

Automaty i Języki Formalne

Soma Dutta

Wydział Matematyki i Informatyki, UWM w Olsztynie
soma.dutta@matman.uwm.edu.pl

Wykład - 3: Automaty skończone

Semestr zimowy 2023

Podsumowanie

- ▶ Widzieliśmy kilka przykładów deterministycznego automatu skończonego (DFA) i tego, jak dla danego języka można skonstruować DFA.
- ▶ Przedstawiono definicję niedeterministycznego automatu skończonego i zaobserwowaliśmy, że różnica między DFA i N DFA polega na pojęciu funkcji przejścia.
- ▶ Dowiedzieliśmy się, że często w przypadku niektórych języków łatwiej jest skonstruować N DFA niż DFA. Możemy jednak również przekonwertować N DFA na DFA, a języki generowane przez N DFA są takie same, jak języki generowane przez DFA.

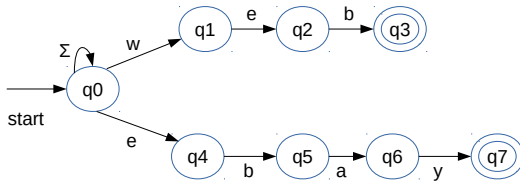
Stosowanie NDFA w praktyce

Search engines Wyszukiwanie słów kluczowych w dokumencie tekstowym

- ▶ Załóżmy, że chcemy wyszukać zestaw słów kluczowych z podanego tekstu. W takich zastosowaniach często przydatne jest zaprojektowanie NDFA, które poprzez wejście w stan końcowy sygnalizuje, gdy odczyta jedno z podanych słów kluczowych. Tekst dokumentu jest podawany, jeden znak na raz, do tego NDFA, który następnie rozpoznaje wystąpienia słów kluczowych w tym tekście.
 - ▶ Pomysł jest taki, że możemy zacząć od stanu początkowego z przejściem do samego siebie dla każdego drukowalnego ASCII postać (ASCII character).
 - ▶ Dla każdego słowa kluczowego $a_1 a_2 \dots a_k$, czyli ciągu symboli, potrzebujemy $k+1$ stanów $q_0, q_1, \dots, q_k$, aby zaczynało się od stanu początkowego i czyta a_1 przesuwa się do q_1 , czyta a_2 przechodzi do q_2 i w ten sposób po przeczytaniu całego słowa dociera do q_k .

NDFA do rozpoznać kluczowe słów web i ebay

- ▶ Przypuszczamy, że chcemy zaprojektować NDFA do rozpoznać wystąpienia słów web i ebay w dokumencie tekstowym.
- ▶ W stanie początkowym wszystkie litery alfabetu, oznaczone Σ , mają pętlę powracającą do tego samego stanu. Stany q_1 do q_3 mają zadanie rozpoznawania 'web', podczas gdy stany q_4 do q_7



rozpoznają 'ebay'.

- ▶ To jest NDFA bo $\delta(q_0, w) = \{q_0, q_1\}$ i $\delta(q_0, e) = \{q_0, q_4\}$ oraz w stanach $q_1, \dots, q_7$ funkcja przejścia nie jest zdefiniowana dla wszystkich symboli wejściowych.
- ▶ Teraz możemy zaimplementować tę technikę wyszukiwania jako program, po prostu konwertując ten NDFA na DFA, co zwykle jest wykonywane w niektórych programach do przetwarzania tekstu.

Automat skończony z pustym przejściem

- ▶ W tym kontekście N DFA może przejść do innego stanu bez konieczności wprowadzania danych wejściowych symbol (tzn. w obecności pustego ciągu). Ta nowa możliwość już tak nie rozszerza klasy języków akceptowalnych przez automaty skończone, ale daje nam dodatkową wygodę programowania.
- ▶ Zwykle taki automat nazywany jest automatem z ϵ -przejściem, które pozwala na przejście do innego stanu po otrzymaniu pustego ciągu znaków (Λ albo możemy używać ϵ) jako dane wejściowe.
- ▶ **Definition:** ϵ -N DFA to pięciokrotka $(Q, \Sigma, \delta, q_0, F)$, w której wszystkie składniki są takie same, jak N DFA, ale funkcja przejścia to $\delta : Q \times \Sigma \cup \{\epsilon\} \mapsto Q$, która stojąc w jednym stanie może przyjąć symbol wejściowy z alfabetu Σ lub pusty ciąg znaków ϵ .

