

Let's look back at our example for the change machine

- We wanted to know if we could get exact change in quarters from a jar of nickels and dimes.
- This is similar to computing the sum of the change mod 25.

Let's look back at our example for the change machine

- We wanted to know if we could get exact change in quarters from a jar of nickels and dimes.
- This is similar to computing the sum of the change mod 25.

Thinking in terms of state of mind:

Let's look back at our example for the change machine

- We wanted to know if we could get exact change in quarters from a jar of nickels and dimes.
- This is similar to computing the sum of the change mod 25.

Thinking in terms of state of mind:

- We need to keep track of the previous value of the sum.

Let's look back at our example for the change machine

- We wanted to know if we could get exact change in quarters from a jar of nickels and dimes.
- This is similar to computing the sum of the change mod 25.

Thinking in terms of state of mind:

- We need to keep track of the previous value of the sum.
- Look at our next coin that is inputted.

Let's look back at our example for the change machine

- We wanted to know if we could get exact change in quarters from a jar of nickels and dimes.
- This is similar to computing the sum of the change mod 25.

Thinking in terms of state of mind:

- We need to keep track of the previous value of the sum.
- Look at our next coin that is inputted.
- Compute the new sum mod 25 and move to that new state of mind.

The Setup

- Since we are working with values mod 25, and only allowing additions of nickels(5) and dimes(10), we need 5 states.

The Setup

- Since we are working with values mod 25, and only allowing additions of nickels(5) and dimes(10), we need 5 states.
- Each state will represent a possible value of the total mod 25, so we have 0, 5, 10, 15, 20.

The Setup

- Since we are working with values mod 25, and only allowing additions of nickels(5) and dimes(10), we need 5 states.
- Each state will represent a possible value of the total mod 25, so we have 0, 5, 10, 15, 20.

The Formalization

The Setup

- Since we are working with values mod 25, and only allowing additions of nickels(5) and dimes(10), we need 5 states.
- Each state will represent a possible value of the total mod 25, so we have 0, 5, 10, 15, 20.

The Formalization

- Our set of states will be 5 states representing each possible value.

$$Q = \{s_{\{0\}}, s_{\{5\}}, s_{\{10\}}, s_{\{15\}}, s_{\{20\}}\}$$

The Setup

- Since we are working with values mod 25, and only allowing additions of nickels(5) and dimes(10), we need 5 states.
- Each state will represent a possible value of the total mod 25, so we have 0, 5, 10, 15, 20.

The Formalization

- Our set of states will be 5 states representing each possible value.
 $Q = \{s_{\{0\}}, s_{\{5\}}, s_{\{10\}}, s_{\{15\}}, s_{\{20\}}\}$
- The alphabet we are looking at is going to contain only symbols representing nickels and dimes, so $\Sigma = \{N, D\}$.

The Setup

- Since we are working with values mod 25, and only allowing additions of nickels(5) and dimes(10), we need 5 states.
- Each state will represent a possible value of the total mod 25, so we have 0, 5, 10, 15, 20.

The Formalization

- Our set of states will be 5 states representing each possible value.
 $Q = \{s_{\{0\}}, s_{\{5\}}, s_{\{10\}}, s_{\{15\}}, s_{\{20\}}\}$
- The alphabet we are looking at is going to contain only symbols representing nickels and dimes, so $\Sigma = \{N, D\}$.
- To transition between states, if we add a nickel our sum increases by 5 mod 25 and a dime increases the sum by 10 mod 25. This will be shown in the state diagram.

The Setup

- Since we are working with values mod 25, and only allowing additions of nickels(5) and dimes(10), we need 5 states.
- Each state will represent a possible value of the total mod 25, so we have 0, 5, 10, 15, 20.

The Formalization

- Our set of states will be 5 states representing each possible value.
 $Q = \{s_{\{0\}}, s_{\{5\}}, s_{\{10\}}, s_{\{15\}}, s_{\{20\}}\}$
- The alphabet we are looking at is going to contain only symbols representing nickels and dimes, so $\Sigma = \{N, D\}$.
- To transition between states, if we add a nickel our sum increases by 5 mod 25 and a dime increases the sum by 10 mod 25. This will be shown in the state diagram.
- The initial sum we will assume is 0, so our initial state $q_0 = s_{\{0\}}$.

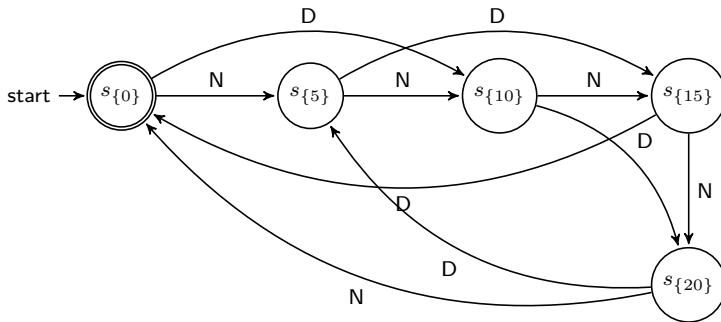
The Setup

- Since we are working with values mod 25, and only allowing additions of nickels(5) and dimes(10), we need 5 states.
- Each state will represent a possible value of the total mod 25, so we have 0, 5, 10, 15, 20.

