Consider the language,

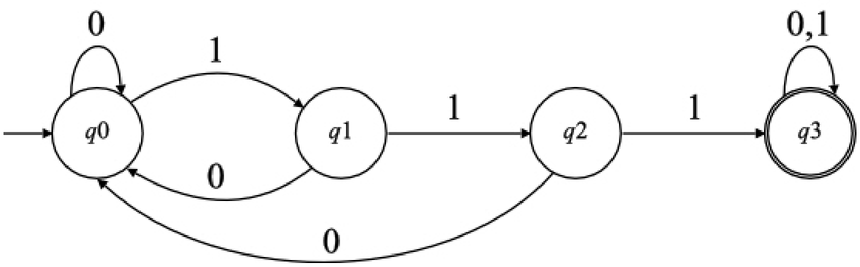
$$A=\left\{\begin{array}{l}\langle R\rangle \mid R \text{ is a regular expression describing a language which contain} \\ \text{at least one string } w \text{ containing } 111 \text{ as its substring}\end{array}\right\}$$

The decidability of the language *A* is proved as follows:

• Define a language *S* such that $S=\{w\in\Sigma^* \mid w \text{ consists } 111 \text{ as a substring}\}$.

• The regular expression (RE) for the language *S* is $(0\cup 1)^*111(0\cup 1)^*$.

• The DFA D_S for the language *S* is shown below:



• Now think about some RE *R* on input alphabet Σ.

• If $S\cap L(R)\neq\phi$, then *R* produces a string containing 111 as a substring. Thus, $\langle R\rangle\in A$.

• Similarly, if $S\cap L(R)\neq\phi$ then R produces a string that does not contain 111. Thus, $\langle R\rangle$ does not belongs to *A*.

• Since $L(R)$ is described by regular language, $L(R)$ is a regular language. Both *S* and $L(R)$ are regular languages.

• $S\cap L(R)$ is regular because, regular languages are closed under intersection. Thus, $S\cap L(R)$ has some DFA $D_{S\cap L(R)}$.

• Theorem 4.4 shows that $E_{DFA}=\{\langle K\rangle \mid K \text{ is a DFA with } L(K)\neq\phi\}$ is decidable. Thus, there exists a Turing Machine *TM* which determines E_{DFA}.

• Relate TM *T* to $D_{S\cap L(R)}$ to determine if $L(R)\cap S\neq\phi$.

Summarization of the above discussion contributes the subsequent Turing machine *M* to decide *A*:

M = "On input $\langle R\rangle$, where R is a regular expression:

• Transform *R* into a DFA D_R by means of the algorithm in the proof of Kleene's Theorem.

• Build a DFA $D_{S\cap L(R)}$ for the language $S\cap L(R)$ from the DFAs D_S and D_R.

• Run TM *T* that decides E_{DFA} on input $\langle D_{S\cap L(R)}\rangle$.

• If *T* accepts, reject. If *T* rejects, accept.

The Turing machine *T* decides *A* . Therefore, the language *A* is decidable.