Przykład: ϵ -N DFA, która akceptuje liczby dziesiętne

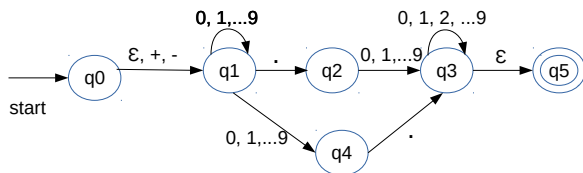
- ▶ Wymagane ciągi powinny składać się z (i) opcjonalnego znaku + lub – (ii) ciągu cyfr (iii) kropki dziesiętnej (iv) innego ciągu cyfr, tak aby przynajmniej ten ciąg lub ciąg w (ii) nie był pusty

Przykład: ϵ -N DFA, która akceptuje liczby dziesiętne

- ▶ Wymagane ciągi powinny składać się z (i) opcjonalnego znaku $+$ lub $-$ (ii) ciągu cyfr (iii) kropki dziesiętnej (iv) innego ciągu cyfr, tak aby przynajmniej ten ciąg lub ciąg w (ii) nie był pusty
- ▶ $\Sigma = \{0, 1, 2, \dots, 9\} \cup \{., +, -\}$ i rozważymy ϵ dla pustych ruchów.

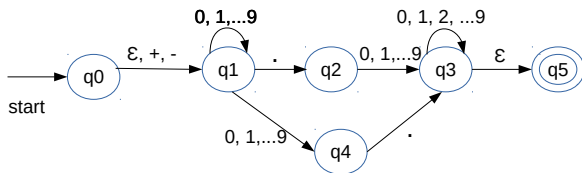
Przykład: ϵ -N DFA, która akceptuje liczby dziesiętne

- ▶ Wymagane ciągi powinny składać się z (i) opcjonalnego znaku $+$ lub $-$ (ii) ciągu cyfr (iii) kropki dziesiętnej (iv) innego ciągu cyfr, tak aby przynajmniej ten ciąg lub ciąg w (ii) nie był pusty
- ▶ $\Sigma = \{0, 1, 2, \dots, 9\} \cup \{., +, -\}$ i rozważymy ϵ dla pustych ruchów.



Przykład: ϵ -NFA, która akceptuje liczby dziesiętne

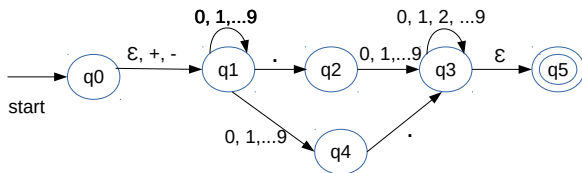
- ▶ Wymagane ciągi powinny składać się z (i) opcjonalnego znaku $+$ lub $-$ (ii) ciągi cyfr (iii) kropki dziesiętnej (iv) innego ciągu cyfr, tak aby przynajmniej ten ciąg lub ciąg w (ii) nie był pusty
- ▶ $\Sigma = \{0, 1, 2, \dots, 9\} \cup \{., +, -\}$ i rozważymy ϵ dla pustych ruchów.



- ▶ Stan q_1 reprezentuje sytuację, w której ciąg wejściowy może być znakiem $(+/-)$ lub bez znaku, z niektórymi cyframi lub nawet bez cyfr. Zwykle część przed przecinkiem. W stanie q_2 możemy już zobaczyć ciągi znaków aż do miejsca dziesiętnego.

