

Projekt pn. „Wzmocnienie potencjału dydaktycznego UMK w Toruniu w dziedzinach matematyczno-przyrodniczych”
realizowany w ramach Poddziałania 4.1.1 Programu Operacyjnego Kapitał Ludzki

TEORIA OBLICZALNOŚCI – ĆWICZENIA 6

Ćwiczenia 6

Zadanie wstępne:

Zapoznaj się z symulatorem maszyny Turinga dostępnym na stronie: <http://morphett.info/turing/turing.html>. Wykorzystaj go do sprawdzania poprawności swoich rozwiązań.

Zadanie 1

Napisz program, który zastępuje w dowolnym tekście zbudowanym z liter A, B, C i D wszystkie literki A literką C . Na przykład, z tekstu $BACA$ ma powstać $BCCC$.

Rozwiązanie:

$$\begin{aligned} q_0 A &\rightarrow q_0 C R \\ q_0 B &\rightarrow q_0 B R \\ q_0 C &\rightarrow q_0 C R \\ q_0 D &\rightarrow q_0 D R \\ q_0 \square &\rightarrow q_0 \square L \\ q_1 B &\rightarrow q_1 B L \\ q_1 C &\rightarrow q_1 C L \\ q_1 D &\rightarrow q_1 D L \\ q_1 \square &\rightarrow q_{ACC} \square R \end{aligned}$$

Zadanie 2

Napisz program, który przenosi pierwszą literkę wyrazu zbudowanego z liter A, B, C i D z początku na koniec. Na przykład, z tekstu DAB ma powstać ABD .

Rozwiązanie:

$q_0 A \rightarrow q_A \square R$	$q_1 C \rightarrow q_1 C L$	$q_B D \rightarrow q_B D R$
$q_0 B \rightarrow q_B \square R$	$q_1 D \rightarrow q_1 D L$	$q_C A \rightarrow q_C A R$
$q_0 C \rightarrow q_C \square R$	$q_1 \square \rightarrow q_{ACC} \square R$	$q_C B \rightarrow q_C B R$
$q_0 D \rightarrow q_D \square R$	$q_A A \rightarrow q_A A R$	$q_C C \rightarrow q_C C R$
$q_A \square \rightarrow q_1 A L$	$q_A B \rightarrow q_A B R$	$q_C D \rightarrow q_C D R$
$q_B \square \rightarrow q_1 B L$	$q_A C \rightarrow q_A C R$	$q_D A \rightarrow q_D A R$
$q_C \square \rightarrow q_1 C L$	$q_A D \rightarrow q_A D R$	$q_D B \rightarrow q_D B R$
$q_D \square \rightarrow q_1 D L$	$q_B A \rightarrow q_B A R$	$q_D C \rightarrow q_D C R$
$q_1 A \rightarrow q_1 A L$	$q_B B \rightarrow q_B B R$	$q_D D \rightarrow q_D D R$
$q_1 B \rightarrow q_1 B L$	$q_B C \rightarrow q_B C R$	

Projekt pn. „Wzmocnienie potencjału dydaktycznego UMK w Toruniu w dziedzinach matematyczno-przyrodniczych”
 realizowany w ramach Poddziałania 4.1.1 Programu Operacyjnego Kapitał Ludzki

TEORIA OBLICZALNOŚCI – ĆWICZENIA 6

Zadanie 3

Napisz program, który łączy dwie liczby binarne rozdzielone znakiem pustym. Liczba pierwsza ma zostać dołączona na koniec drugiej. Na przykład, z danych wejściowych $\square 110011 \square 101 \square$ ma powstać $\square 101110011 \square$.

Rozwiązanie:

q_s	1	\rightarrow	q_1	\square	R	zapamiętaj pierwszą literę (q_s – stan początkowy)
q_s	0	\rightarrow	q_0	\square	R	
q_s	\square	\rightarrow	q_{ACC}	\square	R	
q_1	1	\rightarrow	q_1	1	R	kończymy niesiemy „1”
q_1	0	\rightarrow	q_1	0	R	
q_1	\square	\rightarrow	q_{11}	\square	R	
q_{11}	1	\rightarrow	q_{11}	1	R	
q_{11}	0	\rightarrow	q_{11}	0	R	
q_{11}	\square	\rightarrow	q_w	1	L	
q_0	1	\rightarrow	q_0	1	R	niesiemy „0”
q_0	0	\rightarrow	q_0	0	R	
q_0	\square	\rightarrow	q_{00}	\square	R	
q_{00}	1	\rightarrow	q_{00}	1	R	
q_{00}	0	\rightarrow	q_{00}	0	R	
q_{00}	\square	\rightarrow	q_w	0	L	
q_w	1	\rightarrow	q_w	1	L	wracamy
q_w	0	\rightarrow	q_w	0	L	
q_w	\square	\rightarrow	q_{ww}	\square	L	
q_{ww}	1	\rightarrow	q_{ww}	1	L	
q_{ww}	0	\rightarrow	q_{ww}	0	L	
q_{ww}	\square	\rightarrow	q_s	\square	R	

Zadanie 4

Napisz program, który podwaja każdy bit wejściowej liczby binarnej. Na przykład, z liczby 101 powinniśmy otrzymać 110011.

