Regexp $r = (aa)^* (a+b)(ba)^* \quad \text{and} \quad r^* = (aa)^* (ba)^*$

What happens with $r', N', \text{and } N''$ when $x = bb$, or $x = aabb$?

Can we process $x$, let alone accept it?

Trial computation paths:

1. in $N'$: $(s, a, 1, a, s, e, f, b, 2, b \text{ die.})$
2. in $N''$ only: $(s, e, f, a, \text{ crash, die.})$

Defn: For any states $p, q$ of an NFA $N = (Q, \Sigma, \delta, s, F)$ and string $x \in \Sigma^*$, let $n = |x|$, say that $N$ can process $x$ from $p$ to $q$ if there is a computation path

\[(p, w_1, q_1, w_2, q_2, w_3, q_3, \ldots, q_{m-1}, w_m, q_m)\]

such that:

- each $w_j$ is either a char of $\Sigma$, and $m \geq n$.
- $q_0 = p$.
- For each $j$, $1 \leq j \leq m$, $(q_{j-1}, w_j, q_j)$ is an instruction in $\delta$ of $N$.
- $q_m = q$.
- $X = W_1 \ldots W_m$ (implies $m \geq n$).

And define $L(N) = \{x \in \Sigma^*: N \text{ can process } x \text{ from } p \text{ to } q\}$ (Or just $L(p,q)$).
Then we can formally define \( L(N) = \{ w \mid w \text{ is a string over } \Sigma \} \) for a DFA \( M \). This also works for a NFA, \( M \).

Given \( x \) and \( i \), \( 0 \leq i \leq n \), define

\[
R_i = \{ q : N \text{ can process } x_1 \cdots x_i \text{ from } s \text{ to } q \}
\]

\[
R_0 = \{ q : q \in L_{eq} \}. \text{ In } N \text{ and } N', \ R_0 = \{ s \}. \text{ In } N, \ R_0 = \{ s \}, \quad \text{if } x_i \in F.
\]

**Theorem:** For every NFA \( N = (Q, \Sigma, S, s, F) \), we can build a DFA \( M = (Q, \Sigma, \Delta, S, F) \) such that \( L(M) = L(N) \).

**Proof:** We will maintain the induction invariant that for any \( x \in \Sigma^* \) and all \( i \), \( 0 \leq i \leq n \), \( R_i \) equals the state \( M \) is in upon reading \( x_1 \cdots x_i \). We define \( F = \{ R \subseteq Q : R \text{ includes a } q \in F \} \).

**Basis (i = 0):** Make \( S = R_0 = \{ q : q \in L_{eq} \} \). Text calls this \( E(s) \), the \( E \)-closure of \( s \).

For all \( p \in Q \) and \( c \in \Sigma \), define \( \Delta(p, c) = \bigcup_{p' \in p} \{ q : N \text{ can process } c \text{ from } p \text{ to } q \} \) by first reading \( c \).

Suppose \( M \) is in state \( R_i \) after processing \( x_1 \cdots x_{i-1} \). Take \( c = x_i \). Say \( p' = \Delta(p, c) \). Need to show \( p' = R_i \) which is defined in terms of the NFA.

\[
R_{i+1} = \{ q : N \text{ can process } x_i \cdots x_{i-1} \text{ from } s \text{ to } q \}
\]

[proof converted in words]

(Will write out later) that \( p' = R_i \). Thus the induction "goes through" and \( L(M) = L(N) \).
Example \( N = \begin{array}{c}
\varepsilon \\
1 \\
2 \\
3
\end{array} \rightarrow
\begin{array}{c}
a \\
b \\
an \\
b
\end{array} \rightarrow
\begin{array}{c}
a \\
b \\
a \\
b
\end{array}
\)

\[ \Delta(p, c) = \bigcup_{p \in P} \delta(p, c) \]
where \( \delta(p, c) \) is the RHS of the depth \( a \) \( b \) of \( \Delta \) after the \( \bigcup \).

\[
\begin{array}{c|c|c|c}
S & 1 & 0 & 2 \\
\hline
f & 0 & 2 & 0 \\
S & 1 & 0 & 2 \\
\hline
f & 0 & 2 & 0
\end{array}
\]

Lecture did not finish the example, so once you get the \( S \) table correct it is automatic:

\[ \Delta(s, a) = \delta(s, a) \cup \delta(s, c) = \{ 1 \} \cup \emptyset = \{ 1 \} \]

\[ \Delta(s, b) = \delta(s, b) \cup \delta(s, c) = \emptyset \cup \{ 2 \} = \{ 2 \} \]

\[ \Delta(s, c) = \delta(s, c) = \{ 1 \} \]

\[ \Delta(g, b) = \emptyset \]

\[ \Delta(1, a) = \delta(1, a) = \{ 2 \} \]

\[ \Delta(1, b) = \emptyset \]

\[ \Delta(2, a) = \delta(2, a) = \{ 2 \} \]

\[ \Delta(2, b) = \delta(2, b) = \{ 2 \} \]

\[ \Delta(2, c) = \{ 1 \} \]

\[ \Delta(f, a) = \emptyset \]

\[ \Delta(f, b) = \emptyset \]

\[ \Delta(f, c) = \emptyset \]

\[ \Delta(2a, b) = \{ 2 \} \]

The final PFA. It is minimal.

Note that we could have had as many as \( 2^{16} \) states, but we only needed 5. This way of economically going about the states is an example of breadth-first search (BFS).

**Writing out what was said orally in the NFA-to-DFA Proof:**

To show \( p' = R_j \), we need to show that any \( q \in p' \) belongs to \( R_j \) and vice-versa. So first suppose \( q \in p' \). By \( p' = \Delta(p, c) \), where \( P \) was the state of \( M \) after reading \( X_1 \ldots X_i \). We have that \( N \) can process \( C \) from \( p \) to \( q \) (by first reading \( c \)). By inductive hypothesis, we have \( P = R_{j-1} \), which means that since \( p \in P \), we have \( p \in R_{j-1} \), so \( N \) can process \( X_1 \ldots X_{i-1}C \) from \( s \) to \( p \). Putting those together, \( N \) can process \( X_1 \ldots X_{i-1}C \) from \( s \) to \( q \) when \( p \) is the state it was in just before \( N \) read \( c \). So \( q \in R_{j-1} \). By Ind. Hyp., \( p \in P \). And \( N \) can process \( C \) from \( p \) to \( q \) by first reading \( c \). So \( q \in \Delta(p, c) \) ∈ \( p' \).