Seminarium - Matematyka Dyskretna 29/10

Robbins and Ardila meet Berstel

Mikołaj Kondratek

Wydział Matematyki i Informatyki Uniwersytetu Jagiellońskiego

29 Października 2020

Plan

- Problem
- Tło problemu
- Twierdzenie Zeckendorfa
- Reprezentacja Fibonacciego Liczby
- Transduktor
- "Podziały Fibonacciego"
- Reprezentacja liniowa automatu
- Obliczanie $a(n)$

Problem

Współczynniki przy $\mathbf{x^n}$ *w nieskończonym produkcie*

$$\prod_{n\geq 2}^{\infty}(1 - x^{F_n})$$

Problem

Współczynniki przy $\mathbf{x^n}$ *w nieskończonym produkcie*

$$\prod_{n\geq 2}^{\infty}(1-x^{F_n})$$

$$=(1-x)(1-x^2)(1-x^3)(1-x^5)(1-x^8)\dots$$

Problem

Współczynniki przy $\mathbf{x^n}$ *w nieskończonym produkcie*

$$\prod_{n\geq 2}^{\infty}(1-x^{F_n})$$

$$=(1-x)(1-x^2)(1-x^3)(1-x^5)(1-x^8)\ldots$$

$$=1-x+x^2+x^4+\ldots$$

Problem

Współczynniki przy $\mathbf{x^n}$ *w nieskończonym produkcie*

$$\prod_{n\geq 2}^{\infty}(1-x^{F_n})$$

$$= (1-x)(1-x^2)(1-x^3)(1-x^5)(1-x^8)\dots$$

$$= 1 - x + x^2 + x^4 + \dots$$

*wynoszą zawsze **-1**, **0** albo **1***

Tło problemu

Robbins and Ardila meet Berstel, cd

Tło problemu

Robbins and Ardila meet Berstel, cd

Tło problemu

Robbins and Ardila meet Berstel, cd

Tło problemu

Robbins and Ardila meet Berstel, cd

Tło problemu

Robbins and Ardila meet Berstel, cd

J. Shallit

N. Robbins

F. Ardila

J. Berstel

Twierdzenie Zeckendorfa

Twierdzenie Zeckendorfa

Każda dodatnia liczba całkowita może być przedstawiona jednoznacznie jako suma jednej lub więcej różnych liczb Fibonacciego w taki sposób, że owa suma nie zawiera żadnych dwóch kolejnych liczb Fibonacciego.

Twierdzenie Zeckendorfa

Twierdzenie Zeckendorfa

Każda dodatnia liczba całkowita może być przedstawiona jednoznacznie jako suma jednej lub więcej różnych liczb Fibonacciego w taki sposób, że owa suma nie zawiera żadnych dwóch kolejnych liczb Fibonacciego.

Czyli, jeżeli N jest dowolną dodatnią liczbą całkowitą, istnieją takie liczby całkowite $c_i \geqslant 2$ spełniające $c_{i+1} > c_i + 1$, że:

$$N = \sum_{i=0}^{k} F_{c_i}$$

$$44 =$$

$$44 =$$

$$34 + 8 + 2$$

$$\mathbf{44} =$$

$$\mathbf{34 + 8 + 2}$$

$$34 + 5 + 3 + 2$$

$$\mathbf{44} =$$

$$\mathbf{34} + \mathbf{8} + \mathbf{2}$$

$$34 + 5 + 3 + 2$$
$$21 + 13 + 8 + 2$$

Twierdzenie Zeckendorfa
Przykład

$$\mathbf{44 =}$$

$$\mathbf{34 + 8 + 2}$$

$$34 + 5 + 3 + 2$$
$$21 + 13 + 8 + 2$$
$$21 + 13 + 5 + 3 + 2$$

$\mathbf{44} =$

Reprezentacja Zeckendorfa

$\mathbf{34} + \mathbf{8} + \mathbf{2}$

Reprezentacja "nie-Zeckendorfa"

$34 + 5 + 3 + 2$

$21 + 13 + 8 + 2$

$21 + 13 + 5 + 3 + 2$

Dla $\mathbf{n} = \mathbf{1}, \mathbf{2}, \mathbf{3}$ ✓

Dla $\mathbf{n} = \mathbf{1}, \mathbf{2}, \mathbf{3}$ ✓

Dla $\mathbf{n} = \mathbf{4}$ mamy $4 = 3 + 1$ ✓

Dla $\mathbf{n} = \mathbf{1}, \mathbf{2}, \mathbf{3}$ ✓

Dla $\mathbf{n} = \mathbf{4}$ mamy $4 = 3 + 1$ ✓

Jeśli n jest liczbą Fibonacciego ✓

Dla $\mathbf{n} = \mathbf{1}, \mathbf{2}, \mathbf{3}$ ✓

Dla $\mathbf{n} = \mathbf{4}$ mamy $4 = 3 + 1$ ✓

Jeśli n jest liczbą Fibonacciego ✓

w.p.p. $\exists \mathbf{j}$ takie, że $\mathbf{F_j} < \mathbf{n} < \mathbf{F_{j+1}}$

Dla $\mathbf{n} = \mathbf{1}, \mathbf{2}, \mathbf{3}$ ✓

Dla $\mathbf{n} = \mathbf{4}$ mamy $4 = 3 + 1$ ✓

Jeśli n jest liczbą Fibonacciego ✓

w.p.p. $\exists \mathbf{j}$ takie, że $\mathbf{F_j} < \mathbf{n} < \mathbf{F_{j+1}}$

$\forall \mathbf{a} < \mathbf{n}$, $\mathbf{a}$ jest ok (teza indukcyjna)

Dla $\mathbf{n} = \mathbf{1}, \mathbf{2}, \mathbf{3}$ ✓

Dla $\mathbf{n} = \mathbf{4}$ mamy $4 = 3 + 1$ ✓

Jeśli n jest liczbą Fibonacciego ✓

w.p.p. $\exists \mathbf{j}$ takie, że $\mathbf{F_j} < \mathbf{n} < \mathbf{F_{j+1}}$

$\forall \mathbf{a} < \mathbf{n}$, $\mathbf{a}$ jest ok (teza indukcyjna)

Rozważmy $\mathbf{a} = \mathbf{n} - \mathbf{F_j}$ ($\mathbf{a} < \mathbf{n}$, zatem $\mathbf{a}$ ✓)

Dla $\mathbf{n} = \mathbf{1}, \mathbf{2}, \mathbf{3}$ ✓

Dla $\mathbf{n} = \mathbf{4}$ mamy $4 = 3 + 1$ ✓

Jeśli n jest liczbą Fibonacciego ✓

w.p.p. $\exists \mathbf{j}$ takie, że $\mathbf{F_j} < \mathbf{n} < \mathbf{F_{j+1}}$

$\forall \mathbf{a} < \mathbf{n}$, $\mathbf{a}$ jest ok (teza indukcyjna)

