The idea of the proof is as follows:

• L(R) is a subset of L(S) iff L(R) intersected with the complement of L(S), $L(S)^C$ is the empty set.

• Use the Theorem 4.4 (or its equivalent for regular expressions) to prove that

the condition "L(R) intersection $L(S)^C$ = empty set" is decidable.

Proof:

The following Turing machine T decides the language A.

"On input string *w*,

1. Check that *w* encodes a pair $<R,S>$ where R and S are regular expressions

If it does not, then reject *w*.

2. Translate R into an equivalent DFA DR and S into an equivalent DFA DS.

3. Build a DFA DS^C that accepts the language $L(DS)^C$.

4. Build a DFA D that is the intersection of DR with DS^C that is, L(D) = L(DR) intersection $L(DS^C)$.

5. Now, run the Turing machine T with input < D > to determine if L(D) is empty.

If T accepts < D > (which means that L (D) is empty), then accept *w*.

If T rejects < D > (which means that L (D) is NOT empty), then reject *w*."

Now prove that the Turing machine T decides the language A.

• First of all, prove that T halts on all inputs.

• Let *w* be any word. There are two possibilities.

• Either *w* codifies a pair of regular expressions or it doesn't. Checking this will take a finite amount of time.

• If x doesn't codify a pair of regular expressions, then T stops rejecting *w*.

• Otherwise, assume *w* codifies a pair < R, S >.

• Constructing the DFAs described in steps 2, 3, and 4 above is possible (regular languages are closed under intersection and complementation) and takes finite time.

Hence, in all cases T stops on any input *w* after a finite amount of time.

Hence, it is proved that language A is decidable.