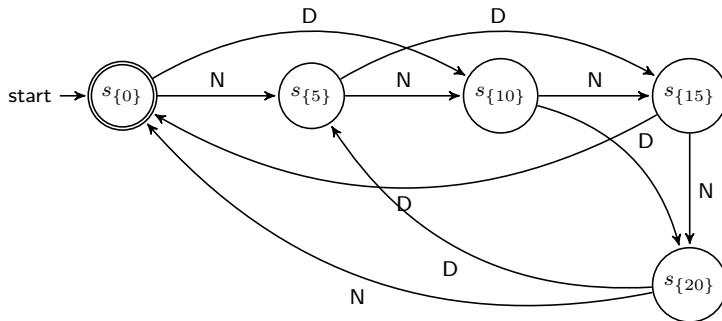
The Formalization

- Our set of states will be 5 states representing each possible value.
 $Q = \{s_{\{0\}}, s_{\{5\}}, s_{\{10\}}, s_{\{15\}}, s_{\{20\}}\}$
- The alphabet we are looking at is going to contain only symbols representing nickels and dimes, so $\Sigma = \{N, D\}$.
- To transition between states, if we add a nickel our sum increases by 5 mod 25 and a dime increases the sum by 10 mod 25. This will be shown in the state diagram.
- The initial sum we will assume is 0, so our initial state $q_0 = s_{\{0\}}$.
- The final states will only consist of the state where we have received exact change from the machine (sum is 0), so $F = \{s_{\{0\}}\}$.

Taking our formal definition and converting it into a state diagram produces the following:



Taking our formal definition and converting it into a state diagram produces the following:



Let us call this DFA machine M.

Some things to note:

- The starting state is labeled with an arrow pointing to it.

Some things to note:

- The starting state is labeled with an arrow pointing to it.
- The final/accepting states are denoted by a double circle around the state.

Some things to note:

- The starting state is labeled with an arrow pointing to it.
- The final/accepting states are denoted by a double circle around the state.
- The arrows between states are each labeled with a symbol from the alphabet and represent where to go to depending on the symbol that is read.

Some things to note:

- The starting state is labeled with an arrow pointing to it.
- The final/accepting states are denoted by a double circle around the state.
- The arrows between states are each labeled with a symbol from the alphabet and represent where to go to depending on the symbol that is read.
- Note: since this is a DFA, each symbol appears exactly once as output from each state (cannot have 2 or more locations or have no location that a symbol transitions to)

Let's look at the computation of this machine on some different input strings.

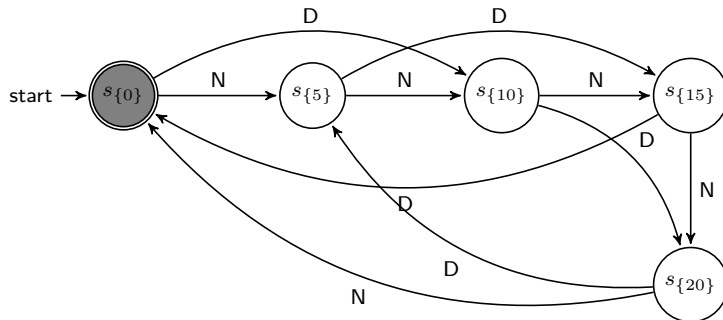
- $w_1 = \text{NNNNDDDD}$
- $w_2 = \text{NDDDD}$
- $w_3 = \lambda$

The way the computation works is that the machine starts in the initial state observing the first character. Based on these observations it transitions to a new state and reads the next character. This continues until no characters remain to be read.

Remember that λ is the empty string and contains no characters. So the computation on λ simply starts the machine and if an accepting/final state can be reached without having to do anything then it is accepted.

