Number of States for DFAs and NFAs

Lecture 5

Binghui Peng

Recap from last lecture

Equivalence of NFAs and DFAs

Theorem: If L is accepted by an NFA, then L is also accepted by a DFA. Given an NFA $(Q, \Sigma, \Delta, s, F)$ with no e-transitions, we construct a DFA $(2^Q, \Sigma, \delta, \{s\}, F')$:

- **States:** Every state in the DFA corresponds to a **subset** of states in the NFA.
- **Transition Function:**

$$\delta(A, \sigma) = \bigcup_{q \in A} \Delta(q, \sigma)$$

- **Final States:**

$$F' = \{A \subseteq Q : A \cap F \neq \emptyset\}$$

Result: The DFA tracks all possible states the NFA could be in simultaneously. If the NFA has n states, the DFA has at most 2^n states.

Summary of Closure Properties

The following table summarizes the state complexity for DFA and NFA constructions. (L_i has n_i states).

Closure Property	**DFA**	**NFA**
$L_1 \cup L_2$	$n_1 n_2$	$n_1 + n_2 + 1$
$L_1 \cap L_2$	$n_1 n_2$	$n_1 n_2$
$L_1 \cdot L_2$	Complex	$n_1 + n_2$
$\overline{L}$	n	$\sim 2^n$
L^*	Complex	$n + 1$

This lecture: NFA with small states

Consider DFAs For $\{aaa\}$

$\Sigma = \{a\}$.

$$L = \{aaa\}$$

With your table draw a DFA for this language.

How many states does it have?

Consider DFAs For $\{aaa\}$

$\Sigma = \{a\}$.

$$L = \{aaa\}$$

With your table draw a DFA for this language.

How many states does it have?

5

Consider DFAs For $\{aaa\}$

$\Sigma = \{a\}$.

$$L = \{aaa\}$$

With your table draw a DFA for this language.

How many states does it have?

5

Is there a DFA for L that has 4 states? Discuss.

Any DFA for $\{aaa\}$ Has ≥ 5 states

Assume there exists a DFA M for $\{aaa\}$ with 4 states.
Input aaa

Starts in state $s = q_0$
An a is processed. Now in state q_1. $q_1 \notin F$.
Another a is processed. Now in state q_2. $q_2 \notin F$.
Another a is processed. Now in state q_3. $q_3 \in F$.

$\delta(q_3, a)$ has to be one of q_0, q_1, q_2, q_3.
Let say it is q_1 (the other cases are similar).
aaa ends in state q_3.
$aaaa$ ends in state q_1.
$aaaaa$ ends in state q_2.
$aaaaaa$ ends in state q_3, so $aaaaaa$ is accepted. **Contradiction.**

What about $\{a^n\}$?

More generally: For all n

What about $\{a^n\}$?

More generally: For all n

There is a DFA for $\{a^n\}$ with $n + 2$ states.

What about $\{a^n\}$?

More generally: For all n

There is a DFA for $\{a^n\}$ with $n + 2$ states.

Any DFA for $\{a^n\}$ has $\geq n + 2$ states.

Consider NFAs For $\{aaa\}$

$\Sigma = \{a\}$.

$$L = \{aaa\}$$

With your table draw an NFA for this language.

Consider NFAs For $\{aaa\}$

$\Sigma = \{a\}$.

$$L = \{aaa\}$$

With your table draw an NFA for this language.

How many states does it have?

5

Is there an NFA for L that has 4 states? Discuss.

Any NFA for $\{aaa\}$ Has ≥ 5 states

Assume there exists a NFA M for $\{aaa\}$ with 4 states.
Input aaa. We look at the **ACCEPTING** path.

Starts in state $s = q_0$
An a is processed.

Any NFA for $\{aaa\}$ Has ≥ 5 states

Assume there exists a NFA M for $\{aaa\}$ with 4 states.
Input aaa. We look at the **ACCEPTING** path.

Starts in state $s = q_0$
An a is processed. Now in state q_1. $q_1 \notin F$.
Another a is processed.

