JAO - Języki, Automaty i Obliczenia - Wykład 2

- **[lemat o pompowaniu]** Jeśli L regularny to istnieje stała c spełniająca : jeżeli $z \in L$, $|z| \geq c$ to istnieje dekompozycja $w = u \cdot v \cdot x$ tak, że $uv^i x \in L$ dla każdego $i \geq 0$

- **[lemat o skończonej liczbie ilorazów]** Jeśli L regularny to liczba ilorazów L jest skończona, liczba stanów automatu jest nie mniejsza niż liczba ilorazów, gdzie ilorazy zdefiniowane są następująco:

$$L/u = \{ w \ : \ uw \in L \}$$

- **[ogłupianie automatu]** Po wczytaniu danego słowa automat nie może rozróżnić więcej następnych słów niż liczba stanów .

Lemat o pompowaniu stosuje sie zazwyczaj w stowarzyszeniu z następującymi faktami (dowodzonymi potem):

Lemat

Język jest akceptowany przez automat skończony wtw gdy jest regularny. Przecięcie i dopełnienie teoriomongościowe języków regularnych jest językiem regularnym.

Przykłady. Następujące języki nie są regularne:

$$\{a^n b^n : \ n \geq 1\}, \ \{a^n : \ n \ \text{ jest liczbą pierwszą}\}$$

$$L \ = \ \text{zbiór binarnych zapisów liczb pierwszych}$$

$$L \ = \ \text{zbiór binarnych zapisów kwadratów liczb naturalnych}$$

Lemat

Niech $0 < r < 1$. $L = \{x \in (0 \cup 1)^+ \ : \ [0.x]_2 > r \ \}$ *jest regularny wtw gdy* r *jest wymierne.*

Lemat

Niech L będzie językiem akceptowanym przez det. automat skończony. Minimalna liczba stanów det. automatu skończnego akceptującego L jest równa liczbie różnych ilorazów L

Uzasadnienie. Wiemy, że L jest akceptowany przez aut. skończony, zatem liczba jego ilorazów jest skończona. Konstruujemy tzw. automat ilorazowy:

$$A = (\Sigma, Q, \delta, q_0, F), \text{ gdzie :}$$

- Q - zbiór ilorazów L;
- $q_0 = L/\varepsilon = L$;
- $F = \{X : \varepsilon \in X\}$;
- $\delta(X, a) = X/a$ dla $a \in \Sigma$.

Niech $\Sigma = \{a, b\}$ oraz

$$L = \Sigma^* \cdot a \cdot \Sigma^n.$$

Wtedy wszystkie ilorazy

$$\{ L/w : w \in \Sigma^{n+1} \}$$

są parami różne. Rozmiar zbioru Σ^{n+1} wynosi 2^{n+1}.
Zatem deterministyczny automat akceptujący ten język musi mieć co najmniej 2^{n+1} różnych stanów. Taki automat istnieje.

Mając wyrażenie regularne możemy je dzielić przez kolejne litery słowa, sprowadzając dzielenie do coraz prostszych wyrażen.

- $(X \cup Y)/a = X/a \cup Y/a$
- $(X \cdot Y)/a = X \cdot (Y/a) \cup X/a \cdot \varepsilon(Y)$,

  gdzie $\varepsilon(Y) = \varepsilon$ if $\varepsilon \in Y$, wpp. $\varepsilon(Y) = \emptyset$
- $X^*/a = X^* \cdot (X/a)$

Dla małych wyrażeń możemy policzyć *ręcznie* wszystkie ilorazy (jako wyrażenia regularne), *wąskim gardłem* jest sprawdzanie, które z tych wyrażeń są parami równe.
W ten sposób mamy pewną bardziej konstruktywną metodę liczenia automatu ilorazowego. Potem podamy algorytm efektywniejszy, działający w czasie wielomianowym ze względu na łączny rozmiar wejścia i wyjścia.

