Formal Languages and Automata

DFA Minimization

Ryan Stansifer

Computer Sciences
Florida Institute of Technology
Melbourne, Florida USA 32901

`http://www.cs.fit.edu/~ryan/`

2 March 2024

Equivalence of DFAs and NFAs

Linz 6th, Section 2.3, pages 58–65
HMU 3rd, Section 2.3.5, pages 60–64
Martin 2nd, Theorem 4.1, page 105–106
Du & Ko 2001, Section 2.4, pages 45–53
Greenlaw & Hoover 1998, Section 4.5, page 107–115
Floyd & Beigel 1994, Section 4.5, pages 247–258
Powerset construction at Wikipedia

Minimization of DFAs

Linz 6th, Section 2.4, pages 66–71
HMU 3rd, Section 4.4, pages 155–165
Kozen 1997, Lecture 13 & 14, pages 77–88
Du & Ko 2001, Section 2.7, pages 69–78
Floyd & Beigel 1994, Section 4.7, pages 258–279
DFA minimization at Wikipedia

Minimization of DFAs

There are different appoarches.

- Hopcroft's algorithm (1971) based on partition refinement
- Moore's algorithm
- Brzozowki's (1962) based on repeating the subset construction

Any DFA defines one language, For a particular language there are many DFA's that accept it.

For reasons of simplicity, a DFA with the fewest number of state may be preferred.

A DFA is minimal if it satisfied two properties:

1. Every state is reachable: for all $q \in Q$ there exists a $w \in \Sigma^*$ such that $\delta^*(q, w) = s$.
2. Every pair of states is distinguishable: for all $q, r \in Q$ such that $q \neq r$ implies there exists $w \in \Sigma^*$ such that $\delta^*(q, w) \in F$ iff $\delta^*(r, w) \notin F$.

Definition

A state $p \in Q$ of a DFA $\langle Q, \Sigma, \delta, q_0, F \rangle$ is said to be *accessible* or *reachable* if for some $w \in \Sigma^*$ it is the case the $\delta^*(q_0, w) = p$.

Definition

A state $p \in Q$ of a DFA $\langle Q, \Sigma, \delta, q_0, F \rangle$ is said to be *inaccessible* if for all $w \in \Sigma^*$ it is the case that $\delta^*(q_0, w) \neq p$.

```
Reach  := {q0} -- start state is reachable
Next   := {q0} -- it has been newly added
loop
  Next   := { δ(q,c) for q ∈ Next for c ∈ Σ} \ Reach ;
  Reach := Reach ∪ Next ;
  exit when Next is empty ;
end loop;
UnReach = Q \ Reach
```

Definition

The states p and q of a DFA $\langle Q, \Sigma, \delta, q_0, F \rangle$ are said to be *indistinguishable*, written $p \approx q$, if for all $x \in \Sigma^*$

$$\delta^*(p, x) \in F \leftrightarrow \delta^*(q, x) \in F$$

If two states are not indistinguishable, then they are distinguishable. Or, equivalently, we may define the following:

Definition

The states p and q of a DFA $\langle Q, \Sigma, \delta, q_0, F \rangle$ are said to be *distinguishable*, written $p \not\approx q$, if for some $x \in \Sigma^*$ one of these equivalent statements hold

$$\delta^*(p, x) \in F \textbf{ xor } \delta^*(q, x) \in F \tag{1}$$

$$\delta^*(p, x) \notin F \leftrightarrow \delta^*(q, x) \in F \tag{2}$$

$$\delta^*(p, x) \in F \leftrightarrow \delta^*(q, x) \notin F \tag{3}$$

Theorem

The relation $p \approx q$ of indistinguishable states is an equivalence relation, i.e., it is reflexive, symmetric, and transitive.

Proof.

It is obviously reflexive and symmetric. That it is transitive is proved by contradiction. Suppose $p \approx q$ and $q \approx r$, but p and r are distinguishable by some string w. Suppose that $\delta^*(p, w) \in F$ and so $\delta^*(r, w)$ must not be in F. Since $p \approx q$, $\delta^*(q, w) \in F$. This contradicts the fact that q and r are indistinguishable. Similarly if $\delta^*(p, w) \notin F$. We conclude $p \approx r$. $\square$

The significance of this is that the indistinguishable relation partitions the set of states of a DFA into equivalence classes.

Notation

We write $[p]_{\approx}$ for the set $\{q \in Q \mid p \approx q\}$

Definition

The DFA $M/\approx$ is defined from a given DFA M as $\langle Q', \Sigma, \delta', q', F' \rangle$ where

- $Q' = \{[p]_\approx \mid p \in Q\}$
- $q' = [q]_\approx$
- $F' = \{[p]_\approx \mid p \in F\}$
- $\delta'([p]_\approx, a) = [\delta(p, a)]_\approx$ for all $a \in \Sigma$

The DFA $M/\approx$ is well-defined since for all $a \in \Sigma$ and all states in Q

$$p \approx q \Rightarrow \delta(p, a) \approx \delta(q, a)$$

Theorem

The language $L(M/\approx) = L(M)$.

Proof.