Any NFA for $\{aaa\}$ Has ≥ 5 states

Assume there exists a NFA M for $\{aaa\}$ with 4 states.
Input aaa. We look at the **ACCEPTING** path.

Starts in state $s = q_0$
An a is processed. Now in state q_1. $q_1 \notin F$.
Another a is processed. Now in state q_2. $q_2 \notin F$.
Another a is processed.

Any NFA for $\{aaa\}$ Has ≥ 5 states

Assume there exists a NFA M for $\{aaa\}$ with 4 states.
Input aaa. We look at the **ACCEPTING** path.

Starts in state $s = q_0$
An a is processed. Now in state q_1. $q_1 \notin F$.
Another a is processed. Now in state q_2. $q_2 \notin F$.
Another a is processed. Now in state q_3. $q_3 \in F$.

Look at q_0, q_1, q_2, q_3.

Any NFA for $\{aaa\}$ Has ≥ 5 states

Assume there exists a NFA M for $\{aaa\}$ with 4 states.
Input aaa. We look at the **ACCEPTING** path.

Starts in state $s = q_0$
An a is processed. Now in state q_1. $q_1 \notin F$.
Another a is processed. Now in state q_2. $q_2 \notin F$.
Another a is processed. Now in state q_3. $q_3 \in F$.

Look at q_0, q_1, q_2, q_3. Two of them have to be the same.

Can use this to find a shorter string that is accepted. **Contradiction.**

DFAs and NFAs for $\{a^n\}$?

For all n

There is a DFA for $\{a^n\}$ with $n+2$ states.

Any DFA for $\{a^n\}$ has $\geq n+2$ states.

There is an NFA for $\{a^n\}$ with $n+2$ states.

Any NFA for $\{a^n\}$ has $\geq n+2$ states.

DFA for $\{a^i : i \neq 1000\}$

$L = \{a^i : i \neq 1000\}$
There is a DFA for this with 1002 states.

Is there a DFA with ≤ 1001 states?

DFA for $\{a^i : i \neq 1000\}$

$L = \{a^i : i \neq 1000\}$
There is a DFA for this with 1002 states.

Is there a DFA with ≤ 1001 states?

No.

If there was, then complement it to get
a DFA for $\{a^{1000}\}$ with ≤ 1001 states.

Contradiction.

NFA for $\{a^i : i \neq 1000\}$

There is an NFA for L that has 1001 states.

NFA for $\{a^i : i \neq 1000\}$

There is an NFA for L that has 1001 states.
Work in groups to see if you can do better, and not just be a few. For definiteness: Can you get an NFA with ≤ 900 states?

NFA for $\{a^i : i \neq 1000\}$

There is an NFA for L that has 1001 states.
Work in groups to see if you can do better, and not just be a few. For definiteness: Can you get an NFA with ≤ 900 states?
VOTE

NFA for $\{a^i : i \neq 1000\}$

There is an NFA for L that has 1001 states.
Work in groups to see if you can do better, and not just be a few. For definiteness: Can you get an NFA with ≤ 900 states?
VOTE

1. There is an NFA for L with ≤ 900 states.

NFA for $\{a^i : i \neq 1000\}$

There is an NFA for L that has 1001 states.
Work in groups to see if you can do better, and not just be a few. For definiteness: Can you get an NFA with ≤ 900 states?

VOTE

1. There is an NFA for L with ≤ 900 states.
2. All NFA's for L have ~ 1000 states.

Break

Revisiting the Language $L = \{a^i : i \neq 1000\}$

In the previous lecture, we discussed the language:

$$L = \{a^i : i \neq 1000\}$$

Revisiting the Language $L = \{a^i : i \neq 1000\}$

In the previous lecture, we discussed the language:

$$L = \{a^i : i \neq 1000\}$$

While any DFA for L requires 1002 states, the NFA complexity is more nuanced.