Rozważmy $\mathbf{a} = \mathbf{n} - \mathbf{F_j}$ ($\mathbf{a} < \mathbf{n}$, zatem $\mathbf{a}$ ✓)

Jednocześnie $\mathbf{a} < \mathbf{F_{j+1}} - \mathbf{F_j} = \mathbf{F_{j-1}}$,

zatem reprezentacja $\mathbf{a}$ nie zawiera $\mathbf{F_{j-1}}$

Dla $\mathbf{n} = \mathbf{1}, \mathbf{2}, \mathbf{3}$ ✓

Dla $\mathbf{n} = \mathbf{4}$ mamy $4 = 3 + 1$ ✓

Jeśli n jest liczbą Fibonacciego ✓

w.p.p. $\exists \mathbf{j}$ takie, że $\mathbf{F_j} < \mathbf{n} < \mathbf{F_{j+1}}$

$\forall \mathbf{a} < \mathbf{n}$, $\mathbf{a}$ jest ok (teza indukcyjna)

Rozważmy $\mathbf{a} = \mathbf{n} - \mathbf{F_j}$ ($\mathbf{a} < \mathbf{n}$, zatem $\mathbf{a}$ ✓)

Jednocześnie $\mathbf{a} < \mathbf{F_{j+1}} - \mathbf{F_j} = \mathbf{F_{j-1}}$,

zatem reprezentacja $\mathbf{a}$ nie zawiera $\mathbf{F_{j-1}}$

Zatem $\mathbf{n}$ da się przedstawić jak sumę $\mathbf{F_j}$ i reprezentacji Zeckendorfa $\mathbf{a}$. ■

Lemat

Suma niepustego zbioru różnych, nie następujący po sobie wyrazów ciągu Fibonacciego, w którym największa liczba ma wartość F_j jest ściśle mniejsza od F_{j+1}.

Lemat

Suma niepustego zbioru różnych, nie następujący po sobie wyrazów ciągu Fibonacciego, w którym największa liczba ma wartość F_j jest ściśle mniejsza od F_{j+1}.

Dowód przez indukcję na j...

Weźmy dwa niepuste zbiory o różnych, nie następujący po sobie wyrazów ciągu Fibonacciego **S** i **T**

Weźmy dwa niepuste zbiory o różnych, nie następujący po sobie wyrazów ciągu Fibonacciego **S** i **T** o tej samej sumie.

Weźmy dwa niepuste zbiory o różnych, nie następujący po sobie wyrazów ciągu Fibonacciego $\mathbf{S}$ i $\mathbf{T}$ o tej samej sumie.

Rozważmy zbiory $\mathbf{S}' = \mathbf{S} \setminus \mathbf{T}$ i $\mathbf{T}' = \mathbf{T} \setminus \mathbf{S}$,

Weźmy dwa niepuste zbiory o różnych, nie następujący po sobie wyrazów ciągu Fibonacciego $\mathbf{S}$ i $\mathbf{T}$ o tej samej sumie.

Rozważmy zbiory $\mathbf{S}' = \mathbf{S} \setminus \mathbf{T}$ i $\mathbf{T}' = \mathbf{T} \setminus \mathbf{S}$, czyli zbiory pozbawione wspólnych elementów zbiorów $\mathbf{S}$ i $\mathbf{T}$.

Weźmy dwa niepuste zbiory o różnych, nie następujący po sobie wyrazów ciągu Fibonacciego $\mathbf{S}$ i $\mathbf{T}$ o tej samej sumie.

Rozważmy zbiory $\mathbf{S}' = \mathbf{S} \setminus \mathbf{T}$ i $\mathbf{T}' = \mathbf{T} \setminus \mathbf{S}$, czyli zbiory pozbawione wspólnych elementów zbiorów $\mathbf{S}$ i $\mathbf{T}$.

Ponieważ $\mathbf{S}$ i $\mathbf{T}$ miały tę samą sumę,

Weźmy dwa niepuste zbiory o różnych, nie następujący po sobie wyrazów ciągu Fibonacciego $\mathbf{S}$ i $\mathbf{T}$ o tej samej sumie.

Rozważmy zbiory $\mathbf{S}' = \mathbf{S} \setminus \mathbf{T}$ i $\mathbf{T}' = \mathbf{T} \setminus \mathbf{S}$, czyli zbiory pozbawione wspólnych elementów zbiorów $\mathbf{S}$ i $\mathbf{T}$.

Ponieważ $\mathbf{S}$ i $\mathbf{T}$ miały tę samą sumę, a z obydwu usunęliśmy te same elementy należące do $\mathbf{S} \cap \mathbf{T}$,

Weźmy dwa niepuste zbiory o różnych, nie następujący po sobie wyrazów ciągu Fibonacciego $\mathbf{S}$ i $\mathbf{T}$ o tej samej sumie.

Rozważmy zbiory $\mathbf{S}' = \mathbf{S} \setminus \mathbf{T}$ i $\mathbf{T}' = \mathbf{T} \setminus \mathbf{S}$, czyli zbiory pozbawione wspólnych elementów zbiorów $\mathbf{S}$ i $\mathbf{T}$.

Ponieważ $\mathbf{S}$ i $\mathbf{T}$ miały tę samą sumę, a z obydwu usunęliśmy te same elementy należące do $\mathbf{S} \cap \mathbf{T}$, zatem suma elementów w $\mathbf{S}'$ i $\mathbf{T}'$ również jest taka sama.

Pokażemy nie wprost, że jeden ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty...

Twierdzenie Zeckendorfa

Dowód unikalności reprezentacji, cd

Pokażemy nie wprost, że jeden ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty...

Załóżmy przeciwnie, zbiory $\mathbf{S}'$ i $\mathbf{T}'$ są niepuste,

Pokażemy nie wprost, że jeden ze zbiorów $\mathbf{S'}$ i $\mathbf{T'}$ jest pusty...

Załóżmy przeciwnie, zbiory $\mathbf{S'}$ i $\mathbf{T'}$ są niepuste, zawierają największe wyrazy, odpowiednio $\mathbf{F_s}$ i $\mathbf{F_t}$.

Pokażemy nie wprost, że jeden ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty...

Załóżmy przeciwnie, zbiory $\mathbf{S}'$ i $\mathbf{T}'$ są niepuste, zawierają największe wyrazy, odpowiednio $\mathbf{F_s}$ i $\mathbf{F_t}$. Bez straty ogólności, $\mathbf{F_s} < \mathbf{F_t}$.

Pokażemy nie wprost, że jeden ze zbiorów $\mathbf{S'}$ i $\mathbf{T'}$ jest pusty...

Załóżmy przeciwnie, zbiory $\mathbf{S'}$ i $\mathbf{T'}$ są niepuste, zawierają największe wyrazy, odpowiednio $\mathbf{F_s}$ i $\mathbf{F_t}$. Bez straty ogólności, $\mathbf{F_s} < \mathbf{F_t}$.

