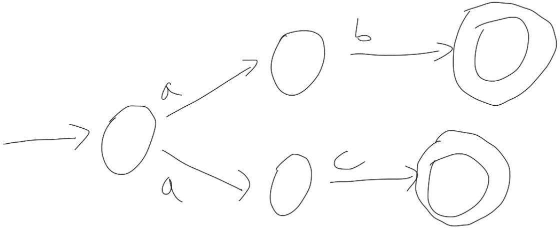
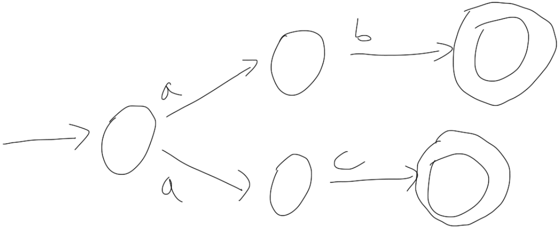
Non-deterministic Finite Automata

Consider the following machine. What is different about it?



Non-deterministic Finite Automata

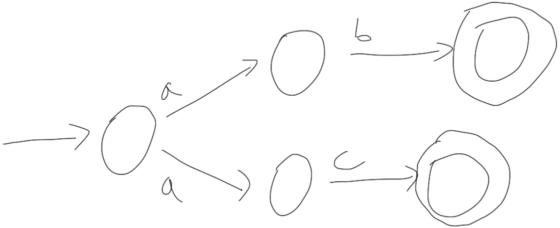
Consider the following machine. What is different about it?



We call this a non-deterministic finite automata (NFA).

Non-deterministic Finite Automata

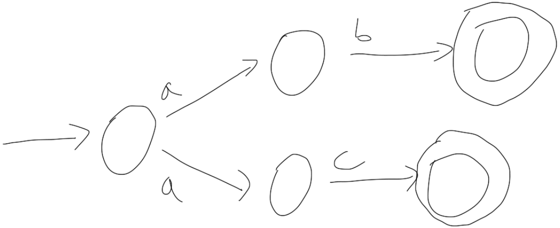
Consider the following machine. What is different about it?



We call this a non-deterministic finite automata (NFA).
We say it accepts input x *if and only if there exists* a path that accepts x.

Non-deterministic Finite Automata

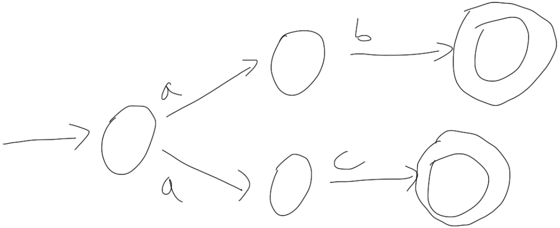
Consider the following machine. What is different about it?



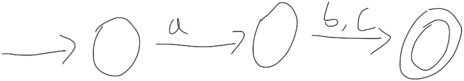
We call this a non-deterministic finite automata (NFA).
We say it accepts input x *if and only if there exists* a path that accepts x.
What is the language of the above machine?

Non-deterministic Finite Automata

Consider the following machine. What is different about it?



We call this a non-deterministic finite automata (NFA).
We say it accepts input x *if and only if there exists* a path that accepts x.
What is the language of the above machine?

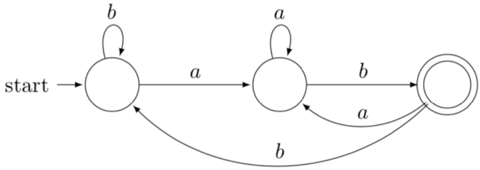


NFA Example

Write a DFA that recognizes the language $L = \{x \mid x \in \{a, b\}^* \text{ and } x \text{ ends with } ab\}$.

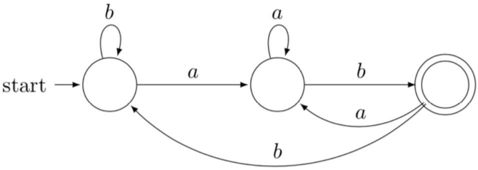
NFA Example

Write a DFA that recognizes the language $L = \{x \mid x \in \{a, b\}^* \text{ and } x \text{ ends with } ab\}$.



NFA Example

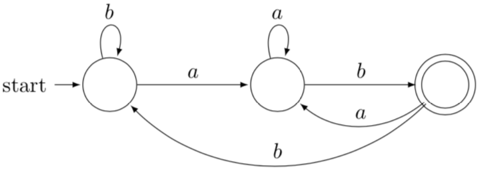
Write a DFA that recognizes the language $L = \{x \mid x \in \{a, b\}^* \text{ and } x \text{ ends with } ab\}$.



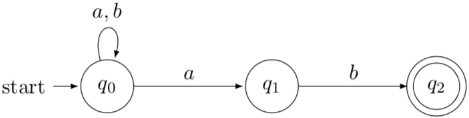
Write an NFA that recognizes the language L.

NFA Example

Write a DFA that recognizes the language $L = \{x \mid x \in \{a, b\}^* \text{ and } x \text{ ends with } ab\}$.



Write an NFA that recognizes the language L.