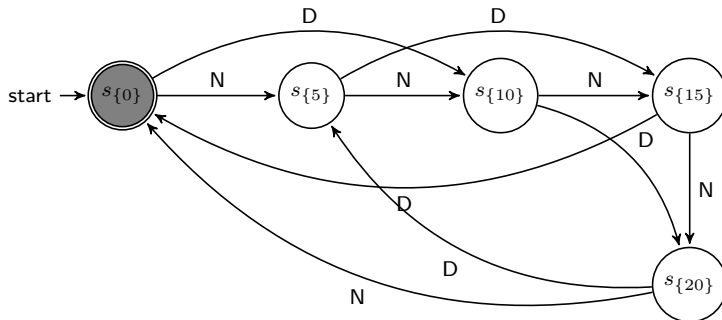
CSE 596 – Automaton Examples

Current state = $s_{\{0\}}$, $w_1 = \text{N N N N D D D D}$



CSE 596 – Automaton Examples

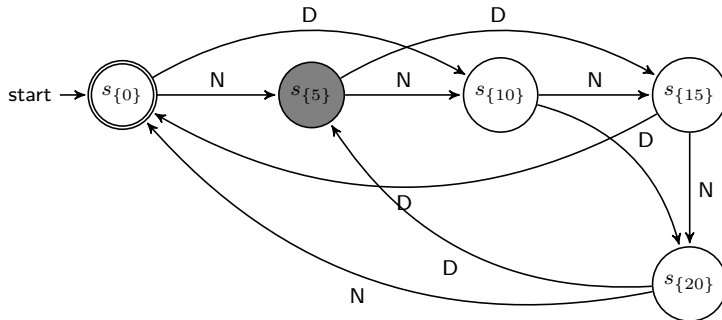
Current state = $s_{\{0\}}$, $w_1 = \text{N N N N D D D D}$



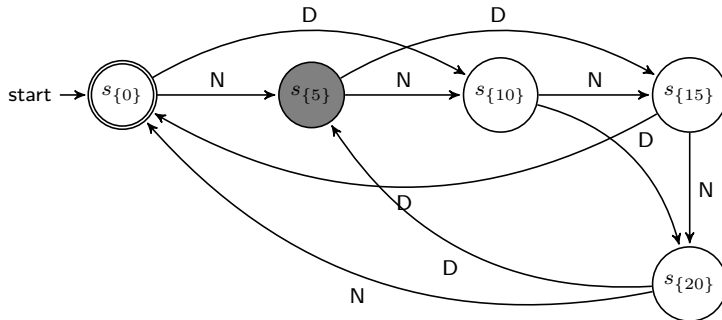
Given this information, we are looking at the first symbol in w_1 which is N and we are in state $s_{\{0\}}$. From our diagram, our transition function takes this as input and outputs a new state. That is, $\delta(s_{\{0\}}, N) = s_{\{5\}}$. So we will transition to state $s_{\{5\}}$ and read the next character in w_1 .

CSE 596 – Automaton Examples

Current state = $s_{\{5\}}$, $w_1 = \text{N} \color{red}{\text{N}} \text{N} \text{N} \text{D} \text{D} \text{D} \text{D}$



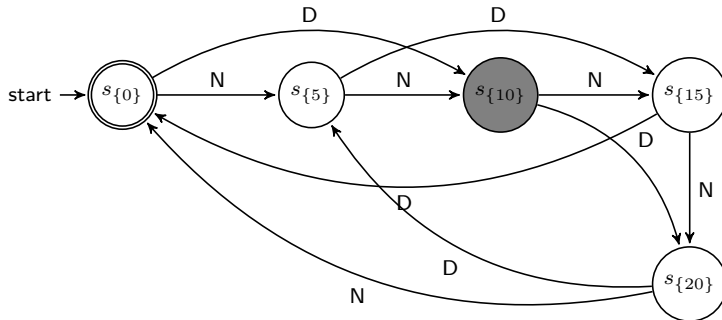
Current state = $s_{\{5\}}$, $w_1 = \text{N} \color{red}{\text{N}} \text{N} \text{N} \text{D} \text{D} \text{D} \text{D}$



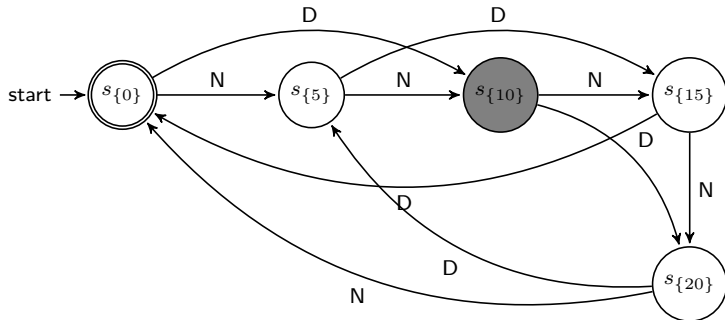
Given this information, $\delta(s_{\{5\}}, \text{N}) = s_{\{10\}}$.

