Deterministic Finite Automata

Lecture 5
We say that $M$ accepts $w \in \Sigma^*$ if $M$, on input $w$, starting from the start state $s$, reaches an accepting state

i.e., $\delta^*(s,w) \in A$

$L(M)$ is the set of all strings accepted by $M$

i.e., $L(M) = \{ w \mid \delta^*(s,w) \in A \}$

Called the language accepted by $M$
Input Accepted by a DFA

What kind of language is accepted by FSM?

- Automatic (it is an automaton after all)!

- We will use: REGULAR (not a coincidence)

Language is regular iff

- it is accepted by a finite state automaton

- it is described by a regular expression
Warning

“$M$ accepts language $L$” does not mean simply that $M$ accepts each string in $L$.

“$M$ accepts language $L$” means $M$ accepts each string in $L$ and no others!

$L(M) = L$
Examples: What is $L(M)$?

odd #0 and odd #1

$0^*11^*$

$(A+B)^*ABBA$
Building DFAs
State = Memory

First, decide on $Q$

The state of a DFA is its entire memory of what has come before

The state must capture enough information to complete the computation on the suffix to come

When designing a DFA, think “what do I need to know at this moment?” That is your state.
DFA Construction Exercise

$L(M) = \{w \mid w \text{ contains } 00 \}$

Is it regular?? $(0+1)^*00(0+1)^*$

What should be in the memory?

\[ s \rightarrow a \]
DFA Construction Exercise

$L(M) = \{ w \mid w \text{ contains } 00 \}$

Is it regular?? $(0+1)^*00(0+1)^*$

What should be in the memory?

![DFA Diagram]

1

s

a
DFA Construction Exercise

$L(M) = \{ w \mid w \text{ contains 00}\}$

Is it regular??

$(0+1)^*00(0+1)^*$

What should be in the memory?

![DFA Diagram]

1

s

0

a

1

1
DFA Construction Exercise

$L(M) = \{ w \mid w \text{ contains } 00 \}$

Is it regular?? $(0+1)^*00(0+1)^*$

What should be in the memory?

![DFA Diagram]
DFA Construction Exercise

\[ L(M) = \{ w \mid w \text{ contains } 00 \} \]

Is it regular??

\[(0+1)^*00(0+1)^*\]

What should be in the memory?

![DFA Diagram]

- **s**: States
- **a**: States
- **b**: States
- **1**: Transition on 1
- **0**: Transition on 0
- **0,1**: Transition on 0 and 1
DFA Construction Exercise

$L(M) = \{ w \mid w \text{ contains } 00 \}$

- **s**: I haven’t seen a 00, previous symbol was 1 or undefined.

- **a**: I haven’t seen a 00, previous symbol was a 0

- **b**: I have seen a 00

- We have exhausted of all strings. Either accepted (with 00) or not.
DFA construction

• Make sure you interpret all the cases!

• How about design a DFA for $L(M) = \{w | w \text{ contains } 001100110011111001101101\}$?

• There is algorithm to minimize the DFA, but when you are asked to do it, try to be clear versus succinct.

• Try to be “stupid”, do brute force!!!

• When you are just trying to prove that a language is regular — $\rightarrow$ DFA for the language exists. Write an algorithm like we did for multiple of 5!
A More Complicated example

\[ L(M) = \{w \mid w \text{ contains } 00 \text{ and then } 11\} \]
A More Complicated example

\[ L(M) = \{ w \mid w \text{ contains 00} \} \]
A More Complicated example

\[ L(M) = \{ w \mid w \text{ contains 11} \} \]
A More Complicated example

$L(M) = \{ w \mid w \text{ contains 00 and then 11} \}$
A More Complicated example

\[ L(M) = \{w \mid w \text{ contains } 00 \text{ and then } 11 \} \]
A More Complicated example

\[ L(M) = \{ w \mid w \text{ contains 00 and then 11} \} \]
A More Complicated example

\[ L(M) = \{ w \mid w \text{ contains } 00 \text{ and then } 11 \} \]
A More Complicated example

$L(M) = \{w \mid w \text{ contains 00 and then 11}\}$
A More Complicated example

$L(M) = \{w \mid w \text{ contains } 00 \text{ and then } 11\}$
A More Complicated example

$L(M) = \{ w \mid w \text{ contains 00 and then 11} \}$
A More Complicated example

$L(M) = \{ w \mid w \text{ contains 00 and then 11} \}$

- If $A$ and $B$ are regular, then $AB$ is regular. Does the same hold for DFA?
A More Complicated example

\[ L(M) = \{ w \mid w \text{ contains 00 and then 11} \} \]

• If A and B are regular, then AB is regular.