Przykład: ϵ -NFA, która akceptuje liczby dziesiętne

- ▶ Wymagane ciągi powinny składać się z (i) opcjonalnego znaku $+$ lub $-$ (ii) ciągi cyfr (iii) kropki dziesiętnej (iv) innego ciągu cyfr, tak aby przynajmniej ten ciąg lub ciąg w (ii) nie był pusty
- ▶ $\Sigma = \{0, 1, 2, \dots, 9\} \cup \{., +, -\}$ i rozważymy ϵ dla pustych ruchów.



- ▶ Stan q_1 reprezentuje sytuację, w której ciąg wejściowy może być znakiem $(+/-)$ lub bez znaku, z niektórymi cyframi lub nawet bez cyfr. Zwykle część przed przecinkiem. W stanie q_2 możemy już zobaczyć ciągi znaków aż do miejsca dziesiętnego.
- ▶ W q_4 na pewno widzieliśmy co najmniej jedną cyfrę, ale nie kropka dziesiętna. W q_3 widzimy liczby z kropką dziesiętną mającą co najmniej jedną cyfrę przed lub po przecinku.

ϵ -domknięcie: rozszerzenie funkcji przejścia na zbiór ciągów

- ▶ Aby zdefiniować funkcję przejścia na zbiorze wszystkich ciągów znaków, musimy najpierw poznać pojęcie ϵ -domknięcia (ϵ -closure).
- ▶ Intuicyjnie ϵ -domknięcie stanu q jest zbiorem wszystkich tych stanów, do których można dotrzeć od stanu q po łuku oznaczonym przez ϵ .
- ▶ **Definicja:** Definiujemy ϵ -domknięcie, oznaczony przez $ECLOSE(q)$, rekurencyjnie w następujący sposób.
 - (i) $q \in ECLOSE(q)$.
 - (ii) Jeśli $p \in ECLOSE(q)$ i $r \in \delta(p, \epsilon)$, wtedy $r \in ECLOSE(q)$.
(Tzn., jeśli stan $p \in ECLOSE(q)$ i następuje przejście ze stanu p do stanu r oznaczony jako ϵ , wówczas $r \in ECLOSE(q)$.)
- ▶ Np. w ostatnim przykładzie, $ECLOSE(q_0) = \{q_0, q_1\}$ i $ECLOSE(q_3) = \{q_3, q_5\}$.
- ▶ **Definicja:** Dla danego $S \subseteq Q$, $ECLOSE(S) = \bigcup_{q \in S} ECLOSE(q)$.

Rozszerzenie funkcji przejścia na zbiór ciągów

- ▶ Załóżmy, że $E = (Q, \Sigma, \delta, q_0, F)$ to ϵ -Ndfa.
- ▶ Aby zdefiniować rozszerzoną funkcję przejścia $\hat{\delta}$ musimy wyjaśnić gdzie można dojść ze stanu q po odczytaniu ciągu ω który może być konkatenacją ciągu symboli zawierającego pusty ciąg znaków ϵ .
- ▶ **Definicja:** $\hat{\delta}$ jest zdefiniowana rekurencyjnie w następujący sposób.
 - (i) Krok podstawowy (Basis case): $\hat{\delta}(q, \epsilon) = ECLOSE(q)$
 - (ii) Krok indukcyjny (Induction step): Przypuszczamy, $\omega = xa$ gdzie $a \in \Sigma$ and $x \in \Sigma^*$. Należy zauważyć, że x może zawierać pusty ciąg ϵ , ale a nie może być ϵ .
Jeśli $\hat{\delta}(q, x) = \{q_1, q_2, \dots, q_k\}$ i $\bigcup_{i=1}^k \delta(q_i, a) = \{r_1, r_2, \dots, r_m\}$, wtedy $\hat{\delta}(q, w) = ECLOSE(\{r_1, r_2, \dots, r_m\})$.
- ▶ **Definicja:** Język ϵ -Ndfa E , oznaczony przez $L(E)$, to zbiór ciągów akceptowane przez E , Tzn., $L(E) = \{\omega \in \Sigma^* : \hat{\delta}(q_0, \omega) \cap F \neq \emptyset\}$.