Equivalence between RG and NFA

Lemma 9.4

If $L = \mathcal{L}(G)$ for some regular grammar G, then there exists an NFA M such that $L = \mathcal{L}(M)$.

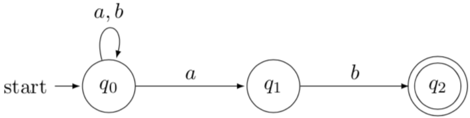
Lemma 9.5

If $L = \mathcal{L}(M)$ for some NFA M, then there exists a regular grammar G such that $L = \mathcal{L}(G)$.

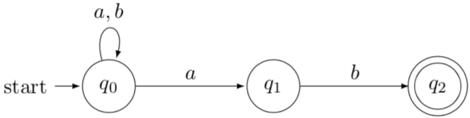
Theorem 9.2

L is regular if and only if there exists an NFA M such that $L = \mathcal{L}(M)$.

Example: RG from NFA



Example: RG from NFA



$G = (\{Q_0, Q_1, Q_2\}, \{a, b\}, q_0, P)$, where:
$P = \{Q_0 \to aQ_0, Q_0 \to bQ_0, Q_0 \to aQ_1, Q_1 \to bQ_2, Q_2 \to \Lambda\}$

Formalizing NFA

Given a set of elements, Q, we denote the set of all subsets of Q, which we call the *power set* of Q, by 2^Q.

Formalizing NFA

Given a set of elements, Q, we denote the set of all subsets of Q, which we call the *power set* of Q, by 2^Q.
Example: $Q = \{A, B, C\}$,
$2^Q = \{\emptyset, \{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}\}$

Formalizing NFA

Given a set of elements, Q, we denote the set of all subsets of Q, which we call the *power set* of Q, by 2^Q.
Example: $Q = \{A, B, C\}$,
$2^Q = \{\emptyset, \{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}\}$

NFAs

An NFA, M, is a quintuple, $(Q, \Sigma, q_0, \delta, A)$, where Q, Σ, q_0 and A are defined as in DFAs, and $\delta : Q \times \Sigma \to 2^Q$

Formalizing NFA

Given a set of elements, Q, we denote the set of all subsets of Q, which we call the *power set* of Q, by 2^Q.
Example: $Q = \{A, B, C\}$,
$2^Q = \{\emptyset, \{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}\}$

NFAs

An NFA, M, is a quintuple, $(Q, \Sigma, q_0, \delta, A)$, where Q, Σ, q_0 and A are defined as in DFAs, and $\delta : Q \times \Sigma \to 2^Q$

δ maps to a set of states, rather than a single state!

Formalizing NFA

Given a set of elements, Q, we denote the set of all subsets of Q, which we call the *power set* of Q, by 2^Q.
Example: $Q = \{A, B, C\}$,
$2^Q = \{\emptyset, \{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}\}$

NFAs

An NFA, M, is a quintuple, $(Q, \Sigma, q_0, \delta, A)$, where Q, Σ, q_0 and A are defined as in DFAs, and $\delta : Q \times \Sigma \to 2^Q$

δ maps to a set of states, rather than a single state!

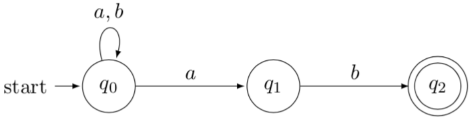
δ^*

For NFA $M = (Q, \Sigma, q_0, \delta, A)$, δ^* is a function that takes a state and a string as input and produces a resulting set of states. That is $\delta^* : Q \times \Sigma^* \to 2^Q$, such that:

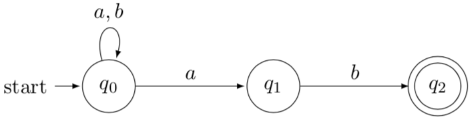
- for any $q \in Q$, $\delta^*(q, \Lambda) = \{q\}$, and
- for any $q \in Q$, any $\sigma \in \Sigma$, and any $x \in \Sigma^*$,

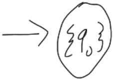
$$\delta^*(q, x\sigma) = \bigcup_{p \in \delta^*(q,x)} \delta(p, \sigma).$$