Let δ be the transition function and q_0 the start state in both (sorry), M and $M/\approx$. Then:

$$
\begin{aligned}
x \in L(M/\approx) \quad & \text{iff} \quad \delta^*(q_0, x) \in F' \\
& \text{iff} \quad [\delta^*(q_0, x)]_\approx \in F' \\
& \text{iff} \quad \delta^*(q_0, x) \in F \\
& \text{iff} \quad x \in L(M)
\end{aligned}
$$

$\square$

DFA Minimization Algorithm

Three algorithms for DFA minimization:

1. Hopcroft's partition refinement
2. Brzozowski: reverse edges, convert to DFA, and do it again
3. Moore's algorithm

Minimization Algorithm

For a DFA $\langle Q, \Sigma, \delta, q_0, F \rangle$:

1. Remove inaccessible state
2. For every pair of states, mark whether or not they are distinguishable.
3. Collapse indistinguishable states.

The states of the minimized DFA are non-empty, pairwise-disjoint subsets of the original DFA.

Marking Algorithm

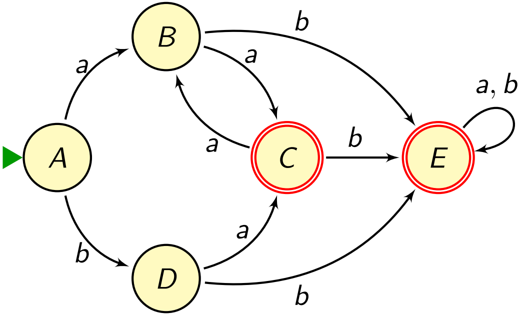
For a DFA $\langle Q, \Sigma, \delta, q_0, F \rangle$ systematically try longer and longer strings to establish that a pair of states is distinguishable.

1. Mark all unordered pairs $\{p, q\} \in Q \times Q$ as indistinguishable.
2. Mark $\{p, q\}$ as distinguishable, if $p \in F$ **xor** $q \in F$.
3. Repeat until no further changes: mark $\{p, q\}$ as distinguishable, if $\{\delta(p, a), \delta(q, a)\}$ is distinguishable for some $a \in \Sigma$.

DFA Minimization
An Example
Combining Indistinquishable States
Linz 6th, Example 2.18, page 69

Minimize a DFA

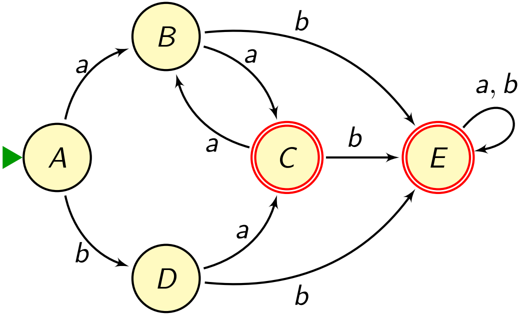
Find a DFA equivalent to the one below with the minimum number of states.



Q	Σ	Q
A	a	B
A	b	D
B	a	C
B	b	E
C	a	B
C	b	E
D	a	C
D	b	E
E	a	E
E	b	E

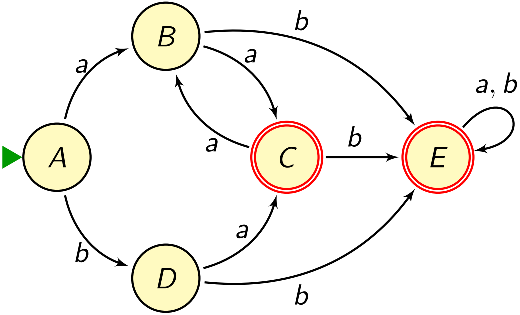
An example DFA. Linz 6th, Figure 2.18, page 69.

Minimize a DFA



A				
	B			
		C		
			D	
				E

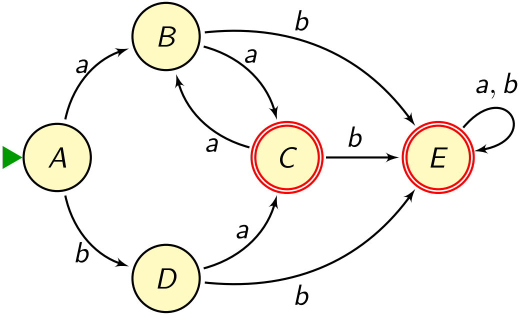
Minimize a DFA



A	$\approx$			
	B			
		C		
			D	
				E

A, B: A and B are both non-final states.

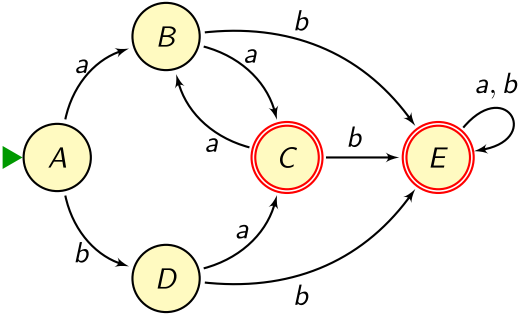
Minimize a DFA



A	$\approx$	$\times$		
	B			
		C		
			D	
				E

A, C: A is not final, but C is final.

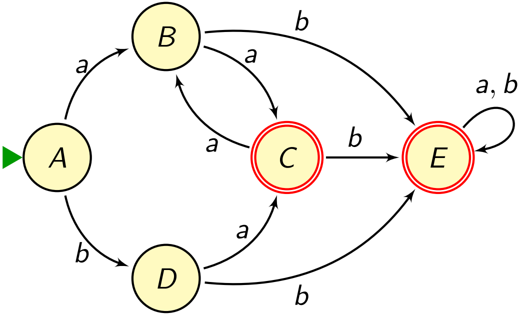
Minimize a DFA



A	≈	×	≈	
	B			
		C		
			D	
				E

A, D: A and D are both not final.