CSE 596 – Automaton Examples

Current state = $s_{\{10\}}$, $w_1 = \text{NNND}$



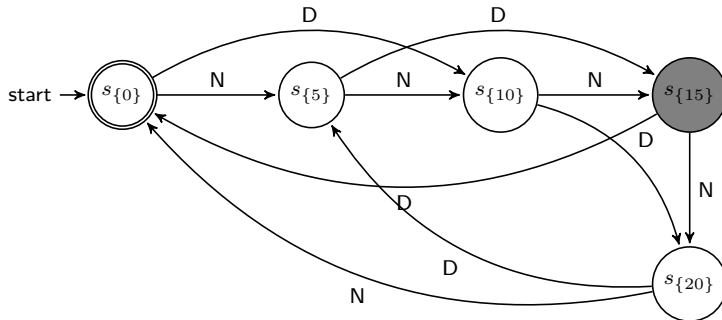
Current state = $s_{\{10\}}$, $w_1 = \text{NNND}$



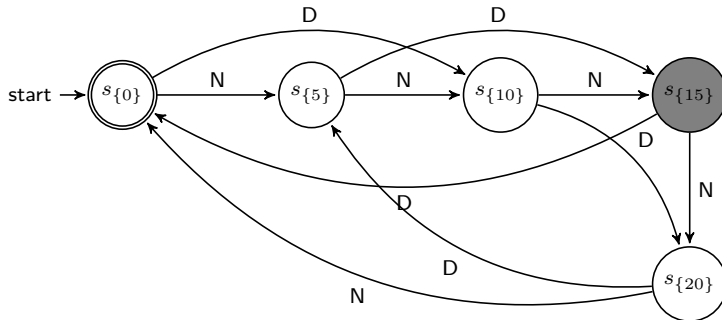
Given this information, $\delta(s_{\{10\}}, N) = s_{\{15\}}$.

CSE 596 – Automaton Examples

Current state = $s_{\{15\}}$, $w_1 = \text{NNN} \color{red}{\text{N}} \text{DDD}$



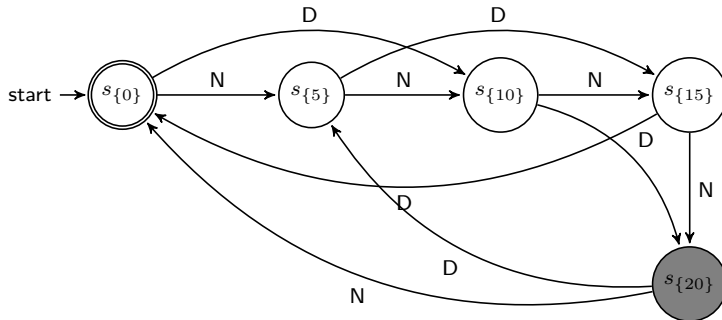
Current state = $s_{\{15\}}$, $w_1 = \text{NNN} \color{red}{\text{N}} \text{DDD}$



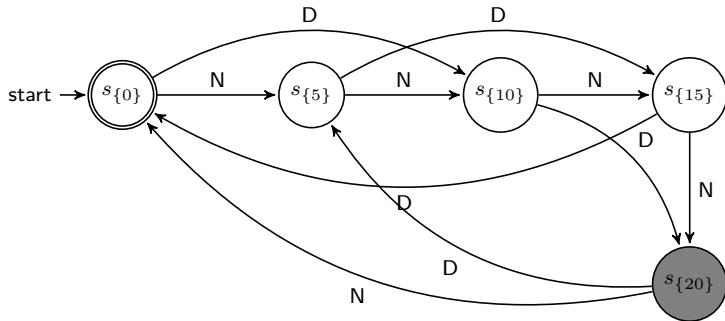
Given this information, $\delta(s_{\{15\}}, N) = s_{\{20\}}$.

CSE 596 – Automaton Examples

Current state = $s_{\{20\}}$, $w_1 = \text{NNNN}\color{red}{D}\text{DD}$



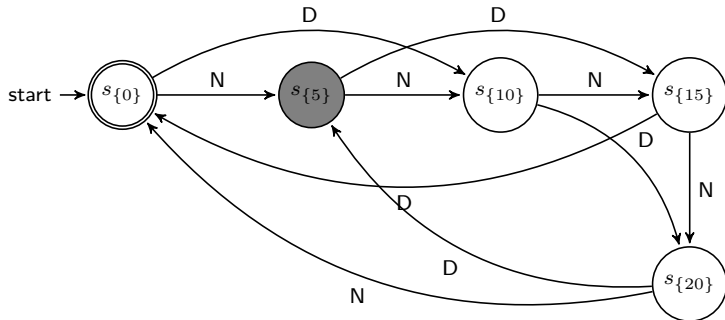
Current state = $s_{\{20\}}$, $w_1 = \text{NNNNDDDD}$



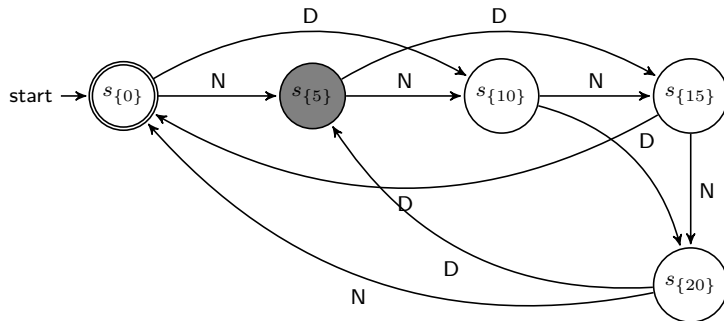
Given this information, $\delta(s_{\{20\}}, D) = s_{\{5\}}$.