• Does the same hold for DFA?

• NO! you cannot glue two DFAs together in general like that.
What about the complement?

$L(M) = \{ w \mid w \text{ contains no } 11 \text{ after } 00 \}$
What about the complement?

$L(M) = \{ w \mid w \text{ contains no } 11 \text{ after } 00 \}$
What about the complement?

$L(M) = \{w \mid w \text{ contains no 11 after 00}\}$

- If $L$ is regular, then $\Sigma^* \setminus L$ is regular
- Make the accepting states into non-accepting
\[ L = \{ w : w \text{ contains } 00 \text{ and } 11 \}? \]

- I want to build a machine that decides if a string contains two zeroes in a row AND two ones in a row.
- I want to run both machines at the same time.
- At the end of the string, if I am on the accept state for machine 1 AND on the accept state for machine 2, then I accept.
- How many states total?
$L = \{ w: w \text{ contains 00 and } 11 \}$?

$L(M_1)$ contains 00

$L(M_2)$ contains 11
\[ L = \{ w : w \text{ contains } 00 \text{ and } 11 \} ? \]

\[ L(M_1) \text{ contains } 00 \]

\[ L(M_2) : \text{contains } 11 \]
$L = \{ w : \text{w contains 00 and 11}\}$?

$L(M_1)$ contains 00

$L(M_2)$ contains 11
\[ L = \{w: w \text{ contains } 00 \text{ and } 0, 11\}? \]
L = \{ w: w contains 00 and 11 \}? 

$L(M_1)$ contains 00 

$L(M_2)$ contains 11
L = \{w: w contains 00 and_{0,11}\}? 

$L(M_1)$ contains 00 

$L(M_2)$ :contains 11
L = \{w: w contains 00 and 11\}?
L = \{w: w contains 00 and 0,11\}?
$L = \{ w: w \text{ contains 00 and 11} \}$?

$L(M_1)$ contains 00

$L(M_2)$ contains 11
L = \{w: w contains 00 and \text{ or } 11\}? 

$L(M_1)$ contains 00 

$L(M_2)$ contains 11
$L = \{ w : w \text{ contains 00 and 11} \}$?

$L(M_1)$ contains 00

$L(M_2)$ contains 11
The Product Construction

Formally, given two DFAs

\[ M_1 = (\Sigma, Q_1, s_1, A_1, \delta_1) \] and \[ M_2 = (\Sigma, Q_2, s_2, A_2, \delta_2) \]

Where \( M_1 \) accepts \( L_1 \)

\( M_2 \) accepts \( L_2 \)

\[ M = (\Sigma, Q, s, A, \delta) \] accepts \( L_1 \cap L_2 \)

\[ Q = Q_1 \times Q_2, \quad s = (s_1, s_2) \]

\[ A = \{(q_1, q_2): q_1 \in A_1 \text{ and } q_2 \in A_2\} \]

\[ \delta: (Q_1 \times Q_2) \times \Sigma \rightarrow Q_1 \times Q_2 \]

\[ \delta( (q_1, q_2), a ) = ( \quad , \quad ) \]
The Product Construction

Formally, given two DFAs

\[ M_1 = (\Sigma, Q_1, s_1, A_1, \delta_1) \] and \[ M_2 = (\Sigma, Q_2, s_2, A_2, \delta_2) \]

Where \( M_1 \) accepts \( L_1 \)

