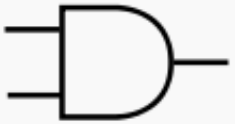
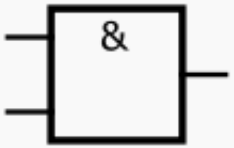
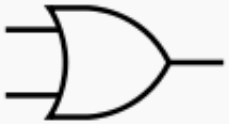
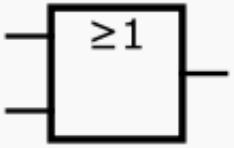
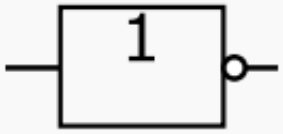
Obliczenia Odwracalne enlacarwdO ainezcilbO

Grzegorz Jakacki, Codility
Marek Rusinowski, Codility

http://www.quietpc.com/gb-en-gbp/products/northbridgecooling/nc-u6

http://www.abchw.com/content/stock-sapphire-cooler

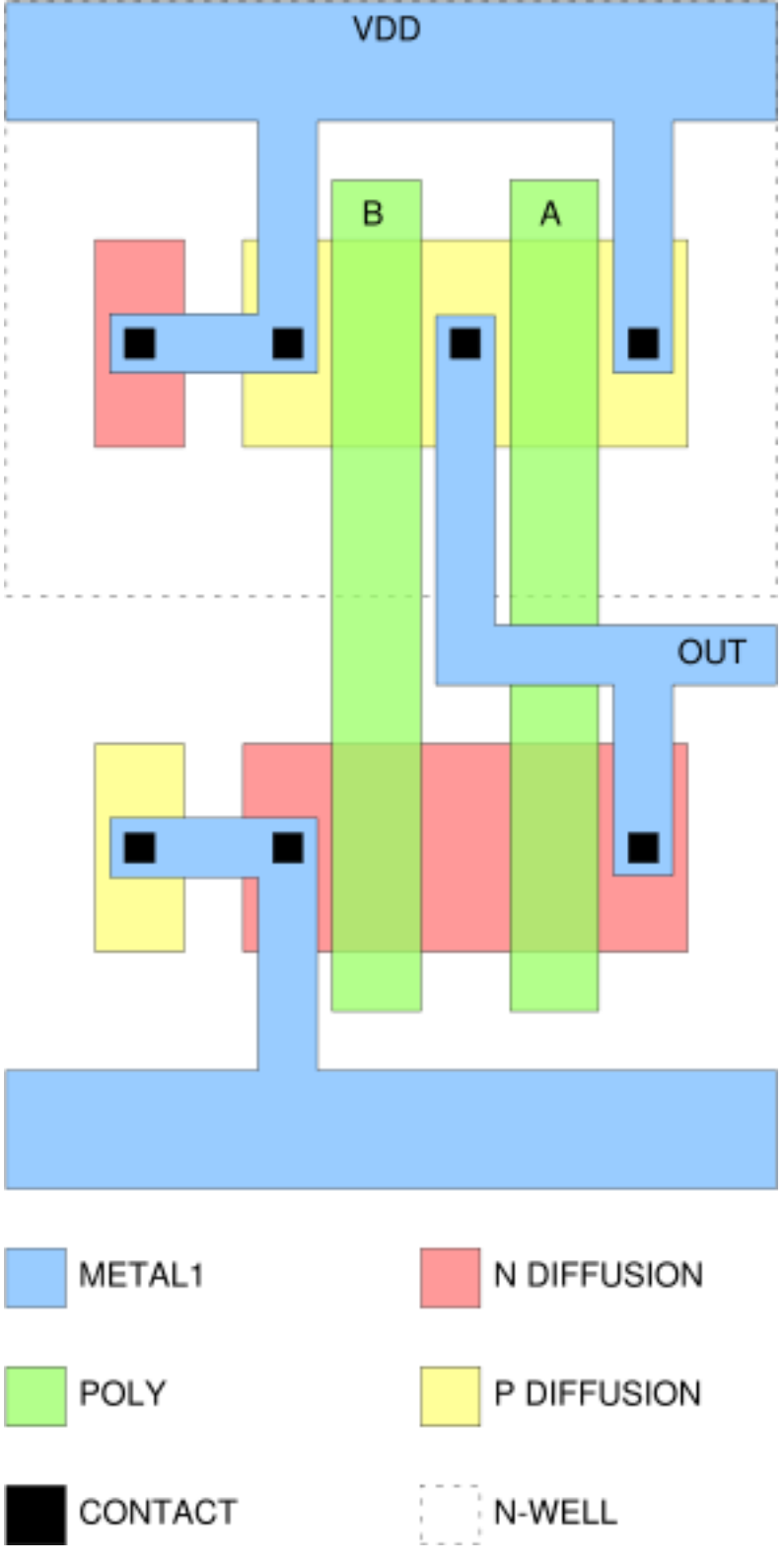
http://www.tech-hounds.com/review10/ReviewsPage1.html