Z *lematu* suma wyrazów w $\mathbf{S'}$ jest ściśle mniejsza od $\mathbf{F_{s+1}}$,

Pokażemy nie wprost, że jeden ze zbiorów $\mathbf{S'}$ i $\mathbf{T'}$ jest pusty...

Załóżmy przeciwnie, zbiory $\mathbf{S'}$ i $\mathbf{T'}$ są niepuste, zawierają największe wyrazy, odpowiednio $\mathbf{F_s}$ i $\mathbf{F_t}$. Bez straty ogólności, $\mathbf{F_s} < \mathbf{F_t}$.

Z *lematu* suma wyrazów w $\mathbf{S'}$ jest ściśle mniejsza od $\mathbf{F_{s+1}}$, a zatem i ściśle mniejsza od $\mathbf{F_t}$,

Pokażemy nie wprost, że jeden ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty...

Załóżmy przeciwnie, zbiory $\mathbf{S}'$ i $\mathbf{T}'$ są niepuste, zawierają największe wyrazy, odpowiednio $\mathbf{F_s}$ i $\mathbf{F_t}$. Bez straty ogólności, $\mathbf{F_s} < \mathbf{F_t}$.

Z *lematu* suma wyrazów w $\mathbf{S}'$ jest ściśle mniejsza od $\mathbf{F_{s+1}}$, a zatem i ściśle mniejsza od $\mathbf{F_t}$, podczas gdy suma elementów w $\mathbf{T}'$ wynosi przynajmniej $\mathbf{F_t}$. Wnioskujemy, że któryś ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty.

Pokażemy nie wprost, że jeden ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty...

Załóżmy przeciwnie, zbiory $\mathbf{S}'$ i $\mathbf{T}'$ są niepuste, zawierają największe wyrazy, odpowiednio $\mathbf{F_s}$ i $\mathbf{F_t}$. Bez straty ogólności, $\mathbf{F_s} < \mathbf{F_t}$.

Z *lematu* suma wyrazów w $\mathbf{S}'$ jest ściśle mniejsza od $\mathbf{F_{s+1}}$, a zatem i ściśle mniejsza od $\mathbf{F_t}$, podczas gdy suma elementów w $\mathbf{T}'$ wynosi przynajmniej $\mathbf{F_t}$. Wnioskujemy, że któryś ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty.

Bez straty ogólności przyjmijmy, że pusty jest $\mathbf{S}'$.

Pokażemy nie wprost, że jeden ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty...

Załóżmy przeciwnie, zbiory $\mathbf{S}'$ i $\mathbf{T}'$ są niepuste, zawierają największe wyrazy, odpowiednio $\mathbf{F_s}$ i $\mathbf{F_t}$. Bez straty ogólności, $\mathbf{F_s} < \mathbf{F_t}$.

Z *lematu* suma wyrazów w $\mathbf{S}'$ jest ściśle mniejsza od $\mathbf{F_{s+1}}$, a zatem i ściśle mniejsza od $\mathbf{F_t}$, podczas gdy suma elementów w $\mathbf{T}'$ wynosi przynajmniej $\mathbf{F_t}$. Wnioskujemy, że któryś ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty.

Bez straty ogólności przyjmijmy, że pusty jest $\mathbf{S}'$. Zatem suma elementów w $\mathbf{S}'$ wynosi $\mathbf{0}$.

Pokażemy nie wprost, że jeden ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty...

Załóżmy przeciwnie, zbiory $\mathbf{S}'$ i $\mathbf{T}'$ są niepuste, zawierają największe wyrazy, odpowiednio $\mathbf{F_s}$ i $\mathbf{F_t}$. Bez straty ogólności, $\mathbf{F_s} < \mathbf{F_t}$.

Z *lematu* suma wyrazów w $\mathbf{S}'$ jest ściśle mniejsza od $\mathbf{F_{s+1}}$, a zatem i ściśle mniejsza od $\mathbf{F_t}$, podczas gdy suma elementów w $\mathbf{T}'$ wynosi przynajmniej $\mathbf{F_t}$. Wnioskujemy, że któryś ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty.

Bez straty ogólności przyjmijmy, że pusty jest $\mathbf{S}'$. Zatem suma elementów w $\mathbf{S}'$ wynosi $\mathbf{0}$. Taką samą sumę ma $\mathbf{T}'$. Ale $\mathbf{T}'$ zawiera jedynie liczby dodatnie,

Pokażemy nie wprost, że jeden ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty...

Załóżmy przeciwnie, zbiory $\mathbf{S}'$ i $\mathbf{T}'$ są niepuste, zawierają największe wyrazy, odpowiednio $\mathbf{F_s}$ i $\mathbf{F_t}$. Bez straty ogólności, $\mathbf{F_s} < \mathbf{F_t}$.

Z *lematu* suma wyrazów w $\mathbf{S}'$ jest ściśle mniejsza od $\mathbf{F_{s+1}}$, a zatem i ściśle mniejsza od $\mathbf{F_t}$, podczas gdy suma elementów w $\mathbf{T}'$ wynosi przynajmniej $\mathbf{F_t}$. Wnioskujemy, że któryś ze zbiorów $\mathbf{S}'$ i $\mathbf{T}'$ jest pusty.

Bez straty ogólności przyjmijmy, że pusty jest $\mathbf{S}'$. Zatem suma elementów w $\mathbf{S}'$ wynosi $\mathbf{0}$. Taką samą sumę ma $\mathbf{T}'$. Ale $\mathbf{T}'$ zawiera jedynie liczby dodatnie, zatem również jest pusty.

Wnioskujemy, że $\mathbf{S}' = \mathbf{T}' = \emptyset$, zatem $\mathbf{S}' = \mathbf{T}'$.
Reprezentacja Zeckendorfa jest unikalna. ■

Reprezentacja Fibonacciego Liczby

Reprezentacja Fibonacciego

- $c_r = 1$
- $c_k = 0$ lub 1

$$\sum_{k=2}^{r} c_k F_k$$

$[22]_F = 110001$

$[22]_F = 1000001$

$[6]_F =$

$[22]_F = 110001$

$[22]_F = 1000001$

$[6]_F = 1001$

$[6]_F =$

$[22]_F = 110001$

$[22]_F = 1000001$

$[6]_F = 1001$

$[6]_F = 111$

Reprezentacja Fibonacciego Liczby

Minimalna Reprezentacja Fibonacciego

- $c_r = 1$
- $c_k = 0$ or 1
- $c_{k-1}c_k = 0$ dla k takich, że $3 \leq k \leq r$

$$\sum_{k=2}^{r} c_k F_k$$

Reprezentacja Fibonacciego Liczby

Minimalna Reprezentacja Fibonacciego

- $c_r = 1$
- $c_k = 0 \text{ or } 1$
- $c_{k-1}c_k = 0$ dla k takich, że $3 \leq k \leq r$

