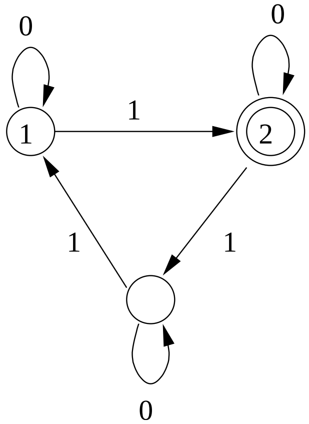
An example of constructing a regular expression from a DFA.

Recall the construction from class. R^k_{ij} is a regular expression that describes the set of strings that takes the machine from state i to state j using only states $1..k$ as intermediate states. We construct these regular expressions as follows.

$$
\begin{aligned}
R^0_{ii} &= \left\{ \begin{array}{ll} \varepsilon & \text{if there is no edge from } i \text{ to } i \\ \varepsilon \cup a_1 \cup a_2 \cup \cdots \cup a_k & \text{if there is an edge from } i \text{ to } i \text{ labelled by characters } a_1, \ldots, a_k \end{array} \right\} \\
\text{For } i \neq j, R^0_{ij} &= \left\{ \begin{array}{ll} \emptyset & \text{if there is no edge from } i \text{ to } j \\ a_1 \cup a_2 \cup \cdots \cup a_k & \text{if there is an edge from } i \text{ to } j \text{ labelled by characters } a_1, \ldots, a_k \end{array} \right\} \\
\text{For } k > 0, R^k_{ij} &= (R^{k-1}_{ij}) \cup (R^{k-1}_{ik})(R^{k-1}_{kk})^*(R^{k-1}_{kj})
\end{aligned}
$$

Example: Consider the following DFA.



Here are the regular expressions R^0_{ij}:

	$j=1$	$j=2$	$j=3$
$i=1$	$0 \cup \varepsilon$	1	$\emptyset$
$i=2$	$\emptyset$	$0 \cup \varepsilon$	1
$i=3$	1	$\emptyset$	$0 \cup \varepsilon$

Here are the regular expressions R^1_{ij}.

	$j=1$	$j=2$	$j=3$
$i=1$	$0 \cup \varepsilon \cup (0 \cup \varepsilon)(0 \cup \varepsilon)^*(0 \cup \varepsilon)$ $\equiv 0^*$	$1 \cup (0 \cup \varepsilon)(0 \cup \varepsilon)^*1$ $\equiv 0^*1$	$\emptyset \cup (0 \cup \varepsilon)(0 \cup \varepsilon)^*\emptyset$ $\equiv \emptyset$
$i=2$	$\emptyset \cup \emptyset(0 \cup \varepsilon)^*(0 \cup \varepsilon)$ $\equiv \emptyset$	$0 \cup \varepsilon \cup \emptyset(0 \cup \varepsilon)^*1$ $\equiv 0 \cup \varepsilon$	$1 \cup \emptyset(0 \cup \varepsilon)^*\emptyset$ $\equiv 1$
$i=3$	$1 \cup 1(0 \cup \varepsilon)^*(0 \cup \varepsilon)$ $\equiv 10^*$	$\emptyset \cup 1(0 \cup \varepsilon)^*1$ $\equiv 10^*1$	$0 \cup \varepsilon \cup 1(0 \cup \varepsilon)^*\emptyset$ $\equiv 0 \cup \varepsilon$

In the above table, I have shown simplified versions of each regular expression. (The real algorithm would not do this, it would just churn away with the big regular expressions, and they would grow larger and larger, but that would be too much to type.) In order to make these simplifications, I used some identities for regular expressions (e.g., $E\emptyset \equiv \emptyset, E \cup \emptyset \equiv E, (E \cup \varepsilon)^* \equiv E^*, (E \cup \varepsilon)E^* \equiv E^*, \ldots$). It's a good exercise to prove that these equivalences hold. (Two regular expressions are equivalent if they represent the same language.)

Here are the regular expressions R^2_{ij}.

	$j=1$	$j=2$	$j=3$
$i=1$	$0^* \cup 0^*1(0\cup\varepsilon)^*\emptyset$ $\equiv 0^*$	$0^*1 \cup 0^*1(0\cup\varepsilon)^*(0\cup\varepsilon)$ $\equiv 0^*10^*$	$\emptyset \cup 0^*1(0\cup\varepsilon)^*1$ $\equiv 0^*10^*1$
$i=2$	$\emptyset \cup (0\cup\varepsilon)(0\cup\varepsilon)^*\emptyset$ $\equiv \emptyset$	$0\cup\varepsilon\cup(0\cup\varepsilon)(0\cup\varepsilon)^*(0\cup\varepsilon)$ $\equiv 0^*$	$1\cup(0\cup\varepsilon)(0\cup\varepsilon)^*1$ $\equiv 0^*1$
$i=3$	$10^* \cup 10^*1(0\cup\varepsilon)^*\emptyset$ $\equiv 10^*$	$10^*1 \cup 10^*1(0\cup\varepsilon)^*(0\cup\varepsilon)$ $\equiv 10^*10^*$	$(0\cup\varepsilon)\cup 10^*1(0\cup\varepsilon)^*1$ $\equiv 0\cup\varepsilon\cup 10^*10^*1$

Finally, instead of writing out the whole table for R^3_{ij}, we only need R^3_{12} for the regular expression that describes the language accepted by the DFA, and it is:

$$R^3_{12} = 0^*10^* \cup 0^*10^*1(0\cup\varepsilon\cup 10^*10^*1)^*10^*10^*.$$

(This last expression can be simplified to $0^*10^*(10^*10^*10^*)^*$.)