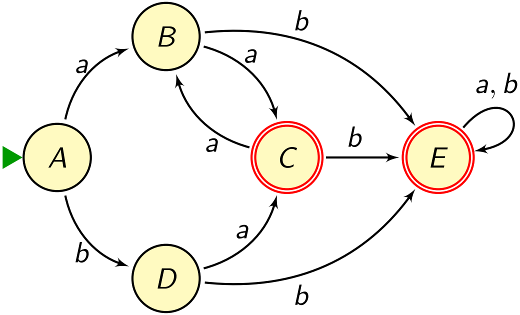
Minimize a DFA



A	≈	×	≈	×
	B			
		C		
			D	
				E

A, E: A is not final, but E is final.

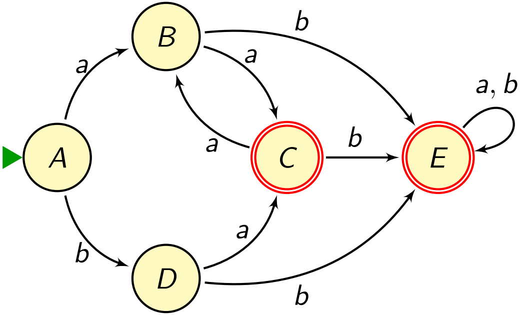
Minimize a DFA



A	≈	×	≈	×
	B	×		
		C		
			D	
				E

B, C: B is not final, but C is final.

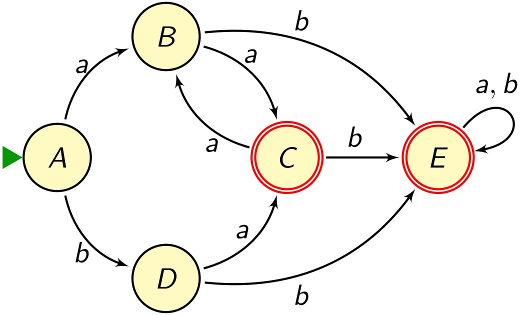
Minimize a DFA



A	$\approx$	$\times$	$\approx$	$\times$
B		$\times$	$\approx$	
C				
D				
E				

B, D: B and D are both not final.

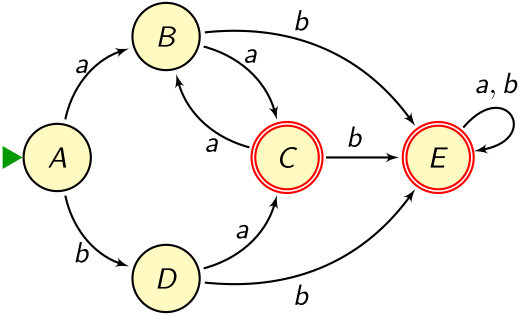
Minimize a DFA



A	≈	×	≈	×
	B	×	≈	×
		C		
			D	
				E

B, E: B is final, but E is not final.

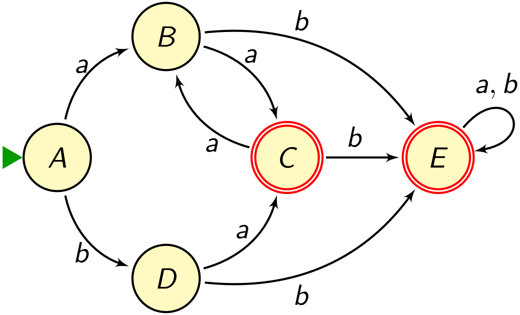
Minimize a DFA



A	≈	×	≈	×
	B	×	≈	×
		C	×	
			D	
				E

C, D: C is final but D is not final.

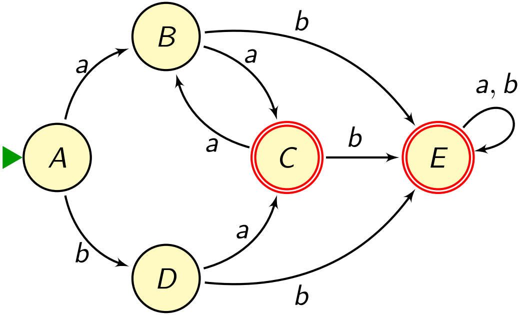
Minimize a DFA



	B	C	D	E
A	$\approx$	$\times$	$\approx$	$\times$
B		$\times$	$\approx$	$\times$
C			$\times$	$\approx$
D				
E				

C, E: Both C and E are final.

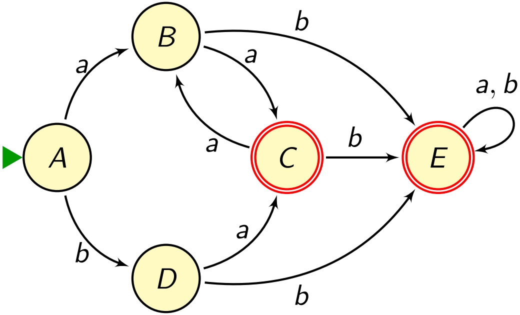
Minimize a DFA



	B	C	D	E
A	≈	×	≈	×
B		×	≈	×
C			×	≈
D				×
E				

D, E: D is not final, but E is final.