Revisiting the Language $L = \{a^i : i \neq 1000\}$

In the previous lecture, we discussed the language:

$$L = \{a^i : i \neq 1000\}$$

While any DFA for L requires 1002 states, the NFA complexity is more nuanced.

Contrary to initial intuition that might suggest a size proportional to 1000, L can be recognized by an NFA with significantly fewer states.

State Complexity of L

Question: How many states are required for an NFA to recognize L?

State Complexity of L

Question: How many states are required for an NFA to recognize L?

Vote:

State Complexity of L

Question: How many states are required for an NFA to recognize L?

Vote:

1. $700 \leq s \leq 900$
2. $400 \leq s \leq 699$
3. $100 \leq s \leq 399$
4. $s \leq 99$

State Complexity of L

Question: How many states are required for an NFA to recognize L?

Vote:

1. $700 \leq s \leq 900$
2. $400 \leq s \leq 699$
3. $100 \leq s \leq 399$
4. $s \leq 99$

Result: There exists an NFA for L with only **70 states**.

Construction Strategy

Construction Strategy

We decompose the problem by constructing two separate NFAs:

Construction Strategy

We decompose the problem by constructing two separate NFAs:

NFA A:

- Rejects a^{1000}.
- Accepts all strings a^n where $n \geq 1001$.
- Behavior on $n \leq 999$ is unspecified.

Construction Strategy

We decompose the problem by constructing two separate NFAs:

NFA A:

- Rejects a^{1000}.
- Accepts all strings a^n where $n \geq 1001$.
- Behavior on $n \leq 999$ is unspecified.

NFA B:

- Rejects a^{1000}.
- Accepts all strings a^n where $n \leq 999$.
- Behavior on $n \geq 1001$ is unspecified.

Construction Strategy

We decompose the problem by constructing two separate NFAs:

NFA A:
- Rejects a^{1000}.
- Accepts all strings a^n where $n \geq 1001$.
- Behavior on $n \leq 999$ is unspecified.

NFA B:
- Rejects a^{1000}.
- Accepts all strings a^n where $n \leq 999$.
- Behavior on $n \geq 1001$ is unspecified.

The final NFA for L is the **union** of NFA A and NFA B.

Construction of NFA A

NFA A

Frobenius Coin Problem: Sums of 32 and 33

Theorem:

Frobenius Coin Problem: Sums of 32 and 33

Theorem:

1. For all $n \geq 992$, there exist $x, y \in \mathbb{N}$ such that $n = 32x + 33y$.

Frobenius Coin Problem: Sums of 32 and 33

Theorem:

1. For all $n \geq 992$, there exist $x, y \in \mathbb{N}$ such that $n = 32x + 33y$.
2. There do not exist $x, y \in \mathbb{N}$ such that $991 = 32x + 33y$.

Frobenius Coin Problem: Sums of 32 and 33

Theorem:

1. For all $n \geq 992$, there exist $x, y \in \mathbb{N}$ such that $n = 32x + 33y$.
2. There do not exist $x, y \in \mathbb{N}$ such that $991 = 32x + 33y$.

Proof (by induction): Base Case: $992 = 32 \times 31 + 33 \times 0$.

Inductive Step: $n = 32x + 33y$

Hypothesis: $n - 1 = 32x' + 33y'$ for some $x', y' \in \mathbb{N}$.

Inductive Step: $n = 32x + 33y$

Hypothesis: $n - 1 = 32x' + 33y'$ for some $x', y' \in \mathbb{N}$.

Case 1: $x' \geq 1$. Then
$n = (n-1) + 1 = 32x' + 33y' + (33 - 32) = 32(x' - 1) + 33(y' + 1)$.

Inductive Step: $n = 32x + 33y$

Hypothesis: $n - 1 = 32x' + 33y'$ for some $x', y' \in \mathbb{N}$.

Case 1: $x' \geq 1$. Then
$n = (n-1) + 1 = 32x' + 33y' + (33 - 32) = 32(x' - 1) + 33(y' + 1)$.