NFA to DFA: Example 1

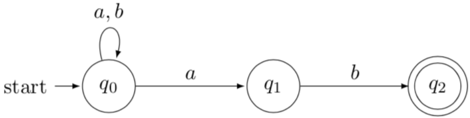


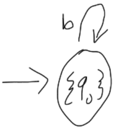
NFA to DFA: Example 1



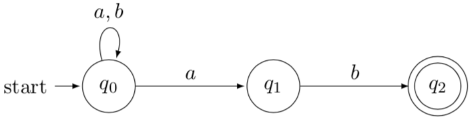


NFA to DFA: Example 1

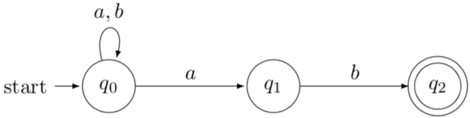


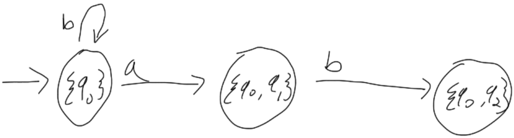


NFA to DFA: Example 1

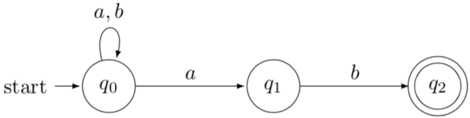


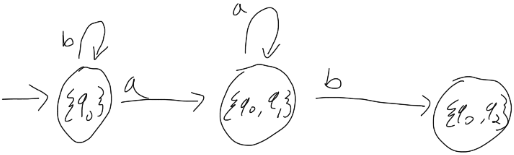
NFA to DFA: Example 1



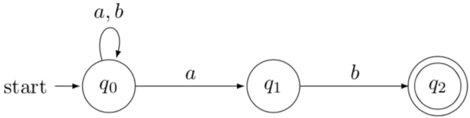


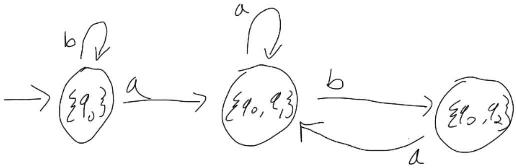
NFA to DFA: Example 1



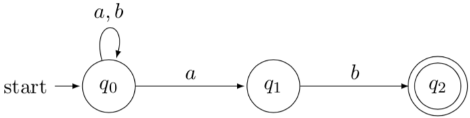


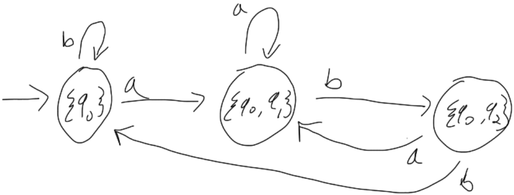
NFA to DFA: Example 1



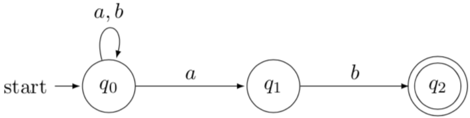


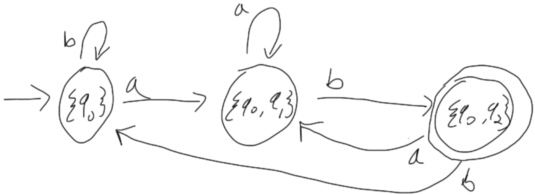
NFA to DFA: Example 1



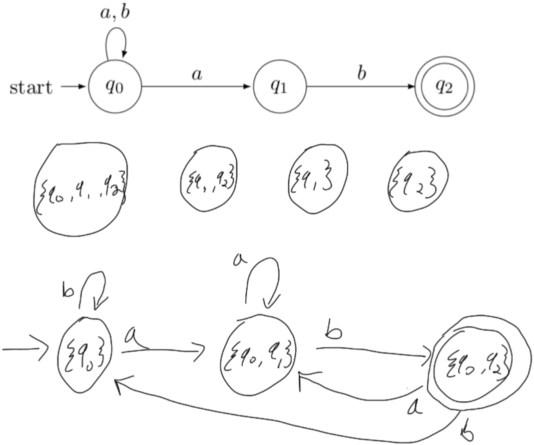


NFA to DFA: Example 1

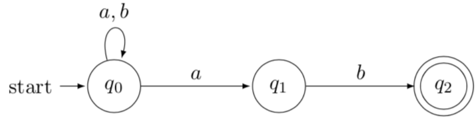


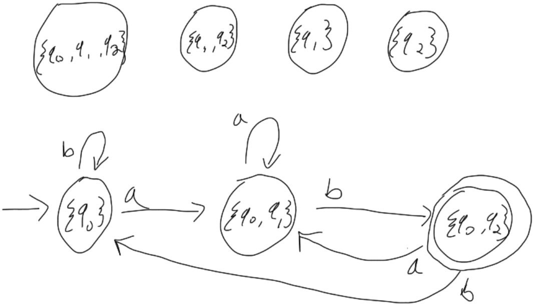


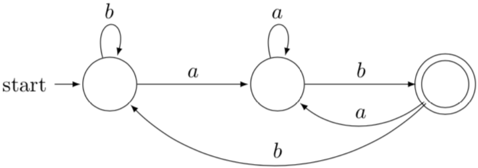
NFA to DFA: Example 1



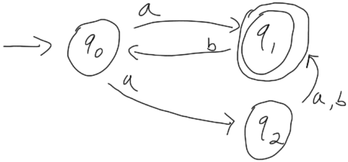
NFA to DFA: Example 1



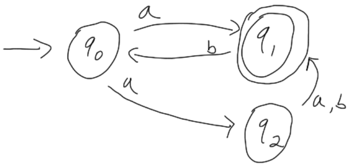




NFA to DFA: Example 2

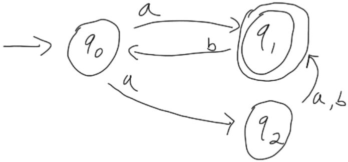


NFA to DFA: Example 2

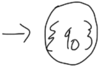


$$(a+aa+ab)(ba+baa+bab)^*$$

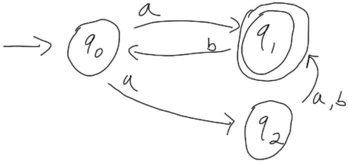