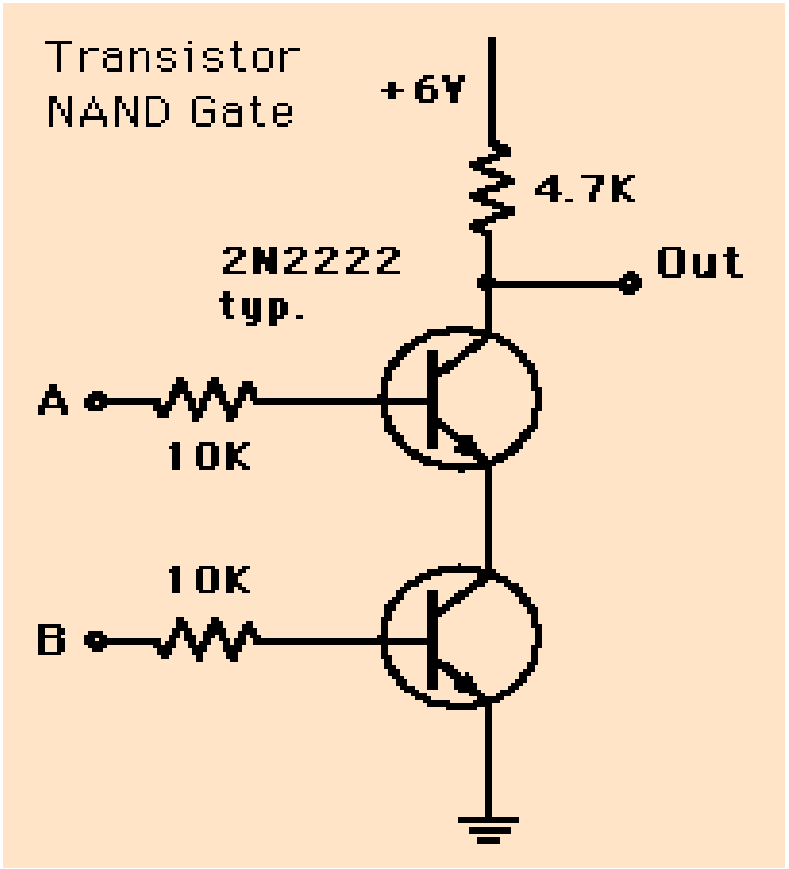
Type	Distinctive shape	Rectangular shape	Boolean algebra between A & B	Truth table
AND			$A \cdot B$	<table><tr><th colspan="2">INPUT</th><th>OUTPUT</th></tr><tr><td>A</td><td>B</td><td>A AND B</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>
OR			$A + B$	<table><tr><th colspan="2">INPUT</th><th>OUTPUT</th></tr><tr><td>A</td><td>B</td><td>A OR B</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>
NOT			$\overline{A}$	<table><tr><th colspan="2">INPUT</th><th>OUTPUT</th></tr><tr><td colspan="2">A</td><td>NOT A</td></tr><tr><td colspan="2">0</td><td>1</td></tr><tr><td colspan="2">1</td><td>0</td></tr></table>