CSE 596 – Automaton Examples

Current state = $s_{\{5\}}$, $w_1 = \text{NNNND}\color{red}{D}D$



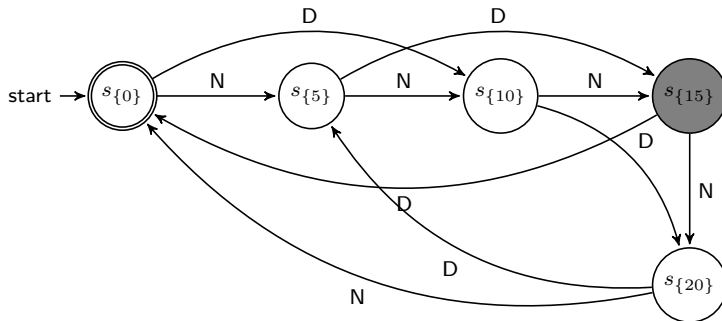
Current state = $s_{\{5\}}$, $w_1 = \text{NNNND}\color{red}{D}D$



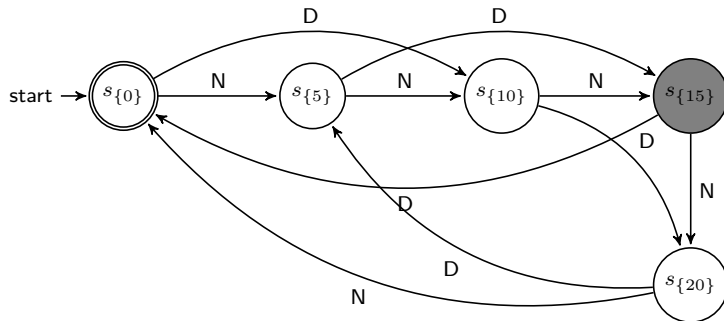
Given this information, $\delta(s_{\{5\}}, D) = s_{\{15\}}$.

CSE 596 – Automaton Examples

Current state = $s_{\{15\}}$, $w_1 = \text{NNNDDDD}$



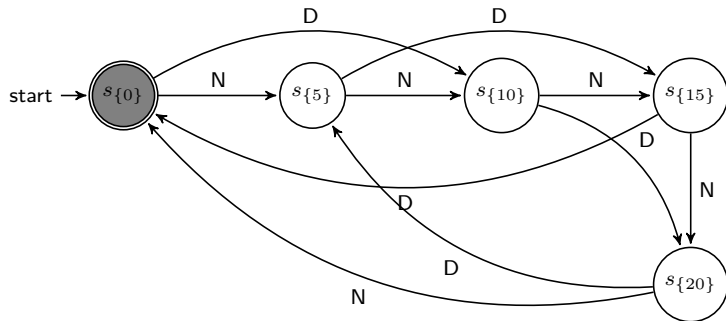
Current state = $s_{\{15\}}$, $w_1 = \text{NNNDDDD}$



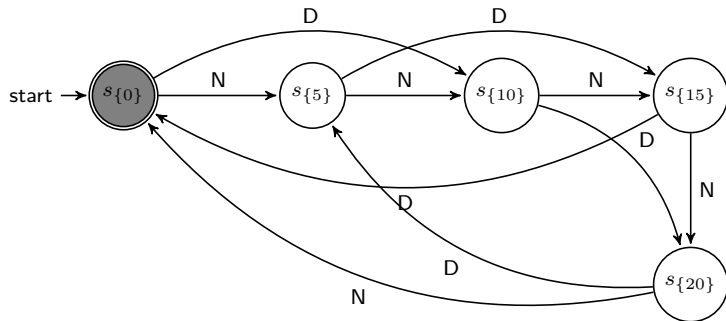
Given this information, $\delta(s_{\{15\}}, D) = s_{\{0\}}$.

CSE 596 – Automaton Examples

Current state = $s_{\{0\}}$, $w_1 = \text{NNNDDDD}$



Current state = $s_{\{0\}}$, $w_1 = \text{NNNDDDD}$



Given this information, there are no more symbols to be read. We are in state $s_{\{0\}} \in F$, so the string w_1 is accepted by this DFA M .