Case 2: $x' = 0$. If $n - 1 = 33y' \geq 992$, then $y' \geq 31$. We use the identity $32 \times 32 - 31 \times 33 = 1$. Then
$n = 33y' + 32(32) - 33(31) = 32(32) + 33(y' - 31)$.

Inductive Step: $n = 32x + 33y$

Hypothesis: $n - 1 = 32x' + 33y'$ for some $x', y' \in \mathbb{N}$.

Case 1: $x' \geq 1$. Then
$n = (n-1) + 1 = 32x' + 33y' + (33 - 32) = 32(x' - 1) + 33(y' + 1)$.

Case 2: $x' = 0$. If $n - 1 = 33y' \geq 992$, then $y' \geq 31$. We use the identity $32 \times 32 - 31 \times 33 = 1$. Then
$n = 33y' + 32(32) - 33(31) = 32(32) + 33(y' - 31)$.

Case 3: $x' = 0$ and $y' \leq 30$. Then $n - 1 \leq 33 \times 30 = 990$, which contradicts $n - 1 \geq 992$.

Non-Representability of 991

Non-Representability of 991

Proof by contradiction: Assume $991 = 32x + 33y$ for some $x, y \in \mathbb{N}$.

Non-Representability of 991

Proof by contradiction: Assume $991 = 32x + 33y$ for some $x, y \in \mathbb{N}$.
Modulo 32, we have:

$$991 \equiv 32x + 33y \pmod{32}$$

Non-Representability of 991

Proof by contradiction: Assume $991 = 32x + 33y$ for some $x, y \in \mathbb{N}$.
Modulo 32, we have:

$$991 \equiv 32x + 33y \pmod{32}$$

$$31 \equiv 0x + 1y \pmod{32}$$

Non-Representability of 991

Proof by contradiction: Assume $991 = 32x + 33y$ for some $x, y \in \mathbb{N}$.
Modulo 32, we have:

$$991 \equiv 32x + 33y \pmod{32}$$

$$31 \equiv 0x + 1y \pmod{32}$$

$$y \equiv 31 \pmod{32} \implies y \geq 31$$

Non-Representability of 991

Proof by contradiction: Assume $991 = 32x + 33y$ for some $x, y \in \mathbb{N}$.
Modulo 32, we have:

$$991 \equiv 32x + 33y \pmod{32}$$

$$31 \equiv 0x + 1y \pmod{32}$$

$$y \equiv 31 \pmod{32} \implies y \geq 31$$

However, if $y \geq 31$, then $33y \geq 33 \times 31 = 1023$, which exceeds 991.
Contradiction.

Application to NFA A

Theorem:

1. For all $n \geq 1001$, there exist $x, y \in \mathbb{N}$ such that $n = 32x + 33y + 9$.
2. There do not exist $x, y \in \mathbb{N}$ such that $1000 = 32x + 33y + 9$.

Application to NFA A

Theorem:

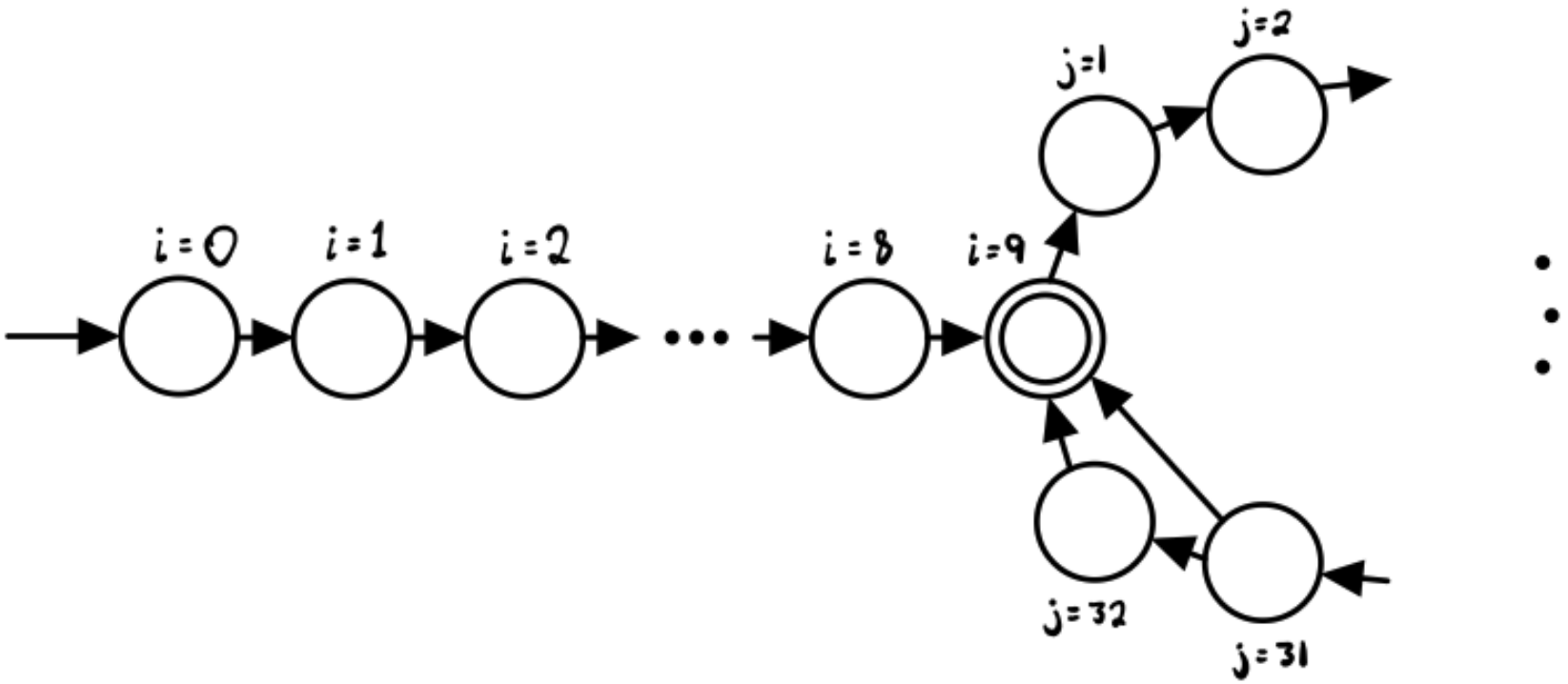
1. For all $n \geq 1001$, there exist $x, y \in \mathbb{N}$ such that $n = 32x + 33y + 9$.
2. There do not exist $x, y \in \mathbb{N}$ such that $1000 = 32x + 33y + 9$.

Proof:

1. If $n \geq 1001$, then $n - 9 \geq 992$. By the previous theorem, $n - 9 = 32x + 33y$.
2. If $1000 = 32x + 33y + 9$, then $991 = 32x + 33y$, which is impossible.

Structure of NFA A

Design: A start state followed by a chain of 9 states, leading into a loop structure that can represent any combination of 32 and 33 transitions.



State Count: 9 (chain) + 33 (loop) = **42 states**.

Construction of NFA B

NFA B

Modular Arithmetic Approach for NFA B

Modular Arithmetic Approach for NFA B

We use the fact that 1000 has specific residues modulo small primes:

Modular Arithmetic Approach for NFA B

We use the fact that 1000 has specific residues modulo small primes:

- $1000 \equiv 0 \pmod 2$. Let N_1 accept strings where $n \not\equiv 0 \pmod 2$.

Modular Arithmetic Approach for NFA B

We use the fact that 1000 has specific residues modulo small primes:

- $1000 \equiv 0 \pmod 2$. Let N_1 accept strings where $n \not\equiv 0 \pmod 2$.
- $1000 \equiv 1 \pmod 3$. Let N_2 accept strings where $n \not\equiv 1 \pmod 3$.