- zliczanie parzystości
- dodawanie
- spójny podciąg długości 3
- dekodowanie/kodowanie
- kontur 2D
- plamy 2D

Uniwersalny zestaw bramek

NAND


VDD
B
A
OUT
METAL1
N DIFFUSION
POLY
P DIFFUSION
CONTACT
N-WELL


Transistor
NAND Gate
+6V
4.7K
2N2222
typ.
Out
A
10K
10K
B

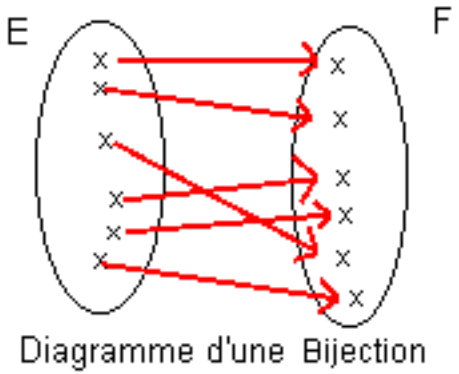
Bramki o większej arności

Sumator

Przerzutnik RS

Układy kombinatoryczne i sekwencyjne

Bijekcja



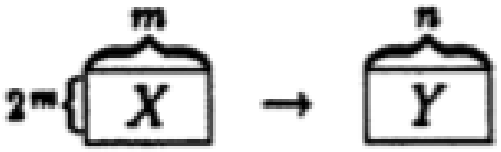
http://www.maths-express.com/bac-exo/bac-s/cour-s/cours-proba/bijection1.gif

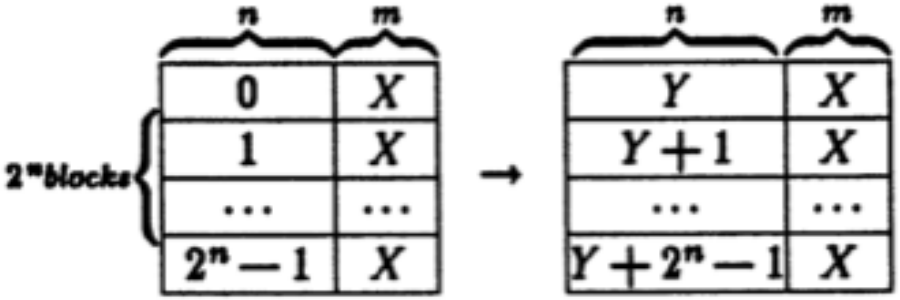
Funkcja obliczana przez kombinatoryczny układ odwracalny musi być bijekcją.

Każdą skończoną funkcję można przerobić na bijekcję rozszerzając dziedzinę i zbiór wartości

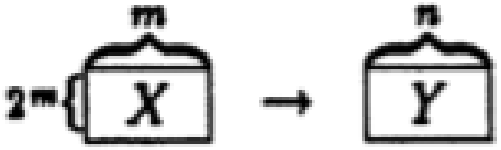
THEOREM 4.1 *For every finite function* $\phi: \mathbf{B}^m \to \mathbf{B}^n$ *there exists an invertible finite function* $f: \mathbf{B}^r \times \mathbf{B}^m \to \mathbf{B}^n \times \mathbf{B}^{r+m-n}$, *with* $r \leq n$, *such that*

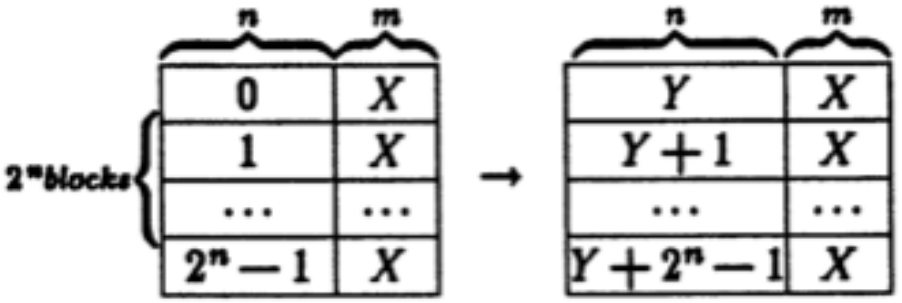
$$f\langle \overbrace{0, \ldots, 0}^{r}, x_1, \ldots, x_m \rangle = \phi_i \langle x_1, \ldots, x_m \rangle, \qquad (i = 1, \ldots, n). \qquad (4.1)$$