NFA to DFA: Example 2



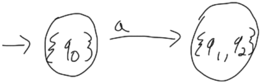
$$(a+aa+ab)(ba+baa+bab)^*$$



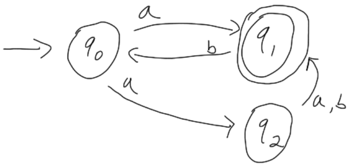
NFA to DFA: Example 2



$$(a + aa + ab)(ba + baa + bab)^*$$

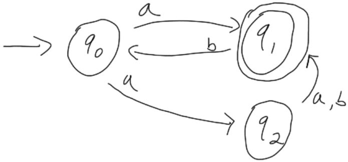


NFA to DFA: Example 2

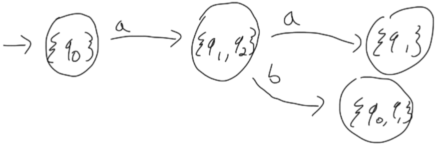


$$(a + aa + ab)(ba + baa + bab)^*$$

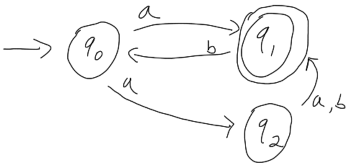
NFA to DFA: Example 2



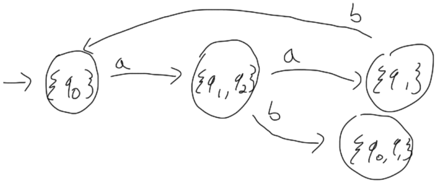
$$(a + aa + ab)(ba + baa + bab)^*$$



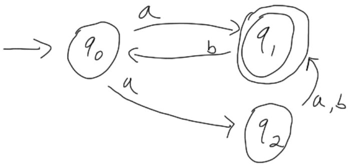
NFA to DFA: Example 2



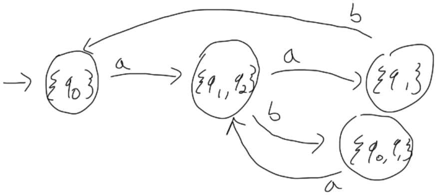
$(a + aa + ab)(ba + baa + bab)^*$



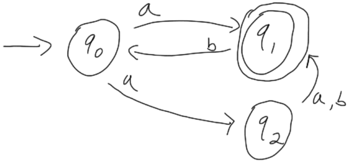
NFA to DFA: Example 2



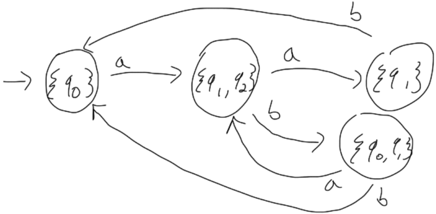
$(a+aa+ab)(ba+baa+bab)^*$



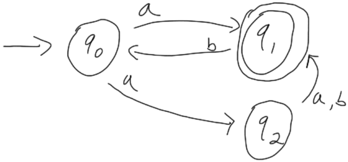
NFA to DFA: Example 2



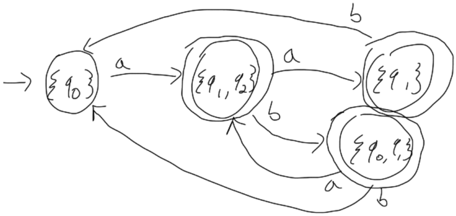
$$(a+aa+ab)(ba+baa+bab)^*$$



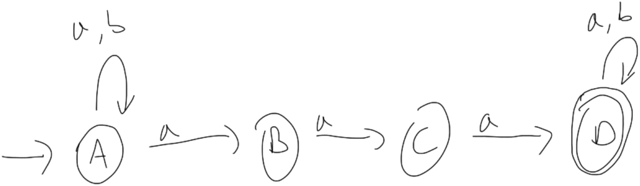
NFA to DFA: Example 2



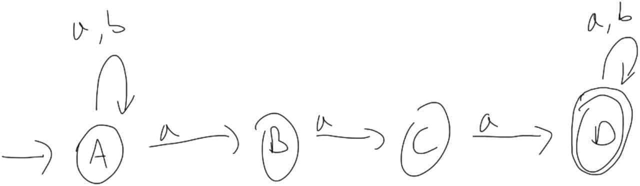
$$(a+aa+ab)(ba+baa+bab)^*$$



NFA to DFA: Example 3

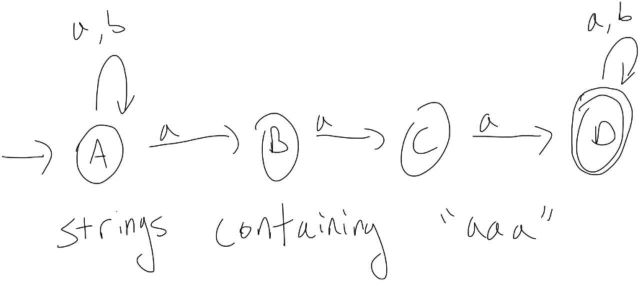


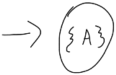
NFA to DFA: Example 3



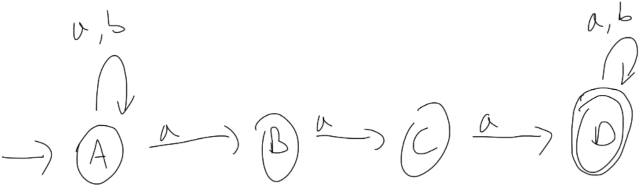
strings containing "aaa"