Dla języka $L = L(A)$ rozważmy jezyki:

1. $\sqrt{L} = \{w \,:\, ww \in L\}$
2. $\sqrt[*]{L} = \{w \,:\, \exists\, k \geq 1\ w^k \in L\}$
3. $L' = \{w \,:\, w^* \subseteq L\}$

Dowodzimy regularności tych języków konstruując automat *transformacyjny* $T(A)$.
Zbiorem stanów $T(A)$ sa wszystkie funkcje (transformacje)

$$f \,:\, Q \to Q$$

Dla $a \in \Sigma$ oznaczmy $\delta_a = \delta(*, a)$.
W automacie funkcyjnym stan początkowy to identyczność.

Funkcja przejść δ' to:

$$\delta'(f, a) = f \cdot \delta_a.$$

Niech F będzie zbiorem stanów akceptujących automatu A, a q_0 jego stanem początkowym.
Biorąc odpowiedni zbiór F' stanów akceptujących pokazać można, że każdy z języków (1-3) jest regularny. Zbiorem stanów akceptujących F' w $T(A)$ jest, zależnie od przypadku:

- $F' = \{ f \ : \ f^2(q_0) \in F \}$
- $F' = \{ f \ : \ f^k(q_0) \in F \text{ dla pewnego } \ k \geq 1 \}$
- $F' = \{ f \ : \ f^2(q_0) \in F \text{ dla każdego } k \geq 1 \}$

Niedeterministyczny automat skończony jest formalnie również opisany piątką obiektów:

$$A \;=\; (\Sigma, Q, \delta, Q_0, F), \text{ gdzie}$$

δ jest funkcją niedeterministyczną (relacją):

$$\delta \; Q \times \Sigma \;\rightarrow\; \text{podzbiory } Q$$

a zamiast jednego stanu początkowego mamy zbiór Q_0 stanów początkowych. Automat może niedeterministycznie rozpocząć działanie w jednym z nich.
Piszemy $q \xrightarrow{a} q'$ gdy $q' \in \delta(q, a)$. Niech $\rightarrow^*$ będzie domknięciem tej relacji ze względu na wszystkie słowa. Podobnie jak poprzednio:

$$L(A) \;=\; \{\, w \in \Sigma^* \;:\; (\,\exists\, q \in Q_0,\; q' \in F\,)\; q \xrightarrow{w} q' \,\}$$

Twierdzenie

Dla każdego automatu niedeterministycznego $A = (\Sigma, Q, Q_0, \delta, F)$ *istnieje równoważny automat deterministyczny (taki, że* $L(A) = L(P(A))$*).*

Konstrukcja automatu potęgowego. Skonstruujemy automat $P(A)$, równoważny A, nazywany automatem potęgowym dla A. Automat $P(A)$ jest reprezentowany przez piątkę

$$P(A) = (\Sigma, Q', q_0', \delta', F'), \text{ gdzie,}$$

- $q_0' = Q_0$,
- $F' = \{X : X \cap F \neq \emptyset\}$,
- $\delta'(X, a) = \{q' \in Q : \exists\, q \in X \; q \xrightarrow{a} q'\}$
- Q' jest rodziną zbiorów stanów osiągalnych z Q_0 za pomocą operacji δ'. Elementy Q' nazywamy warstwami.

Jeśli $A, = (\Sigma, Q, Q_0, \delta, F)$ jest automatem (deterministycznym lub nie) to przez A^R oznaczmy automat który ma taki sam zbiór stanów i alfabet wejściwy Σ i spełnia

$$(L(A)^R = L(A^R).$$

A^R różni sie od A następująco:

- Zbiór stanów początkowych A^R jest równy F;
- Zbiór stanów akceptujących A^R jest równy Q_0;

$$(\forall\, a \in \Sigma)\ q \xrightarrow{a}_{A^R} q' \Leftrightarrow q' \xrightarrow{w}_{A} q$$

Dla dowolnego słowa $w \in \Sigma^*$ zachodzi:

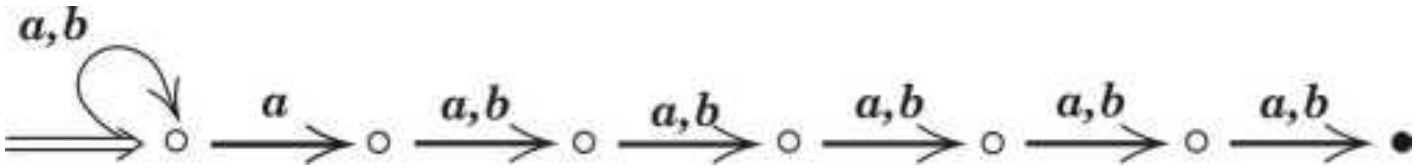
$$q \xrightarrow{w}_{A^R} q' \Leftrightarrow q' \xrightarrow{w}_{A} q.$$

Niech $\Sigma = \{a, b\}$. Poniższy automat A, dla $k = 6$, akceptuje język

$$L = \Sigma^* \cdot a \cdot \Sigma^{k-1}.$$

Automat sprawdza czy k-ta pozycja od końca jest jedynką.

Po odwróceniu strzałek i zamiany "stan akceptujący początkowy" otrzymujemy automat deterministyczny A' akceptujący $L^R = \Sigma^{k-1} \cdot a \cdot \Sigma^*$ mający tylko $k + 1$ stanów.



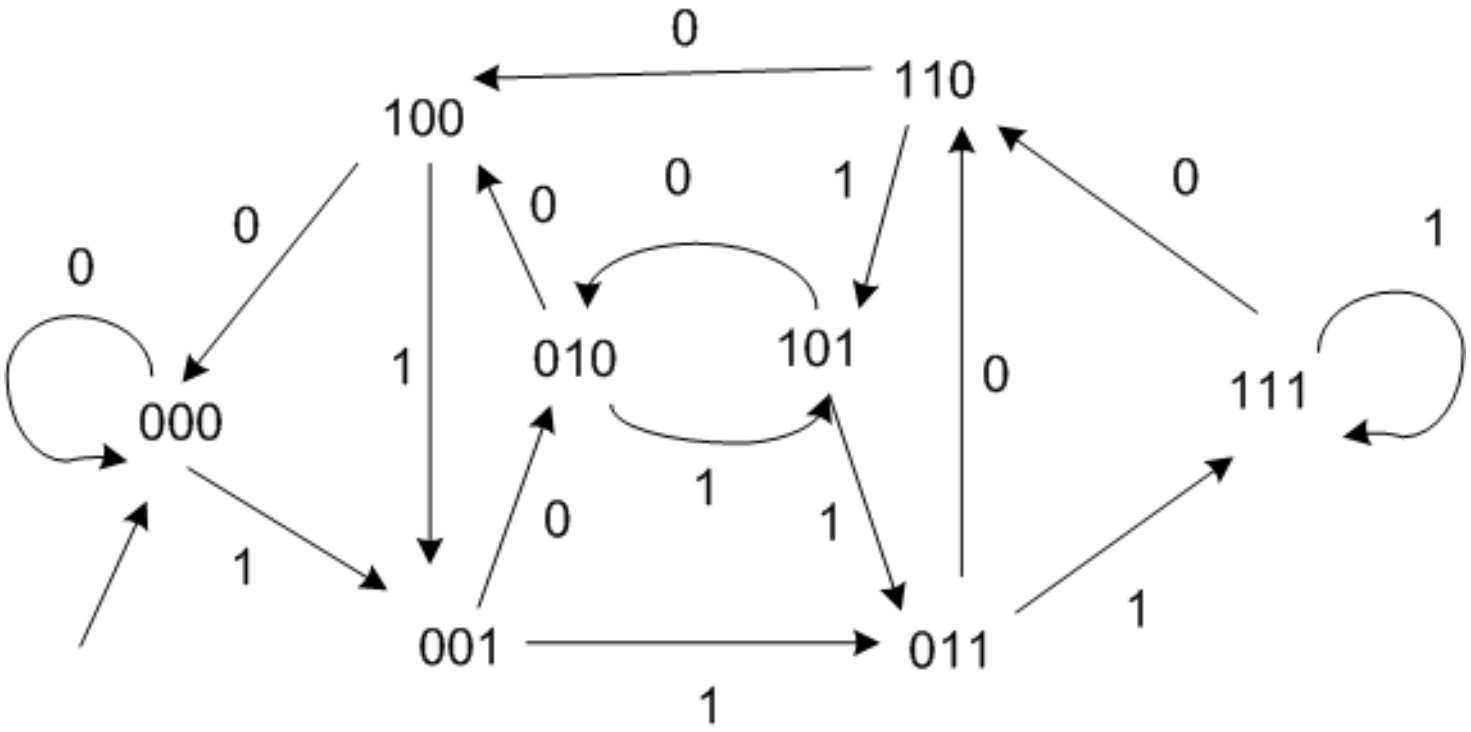
$k = 6$: automat niedeterministyczny mający $n = k + 1$ stanów dla którego minimalny automat det. ma $2^{n-1} = 64$ stany.