Minimize a DFA

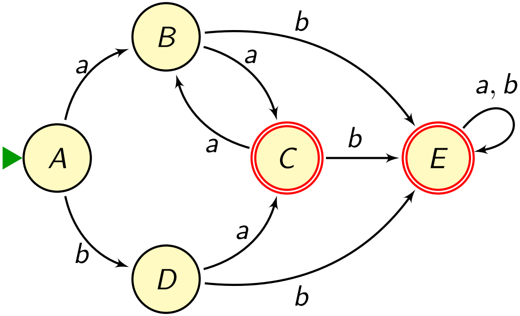


A	≈	×	≈	×
	B	×	≈	×
		C	×	≈
			D	×
				E

The resulting partition: $\{A, B, D\}, \{C, E\}$. The non-final versus the final states.

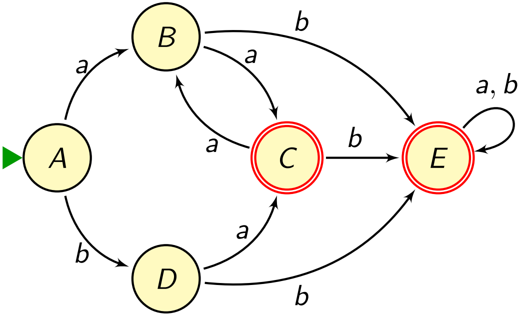
Minimize a DFA

All pairs which were marked distinguishable earlier, remain distinguishable from then on. All the others are re-examined to see if they might become distinguishable.



	B	C	D	E
A		×		×
B		×		×
C			×	
D				×
E				

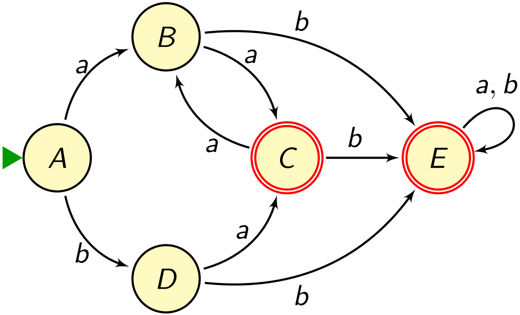
Minimize a DFA



	B	C	D	E
A	×	×		×
B		×		×
C			×	
D				×

A, B: $a \in \Sigma$ distinguishes A from B, as $\delta(A, a) = B \notin F$ but $\delta(B, a) = C \in F$.

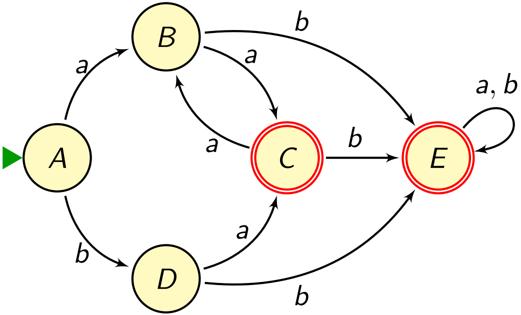
Minimize a DFA



A	×	×	×	×
	B	×		×
		C	×	
			D	×
				E

A, D: $a \in \Sigma$ distinguishes A from D, as $\delta(A, a) = B \notin F$ but $\delta(D, a) = C \in F$.

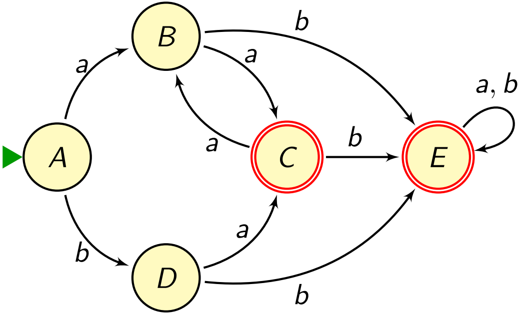
Minimize a DFA



	B	C	D	E
A	×	×	×	×
B		×	≈	×
C			×	
D				×

B and D are indistinguishable, as $\delta(B, a) = \delta(D, a) = C$ and $\delta(B, b) = \delta(D, b) = E$.

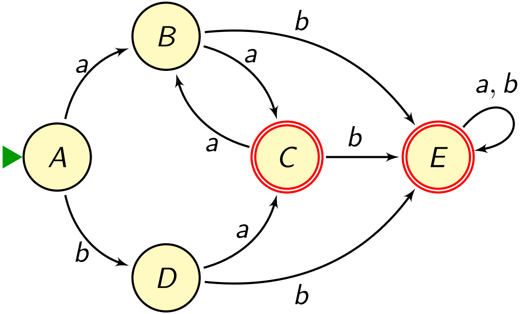
Minimize a DFA



	B	C	D	E
A	×	×	×	×
B		×	≈	×
C			×	×
D				×
E				

C, E: $a \in \Sigma$ distinguishes C from E, as $\delta(C, a) = B$ and $\delta(E, a) = E$ and $B \not\approx E$.

Minimize a DFA

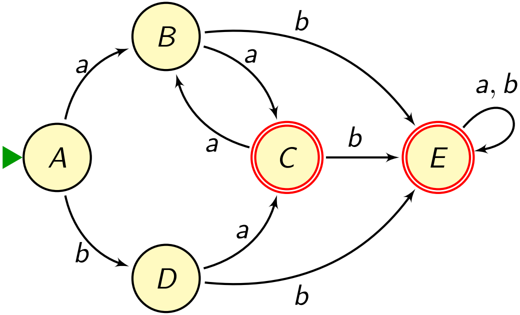


A	×	×	×	×
	B	×	≈	×
		C	×	×
			D	×
				E

The resulting partition: $\{A\}, \{B, D\}, \{C\}, \{E\}$

Minimize a DFA

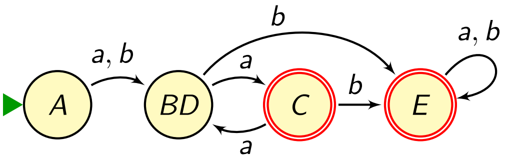
Repeating the step (for strings of length two) creates no changes. The partition $\{A\}, \{B, D\}, \{C\}, \{E\}$ cannot be further refined. The states B and D go to the same states, and so can never be distinguised.