NFA to DFA: Example 3

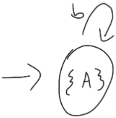




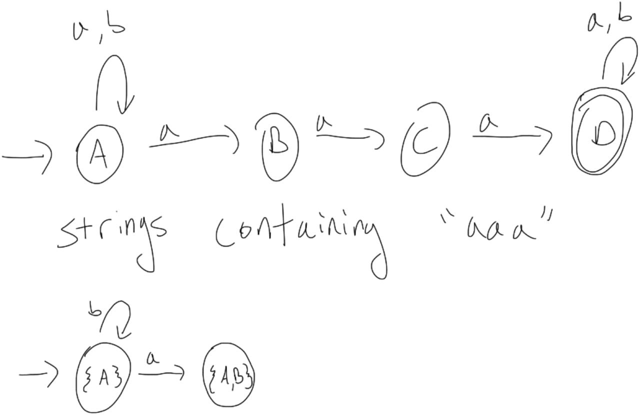
NFA to DFA: Example 3



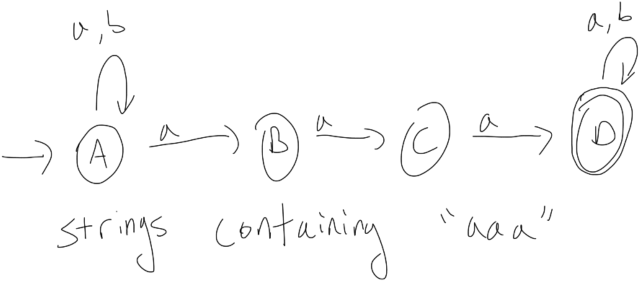
strings containing "aaa"

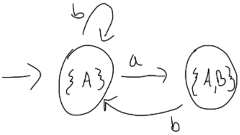


NFA to DFA: Example 3

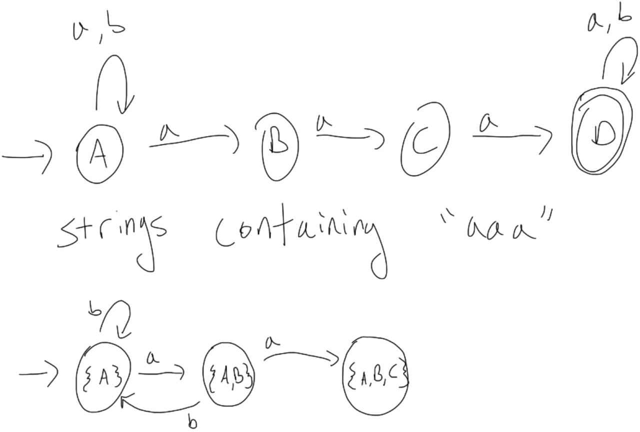


NFA to DFA: Example 3

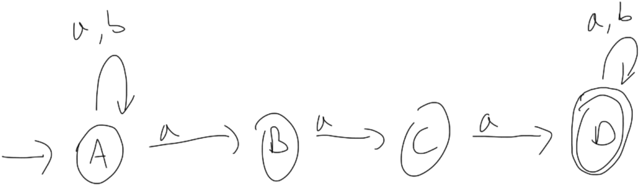




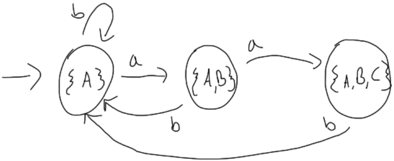
NFA to DFA: Example 3



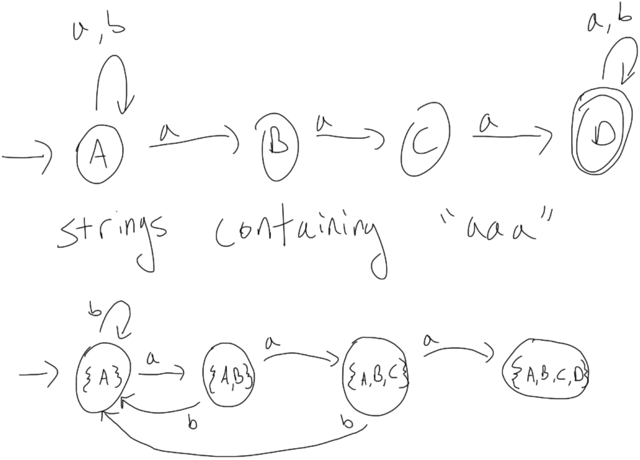
NFA to DFA: Example 3



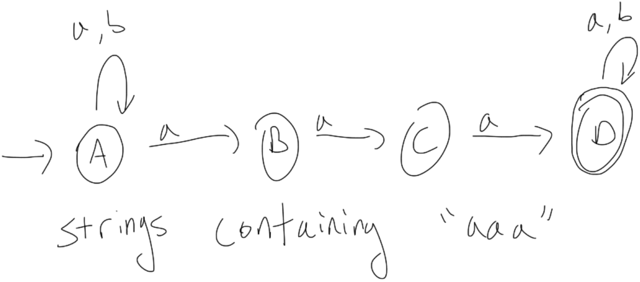
strings containing "aaa"