Szkic rozwiązania:

Projekt pn. „Wzmocnienie potencjału dydaktycznego UMK w Toruniu w dziedzinach matematyczno-przyrodniczych”
realizowany w ramach Poddziałania 4.1.1 Programu Operacyjnego Kapitał Ludzki

TEORIA OBLICZALNOŚCI – ĆWICZENIA 6

Za słowem wejściowym wstawiamy specjalny znak #. Zapamiętujemy pierwszą literę słowa i kasujemy ją. Przechodzimy na koniec słowa i za znakiem # wpisujemy zapamiętaną literę dwukrotnie. Wracamy na początek słowa wejściowego. Zapamiętujemy drugą literę słowa i kasujemy ją. Przechodzimy za znak # i dopisujemy ją na dwukrotnie na końcu nowopowstającego słowa. Wracamy na początek. Postępujemy w ten sposób tak długo, aż słowo wejściowe będzie puste. Wtedy kasujemy znak # i ustawiamy się na początku słowa powstałego za nim.

Zadanie 5

Napisz program, który odwróci kolejność bitów dowolnej liczby binarnej, tzn. z liczby 101100111 ma powstać 111001101.

Szkic rozwiązania:

Przed słowem wejściowym wstawiamy specjalny znak @. Za słowem wejściowym wstawiamy specjalny znak #. Zapamiętujemy i kasujemy ostatnią literę słowa wejściowego i zapisujemy ją za znakiem #. Patrzymy i zapamiętujemy, jaka litera teraz jest przed znakiem # (jest to przedostatnia litera słowa wejściowego). Dopisujemy ją na początku nowopowstałego słowa (za znakiem #). Postępujemy w ten sposób tak długo, aż wyczyścimy słowo wejściowe. Czyścimy znak @ oraz #. Wskazujemy na pierwszą literę nowego słowa.

Zadanie 6

Napisz program, który zbada czy wczytane słowo jest palindromem.

Szkic rozwiązania:

Zapamiętujemy pierwszą literę słowa. Kasujemy tę literę. Idziemy na koniec, sprawdzamy czy ostatnia jest ta sama litera. Jeśli nie, odrzucamy. Jeśli tak, kasujemy ją i wracamy na początek słowa. Kontynuujemy, aż uzyskamy słowo puste. Wtedy akceptujemy.

Zadanie 7

Skonstruuj maszynę Turinga rozpoznającą język zadany gramatyką: $S \rightarrow aTb|b$, $T \rightarrow Ta|\epsilon$.

Rozwiązanie:

Zadana gramatyka generuje następujący język: $L = \{a^n b : n \geq 0\}$.

Zauważmy, że jest to język regularny. Istnieje zatem deterministyczny automat skończony akceptujący język L . Do akceptacji tego języka wystarczy, aby maszyna przeszła taśmę z lewej strony do prawej działając następująco:

- 1) napotykając a , zostaw a idź w prawo.
- 2) napotykając pierwsze b idź w prawo i
 - a. jeśli to ostatnia litera – akceptuj

Projekt pn. „Wzmocnienie potencjału dydaktycznego UMK w Toruniu w dziedzinach matematyczno-przyrodniczych”
realizowany w ramach Poddziałania 4.1.1 Programu Operacyjnego Kapitał Ludzki

TEORIA OBLICZALNOŚCI – ĆWICZENIA 6

b. jeśli są dalej litery, odrzuć.

Zadanie domowe

1. W wejściowym ciągu binarnym zbudowanym maksymalnie z 8 bitów oblicz długość najdłuższego podciągu zbudowanego z samych jedynek. Wynik podaj w postaci cyfry od 0 do 8. Na przykład, dane wejściowe 01011101 powinny zostać przetworzone do postaci 3□01011101.
2. Napisz program, który posortuje bity dowolnej liczby binarnej ustawiając wszystkie 0 po prawej stronie, a 1 po lewej. Na przykład liczba 10111001 musi zostać przekształcona do postaci 11111000.