$\hat{\delta}$ w przykładzie diagramu przejść dla liczb dziesiętnych

- ▶ Np. zgodnie z ostatnim przykładem diagramu przejść dla liczb dziesiętnych, aby obliczyć $\hat{\delta}(q_0, 5.9)$ musimy zrobić następujące kroki.
- ▶ Najpierw obliczymy: $\hat{\delta}(q_0, 5)$ krok po kroku:
 1. $\hat{\delta}(q_0, \epsilon) = \{q_0, q_1\}$
 2. $\delta(q_0, 5) \cup \delta(q_1, 5) = \{q_1, q_4\}$
 3. $\hat{\delta}(q_0, 5) = ECLOSE(\{q_1, q_4\}) = \{q_1, q_4\}$
- ▶ Teraz obliczymy $\hat{\delta}(q_0, 5.)$ krok po kroku.
 1. $\delta(q_1, .) \cup \delta(q_4, .) = \{q_2\} \cup \{q_3\} = \{q_2, q_3\}$.
 2. $\hat{\delta}(q_0, 5.) = ECLOSE(\{q_2, q_3\}) = \{q_2, q_3, q_5\}$
- ▶ W końcu obliczymy $\hat{\delta}(q_0, 5.6)$ krok po kroku:
 1. $\delta(q_2, 6) \cup \delta(q_3, 6) \cup \delta(q_5, 6) = \{q_3\} \cup \{q_3\} \cup \phi = \{q_3\}$.
 2. $\hat{\delta}(q_0, 5.6) = ECLOSE(q_3) = \{q_3, q_5\}$.
- ▶ Oczywiście 5.6 jest akceptowany przez danego ϵ -Ndfa bo $\hat{\delta}(q_0, 5.6) \cap F = \{q_5\}$.

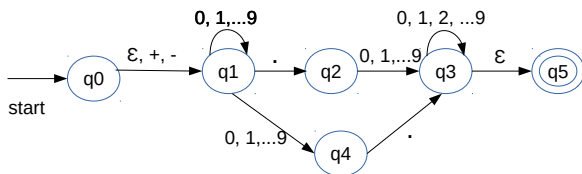
Konwersja ϵ -NFA na odpowiedni DFA

- ▶ **Twierdzenie:** Język L akceptowany przez ϵ -NFA wtedy i tylko wtedy, gdy jest akceptowany przez DFA.
- ▶ Biorąc pod uwagę ϵ -NFA $E = (Q_E, \Sigma, \delta_E, q_0, F)$, możemy skonstruować odpowiedni DFA $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$ taki, że $L(E) = L(D)$.
- ▶ Składniki D są zdefiniowane w następujący sposób.
 - ▶ $Q_D = P(Q_E)$. Chociaż, wszystkie osiągalne stany Q_D są ϵ -domkniętymi podzbiorami Q_E (tzn., rozważmy takie $S \in Q_E$ dla których $S = ECLOSE(S)$).
 - ▶ $q_D = ECLOSE(q_0)$. Warto zauważyć, że w porównaniu do konstrukcji DFA z NFA, tutaj stanem początkowym DFA D powinien być ϵ -domknięty stanu początkowego ϵ -NFA E .
 - ▶ $F_D = \{S \in Q_D : S \cap F_E \neq \emptyset\}$
 - ▶ Dla danego $S \in Q_D$ i $a \in \Sigma$, δ_D jest obliczana w następujący sposób.

Jeśli $S = \{q_1, q_2, \dots, q_k\}$ i $\bigcup_{i=1}^k \delta_E(q_i, a) = \{r_1, r_2, \dots, r_m\}$ wtedy $\delta_D(S, a) = ECLOSE(\{r_1, r_2, \dots, r_m\})$.

Przykład: konwersja ϵ -Ndfa do DFA

- Rozważmy omówiony już przykład ϵ -Ndfa do rozpoznawania liczb dziesiętnych.



