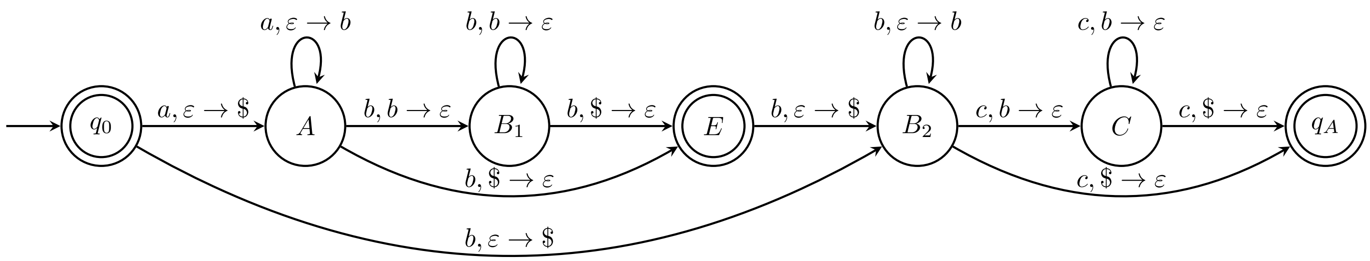
Dustin Newman
CS 181, Discussion 1C
Campbell, Mathur

Assignment Seven

1

The DPDA for this language is $(\{q_0, A, B_1, E, B_2, C, q_A\}, \{a, b, c\}, \{\$, b\}, q_0, \{q_0, E, q_A\})$ where the transition function δ can be seen below:



My DPDA works by recognizing that any string in the language can be partitioned into two substrings: one of $a^i b^i$ and one of $b^j c^j$. Then, $n = i + j$ becomes almost trivial to prove. We only need to handle the first substring if the input string begins with an a. If so, we transition to the DPDA that takes care of that, marking the first a with a \$ from the stack alphabet. We only return from that machine if we have a valid $a^i b^i$ string to state E. Since $j = 0$ is allowed, we accept on E. If we see another b, we know we also have a $b^j c^j$ substring and so we handle that similarly in the second subset of the machine.

2

$$
\begin{aligned}
\underline{v} + v \times v + v &\rightarrowtail F + v \times v + v \\
\underline{F} + v \times v + v &\rightarrowtail T + v \times v + v \\
\underline{T} + v \times v + v &\rightarrowtail E + v \times v + v \\
E + \underline{v} \times v + v &\rightarrowtail E + F \times v + v \\
E + \underline{F} \times v + v &\rightarrowtail E + T \times v + v \\
E + T \times \underline{v} + v &\rightarrowtail E + T \times F + v \\
E + \underline{T \times F} + v &\rightarrowtail E + T + v \\
\underline{E + T} + v &\rightarrowtail E + v \\
E + v &\rightarrowtail E + F \\
E + \underline{F} &\rightarrowtail E + T \\
\underline{E + T} &\rightarrowtail E
\end{aligned}
$$

3

3.1

$$
\begin{aligned}
\underline{()}()()() &\rightarrowtail S()()() \\
S\underline{()}()() &\rightarrowtail SS()() \\
\underline{SS}()() &\rightarrowtail S()() \\
S\underline{()}() &\rightarrowtail SS() \\
SS\underline{()} &\rightarrowtail SSS \\
S\underline{SS} &\rightarrowtail SS \\
\underline{SS} &\rightarrowtail S
\end{aligned}
$$

3.2

$$
\begin{aligned}
\underline{()}()()() &\rightarrowtail S()()() \\
S\underline{()}()() &\rightarrowtail SS()() \\
SS\underline{()}() &\rightarrowtail SSS() \\
SSS\underline{()} &\rightarrowtail SSSS \\
SS\underline{SS} &\rightarrowtail SSS \\
S\underline{SS} &\rightarrowtail SS \\
\underline{SS} &\rightarrowtail S
\end{aligned}
$$