revcomp-rep





AND/OR

00 00
01 01
10 01
11 11

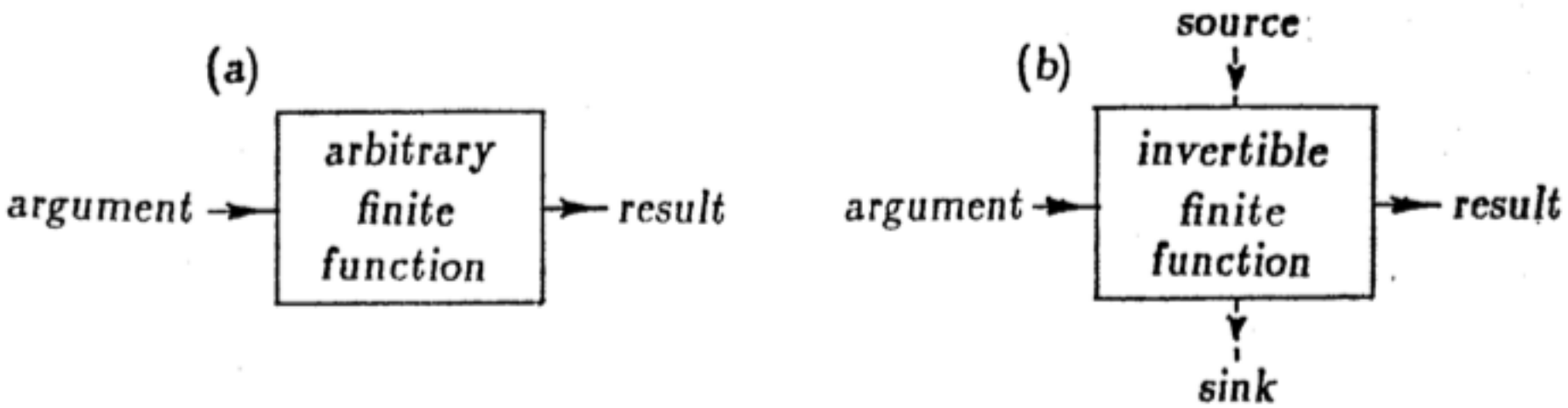
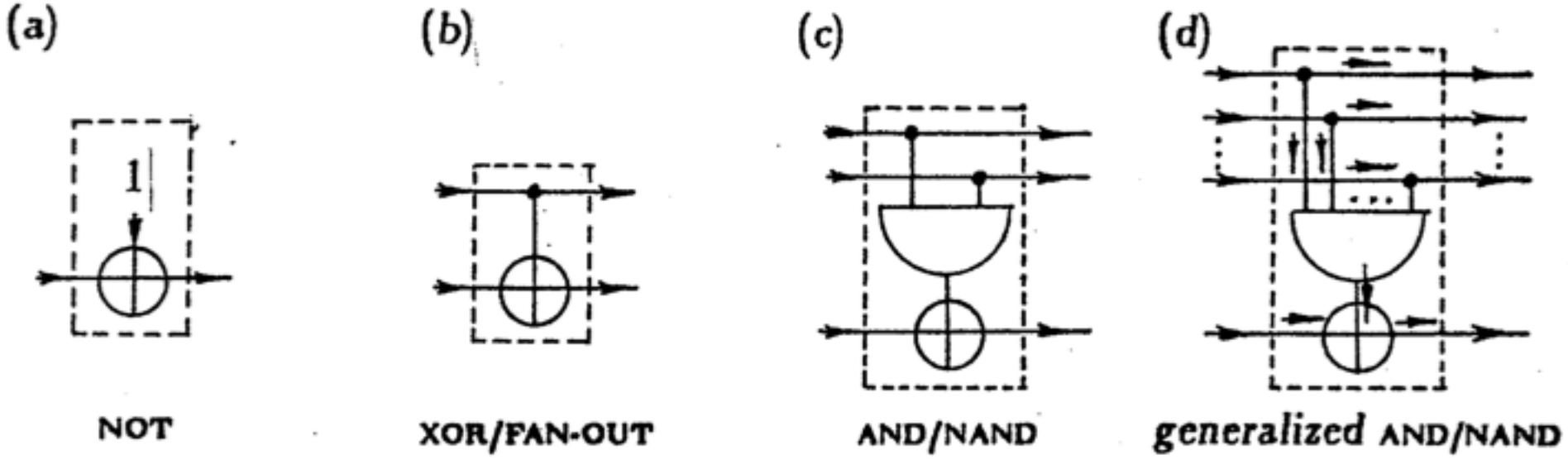