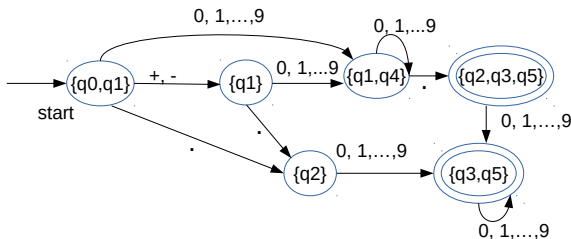
- W odpowiednim DFA $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$, stan początkowy $q_D = ECLOSE(q_0) = \{q_1, q_2\}$.
- Naszym pierwszym zadaniem jest znalezienie przejść z q_0 i q_1 na różnych symbolach Σ . Dla $+$ i $-$ q_1 nie ma żadnego ruchu, a od q_0 odczytania $+$ lub $-$ przesuwają się do q_1 . Zatem, aby obliczyć $\delta_D(\{q_0, q_1\}, +)$ zaczynamy od $\{q_1\}$, a następnie rozważamy $ECLOSE(\{q_1\})$. Ponieważ nie ma ϵ -przejścia z q_1 , mamy $ECLOSE(\{q_1\}) = \{q_1\}$, a $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$. Podobnie, można obliczyć, że $\delta_D(\{q_0, q_1\}, -) = \{q_1\}$.

- ▶ Następnie musimy obliczyć $\delta_D(\{q_0, q_1\}, .)$. Ponieważ od q_0 nie ma ruchu na symbolu '.' i q_1 przesuwa się do q_2 po przeczytaniu symbolu ., musimy wziąć pod uwagę $\{q_2\}$, a tak więc obliczymy $ECLOSE(\{q_2\}) = \{q_2\}$. Stąd, $\delta_D(\{q_0, q_1\}, .) = \{q_2\}$.

- ▶ Następnie musimy obliczyć $\delta_D(\{q_0, q_1\}, .)$. Ponieważ od q_0 nie ma ruchu na symbolu '.' i q_1 przesuwa się do q_2 po przeczytaniu symbolu ., musimy wziąć pod uwagę $\{q_2\}$, a tak więc obliczymy $ECLOSE(\{q_2\}) = \{q_2\}$. Stąd, $\delta_D(\{q_0, q_1\}, .) = \{q_2\}$.
- ▶ Teraz obliczmy $\delta_D(\{q_0, q_1\}, 0)$, ponieważ w przypadku pozostałych cyfr obliczenia będą takie same. Od q_0 nie ma ruchu dla cyfr, ale od q_1 po odczytaniu 0 możliwe są ruchy albo do samego q_1 , albo do q_4 . Ponieważ żaden z tych stanów nie ma ϵ -przejścia, mamy $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}$ i podobnie dla pozostałych cyfr.

- ▶ Następnie musimy obliczyć $\delta_D(\{q_0, q_1\}, .)$. Ponieważ od q_0 nie ma ruchu na symbolu '.', a q_1 przesuwa się do q_2 po przeczytaniu symbolu '.', musimy wziąć pod uwagę $\{q_2\}$, a tak więc obliczymy $ECLOSE(\{q_2\}) = \{q_2\}$. Stąd, $\delta_D(\{q_0, q_1\}, .) = \{q_2\}$.
- ▶ Teraz obliczmy $\delta_D(\{q_0, q_1\}, 0)$, ponieważ w przypadku pozostałych cyfr obliczenia będą takie same. Od q_0 nie ma ruchu dla cyfr, ale od q_1 po odczytaniu 0 możliwe są ruchy albo do samego q_1 , albo do q_4 . Ponieważ żaden z tych stanów nie ma ϵ -przejścia, mamy $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}$ i podobnie dla pozostałych cyfr.
- ▶ Pozostałe przejścia można sprawdzić w podobny sposób i warto to sprawdzić samodzielnie. Oto Schemat przejścia odpowiedniego DFA. Należy uważać, że wszystkie ϵ -przejścia są eliminowane.