Najmniejszy automat deterministyczny akceptujący $L(A)$ ma 2^k stanów, co wynika z konstrukcji potęgowej (podzbiorowej), gdyż mamy 2^k warstw osiągalnych z $\{q_0\}$. Każda warstawa jest podzbiorem $[0..k]$ zawierającym 0. Zamiast konstrukcji potęgowej możemy skonstruować automat deterministyczny pamiętający ostatnie k wczytanych symboli:

- $Q = \Sigma^k$;
- $q_0 = 0^k$;
- $\delta(a_1 a_2 \ldots a_k, s) = (a_2 a_3 \ldots a_k s)$;
- $F = \{ a_1 a_2 \ldots a_k \ : \ a_1 = 1 \}$.

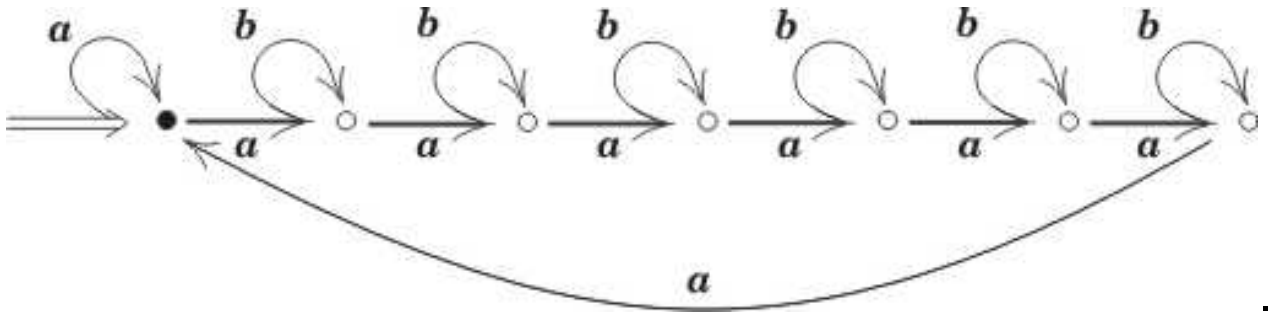
Jeśli alfabet jest binarny to graf automatu jest grafem DeBruijna.

Graf de Bruijna jako automat deterministyczny akceptujący $\{0,1\}^* 1 \{0,1\}^2$.



Stany akceptujące zaczynaja się jedynką.

Niech $\Sigma = \{a, b\}$. Poniższy niedeterministyczny automat A akceptuje język trudny dla automatu deterministycznego. Stan akceptujący jest jednocześnie jedynym stanem początkowym.



Jeśli A ma n stanów to najmniejszy automat deterministyczny akceptujący $L(A)$ ma dokładnie 2^n stanów (ekstremalny przypadek).

Zbiór stanów A to $Q = \{1, 2, \ldots n\}$. Pokażemy, że ze zbioru $\{1\}$ można osiągnąć w $P(A)$ każdy podzbiór Q.