\( M_2 \) accepts \( L_2 \)

\[ M = (\Sigma, Q, s, A, \delta) \] accepts \( L_1 \cap L_2 \)

\[ Q = Q_1 \times Q_2, \quad s = (s_1, s_2) \]

\[ A = \{(q_1, q_2): q_1 \in A_1 \text{ and } q_2 \in A_2\} \]

\[ \delta: (Q_1 \times Q_2) \times \Sigma \to Q_1 \times Q_2 \]

\[ \delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a)) \]
$L = \{w: w \text{ contains } 00 \text{ and } 011\}$?
$L = \{w: w \text{ contains 00 and 0,11}\}$?
L = \{w: w contains 00 and 11\}?
The Product Construction

Formally, given two DFAs

\[ M_1 = (\Sigma, Q_1, s_1, A_1, \delta_1) \text{ and } M_2 = (\Sigma, Q_2, s_2, A_2, \delta_2) \]

Where \( M_1 \) accepts \( L_1 \)

\( M_2 \) accepts \( L_2 \)

\[ L_1 \cup L_2 \]

\[ M = (\Sigma, Q, s, A, \delta) \text{ accepts } L_1 \land L_2 \]

\[ Q = Q_1 \times Q_2, \ s = (s_1, s_2) \]

\[ A = \{(q_1, q_2): q_1 \in A_1 \text{ and } q_2 \in A_2\} \]

\[ \delta: (Q_1 \times Q_2) \times \Sigma \rightarrow Q_1 \times Q_2 \]

\[ \delta( (q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a)) \]
The Product Construction

Formally, given two DFAs

\[ M_1 = (\Sigma, Q_1, s_1, A_1, \delta_1) \] and \[ M_2 = (\Sigma, Q_2, s_2, A_2, \delta_2) \]

Where \( M_1 \) accepts \( L_1 \)

\( M_2 \) accepts \( L_2 \)

\[ M = (\Sigma, Q, s, A, \delta) \] accepts \( L_1 \cap L_2 \)

\[ Q = Q_1 \times Q_2, \ s = (s_1, s_2) \]

\[ A = \{(q_1, q_2) : q_1 \in A_1 \ \text{or} \ q_2 \in A_2 \} \]

\[ \delta: (Q_1 \times Q_2) \times \Sigma \rightarrow Q_1 \times Q_2 \]

\[ \delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a)) \]
The Product Construction: Question

\[ M_1 = (\Sigma, Q_1, s_1, A_1, \delta_1) \text{ and } M_2 = (\Sigma, Q_2, s_2, A_2, \delta_2) \]

Where \( M_1 \) accepts \( L_1 \)

\( M_2 \) accepts \( L_2 \)

\[ M = (\Sigma, Q, s, A, \delta) \text{ accepts } L_1 \setminus L_2 \]

\( Q = Q_1 \times Q_2, \ s = (s_1, s_2) \)

\( A = \{ \} \) ?

\( \delta: (Q_1 \times Q_2) \times \Sigma \rightarrow Q_1 \times Q_2 \)

\( \delta( (q_1, q_2), a) = ( \ , \ ) \) ?
The Product Construction: Question

\[ M_1 = (\Sigma, Q_1, S_1, A_1, \delta_1) \text{ and } M_2 = (\Sigma, Q_2, S_2, A_2, \delta_2) \]

Where \( M_1 \) accepts \( L_1 \)

\( M_2 \) accepts \( L_2 \)

\[ M = (\Sigma, Q, S, A, \delta) \text{ accepts } L_1 \setminus L_2 \]

\[ Q = Q_1 \times Q_2, \ s = (s_1, s_2) \]

\[ A = \{ (q_1, q_2) : q_1 \in A_1 \text{ but not } q_2 \in A_2 \} \]

\[ \delta : (Q_1 \times Q_2) \times \Sigma \rightarrow Q_1 \times Q_2 \]

\[ \delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a)) \]
Closure Properties of Regular Languages

• Union: trivial for regular expressions, easy for DFAs via product

• Complement: easy for DFAs, hard for regular expressions

• Intersection: easy for DFAs via product, hard for regular expressions

• Difference: easy for DFAs via product, hard for regular expressions

• Concatenation: easy for regular expressions, hard for DFA’s