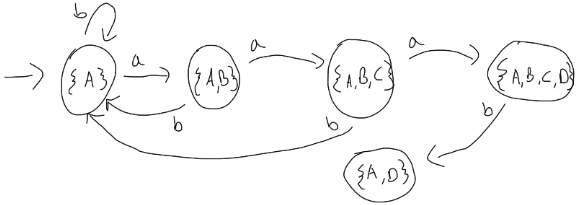


NFA to DFA: Example 3

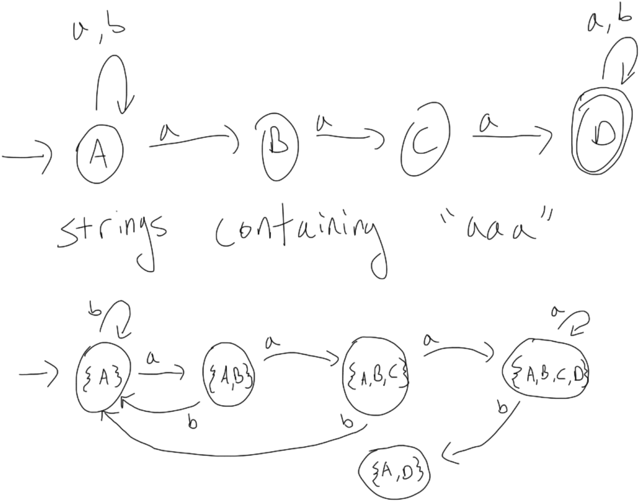


NFA to DFA: Example 3

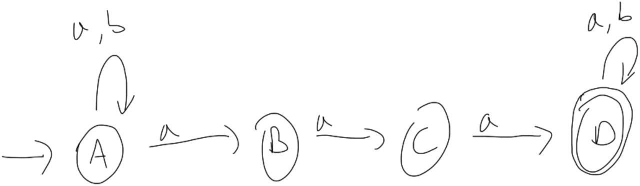




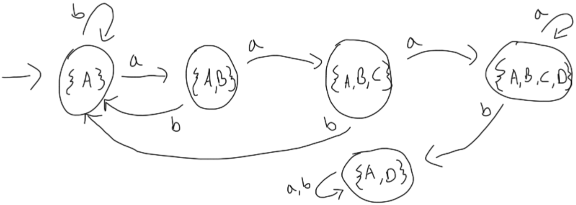
NFA to DFA: Example 3



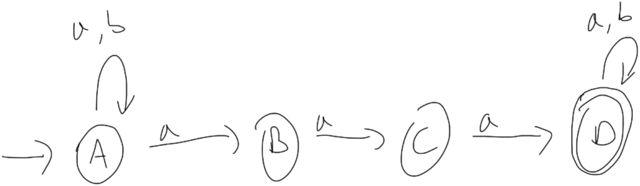
NFA to DFA: Example 3



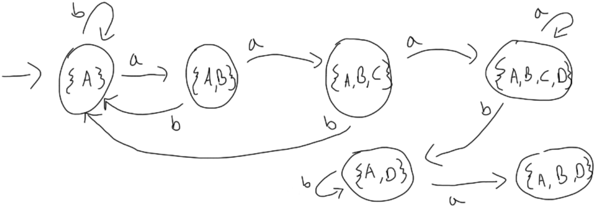
strings containing "aaa"



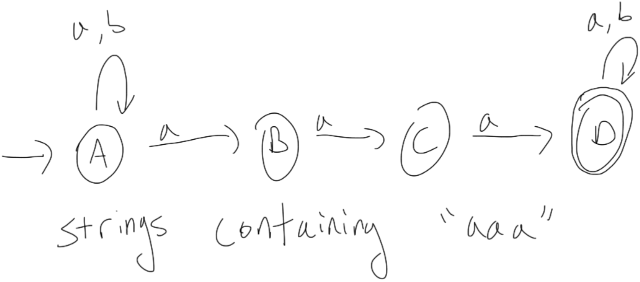
NFA to DFA: Example 3

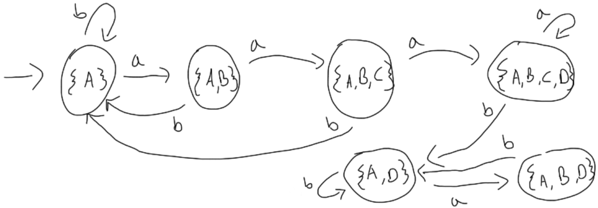


strings containing "aaa"