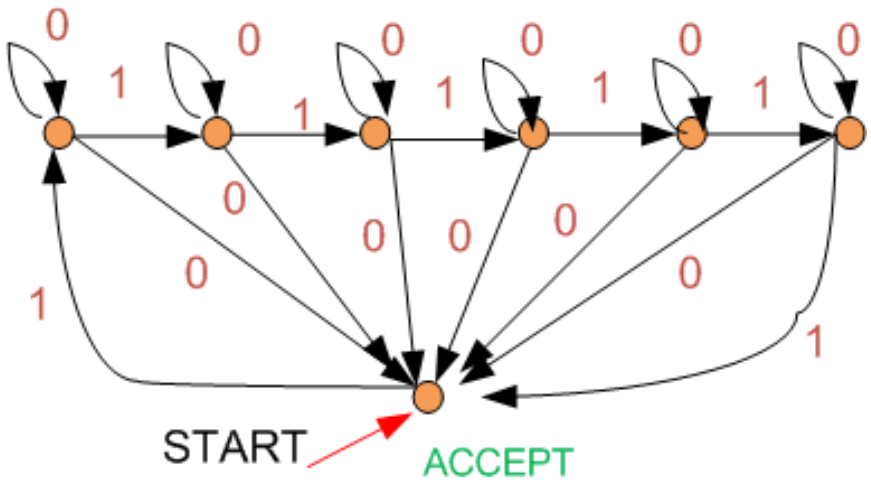
Przykład. $n = 6$, generacja $\{1, 3, 5\}$:

$$\{1\} \xrightarrow{a^4} \{1,2,3,4,5\} \xrightarrow{ab} \{2,3,4,5,6\} \xrightarrow{aa} \{1,2,4,5,6\}$$

$$\xrightarrow{b} \{2,4,5,6\} \xrightarrow{aa} \{1,2,4,6\} \xrightarrow{b} \{2,4,6\} \xrightarrow{a} \{1,3,5\}.$$

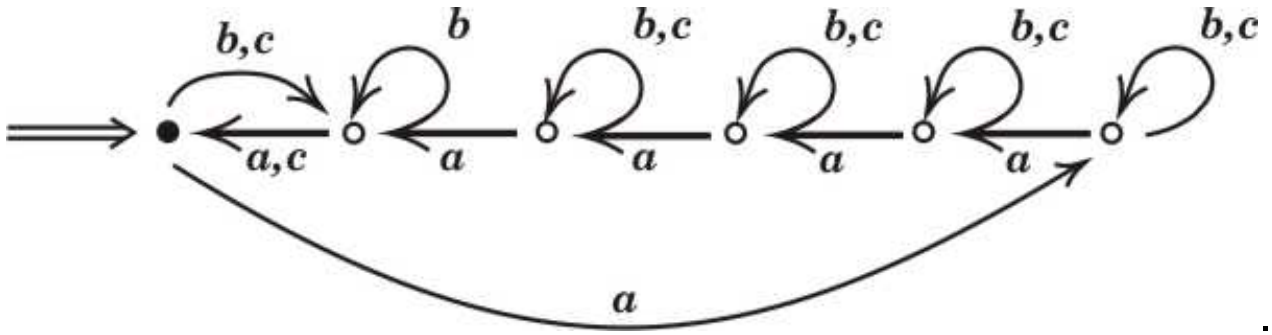
Podzbiory Q utożsamiamy z ciągami binarnymi je reprezentującymi. Mamy operację cyklicznego przestawiania ciągu (ostatni element na początek) za pomoca przejścia a lub ab. Poza tym możemy zamieniać pierwszy bit na zero poprzez przejście b. Teraz wystarczy jedynie początkowo wygenerować zbiór Q za pomocą a^n, otrzymując 1^n. Potem przesuwając cyklicznie i zerując pierwszy bit otrzymamy dowolną konfigurację.

Niech $\Sigma = \{a, b\}$. Inny przykład niedeterministycznego automatu akceptującego język trudny dla automatu deterministycznego. Stan akceptujący jest jednocześnie jedynym stanem początkowym.