- ▶ Następnie musimy obliczyć $\delta_D(\{q_0, q_1\}, .)$. Ponieważ od q_0 nie ma ruchu na symbolu '.' i q_1 przesuwa się do q_2 po przeczytaniu symbolu ., musimy wziąć pod uwagę $\{q_2\}$, a tak więc obliczymy $ECLOSE(\{q_2\}) = \{q_2\}$. Stąd, $\delta_D(\{q_0, q_1\}, .) = \{q_2\}$.
- ▶ Teraz obliczmy $\delta_D(\{q_0, q_1\}, 0)$, ponieważ w przypadku pozostałych cyfr obliczenia będą takie same. Od q_0 nie ma ruchu dla cyfr, ale od q_1 po odczytaniu 0 możliwe są ruchy albo do samego q_1 , albo do q_4 . Ponieważ żaden z tych stanów nie ma ϵ -przejścia, mamy $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}$ i podobnie dla pozostałych cyfr.
- ▶ Pozostałe przejścia można sprawdzić w podobny sposób i warto to sprawdzić samodzielnie. Oto Schemat przejścia odpowiedniego DFA. Należy uważać, że wszystkie ϵ -przejścia są eliminowane.



Automat skończony z osobną funkcją wyjściową

- ▶ Automaty skończone, które rozważaliśmy wcześniej, mają charakter binarny wyjście, tj. albo akceptują ciąg znaków, albo nie akceptują ciągu. Ten Akceptowalność została zdecydowana na podstawie osiągalności stanu końcowego od stanu początkowego. Teraz, usuwamy to ograniczenie i rozważamy sytuacje, w których wyjścia można wybrać z innego alfabetu.
- ▶ Zobaczymy dwa takie modele.
 - ▶ **Mealy machine**: W jednym przypadku wartość funkcji wyjściowej jest funkcją bieżącego stanu i bieżącego symbolu wejściowego. Model ten jest zwykle nazywany maszyną Mealy'ego.
 - ▶ **Moore machine**: W drugim przypadku funkcja wyjściowa zależy tylko od stanu bieżącego i jest niezależny od symbolu wejściowego. Ten ograniczony model jest znany jako maszyna Moore'a.

Maszyna Moore'a i Maszyna Mealy'ego

► **Definicja:** Maszyna Moore'a to sześciokrotka $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ gdzie

- (i) Q jest skończonym zbiorem stanów,
- (ii) Σ jest alfabetem wejściowym,
- (iii) Δ jest alfabetem wyjściowym,
- (iv) $\delta : Q \times \Sigma \mapsto Q$ to funkcja przejścia,
- (v) λ jest funkcją wyjściową odwzorowującą Q na Δ ,
- (vi) $q_0 \in Q$ to stan początkowy.

► **Definicja:** Maszyna Mealy'ego to sześciokrotka $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ gdzie wszystkie składniki z wyjątkiem λ mają takie samo znaczenie jak w maszynie Moore'a. Tutaj funkcja wyjściowa λ odwzorowuje $Q \times \Sigma$ na Δ .

Przykład: Maszyna Moore'a

- $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ gdzie $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \Delta = \{0, 1\}$, a $\delta : Q \times \Sigma \mapsto Q$ i $\lambda : Q \mapsto \Delta$ są zdefiniowane w następujący sposób.

Stan bieżący	Stan następny		Wyjścia
	$a = 0$	$a = 1$	
$\rightarrow q_0$	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

Przykład: Maszyna Moore'a

- $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ gdzie $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \Delta = \{0, 1\}$, a $\delta : Q \times \Sigma \mapsto Q$ i $\lambda : Q \mapsto \Delta$ są zdefiniowane w następujący sposób.

Stan bieżący	Stan następny		Wyjścia
	$a = 0$	$a = 1$	
$\rightarrow q_0$	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

- Rozważmy ciąg wejściowy 0111. Przejście pomiędzy stanami jest następujące. $q_0 \xrightarrow{0} q_3 \xrightarrow{1} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_2$

Przykład: Maszyna Moore'a

- $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ gdzie $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \Delta = \{0, 1\}$, a $\delta : Q \times \Sigma \mapsto Q$ i $\lambda : Q \mapsto \Delta$ są zdefiniowane w następujący sposób.

