Kleene star:
- add a new starting state which is also final
- add jumps from new starting state to old starting state
- add jumps from final states to old starting state

\[ a^* = \{ \lambda, a, aa, \ldots \} \]
\[ (a, b)^* = \{ \lambda, a, b, aa, ab, ba, bb, \ldots \} \]
09/17/18

\[ a \cup b^* \]

\[ L = \{ \Lambda, a, b, bb, bbb, \ldots \} = \{ a \} \cup \{ \Lambda, b, bb, bbb, \ldots \} = a \cup b^* \]
$ab \cup (ab)^*$
ab ∪ (ab)^* = \{ab, ∅, abab, ababab, \ldots\}

L = \{\∅, ab, abab, ababab, \ldots\} = ab ∪ ab^*
Regular expression:

\( a, b \) mean \{a, b\}...

\( L_1L_2 \) means composition

\( L_1 \cup L_2 \) union

\( L^* \) Kleene star

Regular expression = anything obtained from letters by using composition, union, and Kleene star

\[ \text{Regular expression} \rightarrow \text{Finite Automaton} \]

\[ \text{RL} \rightarrow \text{NDA} \rightarrow \text{DA} \rightarrow \text{deterministic automata} \]

Regular language \( \rightarrow \) non-deterministic automata
We have a deterministic finite automaton.

- We want to describe a regular expression that covers exactly all the words accepted by this automaton and only these words.

\[ L = \text{even binary integers ending in 0}. \]

\[ \text{Step 1: We add new starting state, with jump to old starting state,} \]

\[ \text{Step 2: We add new final state, with jumps from old final states} \]
\[ R_{ij} = R_{ij} \cup (R_{ik} R_{kk} R_{kj}) \]

\[ K = \text{start} \]
\[ R'_{ij} = R_{ij} \cdot U \left( R_{ik} R_{kj}^* R_{kj} \right) \]

\[ R'_{ns,s} = R_{ns,s} \cdot U \left( R_{ns, sta} R_{sta}^* R_{sta} \right) \]

\[ i = ns \]

\[ j = s \rightarrow k = sta \]

\[ (a+b)^2 = a^2 + b^2 + 2ab \]

\[ a = 3 \]

\[ (3+y)^2 = 3^2 + 4^2 + 2 \cdot 3 \cdot y \]

\[ b = 4 \]

\[ R'_{ns,s} = \phi \cdot U \left( \Lambda \phi^* \phi \right) \]

\[ \Lambda \Lambda \phi = 0 \]

\[ A^* = \{ \Lambda, (a \in A), a, a_2, (a_i \in A), a_1 a_2 a_3 (a_i \in A), \ldots \} \]

\[ \Lambda \Lambda = A \]

\[ A U \phi = A \]

\[ R'_{ns,s} = 0 \]
\[ R'_{ij} = R_{ij} \cup (R_{ik} R_{k\kappa}^* R_{\kappa j}) \]

\[ i = n_s, \quad j = n_f, \quad \kappa = \text{start} \]

\[ R'_{ns, n_f} = R_{ns, n_f} \cup (R_{ns, \text{start}} R_{\text{start}, n_f}^* R_{\text{start}, n_f}) \]

\[ R'_{ns, n_f} = \emptyset \cup (\emptyset^* \emptyset) = \emptyset \]

\[ A \phi = \emptyset \]

\[ R'_{ns, i} = R_{ns, i} \cup (R_{ns, \text{start}} R_{\text{start}, i}^* R_{\text{start}, i}) \]

\[ \emptyset \cup 1 = 1 \]