Jeśli A ma n stanów to najmniejszy automat deterministyczny akceptujący $L(A)$ ma dokładnie 2^n stanów

Poniższy automat deterministyczny A ma n stanów. Stan akceptujący jest jednocześnie jedynym stanem początkowym.



Dla tego automatu niedeterministyczny automat odwrócony A^R ma też n stanów, ale deterministyczny automat równoważny A^R musi mieć co najmniej 2^n stanów. Jest to przypadek ekstremalny dla operacji odwracania automatu deterministycznego.

Zamek w sejfie jest okręgiem na którym są cyklicznie ułożone przyciski binarne: 1 (otwarty przycisk), 0 (zamknięty). Przed wykonaniem każdej operacji zamek losowo rotuje (przesuwa się cyklicznie). Przyciski są nieodróżnialne a ich stan nie jest znany (poza momentem otwarcia zamka).
Pojedyncza operacja to ciąg n-bitowy (i-ty bit jest równy jeden, $\bar{A}_i(n) = 1$, gdy wciskamy przycisk i). Wciśnięcie przycisku zmienia jego stan zerojednkowe na przeciwny. Stan zamka to klasa równoważności cyklicznej słów binarnych długości n. Zamek się otwiera automatycznie, gdy wszystkie przyciski mają tę samą wartość 1 (są otwarte) lub 0 (są zamknięte).

Problem: podać sztywny ciąg operacji $\bar{A}(n)$ który zawsze otwiera zamek. Niezależnie od losowych rotacji i losowej konfiguracji początkowej w pewnym momencie wszystkie przyciski są w tym samym stanie.

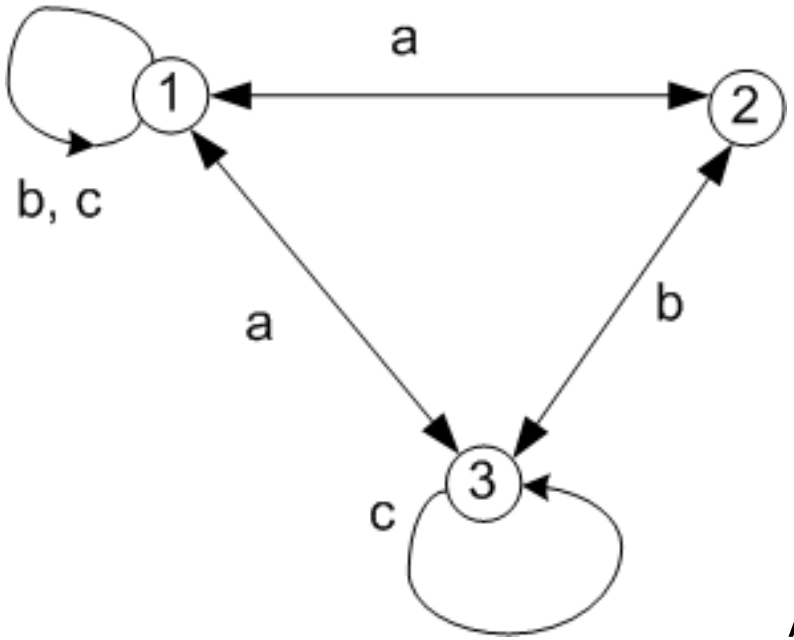
Rozważymy na razie tylko przypadek $n = 4$, konfiguracje, w których nie wszystkie przyciski są w tym samym stanie Możliwe konfiguracje (jako reprezentanci klas cyklicznej równoważości to
1 : ○ ○ ○ ●,
2 : ○ ● ○ ●,
3 : ○ ○ ● ● .

Na przykład konfiguracja 1 oznacza sytuację gdy jeden przycisk jest inny od wszystkich pozostałych, chociaż nie wiemy jaki ma on status.

Możemy wykonać 3 sensowne ruchy:

a: zmienić stan jednego przycisku (równoważne zmianie trzech)
b: zmienić stan 2 sąsiednich przycisków
c: zmienić stan 2 niesąsiednich przycisków

Po zdeterminizowaniu poniższego automatu znajdujemy, że najkrótszy ciąg otwierający zawsze sejf dla $n = 4$ to 7-elementowy ciąg *cbcacbc*, prowadzi on do stanu $\emptyset$ w zdeterminizowanym automacie.