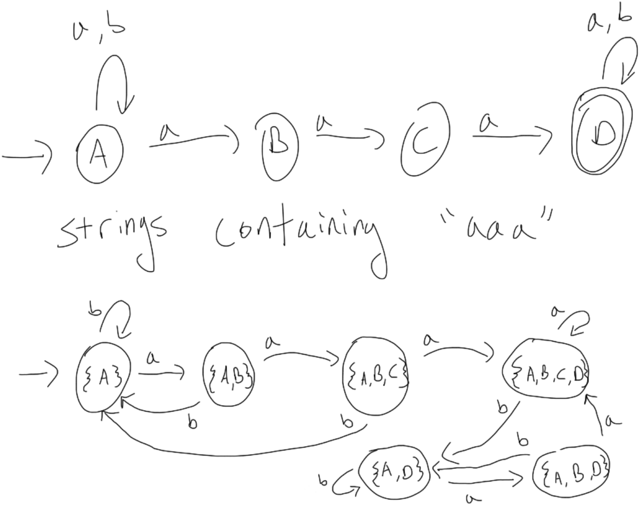


NFA to DFA: Example 3

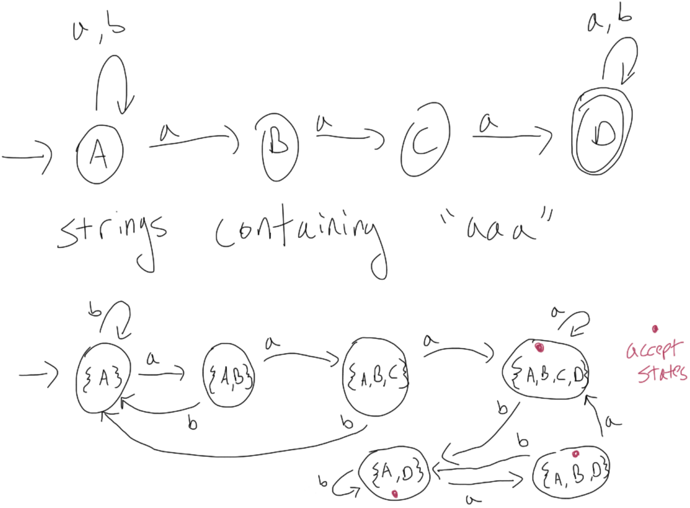




NFA to DFA: Example 3



NFA to DFA: Example 3



NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M)$, then $w \in \mathcal{L}(M')$

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M)$, then $w \in \mathcal{L}(M')$
Because M is an NFA, $\exists(q_0, q_1, \ldots, q_k)$, s.t. $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M)$, then $w \in \mathcal{L}(M')$
Because M is an NFA, $\exists (q_0, q_1, \ldots, q_k)$, s.t. $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.
Let $T_0, T_1, \ldots, T_k$, be the states in M' such that $T_{i+1} = \delta'(T_i, w_i)$.
We claim that $T_k \in A'$.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M)$, then $w \in \mathcal{L}(M')$
Because M is an NFA, $\exists(q_0, q_1, \ldots, q_k)$, s.t. $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.
Let $T_0, T_1, \ldots, T_k$, be the states in M' such that $T_{i+1} = \delta'(T_i, w_i)$.
We claim that $T_k \in A'$.
First, we show that $q_i \in T_i$.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M)$, then $w \in \mathcal{L}(M')$
Because M is an NFA, $\exists (q_0, q_1, \ldots, q_k)$, s.t. $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.
Let $T_0, T_1, \ldots, T_k$, be the states in M' such that $T_{i+1} = \delta'(T_i, w_i)$.
We claim that $T_k \in A'$.
First, we show that $q_i \in T_i$.
Proof by induction: clearly this holds for q_0, since $T_0 = \{q_0\}$

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M)$, then $w \in \mathcal{L}(M')$
Because M is an NFA, $\exists (q_0, q_1, \ldots, q_k)$, s.t. $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.
Let $T_0, T_1, \ldots, T_k$, be the states in M' such that $T_{i+1} = \delta'(T_i, w_i)$.
We claim that $T_k \in A'$.
First, we show that $q_i \in T_i$.
Proof by induction: clearly this holds for q_0, since $T_0 = \{q_0\}$
Assume it holds for q_i. Recall: $T_{i+1} = \bigcup_{q \in T_i} \delta(q, w_i)$.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M)$, then $w \in \mathcal{L}(M')$
Because M is an NFA, $\exists (q_0, q_1, \ldots, q_k)$, s.t. $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.
Let $T_0, T_1, \ldots, T_k$, be the states in M' such that $T_{i+1} = \delta'(T_i, w_i)$.
We claim that $T_k \in A'$.
First, we show that $q_i \in T_i$.
Proof by induction: clearly this holds for q_0, since $T_0 = \{q_0\}$
Assume it holds for q_i. Recall: $T_{i+1} = \bigcup_{q \in T_i} \delta(q, w_i)$.
Since $q_{i+1} \in \delta(q_i, w_i)$, and $q_i \in T_i$, it follows that $q_{i+1} \in T_{i+1}$.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M)$, then $w \in \mathcal{L}(M')$
Because M is an NFA, $\exists (q_0, q_1, \ldots, q_k)$, s.t. $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.
Let $T_0, T_1, \ldots, T_k$, be the states in M' such that $T_{i+1} = \delta'(T_i, w_i)$.
We claim that $T_k \in A'$.
First, we show that $q_i \in T_i$.
Proof by induction: clearly this holds for q_0, since $T_0 = \{q_0\}$
Assume it holds for q_i. Recall: $T_{i+1} = \bigcup_{q \in T_i} \delta(q, w_i)$.
Since $q_{i+1} \in \delta(q_i, w_i)$, and $q_i \in T_i$, it follows that $q_{i+1} \in T_{i+1}$.
Finally, since $q_k \in T_k$, and $q_k \in A$, it follows that $T_k \in A'$.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M')$, then $w \in \mathcal{L}(M)$