$$\sum_{k=2}^{r} c_k F_k$$

Opis wyrażeniem regularnym

$$C_F := \epsilon + 1(0 + 01)*$$

$\cancel{(22)_F = 110001}$

$(22)_F = 1000001$

$(6)_F = 1001$

$\cancel{(6)_F = 111}$

Robbins and Ardila meet Berstel

Automat skończony

$$M = (Q, \sum, \delta, q_0, F)$$

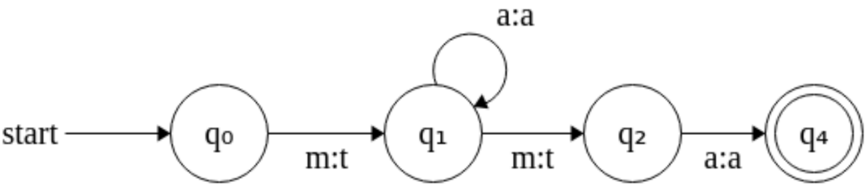
Automat skończony

$M = (Q, \sum, \delta, q_0, F)$

Transduktor

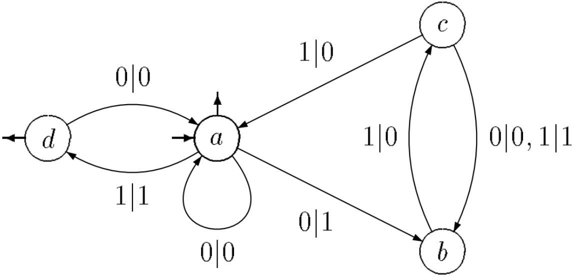
$T = (Q, \sum_1, \sum_2, \delta, \sigma, q_0, F)$

Transduktor
Transduktor ma*ma

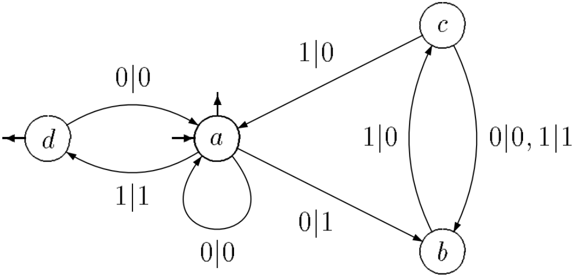


Transduktor

Transduktor Berstela (aka Fibonacci normalizer)



Wejściowe słowo musi zaczynać się od 0.

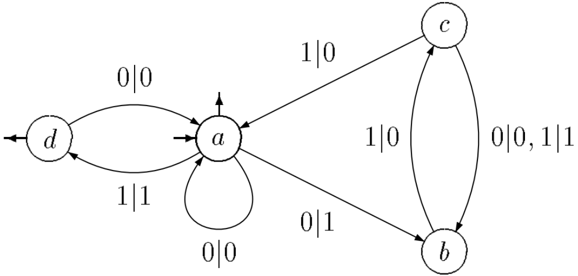


Wejściowe słowo musi zaczynać się od 0.

$[10]_F =$	**0**	**1**	**1**	**1**	**0**
	a 0				

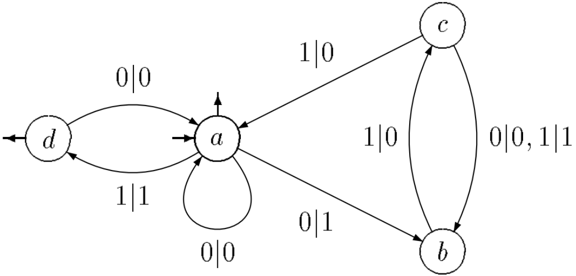
Transduktor

Transduktor Berstela (aka Fibonacci normalizer)



Wejściowe słowo musi zaczynać się od 0.

$[10]_F =$	**0**	**1**	**1**	**1**	**0**
	a 0	d 01			

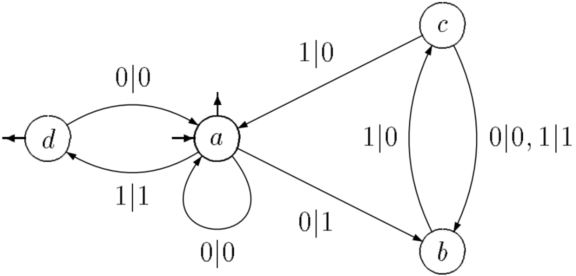


Wejściowe słowo musi zaczynać się od 0.

$[10]_F =$	**0**	**1**	**1**	**1**	**0**
	a 0	d 01	error		

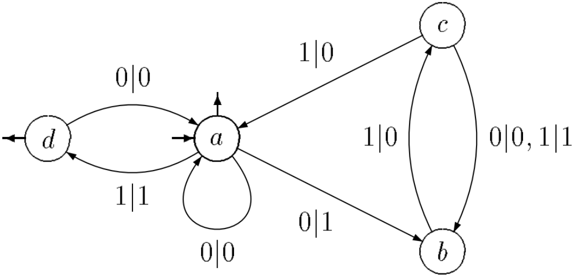
Transduktor

Transduktor Berstela (aka Fibonacci normalizer)



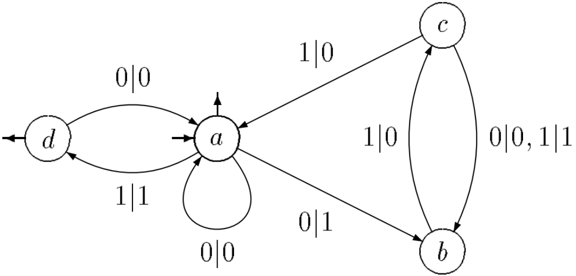
Wejściowe słowo musi zaczynać się od 0.

$[10]_F =$	**0**	**1**	**1**	**1**	**0**
	a 0	d 01	error		
	b 1				



Wejściowe słowo musi zaczynać się od 0.

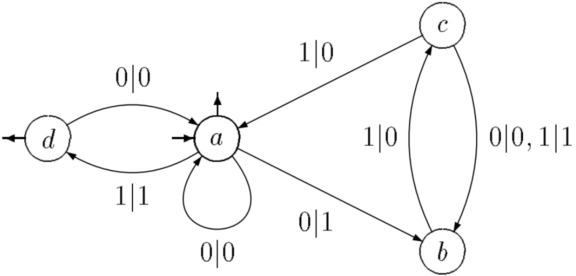
$[10]_F =$	**0**	**1**	**1**	**1**	**0**
	a 0	d 01	error		
	b 1	c 10			



Wejściowe słowo musi zaczynać się od 0.