A	×	×	×	×
	B	×	≈	×
		C	×	×
			D	×
				E

Minimize a DFA (Solution)

The minimized DFA has a state for each equivalence class produced by the marking algorithm. (One state fewer.) The equivalence class with the original start state is the start state of the minimized DFA. An equivalence class of final states in the original DFA becomes a final state of the minimize DFA.



Q	Σ	Q
A	a	BD
A	b	BD
BD	a	C
BD	b	E
C	a	BD
C	b	E
E	a	E
E	b	E

DFA Minimization

An Example

Combining Indistinquishable States

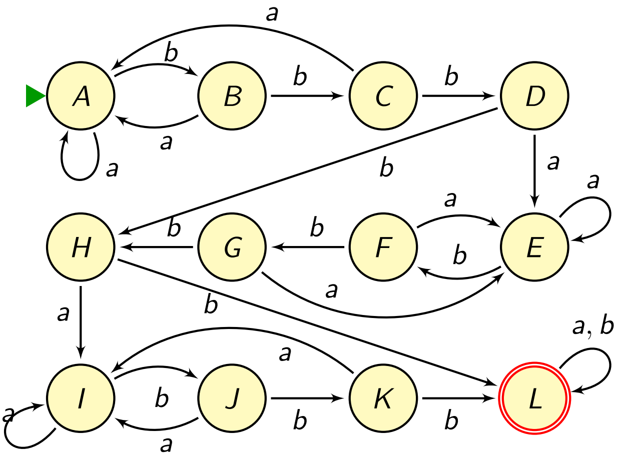
From notes from Univ of Innsbruck

Example

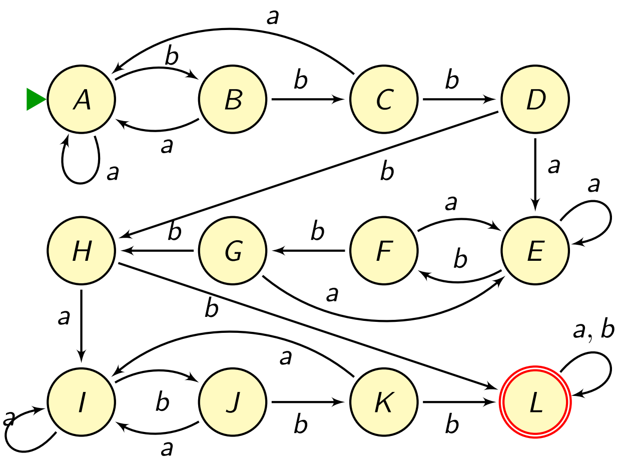
Design a DFA over $\{a, b\}$ containing at least three occurrences of three consecutive b's, overlapping permitted.

Creating a DFA for this language is not so hard. Then minimize it.

Here is a DFA over $\{a, b\}$ containing at least three occurrences of three consecutive b's, overlapping permitted.



Now find an equivalent DFA with the minimum number of states.

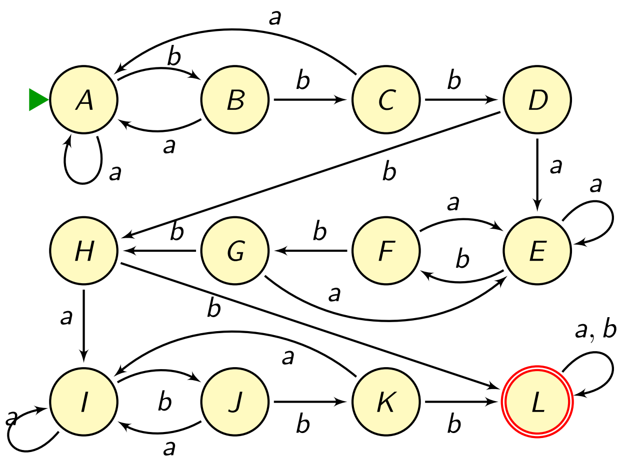


	B	C	D	E	F	G	H	I	J	K	L
A	≈	≈	≈	≈	≈	≈	≈	≈	≈	≈	×
B		≈	≈	≈	≈	≈	≈	≈	≈	≈	×
C			≈	≈	≈	≈	≈	≈	≈	≈	×
D				≈	≈	≈	≈	≈	≈	≈	×
E					≈	≈	≈	≈	≈	≈	×
F						≈	≈	≈	≈	≈	×
G							≈	≈	≈	≈	×
H								≈	≈	≈	×
I									≈	≈	×
J										≈	×
K											×