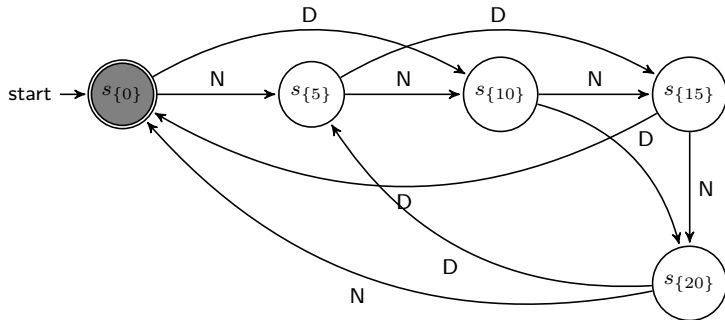
Since there was an accepting computation of w_1 on M , we say w_1 belongs to the language of this DFA M , so $w_1 \in L(M)$.

Since there was an accepting computation of w_1 on M , we say w_1 belongs to the language of this DFA M , so $w_1 \in L(M)$.

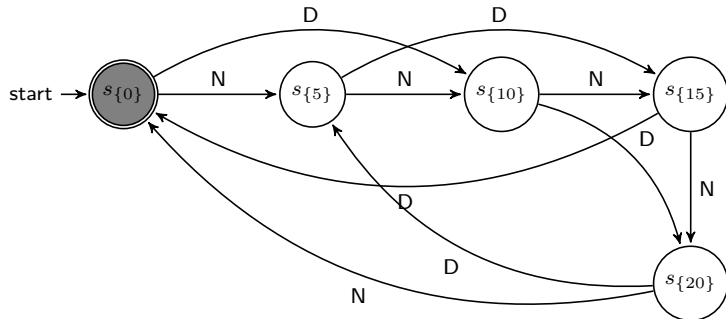
Now we can look at another string, $w_2 = NDDD$, and see if w_2 belongs to $L(M)$.

CSE 596 – Automaton Examples

Current state = $s_{\{0\}}$, $w_2 = \mathbf{N}$ DDD



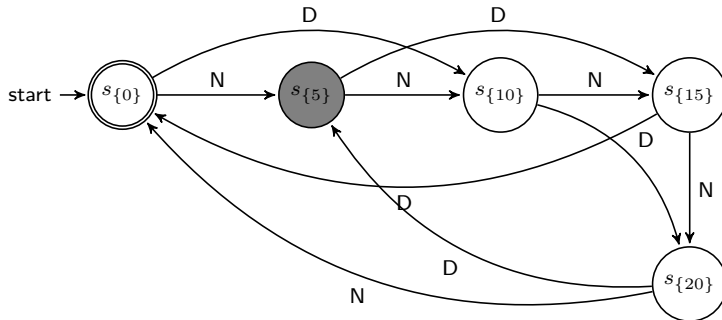
Current state = $s_{\{0\}}$, $w_2 = \mathbf{N}$ DDD



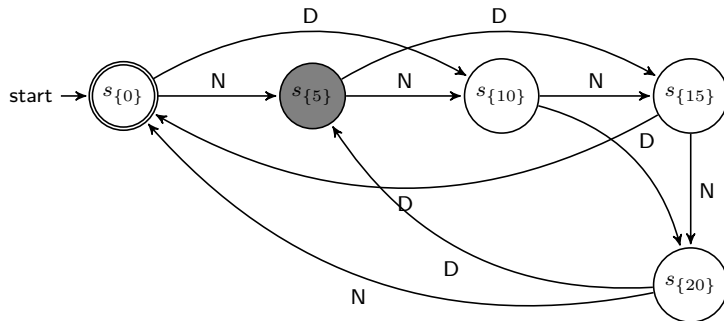
Given this information, we are looking at the first symbol in w_2 which is N and we are in state $s_{\{0\}}$. Our transition function takes this as input and gives us $\delta(s_{\{0\}}, N) = s_{\{5\}}$. So we will transition to state $s_{\{5\}}$ and read the next character in w_2 .

CSE 596 – Automaton Examples

Current state = $s_{\{5\}}$, $w_2 = \text{N}\color{red}{\text{D}}\text{DD}$



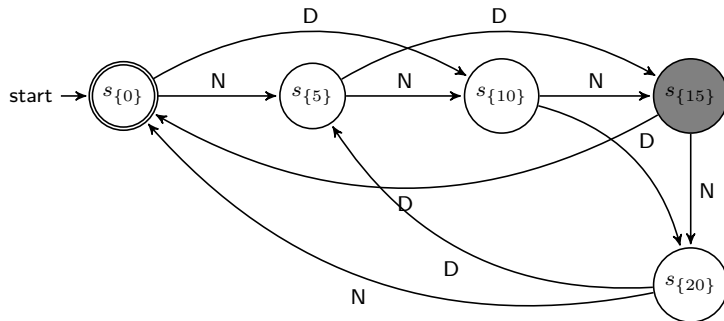
Current state = $s_{\{5\}}$, $w_2 = \text{NDDDD}$



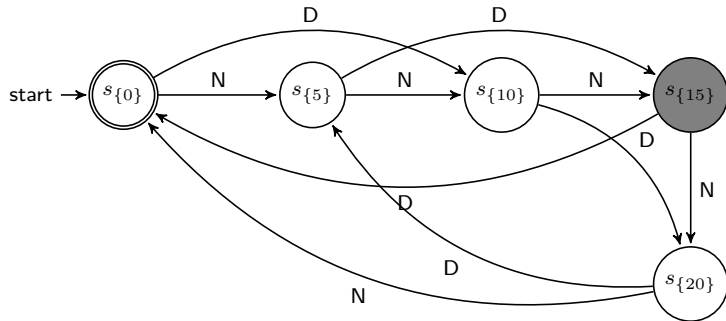
Given this information, $\delta(s_{\{5\}}, D) = s_{\{15\}}$.