FIG. 4.2 *Any finite function (a) can be realized as an invertible finite function (b) having a number of auxiliary input lines which are fed with constants and a number of auxiliary output lines whose values are disregarded.*

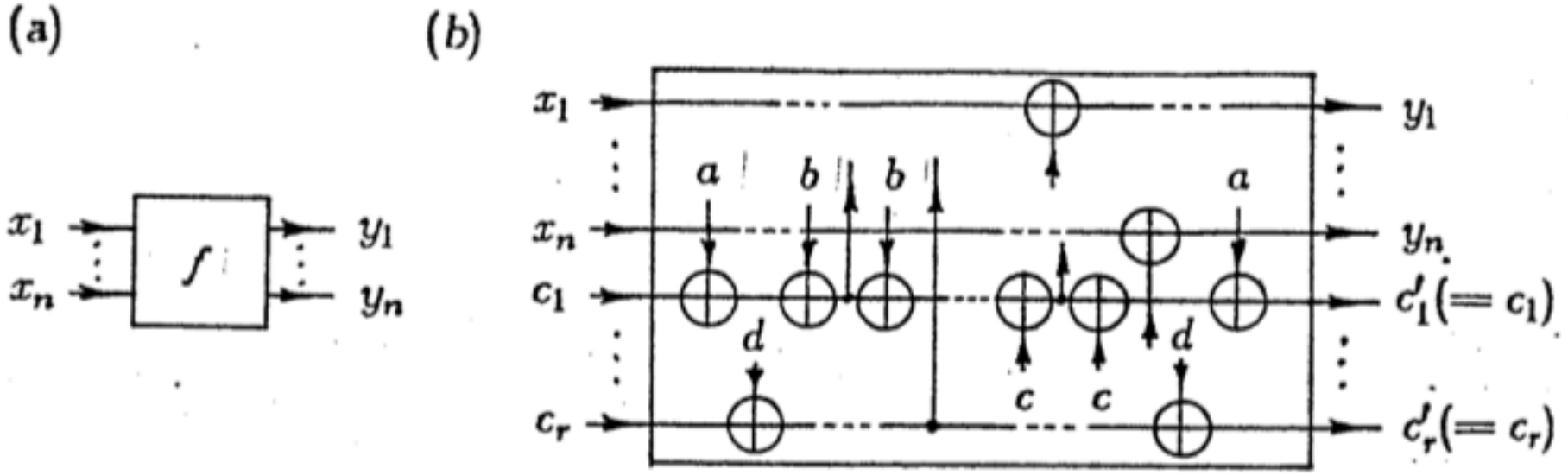
UWAGA: Konstrukcja z dowodu nie jest optymalna, czasami da się uzyskać dużo mniejszy "sink".
Jak? Problem otwarty.

Czy da się uzyskać dowolną funkcję odwracalną z małego zbioru bramek w ten sposób, żeby z konstrukcji “sink” nie występował?



Nie istnieje skończony uniwersalny zbiór bramek dla logiki odrwacalnej dający układy bez wychodzących drutów śmieciowych

Realizacja z dodatkowymi drutami stałymi na wejściu i wyjściu ("pamięć dodatkowa")



Ile potrzeba takich drutów?

ile potrzeba dodatkowej pamięci?