The first iteration merely separates the final from the non-final states. The resulting partition:

$$\{A, B, C, D, E, F, G, H, I, J, K\}, \{L\}$$

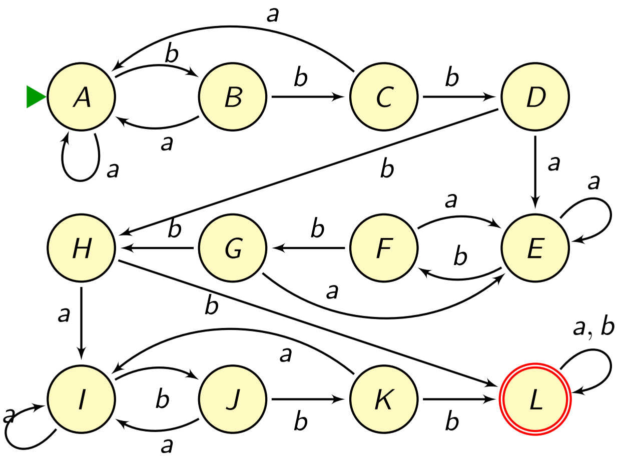
Second Iteration



	B	C	D	E	F	G	H	I	J	K	L
A	≈	≈	≈	≈	≈	≈		≈	≈		×
B		≈	≈	≈	≈	≈		≈	≈		×
C			≈	≈	≈	≈		≈	≈		×
D				≈	≈	≈		≈	≈		×
E					≈	≈		≈	≈		×
F						≈		≈	≈		×
G								≈	≈		×
H											×
I									≈		×
J											×
K											×

for all $s, s' \in \{A, B, C, D, E, F, G, I, J\}$ and for $\star \in a, b$, $\delta(s, \star)$ and $\delta(s', \star)$ are non-final.

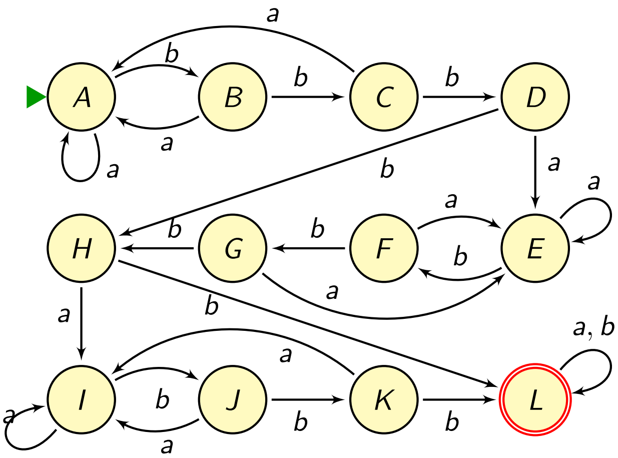
Second Iteration



A	≈	≈	≈	≈	≈	≈	×	≈	≈	×	×
B		≈	≈	≈	≈	≈	×	≈	≈	×	×
C			≈	≈	≈	≈	×	≈	≈	×	×
D				≈	≈	≈	×	≈	≈	×	×
E					≈	≈	×	≈	≈	×	×
F						≈	×	≈	≈	×	×
G							×	≈	≈	×	×
H								×	×	≈	×
I									≈	×	×
J										×	×
K											×
											L

$\delta(H, b)$ and $\delta(K, b)$ is final, but for $s' \in \{A \cdots G, I, J\}$, $\delta(s', b)$ is non-final.

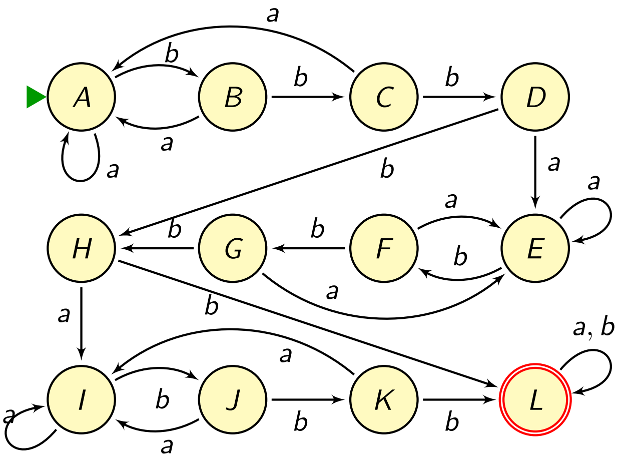
Second Iteration



	B	C	D	E	F	G	H	I	J	K	L
A	≈	≈	≈	≈	≈	≈	×	≈	≈	×	×
B		≈	≈	≈	≈	≈	×	≈	≈	×	×
C			≈	≈	≈	≈	×	≈	≈	×	×
D				≈	≈	≈	×	≈	≈	×	×
E					≈	≈	×	≈	≈	×	×
F						≈	×	≈	≈	×	×
G							×	≈	≈	×	×
H								×	×	≈	×
I									≈	×	×
J										×	×
K											×
L											

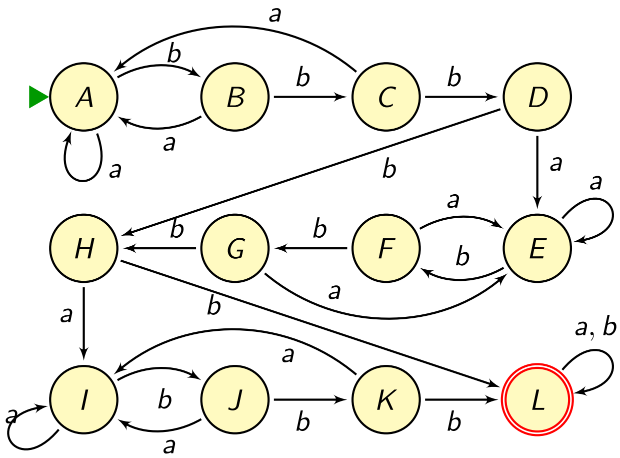
The resulting partition: $\{A, B, C, D, E, F, G, I, J\}, \{H, K\}, \{L\}$