CSE 596 – Automaton Examples

Current state = $s_{\{15\}}$, $w_2 = \text{ND}\color{red}{D}D$



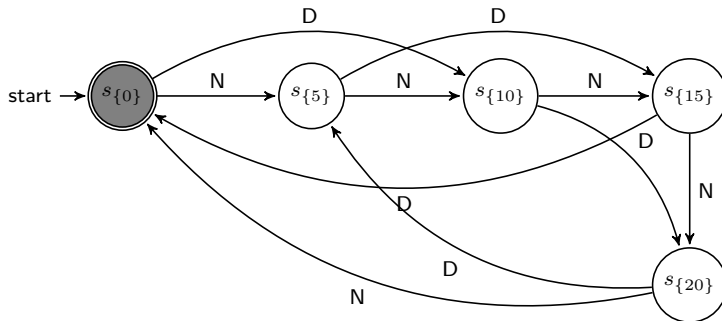
Current state = $s_{\{15\}}$, $w_2 = \text{ND}\color{red}{D}D$



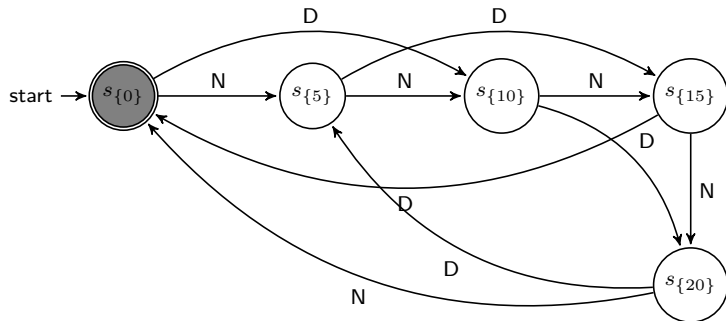
Given this information, $\delta(s_{\{15\}}, D) = s_{\{0\}}$.

CSE 596 – Automaton Examples

Current state = $s_{\{0\}}$, $w_2 = \text{NDDD}$



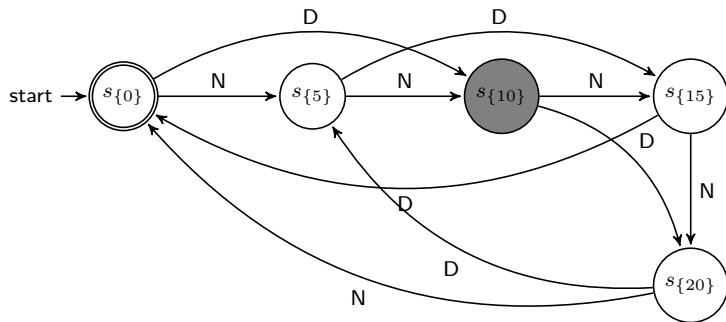
Current state = $s_{\{0\}}$, $w_2 = \text{NDDD}$



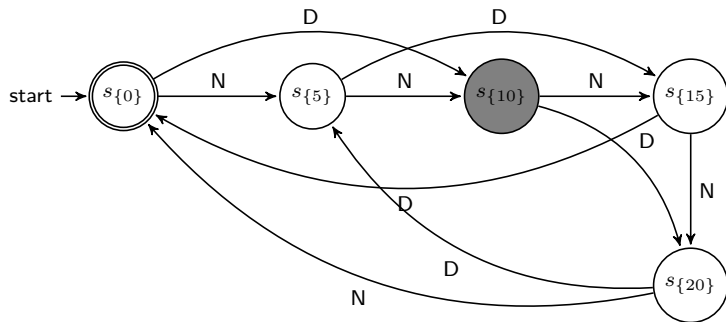
Given this information, $\delta(s_{\{0\}}, \text{D}) = s_{\{10\}}$.