Modular Arithmetic Approach for NFA B

We use the fact that 1000 has specific residues modulo small primes:

- $1000 \equiv 0 \pmod 2$. Let N_1 accept strings where $n \not\equiv 0 \pmod 2$.
- $1000 \equiv 1 \pmod 3$. Let N_2 accept strings where $n \not\equiv 1 \pmod 3$.
- $1000 \equiv 0 \pmod 5$. Let N_3 accept strings where $n \not\equiv 0 \pmod 5$.

Modular Arithmetic Approach for NFA B

We use the fact that 1000 has specific residues modulo small primes:

- $1000 \equiv 0 \pmod 2$. Let N_1 accept strings where $n \not\equiv 0 \pmod 2$.
- $1000 \equiv 1 \pmod 3$. Let N_2 accept strings where $n \not\equiv 1 \pmod 3$.
- $1000 \equiv 0 \pmod 5$. Let N_3 accept strings where $n \not\equiv 0 \pmod 5$.
- $1000 \equiv 6 \pmod 7$. Let N_4 accept strings where $n \not\equiv 6 \pmod 7$.

Modular Arithmetic Approach for NFA B

We use the fact that 1000 has specific residues modulo small primes:

- $1000 \equiv 0 \pmod 2$. Let N_1 accept strings where $n \not\equiv 0 \pmod 2$.
- $1000 \equiv 1 \pmod 3$. Let N_2 accept strings where $n \not\equiv 1 \pmod 3$.
- $1000 \equiv 0 \pmod 5$. Let N_3 accept strings where $n \not\equiv 0 \pmod 5$.
- $1000 \equiv 6 \pmod 7$. Let N_4 accept strings where $n \not\equiv 6 \pmod 7$.
- $1000 \equiv 10 \pmod{11}$. Let N_5 accept strings where $n \not\equiv 10 \pmod{11}$.

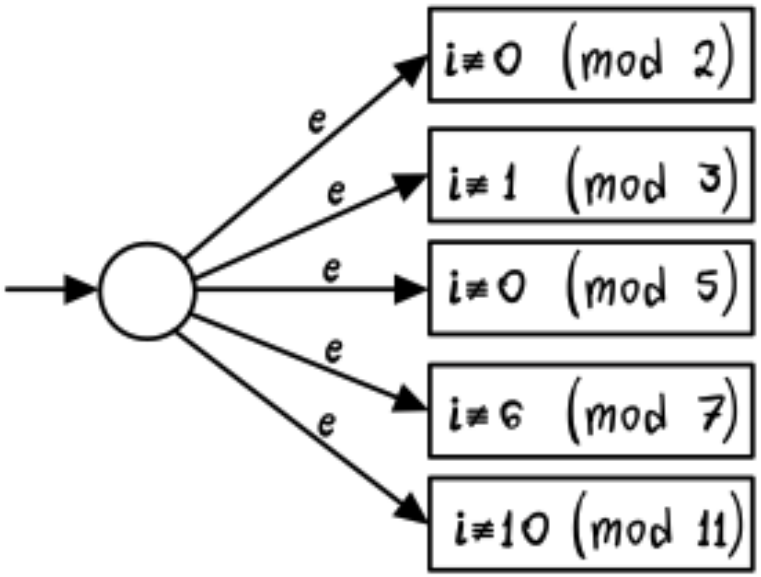
Modular Arithmetic Approach for NFA B

We use the fact that 1000 has specific residues modulo small primes:

- $1000 \equiv 0 \pmod 2$. Let N_1 accept strings where $n \not\equiv 0 \pmod 2$.
- $1000 \equiv 1 \pmod 3$. Let N_2 accept strings where $n \not\equiv 1 \pmod 3$.
- $1000 \equiv 0 \pmod 5$. Let N_3 accept strings where $n \not\equiv 0 \pmod 5$.
- $1000 \equiv 6 \pmod 7$. Let N_4 accept strings where $n \not\equiv 6 \pmod 7$.
- $1000 \equiv 10 \pmod{11}$. Let N_5 accept strings where $n \not\equiv 10 \pmod{11}$.

NFA B is the union of these modular machines.