Third Iteration $\{A, B, C, D, E, F, G, I, J\}, \{H, K\}, \{L\}$



	B	C	D	E	F	G	H	I	J	K	L
A							×			×	×
B							×			×	×
C							×			×	×
D							×			×	×
E							×			×	×
F							×			×	×
G							×			×	×
H								×	×		×
I										×	×
J										×	×
K											×
L											

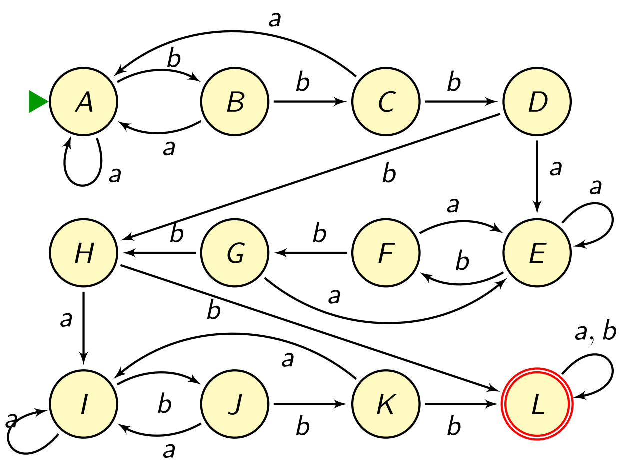
Third Iteration $\{A, B, C, D, E, F, G, I, J\}, \{H, K\}, \{L\}$



	B	C	D	E	F	G	H	I	J	K	L
A	≈	≈		≈	≈		×	≈		×	×
B		≈		≈	≈		×	≈		×	×
C				≈	≈		×	≈		×	×
D				≈	≈		×	≈		×	×
E					≈	≈	×	≈	≈	×	×
F						≈	×	≈	≈	×	×
G							×	≈		×	×
H								×	×	≈	×
I										×	×
J										×	×
K											×
L											

A, B, C, E, F, I are all indistinquishable.

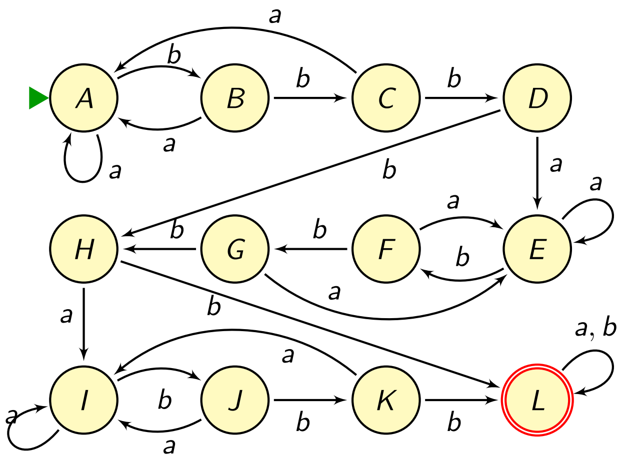
Third Iteration $\{A, B, C, D, E, F, G, I, J\}, \{H, K\}, \{L\}$



A	≈	≈		≈	≈		×	≈		×	×
	B	≈		≈	≈		×	≈		×	×
		C		≈	≈		×	≈		×	×
			D	≈	≈	≈	×	≈	≈	×	×
				E	≈	≈	×	≈	≈	×	×
					F	≈	×	≈	≈	×	×
						G	×	≈	≈	×	×
							H	×	×	≈	×
								I		×	×
									J	×	×
										K	×
											L

A, B, C, E, F, I are all indistinquishable. D, G, J are all indistinquishable.

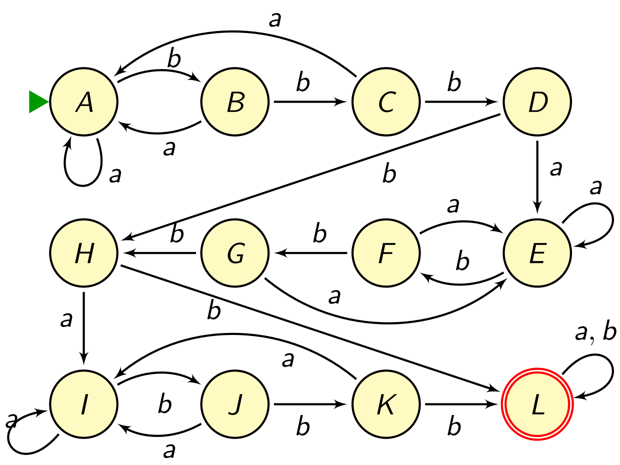
Third Iteration $\{A, B, C, D, E, F, G, I, J\}, \{H, K\}, \{L\}$



	B	C	D	E	F	G	H	I	J	K	L
A	≈	≈		≈	≈		×	≈		×	×
B		≈		≈	≈		×	≈		×	×
C				≈	≈		×	≈		×	×
D				≈	≈	≈	×	≈	≈	×	×
E					≈	≈	×	≈	≈	×	×
F						≈	×	≈	≈	×	×
G							×	≈	≈	×	×
H								×	×	≈	×
I										×	×
J										×	×
K											×
L											