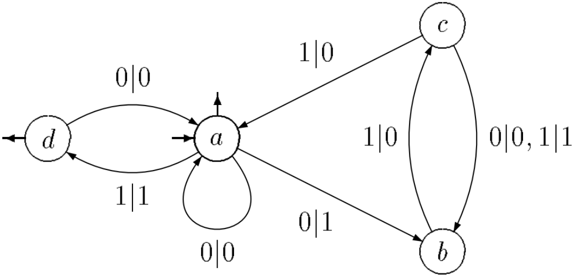
$[10]_F =$	**0**	**1**	**1**	**1**	**0**
	a 0	d 01	error		
	b 1	c 10	b 101		

Transduktor Berstela (aka Fibonacci normalizer)



Wejściowe słowo musi zaczynać się od 0.

$[10]_F =$	**0**	**1**	**1**	**1**	**0**
	a 0	d 01	error		
	b 1	c 10	b 101	c 1010	

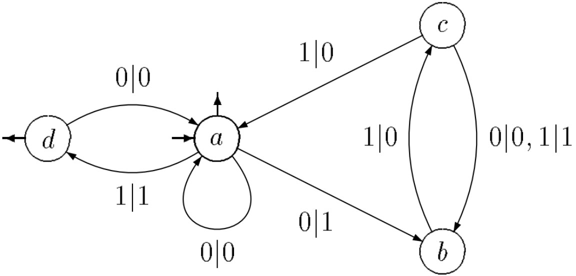


Wejściowe słowo musi zaczynać się od 0.

$[10]_F =$	**0**	**1**	**1**	**1**	**0**
	a 0	d 01	error		
	b 1	c 10	b 101	c 1010	b ~~10100~~

Transduktor

Transduktor Berstela (aka Fibonacci normalizer)

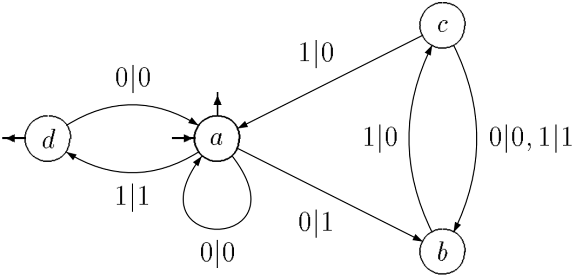


Wejściowe słowo musi zaczynać się od 0.

$[10]_F =$	**0**	**1**	**1**	**1**	**0**
	a 0	d 01	error		
	b 1	c 10	b 101	c 1010	b ~~10100~~
	-	-			

Transduktor

Transduktor Berstela (aka Fibonacci normalizer)

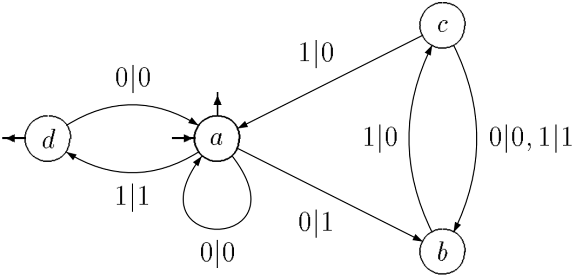


Wejściowe słowo musi zaczynać się od 0.

$[10]_F =$	**0**	**1**	**1**	**1**	**0**
	a 0	d 01	error		
	b 1	c 10	b 101	c 1010	b ~~10100~~
	-	-	a 100		

Transduktor

Transduktor Berstela (aka Fibonacci normalizer)

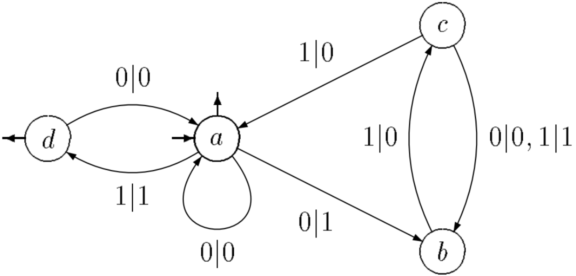


Wejściowe słowo musi zaczynać się od 0.

$[10]_F =$	**0**	**1**	**1**	**1**	**0**
	a 0	d 01	error		
	b 1	c 10	b 101	c 1010	b ~~10100~~
	-	-	a 100	d 1001	

Transduktor

Transduktor Berstela (aka Fibonacci normalizer)

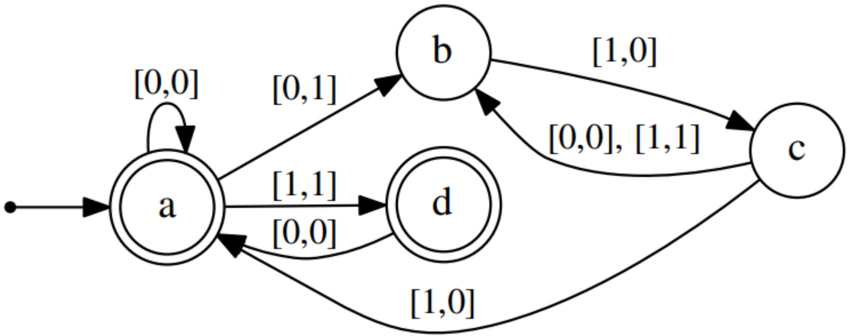


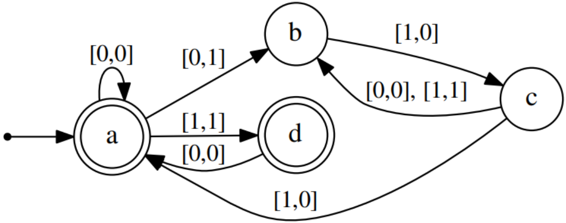
Wejściowe słowo musi zaczynać się od 0.

$[10]_F =$	**0**	**1**	**1**	**1**	**0**
	a 0	d 01	error		
	b 1	c 10	b 101	c 1010	b ~~10100~~
	-	-	a 100	d 1001	a 10010

Transduktor

Automat Berstela (M)

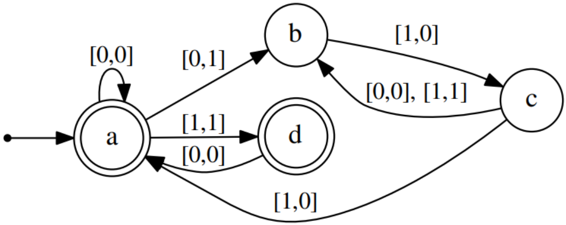




Jeśli słowo literowane na pierwszej współrzędnej $x = [n]_F$,
jest równoważne (ma tę samą wartość), co kanoniczna (minimalna) reprezentacja słowa literowana na drugiej współrzędnej $y = (m)_F$
to automat akceptuje $(n \stackrel{?}{=} m)$.

