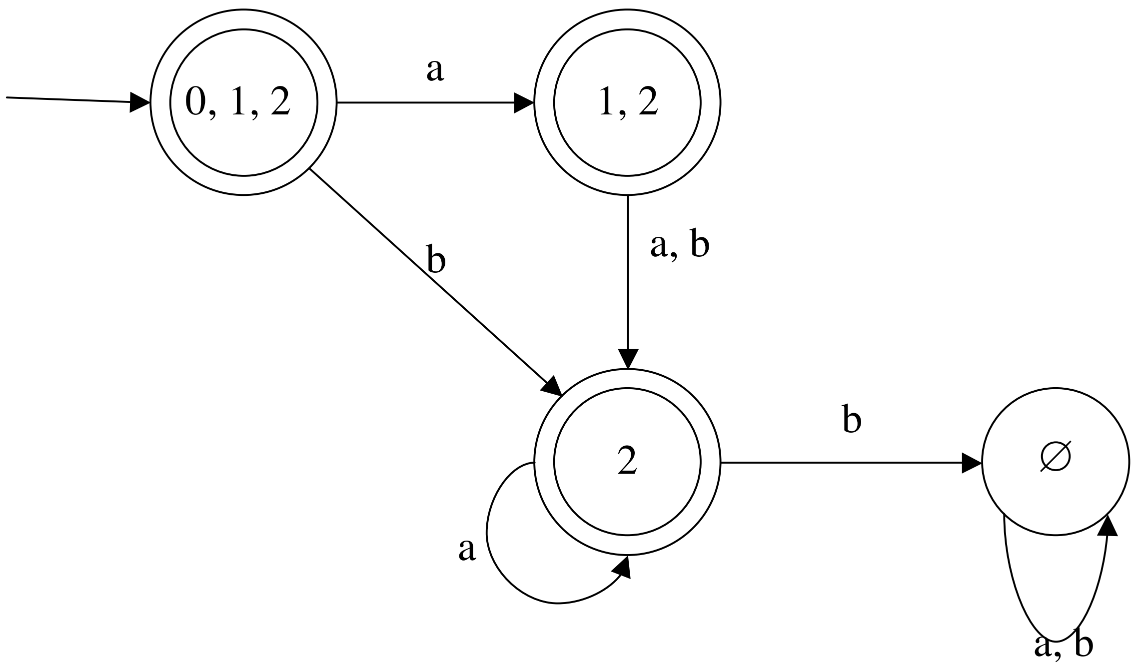
CS 81 Optional Assignment 11 for Wed., May 4 Solutions

Hein p 681-683:

3.c, d (using any method you wish)

Result of transforming to a DFA using subset construction:



Hein p 693-695:

1.f Find a regular grammar for a*bc* + ac

S → aT | cU
T → aT | bV
U → Λ
V→ cV |Λ

3.c Find a regular grammar for all strings of a's and b's that contain the substring aba.

The regular expression is (a + b)* aba (a + b)*.

S → aS | bS | aU
U → bV
V → aW
W → aW | bW | Λ

7.d Use the pumping lemma to show that this language is not regular:

$L = \{a^n b^k \mid n, k \in N \text{ and } n > k\}$

Suppose L is regular. Let m be the number that exists according to the pumping lemma, such that:

If $z \in L$ and $|z| \geq m$, then for some u, v, w $z = uvw$ where $|uv| \leq n$ $v \neq \Lambda$ and $(\forall m \in N)\ uv^m w \in L$.

Let $z = a^m b^{m+1} \in L$ and let uvw be as above. Clearly v cannot consist of a mixture of a's and b's, otherwise $uv^2w \notin L$.

Moreover, v cannot consist of only a's, otherwise $uv^2w \notin L$.