Dla każdej funkcji obliczalnej
istnieje maszyna Turinga, która ją oblicza.

Dla każdej funkcji obliczalnej i odwracalnej
istnieje odwracalna maszyna Turinga,
która ją oblicza. Ale taka maszyna może
potrzebować więcej pamięci i czasu.

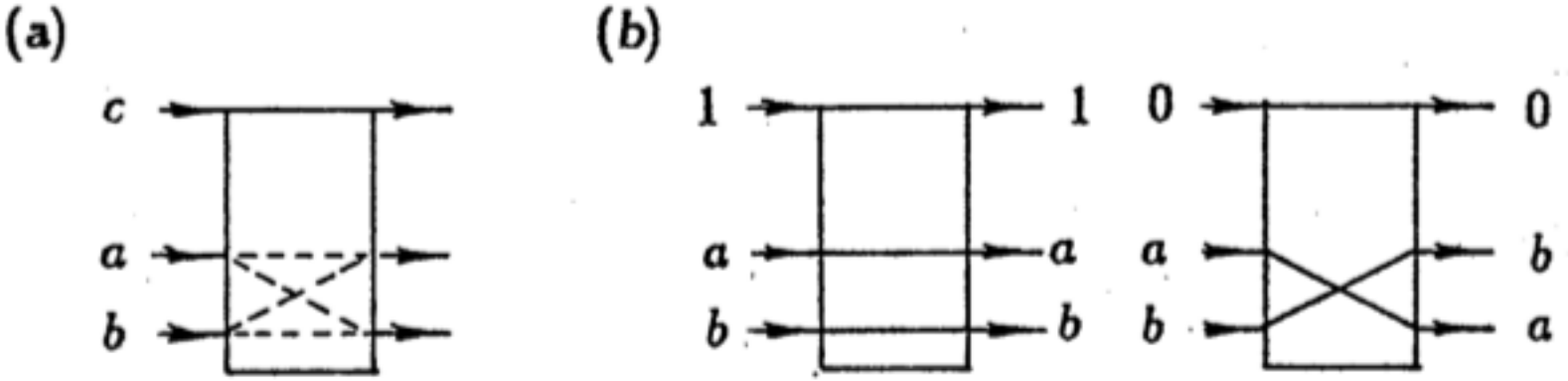
Ile płacimy za odwracalność?

(Bennett 1988)

Dla dowolnego e>0:
Maszyna działająca w czasie T i pamięci M
może być symulowana na maszynie odwracalnej