Structure of NFA B



State Count: 1 (start) $+ 2 + 3 + 5 + 7 + 11 =$ **29states**.

Verification of NFA B

Theorem: NFA B rejects a^{1000} and accepts all a^i for $0 \leq i \leq 999$.

Verification of NFA B

Theorem: NFA B rejects a^{1000} and accepts all a^i for $0 \leq i \leq 999$.

Proof: If a^i is rejected by NFA B, it must be rejected by all component machines:

$$i \equiv 0 \pmod 2, \quad i \equiv 1 \pmod 3, \quad i \equiv 0 \pmod 5, \quad i \equiv 6 \pmod 7,$$

Verification of NFA B

Theorem: NFA B rejects a^{1000} and accepts all a^i for $0 \leq i \leq 999$.

Proof: If a^i is rejected by NFA B, it must be rejected by all component machines:

$$i \equiv 0 \pmod 2, \quad i \equiv 1 \pmod 3, \quad i \equiv 0 \pmod 5, \quad i \equiv 6 \pmod 7,$$

By the Chinese Remainder Theorem:

- $i \equiv 0 \pmod 2$ and $i \equiv 1 \pmod 3 \implies i \equiv 4 \pmod 6$
- $i \equiv 0 \pmod 5$ and $i \equiv 6 \pmod 7 \implies i \equiv 20 \pmod{35}$
- Combining with $i \equiv 10 \pmod{11}$ yields $i \equiv 1000 \pmod{2310}$.

Verification of NFA B

Theorem: NFA B rejects a^{1000} and accepts all a^i for $0 \leq i \leq 999$.

Proof: If a^i is rejected by NFA B, it must be rejected by all component machines:

$$i \equiv 0 \pmod 2, \quad i \equiv 1 \pmod 3, \quad i \equiv 0 \pmod 5, \quad i \equiv 6 \pmod 7,$$

By the Chinese Remainder Theorem:

- $i \equiv 0 \pmod 2$ and $i \equiv 1 \pmod 3 \implies i \equiv 4 \pmod 6$
- $i \equiv 0 \pmod 5$ and $i \equiv 6 \pmod 7 \implies i \equiv 20 \pmod{35}$
- Combining with $i \equiv 10 \pmod{11}$ yields $i \equiv 1000 \pmod{2310}$.

Thus, any rejected string must have length $i \geq 1000$. For $i \leq 999$, a^i must be accepted.

Summary: NFA for $\{a^i : i \neq 1000\}$

1. **NFA A** (42 states): Accepts strings ≥ 1001, rejects 1000.
2. **NFA B** (29 states): Accepts strings ≤ 999, rejects 1000.

Summary: NFA for $\{a^i : i \neq 1000\}$

1. **NFA A** (42 states): Accepts strings ≥ 1001, rejects 1000.
2. **NFA B** (29 states): Accepts strings ≤ 999, rejects 1000.

The union NFA uses:

$$42 + 29 - 1(\text{shared start state}) = \textbf{70states}.$$

Summary: NFA for $\{a^i : i \neq 1000\}$

1. **NFA A** (42 states): Accepts strings ≥ 1001, rejects 1000.
2. **NFA B** (29 states): Accepts strings ≤ 999, rejects 1000.

The union NFA uses:

$$42 + 29 - 1(\text{shared start state}) = \textbf{70states}.$$

Moral: Lower bounds are difficult to prove because they must account for non-obvious algorithmic techniques.

Complexity Theory Perspective

- This construction highlights the gap between determinism (DFA) and nondeterminism (NFA).
- Even in a simple context like a one-letter alphabet, nondeterministic "guessing" or modular decomposition can lead to exponential savings.
- This relates to the broader question of **P vs NP**: we cannot easily rule out clever shortcuts for complex problems.

Conclusion

- The study of state complexity provides insights into the power of nondeterminism.
- Modular arithmetic and number theory (Frobenius Theorem, CRT) are powerful tools for automaton design.
- The difficulty of proving tight lower bounds reflects the fundamental challenges in complexity theory.