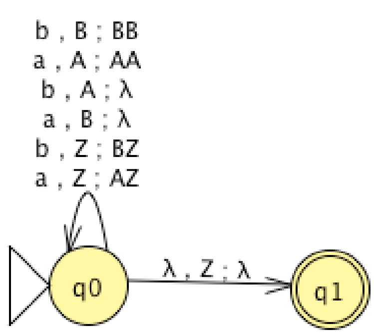
Hence v must consist of only b's. But then $uv^0w \notin L$.

Hein p 715-716:

1.b Find a pda for all strings over a, b with the same number of a's and b's.
I used Jflap for my PDA's. The notation is slightly different from Hein's.
The control states are shown as nodes. Each arc is labeled

$\sigma, \rho; \gamma$ where $\sigma \in \Sigma$ is the symbol read,
$\rho \in \Gamma$ is the symbol read and removed from the top of the statck,
and $\gamma \in \Gamma^*$ is the string of symbols written to the the stack, with the leftmost symbol of γ at the top.

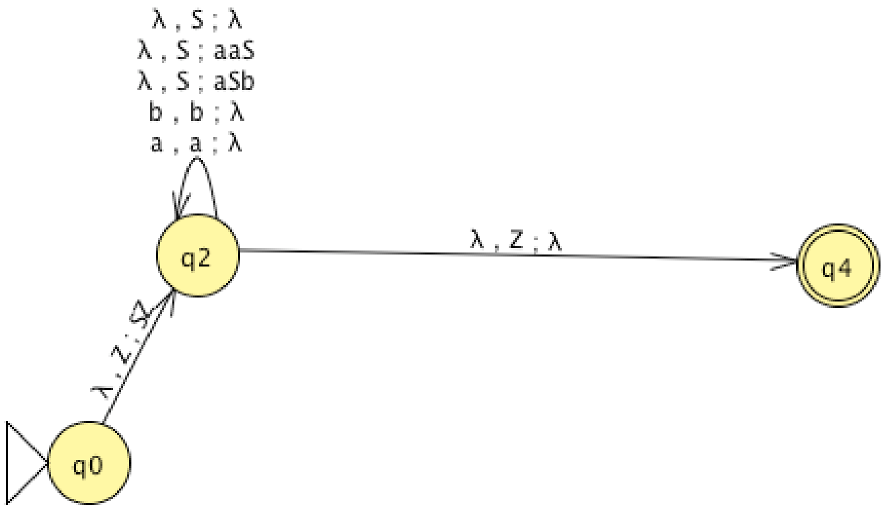
λ denotes the empty string.
These pda's accept by final state, at Z is the initial stack symbol.



The stack in this case maintains the difference between the number of a's and b's read, on top of the initial stack symbol Z. If there were more a's, then the stack will contain the difference in A's, while if there were more b's, then the stack will contain the difference in B's. When all input is read, a transition to the accepting state can take place iff the initial stack symbol Z is at the top.

7.b Construct a pda that accepts the language generated by this grammar:

$S \rightarrow \Lambda \mid aSb \mid aaS$



We use the "produce-match" technique. The start symbol of the grammar is pushed onto the stack. Then for each production, a non-deterministic move can be made when the left-hand side of the production is on top of the stack. The right-hand side is then pushed onto the stack. When a terminal symbol is on top of the stack, the corresponding symbol can be read from the input. Whenever the stack contains only the start symbol Z, this indicates that all of the input has been matched against a string produced in the language. In that case a transition to the accepting state can be made.

Hein p 755-716:

2.b Construct a Chomsky normal form for this grammar:

$S \rightarrow abC \mid babaS \mid de$
$C \rightarrow aCa \mid b$

Step 1: No change
Step 2: No change

Step 3: Replace terminals on the right-hand side with auxiliaries.

$S \rightarrow ABC \mid BABAS \mid DE$
$A \rightarrow a$
$B \rightarrow b$
$C \rightarrow ACA \mid B$
$D \rightarrow d$
$E \rightarrow e$

Step 4: Add new auxiliaries to account for right-hand sides longer than 2.