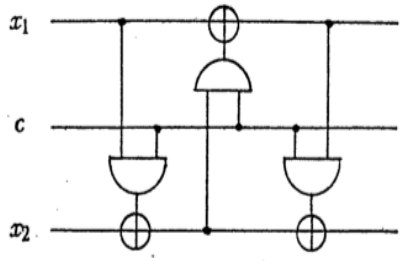
$T' = O(T^{1+e})$ $M' = O(M * \log T)$

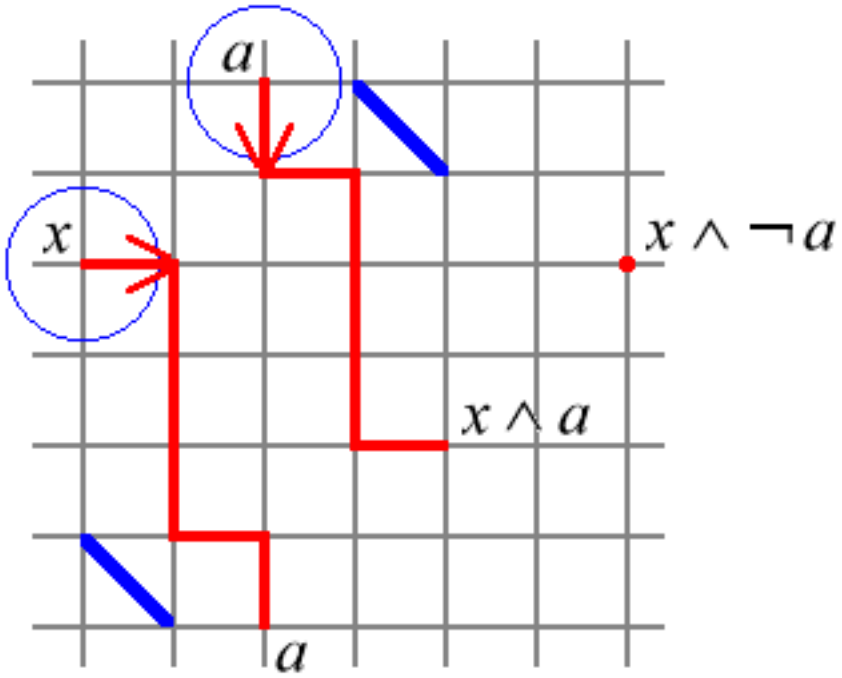
Zachowanie liczby jedynek

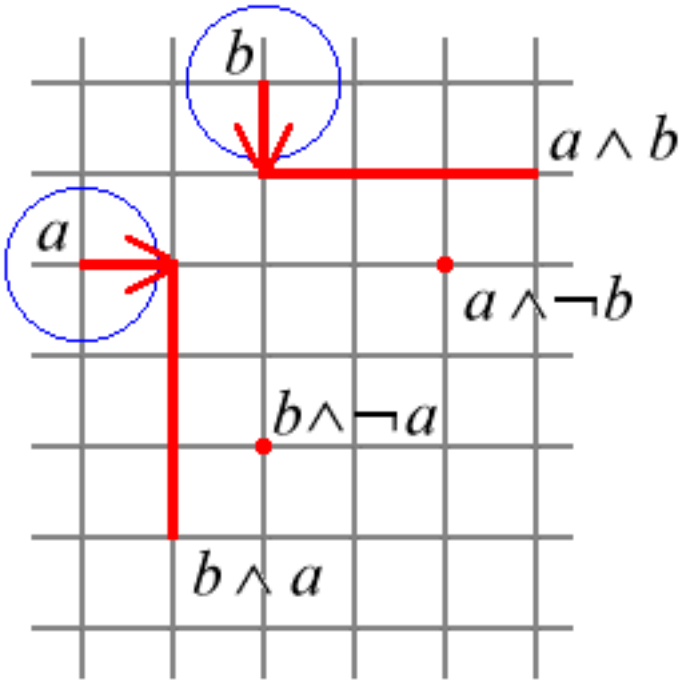


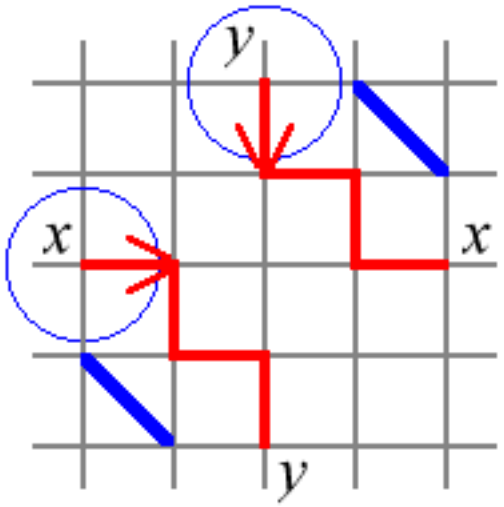
c	x_1	x_2		c'	y_1	y_2
0	0	0		0	0	0
0	0	1		0	1	0
0	1	0		0	0	1
0	1	1	→	0	1	1
1	0	0		1	0	0
1	0	1		1	0	1
1	1	0		1	1	0
1	1	1		1	1	1