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M')$, then $w \in \mathcal{L}(M)$
Let $T_0, \ldots, T_k$ be the sequence of states such that $T_{i+1} = \delta'(T_i, w_i)$.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M')$, then $w \in \mathcal{L}(M)$
Let $T_0, \ldots, T_k$ be the sequence of states such that $T_{i+1} = \delta'(T_i, w_i)$.
Claim $\exists(q_0, \ldots, q_k)$, such that $q_i \in Q$, $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M')$, then $w \in \mathcal{L}(M)$
Let $T_0, \ldots, T_k$ be the sequence of states such that $T_{i+1} = \delta'(T_i, w_i)$.
Claim $\exists (q_0, \ldots, q_k)$, such that $q_i \in Q$, $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.
We start by choosing q_k and work backwards.
To choose q_k, note that because $T_k \in A'$, $\exists q_k \in T_k$ s.t. $q_k \in A$. Choose any such q_k.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M')$, then $w \in \mathcal{L}(M)$
Let $T_0, \ldots, T_k$ be the sequence of states such that $T_{i+1} = \delta'(T_i, w_i)$.
Claim $\exists(q_0, \ldots, q_k)$, such that $q_i \in Q$, $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.
We start by choosing q_k and work backwards.
To choose q_k, note that because $T_k \in A'$, $\exists q_k \in T_k$ s.t. $q_k \in A$. Choose any such q_k.
For $i < k$, assume q_{i+1} has already be chosen.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M')$, then $w \in \mathcal{L}(M)$
Let $T_0, \ldots, T_k$ be the sequence of states such that $T_{i+1} = \delta'(T_i, w_i)$.
Claim $\exists(q_0, \ldots, q_k)$, such that $q_i \in Q$, $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.
We start by choosing q_k and work backwards.
To choose q_k, note that because $T_k \in A'$, $\exists q_k \in T_k$ s.t. $q_k \in A$. Choose any such q_k.
For $i < k$, assume q_{i+1} has already be chosen.
Since $T_{i+1} = \bigcup_{q \in T_i} \delta(q, w_i)$, there exists some $q_i \in T_i$ such that $q_{i+1} \in \delta(q_i, w_i)$.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M')$, then $w \in \mathcal{L}(M)$
Let $T_0, \ldots, T_k$ be the sequence of states such that $T_{i+1} = \delta'(T_i, w_i)$.
Claim $\exists(q_0, \ldots, q_k)$, such that $q_i \in Q$, $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.
We start by choosing q_k and work backwards.
To choose q_k, note that because $T_k \in A'$, $\exists q_k \in T_k$ s.t. $q_k \in A$. Choose any such q_k.
For $i < k$, assume q_{i+1} has already be chosen.
Since $T_{i+1} = \bigcup_{q \in T_i} \delta(q, w_i)$, there exists some $q_i \in T_i$ such that $q_{i+1} \in \delta(q_i, w_i)$.
Choose any such q_i, and repeat.

NFA to DFA: Formalization

Theorem

For any NFA $M = (\Sigma, Q, q_0, A, \delta)$, there exists a DFA, $M' = (\Sigma, Q', S', A', \delta')$, such that $\mathcal{L}(M') = \mathcal{L}(M)$.

We prove it by demonstrating an algorithm that constructs M' from M.
$M' = (\Sigma, Q', \{q_0\}, A', \delta')$, where:
$Q' = 2^Q$, $A' = \{T \in Q' \mid \exists t \in T \text{ s.t. } t \in A\}$, and
$\delta'(T, x) = \bigcup_{q \in T} \delta(q, x)$

Claim: If $w = w_0 \cdots w_{k-1} \in \mathcal{L}(M')$, then $w \in \mathcal{L}(M)$
Let $T_0, \ldots, T_k$ be the sequence of states such that $T_{i+1} = \delta'(T_i, w_i)$.
Claim $\exists (q_0, \ldots, q_k)$, such that $q_i \in Q$, $q_{i+1} \in \delta(q_i, w_i)$, and $q_k \in A$.
We start by choosing q_k and work backwards.
To choose q_k, note that because $T_k \in A'$, $\exists q_k \in T_k$ s.t. $q_k \in A$. Choose any such q_k.
For $i < k$, assume q_{i+1} has already be chosen.
Since $T_{i+1} = \bigcup_{q \in T_i} \delta(q, w_i)$, there exists some $q_i \in T_i$ such that $q_{i+1} \in \delta(q_i, w_i)$.
Choose any such q_i, and repeat.
Since $T_0 = \{q_0\}$, we can choose q_0 as our start state.