A, B, C, E, F, I are all indistinquishable. D, G, J are all indistinquishable. The resulting partition: $\{A, B, C, E, F, I\}, \{D, G, J\}, \{H, K\}, \{L\}$

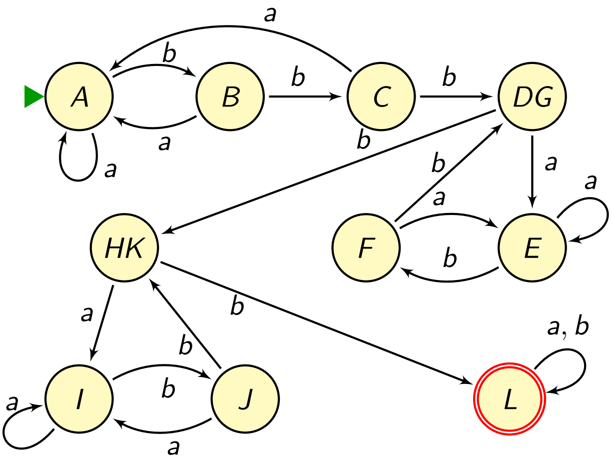
Sixth And Final Iteration



	B	C	D	E	F	G	H	I	J	K	L
A	×	×	×	×	×	×	×	×	×	×	×
B		×	×	×	×	×	×	×	×	×	×
C			×	×	×	×	×	×	×	×	×
D				×	×	≈	×	×	×	×	×
E					×	×	×	×	×	×	×
F						×	×	×	×	×	×
G							×	×	×	×	×
H								×	×	≈	×
I									×	×	×
J										×	×
K											×

The resulting partition: $\{A\}, \{B\}, \{C\}, \{E\}, \{F\}, \{D, G\}, \{I\}, \{J\}, \{H, K\}, \{L\}$

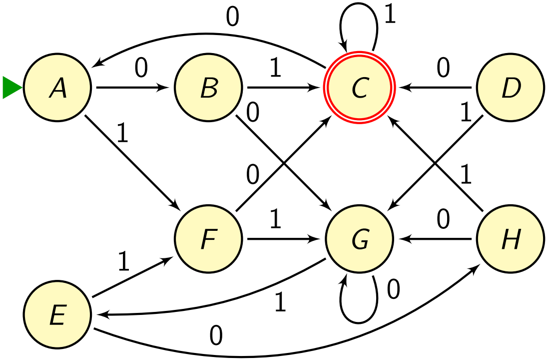
Minimum DFA



The state DG mean that the first set of three consecutive b's has been seen, and two of the three b's in the second set have been seen.
The state HK mean that the second set of three consecutive b's has been seen, and two of the three b's in the third and final set have been seen.

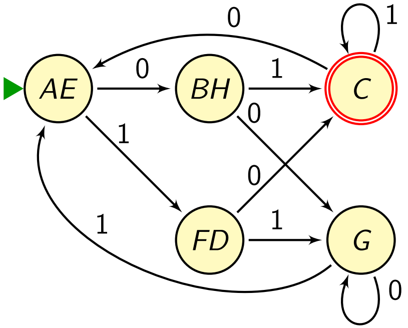
DFA Minimization
An Example
Combining Indistinquishable States
HMU 3rd, Example 4.18, page 156

The example ignores an inaccessible state which, in this case, goes way.



Q	Σ	Q
A	0	B
A	1	F
B	0	G
B	1	C
C	0	A
C	1	C
D	0	C
D	1	G
E	0	H
E	1	F
F	0	C
F	1	G
G	0	G
G	1	E
H	0	G
H	1	C

An example DFA. HMU 4.8.



The final partition $\{A, E\}, \{B, H\}, \{C\}, \{F, D\}, \{G\}$.

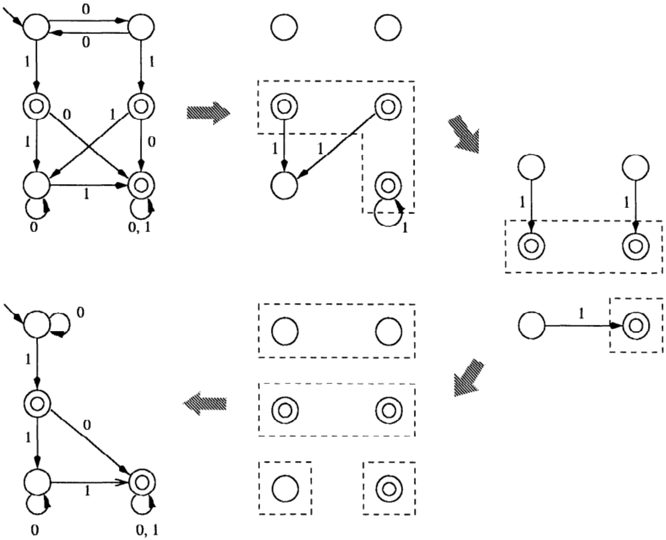


Figure 2.51: The third method.

Ding-Zhu Du & Ker-I Ko, Figure 2.51, page 77

Theorem

Let M be a minimal DFA for $L \subset \Sigma^*$. Making the non-final states final and the final states non-final results in minimal DFA M' for $\bar{L}$.

Proof.

The proof is by contradition. If M' were not minimal, there there would be another DFA M'' with few states. And it's complement would be a DFA for L. This DFA would have few states than M and that is a contradition. $\square$