Stan bieżący	Stan następny		Wyjścia
	$a = 0$	$a = 1$	
$\rightarrow q_0$	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

- Rozważmy ciąg wejściowy 0111. Przejście pomiędzy stanami jest następujące. $q_0 \xrightarrow{0} q_3 \xrightarrow{1} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_2$
- Natomiast przy danym ciągu wejściowym ciąg wyjściowy jest przetwarzany w następujący sposób.

$\lambda(q_0) \xrightarrow{0} \lambda(q_3) \xrightarrow{1} \lambda(q_0) \xrightarrow{1} \lambda(q_1) \xrightarrow{1} \lambda(q_2)$

↓ ↓ ↓ ↓ ↓

0 0 0 1 0

Przykład: Maszyna Mealy'ego

- $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ gdzie $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \Delta = \{0, 1\}$, a $\delta : Q \times \Sigma \mapsto Q$ i $\lambda : Q \times \Sigma \mapsto \Delta$ są zdefiniowane w następujący sposób.

Stan bieżący	δ : Stan następujący			
	$a = 0$		$a = 1$	
	Stan	Wyjścia	Stan	Wyjścia
$\rightarrow q_0$	q_2	0	q_1	0
q_1	q_0	1	q_3	0
q_2	q_1	1	q_0	1
q_3	q_3	1	q_2	0

Przykład: Maszyna Mealy'ego

- $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ gdzie $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \Delta = \{0, 1\}$, a $\delta : Q \times \Sigma \mapsto Q$ i $\lambda : Q \times \Sigma \mapsto \Delta$ są zdefiniowane w następujący sposób.

Stan bieżący	δ : Stan następujący			
	$a = 0$		$a = 1$	
	Stan	Wyjścia	Stan	Wyjścia
$\rightarrow q_0$	q_2	0	q_1	0
q_1	q_0	1	q_3	0
q_2	q_1	1	q_0	1
q_3	q_3	1	q_2	0

- Rozważmy ciąg wejściowy 0111. Przejście pomiędzy stanami jest następujące. $q_0 \xrightarrow{0} q_2 \xrightarrow{1} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_3$

Przykład: Maszyna Mealy'ego

- ▶ $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ gdzie $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \Delta = \{0, 1\}$, a $\delta : Q \times \Sigma \mapsto Q$ i $\lambda : Q \times \Sigma \mapsto \Delta$ są zdefiniowane w następujący sposób.

Stan bieżący	δ : Stan następujący			
	$a = 0$		$a = 1$	
	Stan	Wyjścia	Stan	Wyjścia
$\rightarrow q_0$	q_2	0	q_1	0
q_1	q_0	1	q_3	0
q_2	q_1	1	q_0	1
q_3	q_3	1	q_2	0

- ▶ Rozważmy ciąg wejściowy 0111. Przejście pomiędzy stanami jest następujące. $q_0 \xrightarrow{0} q_2 \xrightarrow{1} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_3$
- ▶ Natomiast przy danym ciągu wejściowym ciąg wyjściowy jest przetwarzany w następujący sposób.

$$\lambda(q_0, 0) \xrightarrow{\delta} \lambda(q_2, 1) \xrightarrow{\delta} \lambda(q_0, 1) \xrightarrow{\delta} \lambda(q_1, 1) \xrightarrow{\delta} q_3$$

↓
0

↓
1

↓
0

↓
0

Obserwacje

- ▶ W przypadku maszyny Moore'a, jeśli ciąg wejściowy ma długość n , ciąg wyjściowy ma długość $n + 1$. Dla wszystkich ciągów wyjściowych pierwszym symbolem jest zawsze $\lambda(q_0)$.
- ▶ W drugiej stronie, dla maszyny Mealy'ego, jeśli ciąg wejściowy ma długość n wtedy ciąg wyjściowy jest również w tej samej długości.
- ▶ Istnieje możliwość przekształcenia maszyny Mealey w maszynę Moore i maszyny Moore w maszynę Mealy.