Transduktor

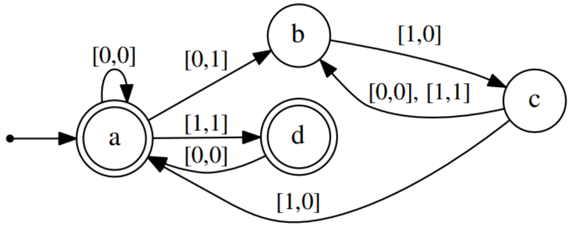
Automat Berstela, cd 2



$$x \in \{0,1\}^* \wedge y \in 0^* C_F \wedge [x]_F = [y]_F$$

Transduktor

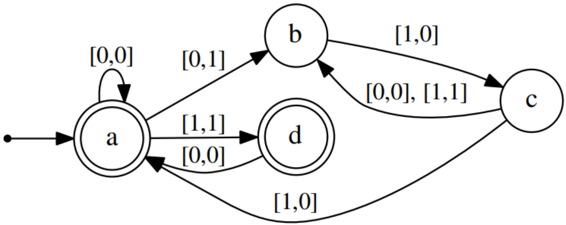
Automat Berstela, przykład



$[?]_F =$	0	1	0	1	1
$(8)_F =$	1	0	0	0	0
a	b				

Transduktor

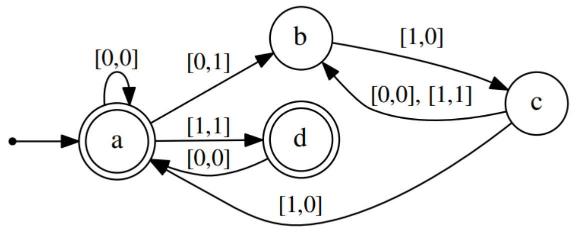
Automat Berstela, przykład



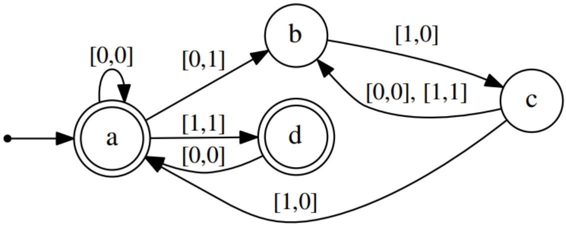
$$\begin{array}{rccccc} [?]_F = & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ (8)_F = & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathrm{a} & \mathrm{b} & \mathrm{c} & & & \end{array}$$

Transduktor

Automat Berstela, przykład



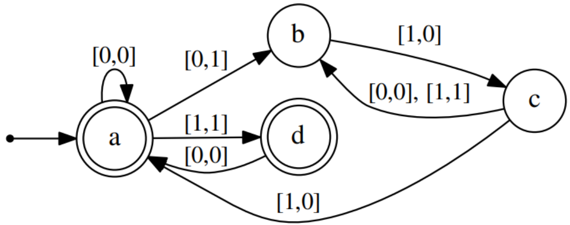
$$
\begin{array}{rccccc}
[?]_F = & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\
(8)_F = & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\hline
\mathsf{a} & \mathsf{b} & \mathsf{c} & \mathsf{b} & &
\end{array}
$$



$$\begin{array}{lccccc} [?]_F = & 0 & 1 & 0 & 1 & 1 \\ (8)_F = & 1 & 0 & 0 & 0 & 0 \\ \hline \text{a} & \text{b} & \text{c} & \text{b} & \text{c} & \end{array}$$

Transduktor

Automat Berstela, przykład



$[?]_F =$	**0**	**1**	**0**	**1**	**1**
$(8)_F =$	**1**	**0**	**0**	**0**	**0**
a	b	c	b	c	a

Liczba podziałów Fibonacciego liczby

Przyjmijmy, że y jest w postaci minimalnej reprezentacji Fibonacciego liczby n. Definiujemy $\mathbf{r(n)}$, liczbę takich x, ze automat Berstela M akceptuje $x \times y$.

Podziały Fibonacciego

Liczba podziałów Fibonacciego liczby

Przyjmijmy, że y jest w postaci minimalnej reprezentacji Fibonacciego liczby n. Definiujemy $\mathbf{r(n)}$, liczbę takich x, ze automat Berstela M akceptuje $x \times y$.

Przykład: $r(8) =$

Podziały Fibonacciego

Liczba podziałów Fibonacciego liczby

Przyjmijmy, że y jest w postaci minimalnej reprezentacji Fibonacciego liczby n. Definiujemy $\mathbf{r(n)}$, liczbę takich x, ze automat Berstela M akceptuje $x \times y$.

Przykład: $r(8) = 3$, bo... $10000, 1100, 1011$

Podziały Fibonacciego

Liczba parzystych podziałów Fibonacciego liczby

Definiujemy $\mathbf{r_e(n)}$, liczbę podziałów o parzystej liczbie jedynek.

Podziały Fibonacciego

Liczba parzystych podziałów Fibonacciego liczby

Definiujemy $\mathbf{r_e(n)}$, liczbę podziałów o parzystej liczbie jedynek.

Przykład: $r_e(8) =$

Liczba parzystych podziałów Fibonacciego liczby

Definiujemy $\mathbf{r_e(n)}$, liczbę podziałów o parzystej liczbie jedynek.

Przykład: $r_e(8) = 1$, bo...1100

Podziały Fibonacciego

Liczba parzystych podziałów Fibonacciego liczby

Definiujemy $\mathbf{r_e(n)}$, liczbę podziałów o parzystej liczbie jedynek.

Przykład: $r_e(8) = 1$, bo. . . 1100

Liczba nieparzystych podziałów Fibonacciego liczby

Definiujemy $\mathbf{r_o(n)}$, liczbę takich o nieparzystej liczbie jedynek.

Podziały Fibonacciego

Liczba parzystych podziałów Fibonacciego liczby

Definiujemy $\mathbf{r_e(n)}$, liczbę podziałów o parzystej liczbie jedynek.

Przykład: $r_e(8) = 1$, bo. . . 1100

Liczba nieparzystych podziałów Fibonacciego liczby

Definiujemy $\mathbf{r_o(n)}$, liczbę takich o nieparzystej liczbie jedynek.

Przykład: $r_o(8) =$

Podziały Fibonacciego

Liczba parzystych podziałów Fibonacciego liczby

Definiujemy $\mathbf{r_e(n)}$, liczbę podziałów o parzystej liczbie jedynek.

Przykład: $r_e(8) = 1$, bo... 1100

Liczba nieparzystych podziałów Fibonacciego liczby

Definiujemy $\mathbf{r_o(n)}$, liczbę takich o nieparzystej liczbie jedynek.

Przykład: $r_o(8) = 2$, bo... $10000, 1011$

Podziały Fibonacciego

Liczba parzystych podziałów Fibonacciego liczby

Definiujemy $\mathbf{r_e(n)}$, liczbę podziałów o parzystej liczbie jedynek.

Przykład: $r_e(8) = 1$, bo... 1100

Liczba nieparzystych podziałów Fibonacciego liczby

Definiujemy $\mathbf{r_o(n)}$, liczbę takich o nieparzystej liczbie jedynek.