Automat niedeterministyczny dla $n = 4$, wszystkie stany są akceptujące i początkowe. Ciąg otwierający sejf to słowo nieakceptowane przez ten automat.

Rozważmy problem dla dowolnego n, tym razem żądamy by wszystkie przyciski znalazły sie jednocześnie w stanie 1 (otwarte).

Problem: podać sztywny ciąg operacji $\bar{A}(n)$ który zawsze otwiera zamek. Niezależnie od losowych rotacji i losowej konfiguracji początkowej w pewnym momencie wszystkie przyciski są w stanie 1.

Rozważamy ciągi typu $\bar{X}(n)$, które mają tutaj długość $2^n - 1$. $\bar{X}_{last}(n)$ oznacza ostatni element ciągu.

Zdefiniujmy operację

$$\bar{X}(n) \otimes \bar{X}'(n) \;=\; (\; \bar{X}_1(n)\bar{X}'_1(n),\; \bar{X}_2(n)\bar{X}'_2(n),\; \ldots \bar{X}_{last}(n)\bar{X}'_{last}(n)\;).$$

Przykład:

$$(01,\; 11,\; 10) \otimes (10,\; 11,\; 00) \;=\; (0110,\; 1111,\; 1000).$$

Wzory rekurencyjne. Niech $\bar{Z}(n)$ będzie ciągiem, którego wszystkie elementy są równe 0^n. $\bar{A}(n)$ będzie rozwiązaniem (ciągiem otwierającym) dla n będących potęgami dwójki, elementy $\bar{A}(n)$ są słowami binarnymi n-bitowymi. Długość ciągu to $2^n - 1$.

Niech $\bar{A}(2) = (11, 01, 11)$ oraz niech dla $n \geq 2$:

$$\bar{B}(n) = \bar{A}(n) \otimes \bar{A}(n)$$

$$\bar{C}(n) = \bar{Z}(n) \otimes \bar{A}(n)$$

$$\bar{A}(2n) = (\ \bar{B}(n),\ \bar{C}_1(n),\ \bar{B}(n),\ \bar{C}_2(n),\ \ldots \bar{B}(n);\ \bar{C}_{last}(n),\ \bar{B}(n)\ ).$$

Fakt. Problem cyklicznego zamka ma rozwiązanie wtedy i tylko wtedy gdy n jest potęgą dwójki.

Uzasadnienie.

(n jest potęgą dwójki) Sekwencja $\bar{B}(n)$ "utożsamia" przyciski znajdujące się naprzeciwko siebie(względem środka cyklicznego zamka). Inaczej mówiąc przyciski $i, i + n/2$ są jednocześnie wciskane albo jednocześnie nie wciskane. Natomiast sekwencja $\bar{C}(n)$ ustawia przciski i, dla $i > n/2$ tak, aby każde i było w tym samym stanie co $i + n/2$.

(n nie jest potęgą dwójki) Pokażemy niemożliwość rozwiązania na przykładzie $n = 12$, rozważmy tylko przyciski 1, 5, 9 rozłożone równomiernie na cyklu. Jeśli przyciski 1 i 5 mają różne wartości, to dla każdej operacji (na wszystkich przyciskach) tak możemy pokręcić zamkiem, że 1 i 5 dalej będą mieć różne wartości. Czyli jest ciąg rotacji będący kontrprzykładem dla każdego ciągu $\bar{A}(n)$.

Komentarz. Pokazaliśmy, że dla potęg 2 istnieje rozwiązanie długści co najwyżej $2^n - 1$. Jeśli chcemy tylko dojść do konfiguracji że wszystkie przyciski mają tę samą warość to np. dla $n = 4$ wystarczy i potrzeba 7 ruchów, zamiast 15, a dla $n = 8$ wystarczy 125 ruchów, poczas gdy $2^8 - 1 = 255$.

Pytanie: jakie są naprawdę minimalne długości rozwiązań ?