$S \to AF \mid BH \mid DE$
$A \to a$
$B \to b$
$C \to AG \mid B$
$D \to d$
$E \to e$
$F \to BC$
$G \to CA$
$H \to AI$
$I \to BJ$
$J \to AS$

4.b

Suppose that $L = \{a^i b^j c^k \mid 0 < i < j < k\}$ is context free.
Let m be the m that exists according to the pumping lemma, so that for any
$(\forall z \in L)\ |z| \geq m \ \to (\exists\ u, v, w, x, y)\ z = uvwxy \wedge |vx| \geq 1 \wedge |vwx| \leq m \ \wedge$
$(\forall k \in N)\ uv^kwx^ky \in L$.

Let $z = a^m\ b^{m+1}\ c^{m+2} \in L$. Clearly $|z| \geq m$, so
let u, v, w, x, y be the strings that exist such that
$z = uvwxy \wedge |vx| \geq 1 \wedge |vwx| \leq m \ \wedge (\forall k \in N)\ uv^kwx^ky \in L$.

Because $|vwx| \leq m$, $vwx \in \{a\}^*\{b\}^*$ or $vwx \in \{b\}^*\{c\}^*$ based on the number of symbols of each type in z.

If $vwx \in \{a\}^*\{b\}^*$, we get a contradiction, since in pumping for large enough k, the number of c's is exceeded by the number of a's and b's.

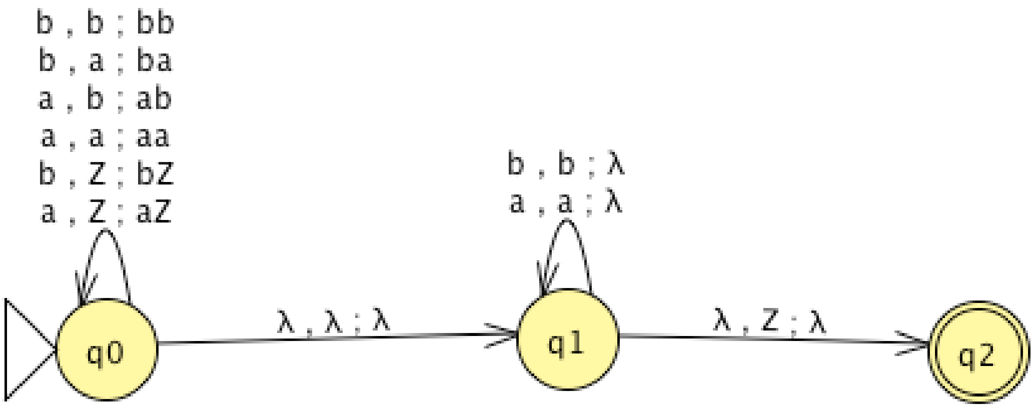
Thus $vwx \in \{b\}^*\{c\}^*$.

If $vwx \in \{b\}^*$, we get a contradiction setting $k = 0$, for then uwy too few b's compared to the number of a's.

If $vwx \in \{c\}^*$, then setting $k = 0$, we get too few c's compared to the number of b's.

Therefore $vwx \in \{b\}\{b\}^*\{c\}\{c\}^*$. But then setting $k = 0$, we get too few b's or c's compared to the number of a's.

Construct a pda that accepts the language $\{ww^R \mid w \in \{a, b\}^*\}$.



There are two modes, represented by q_0 and q_1 respectively. In mode q_0, symbols read from the input are accumulated on the stack, representing the reverse of the string that has been read. Then the pda "guesses" it is time to start matching the remainder of the input against the reverse of the first portion, it shifts to mode q_1, where it matches the remaining input symbols with the corresponding stack symbols. When the stack is empty except for the initial stack symbol Z, the machine guesses that all the input is matched and moves to the accepting state q_2. Note that there can be a correct sequences of such guesses iff the input is of the form ww^R.