Problem 1

p	$\neg p$	$(\neg p \rightarrow p)$	$(p \rightarrow (\neg p \rightarrow p))$
T	F	T	T
F	T	F	T

Problem 2

If we want to convert $((\forall x)F(x) \leftrightarrow \neg(\exists x)\neg F(x))$, then we can convert it to it's conjunctive/disjunctive form and examine its truth table. Such an equivalent formula would look like this:

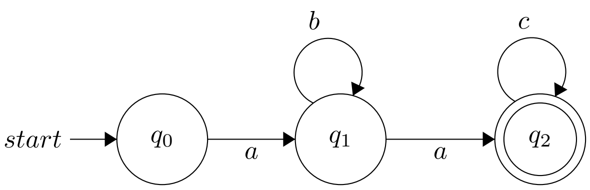
$$(F(a) \wedge F(b)) \leftrightarrow \neg(\neg F(a) \vee \neg F(b))$$

We can show it is tautological by calculating its truth table:

$F(A)$	$F(b)$	$(F(a) \wedge F(b)$	$\neg F(a)$	$\neg F(b)$	$(\neg F(a) \vee \neg F(b))$	$\neg(\neg F(a) \vee \neg F(b))$	$(F(a) \wedge F(b)) \leftrightarrow \neg(\neg F(a) \vee \neg F(b))$
T	T	T	F	F	F	T	T
T	F	F	F	T	T	F	T
F	T	F	T	F	T	F	T
F	F	F	T	T	T	F	T

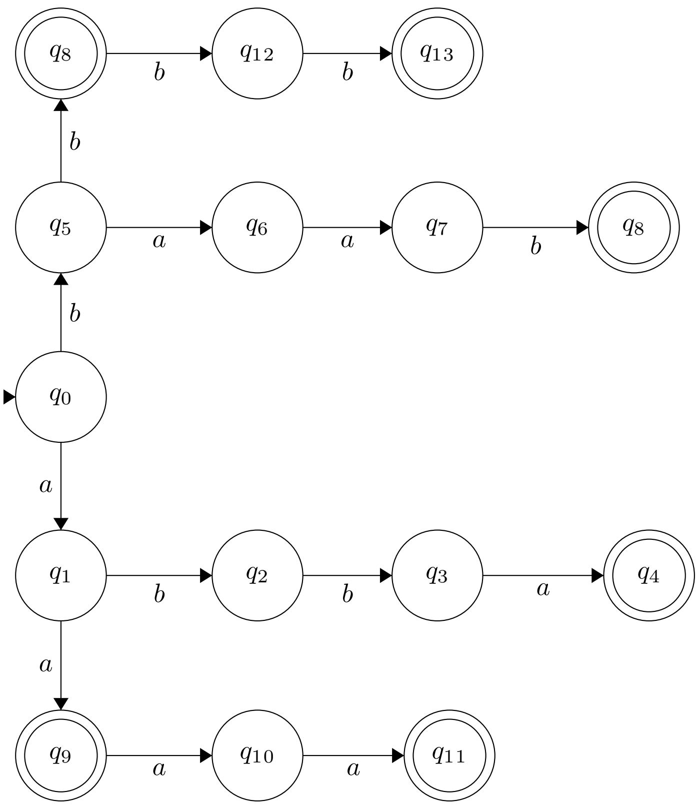
This truth table shows that the formula is tautological. We also could have shown this by using DeMorgan's Law to convert $\neg(\neg F(a) \vee \neg F(b))$ to it's equivalent $(F(a) \wedge F(b))$. Then since both sides of the bi-implication were the same, the formula would have been trivially tautological.

Problem 3



Problem 4

There are a finite number of possible palindromes under $\Sigma = \{a, b\}$. They are: aa, bb, abba, baab, aaaa, bbbb. These can be matched with the following FSM:

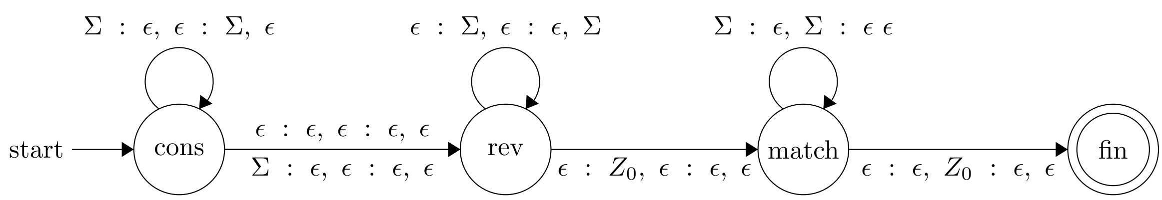


Problem 5

A two-stack pushdown automata could be used to recognize the repeating language $L = ww$. The problem with matching the repeating language in a standard pushdown automata is the stack has a last-in, first-out behavior. With the addition of a second stack, we can reverse the order of the first stack, which will allow us to match the symbols in order. This can be achieved by non-deterministically assuming that we have reached the middle of the string and reversing the stack and proceeding to match the symbols in order.

This non-determinism can be achieved by running multiple automatas in parallel, each with their own pair of stacks.

Below is an attempt at designing a machine that might be able to accept the language $L = ww$. The transitions in the machine are labeled like so: $X : G_1, G_2 : P_1, P_2$ where X is a symbol in Σ, G_n refers to popping the first symbol off of the stack n, and P_n refers to pushing a symbol onto the stack n.



States:

- cons: Consumes symbols in the alphabet Σ and pushes those onto the first stack
- rev: Consumes symbols from the first stack and pushes them onto the second stack (reverses stack)
- match: Consumes a symbol from the input and matches it to the symbol on the top of the second stack.
- fin: Final accepting state

The transition from cons to rev is a pair of non-deterministic transitions that have the effect of "guessing" when we have reached the middle of our input string.