Candidate Number: 2031B
Paper 2
Question 10

a.

i. abc is not accepted because there are are no transitions labelled “c”, and so after the final input “c” the system can’t be in any of states D, E, or F.
bba is accepted via the route:
D -ε-> E -b-> D - ε -> E -b-> D -a-> E - ε -> F

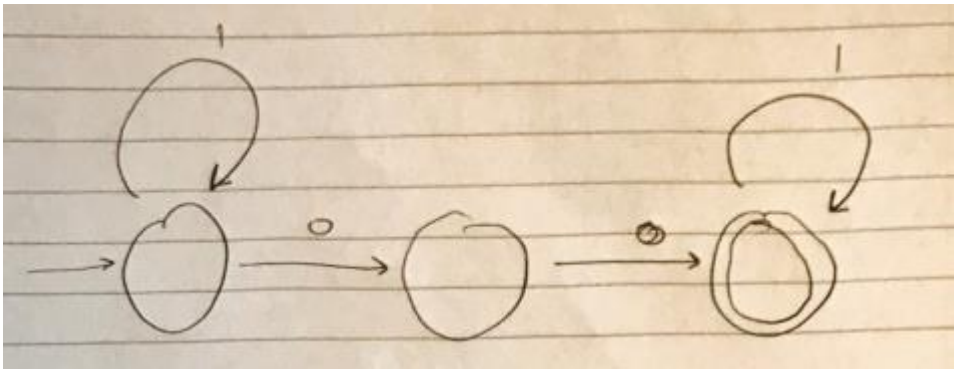
ii.

State	Input	Next state
{}	a	{}
{}	b	{}
{}	c	{}
{D}	a	{D,E,F}
{D}	b	{E,F}
{D}	c	{}
{D,E}	a	{D,E,F}
{D,E}	b	{D,E,F}
{D,E}	c	{}
{D,E,F}	a	{D,E,F}
{D,E,F}	b	{D,E,F}
{D,E,F}	c	{}
{D,F}	a	{D,E,F}
{D,F}	b	{E,F}
{D,F}	c	{}
{E}	a	{D}
{E}	b	{D,E,F}
{E}	c	{}
{E,F}	a	{D,E,F}
{E,F}	b	{D,E,F}
{E,F}	c	{}
{F}	a	{D,E,F}
{F}	b	{F}
{F}	c	{}

The starting state is {D, E, F}. The accepting states are {D, E, F}, {D, F}, {E, F}, and {F}

iii. (a|b)*
This is because any string of characters which does not contain a c will end up keeping the system in state {D, E, F} which is a finishing state. Therefore any combination of a’s and b’s is accepted.

b.

i. 

Candidate Number: 2031B
Paper 2
Question 10

ii. L1 accepts strings of the form 1111…1111001111…1111. Let's say j 1's followed by 00 followed by k 1's. Adding 1 to this will give 1111…1111010000…0000 which is j 1's followed by 01 followed by k 0's. If j is 0, the leading 0 is stripped, giving 1 followed by k 0's. Therefore the regex is

((11*0)| ε)10*

Which can be simplified to
1(1*01)0*