Przykład: $r_o(8) = 2$, bo... $10000, 1011$

Spostrzeżenie

$\mathbf{r(n) = r_e(n) + r_o(n)}$

Podziały Fibonacciego

Spostrzeżenie Robbinsa

$$\begin{cases} r(n) = r_e(n) + r_o(n) \\ a(n) = r_e(n) - r_o(n) \end{cases} \tag{1}$$

Współczynnik przy x^n

- $\prod_{n\geq 2}^{\infty}(1 - x^{F_n})$

Podziały Fibonacciego

Spostrzeżenie Robbinsa

$$\begin{cases} r(n) = r_e(n) + r_o(n) \\ a(n) = r_e(n) - r_o(n) \end{cases} \tag{1}$$

Współczynnik przy x^n

- $\prod_{n\geq 2}^{\infty}(1 - x^{F_n})$
  $= (1-x)(1-x^2)(1-x^3)(1-x^5)(1-x^8)\dots$

Podziały Fibonacciego

Spostrzeżenie Robbinsa

$$\begin{cases} r(n) = r_e(n) + r_o(n) \\ a(n) = r_e(n) - r_o(n) \end{cases} \tag{1}$$

Współczynnik przy x^n

- $\prod_{n\geq 2}^{\infty}(1 - x^{F_n})$

  $= (1-x)(1-x^2)(1-x^3)(1-x^5)(1-x^8)\dots$

  $= \sum_{n\geq 0} a(n)x^n$

Spostrzeżenie Robbinsa

$$\begin{cases} r(n) = r_e(n) + r_o(n) \\ a(n) = r_e(n) - r_o(n) \end{cases} \tag{1}$$

Współczynnik przy x^n

- $\prod_{n\geq 2}^{\infty}(1 - x^{F_n})$
  $= (1 - x)(1 - x^2)(1 - x^3)(1 - x^5)(1 - x^8)\dots$
  $= \sum_{n\geq 0} a(n)x^n$
- $\mathbf{a(n) = 2r_e(n) - r(n)}$

Reprezentacja liniowa automatu — macierz

Niech $A = (S, \sum, \delta, I, F)$,

Reprezentacja liniowa automatu — macierz

Niech $A = (S, \sum, \delta, I, F)$, a $S = s_1, \ldots, s_n$.

Reprezentacja liniowa automatu — macierz

Niech $A = (S, \sum, \delta, I, F)$, a $S = s_1, \ldots, s_n$.

Dla każdego symbolu $a \in \sum$ definiujemy macierz $M(a)$ na nieujemnych liczbach całkowitych w sposób następujący:

$$M_{ij} = \begin{cases} 1 & \text{if} s_j \in \delta(s_i, a) \\ 0 & \text{wpp} \end{cases}$$

Dodatkowo, definiujemy wektory v_I oraz v_F odpowiednio w sposób następujący. . .

Dodatkowo, definiujemy wektory v_I oraz v_F odpowiednio w sposób następujący...

Reprezentacja liniowa automatu - wektor v_I

i-ty element v_I wynosi 1 wtedy i tylko wtedy, gdy s_i jest stanem początkowym, wpp jest 0.

Reprezentacja liniowa automatu

Dodatkowo, definiujemy wektory v_I oraz v_F odpowiednio w sposób następujący. . .

Reprezentacja liniowa automatu - wektor v_I

i-ty element v_I wynosi 1 wtedy i tylko wtedy, gdy s_i jest stanem początkowym, wpp jest 0.

Reprezentacja liniowa automatu - wektor v_F

i-ty element v_F wynosi 1 wtedy i tylko wtedy, gdy s_i jest stanem końcowym, wpp jest 0.

Reprezentacja liniowa automatu

Dodatkowo, definiujemy wektory v_I oraz v_F odpowiednio w sposób następujący. . .

Reprezentacja liniowa automatu - wektor v_I

i-ty element v_I wynosi 1 wtedy i tylko wtedy, gdy s_i jest stanem początkowym, wpp jest 0.

Reprezentacja liniowa automatu - wektor v_F

i-ty element v_F wynosi 1 wtedy i tylko wtedy, gdy s_i jest stanem końcowym, wpp jest 0.

Reprezentacja liniowa automatu

Reprezentacja liniowa automatu Berstela

Definiujemy macierze o wartościach $(\mu(a))_{ij}$ wynoszących liczbę ścieżek $[*, a]$ (gdzie $*$ oznacza symbol dowolny) ze stanu i do stanu j.
Otrzymujemy reprezentację liniową (v, μ, w).

Reprezentacja liniowa automatu Berstela

Definiujemy macierze o wartościach $(\mu(a))_{ij}$ wynoszących liczbę ścieżek $[*, a]$ (gdzie $*$ oznacza symbol dowolny) ze stanu i do stanu j.
Otrzymujemy reprezentację liniową (v, μ, w).

$$v = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}; \quad \mu(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}; \quad \mu(1) = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad w = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Reprezentacja liniowa automatu Berstela

Definiujemy macierze o wartościach $(\mu(a))_{ij}$ wynoszących liczbę ścieżek $[*, a]$ (gdzie $*$ oznacza symbol dowolny) ze stanu i do stanu j.
Otrzymujemy reprezentację liniową (v, μ, w).

$$v = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}; \quad \mu(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}; \quad \mu(1) = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad w = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Morfizm

$$\mu(x)\mu(y) = \mu(xy)$$

Reprezentacja liniowa automatu Berstela

Definiujemy macierze o wartościach $(\mu(a))_{ij}$ wynoszących liczbę ścieżek $[*, a]$ (gdzie $*$ oznacza symbol dowolny) ze stanu i do stanu j.
Otrzymujemy reprezentację liniową (v, μ, w).

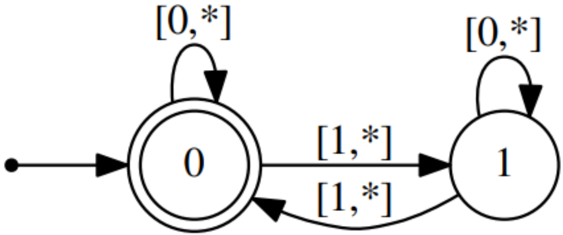
$$v = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}; \quad \mu(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}; \quad \mu(1) = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad w = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Morfizm

$$\mu(x)\mu(y) = \mu(xy)$$

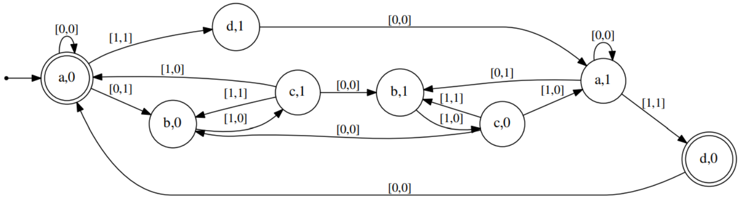
$$\text{Jeśli } [n]_F = x, \text{ to } r(n) = v\mu(x)w$$

Reprezentacja liniowa automatu
Automat "parzystości"