CSE 596 – Automaton Examples

Current state = $s_{\{5\}}$, $w_2 = \text{NDDD}$



Current state = $s_{\{5\}}$, $w_2 = N D D D$



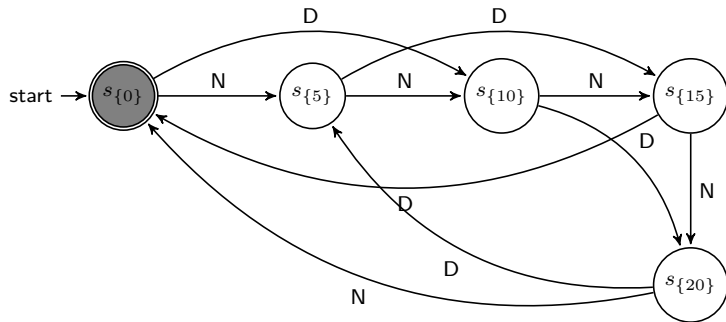
Given this information, there are no more symbols to be read. We are in state $s_{\{10\}} \notin F$, so $s_{\{10\}}$ is not an accepting state. Therefore, the string w_2 is accepted by M .

Since the computation of M on string w_2 did not end in an accepting state, this is not an accepting computation for M , and so $w_2 \notin L(M)$.

Since the computation of M on string w_2 did not end in an accepting state, this is not an accepting computation for M , and so $w_2 \notin L(M)$. Now let's look at the last string $w_3 = \lambda$.

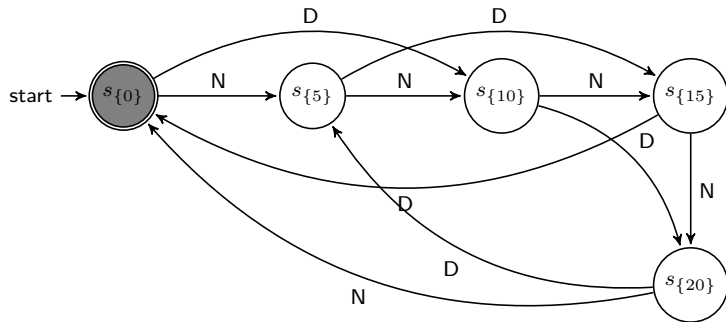
CSE 596 – Automaton Examples

Current state = $s_{\{0\}}$, $w_3 = \lambda$



CSE 596 – Automaton Examples

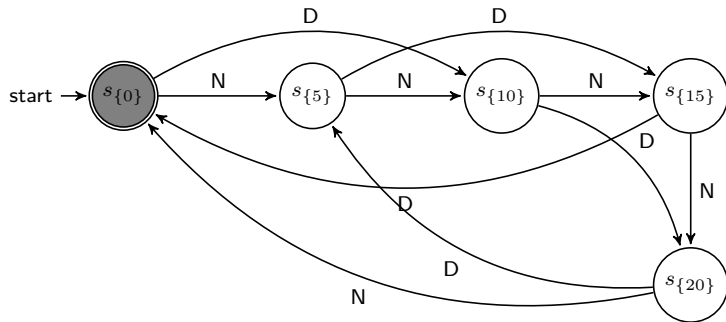
Current state = $s_{\{0\}}$, $w_3 = \lambda$



Given this setup, our machine is starting in the initial state and there are no symbols to be read $w_3 = \lambda$.

CSE 596 – Automaton Examples

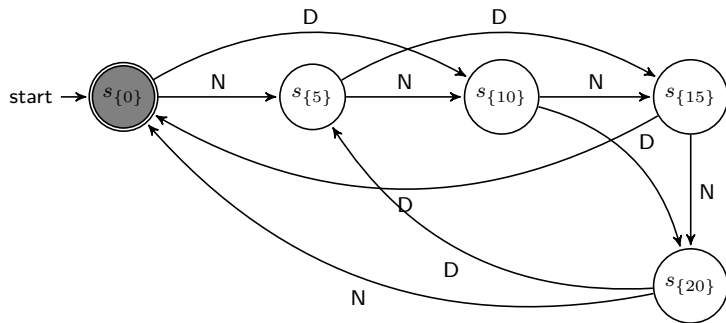
Current state = $s_{\{0\}}$, $w_3 = \lambda$



Given this setup, our machine is starting in the initial state and there are no symbols to be read $w_3 = \lambda$. Therefore, our computation is complete. So in this case, by doing nothing we accept this as a valid string.

CSE 596 – Automaton Examples

Current state = $s_{\{0\}}$, $w_3 = \lambda$



Given this setup, our machine is starting in the initial state and there are no symbols to be read $w_3 = \lambda$. Therefore, our computation is complete.

So in this case, by doing nothing we accept this as a valid string.

The intuition is there as well. If we haven't put any change into the machine, we have received exact change in the sense that we have not left any remainder of money in the change machine.

After looking at the computation of machine M on w_3 , we see that M accepted $w_3 = \lambda$, so $\lambda \in L(M)$.

Bibliography

Problem based on slides from Dr. Ogihara's notes on DFAs
(<http://www.cs.miami.edu/~ogihara/csc527/new01-1.pdf>).
Accessed 09-08-2012.