Definition of GNFA I

- Here we give the formal definition of generalized NFA
- Between any two states: a regular expression
- \((Q, \Sigma, \delta, q_{\text{start}}, q_{\text{accept}})\)
- Single accept state. No longer a set \(F\)
- The \(\delta\) function:

\[
(Q - \{q_{\text{accept}}\}) \times (Q - \{q_{\text{start}}\}) \rightarrow R
\]

\(R\): all regular expressions over \(\Sigma\)

- DFA \(\rightarrow\) GNFA
Definition of GNFA II

Two new states: $q_{\text{start}}, q_{\text{accept}}$

$q_{\text{start}} \rightarrow q_0$ with $\epsilon$

any $q \in F \rightarrow q_{\text{accept}}$ with $\epsilon$

- In the definition, between any two states there is an expression

But what if in the graph two states are not connected?

$\emptyset \in R$ so if no connection, we simply consider $\emptyset$ as the expression between two states
• $q_{\text{rip}}$ is the state being removed

• The new regular expression between $q_i$ and $q_j$ is
In our example
3-state DFA → 5-state GNFA → 4-state · · · → 2-state GNFA → regular expression

In the procedure any any \((i, j)\) related to \(q_{rip}\) considered

Algorithm: convert\((G)\)
  1. \(k: \# \) of \(G\)
  2. If \(k = 2\)
If $k > 2$, choose any $q_{rip} \in Q \setminus \{q_s, q_a\}$ for removal

$Q' = Q - \{q_{rip}\}$

$\forall q_i \in Q' - \{q_{accept}\}, q_j \in Q' - \{q_{start}\}$

$\delta'(q_i, q_j) = R_1 R_2^* R_3 \cup R_4$,

where

$R_1 = \delta(q_i, q_{rip})$, $R_2 = \delta(q_{rip}, q_{rip})$,

$R_3 = \delta(q_{rip}, q_j)$, $R_4 = \delta(q_i, q_j)$
Run convert($G'$), where

$$G' = (Q', \Sigma, \delta', q_s, q_a)$$

- You can see we have a recursive setting
  The process stops when $k = 2$
- Why in the textbook we modify DFA to GNFA?
  Is it ok to use NFA?
  Seems ok??