Reprezentacja liniowa automatu

Automat Berstela * automat "parzystości"



Reprezentacja liniowa automatu

Automat Berstela * automat "parzystości, reprezentacja liniowa"

$$v = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad \mu'(0) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad \mu'(1) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad w' = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Reprezentacja liniowa automatu
Automat automatów..."

$$v'' = \begin{bmatrix} 2v & -v' \end{bmatrix}; \quad \mu''(0) = \begin{bmatrix} \mu(0) & 0 \\ 0 & \mu'(0) \end{bmatrix}; \quad \mu''(1) = \begin{bmatrix} \mu(1) & 0 \\ 1 & \mu'(1) \end{bmatrix}; \quad w'' = \begin{bmatrix} w \\ w' \end{bmatrix}$$

$$v'' = \begin{bmatrix} 2v & -v' \end{bmatrix}; \quad \mu''(0) = \begin{bmatrix} \mu(0) & 0 \\ 0 & \mu'(0) \end{bmatrix}; \quad \mu''(1) = \begin{bmatrix} \mu(1) & 0 \\ 1 & \mu'(1) \end{bmatrix}; \quad w'' = \begin{bmatrix} w \\ w' \end{bmatrix}$$

A po minimalizacji...

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}; \quad \gamma(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & -1 & 0 \end{bmatrix}; \quad \gamma(1) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad z = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 0 \end{bmatrix}$$

Koniec dowodu (semigroup trick)

Wykonując BFSa (przeszukiwanie wszerz) sprawdzamy, że zbiór wszystkich produktów $y\gamma(x), x \in 0, 1^*$ jest skończony.

Koniec dowodu (semigroup trick)

Wykonując BFSa (przeszukiwanie wszerz) sprawdzamy, że zbiór wszystkich produktów $y\gamma(x), x \in 0, 1^*$ jest skończony.

Powstała semigrupa S zawiera jedynie 15 elementów, można je wszystkie sprawdzić.

Koniec dowodu (semigroup trick)

Wykonując BFSa (przeszukiwanie wszerz) sprawdzamy, że zbiór wszystkich produktów $y\gamma(x), x \in 0, 1^*$ jest skończony.

Powstała semigrupa S zawiera jedynie 15 elementów, można je wszystkie sprawdzić. Elementami S są wektory. Łatwo sprawdzić, że iloczyn skalarny każdego z nich z z daje tylko -**1**, **0** albo **1**. ∎

Koniec dowodu (semigroup trick)

Wykonując BFSa (przeszukiwanie wszerz) sprawdzamy, że zbiór wszystkich produktów $y\gamma(x), x \in 0, 1^*$ jest skończony.

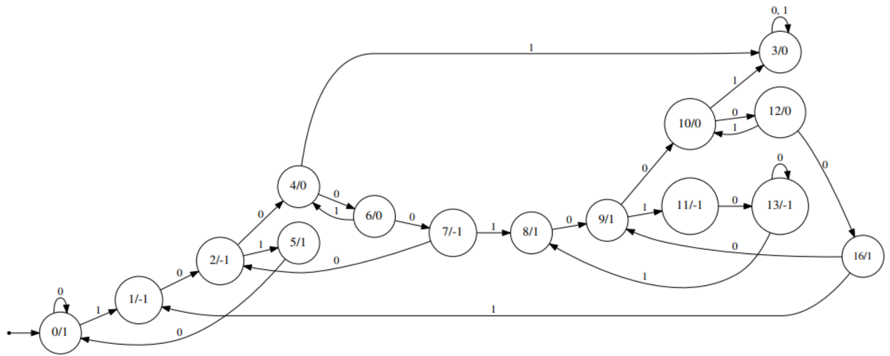
Powstała semigrupa S zawiera jedynie 15 elementów, można je wszystkie sprawdzić. Elementami S są wektory. Łatwo sprawdzić, że iloczyn skalarny każdego z nich z z daje tylko **-1**, **0** albo **1**. ■

Obliczanie $a(n)$

Reprezentacja liniowa (y, γ, z) jest prostym algorytmem obliczania $a(n)$.

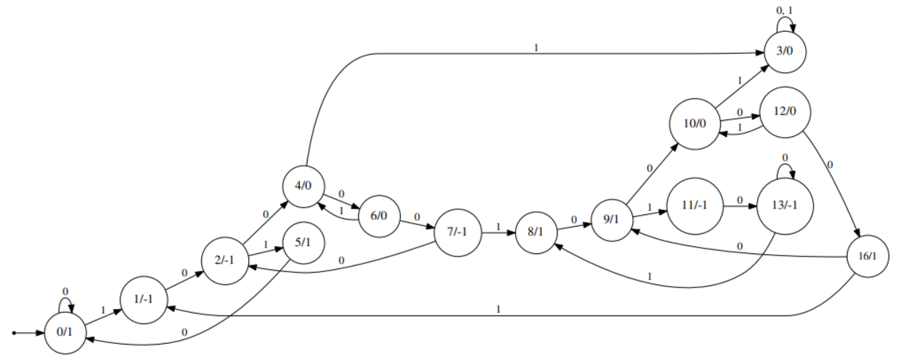
Obliczanie $a(n)$

Automat



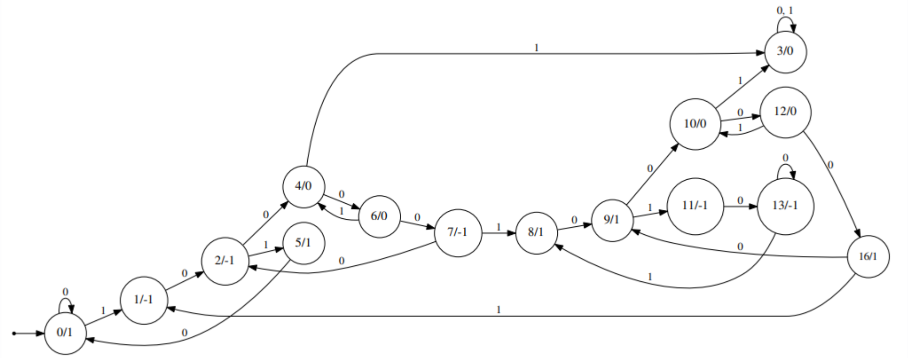
Obliczanie $a(n)$

Automat



Ciekawostka

Dla "prawie" wszystkich n mamy $a(n) = 0$.



Ciekawostka

Dla "prawie" wszystkich n mamy $a(n) = 0$.

Zauważmy, że 01001001 jest słowem synchronizującym.

Źródła

1. J. Shallit - Robbins and Ardila meet Berstel
2. F. M. Ardila - The coefficients of a Fibonacci power series
3. J. Berstel - An exercise on Fibonacci representations
4. N. Robbins - Fibonacci partitions
5. P. Bernacki - Zastosowanie automatów skończonych w przetwarzaniu archiwalnych tekstów języka polskiego

Koniec