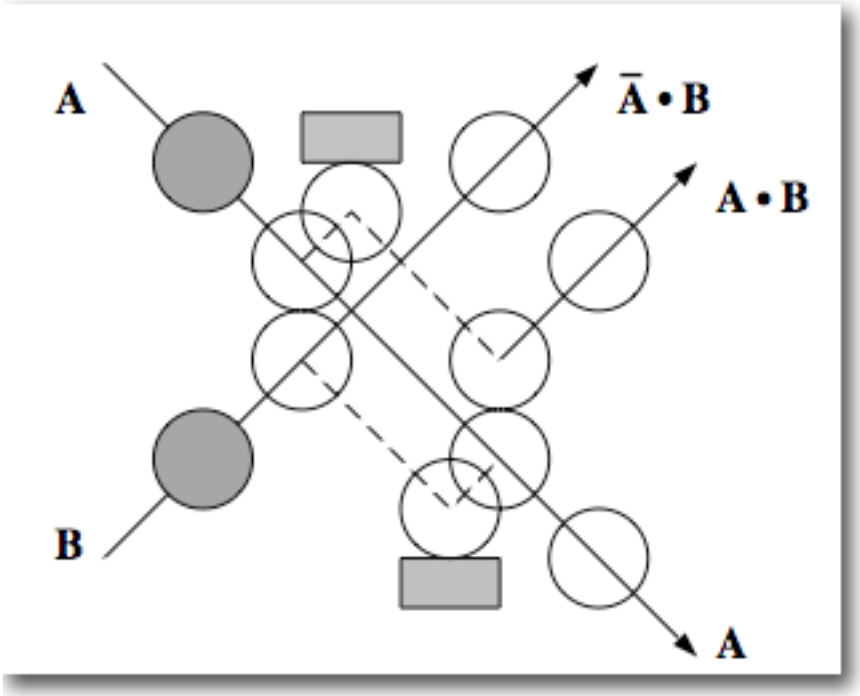
.



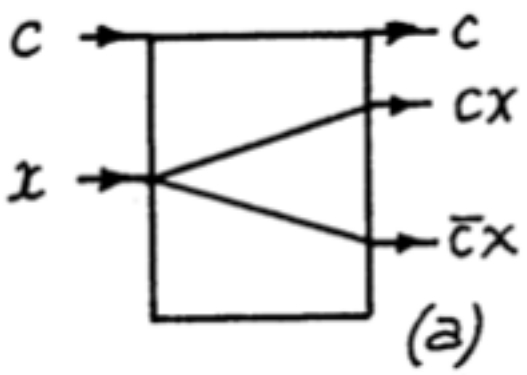


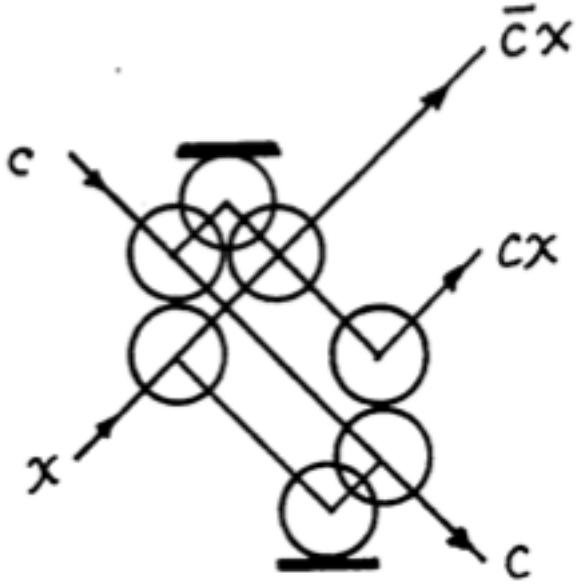


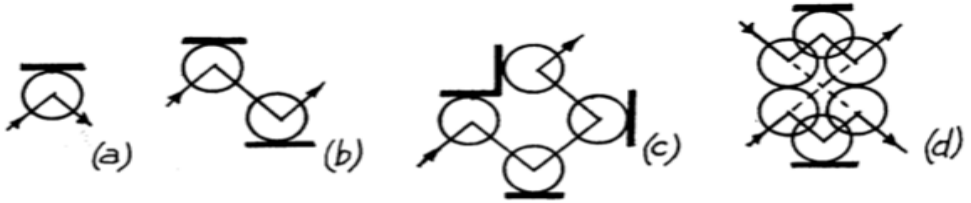


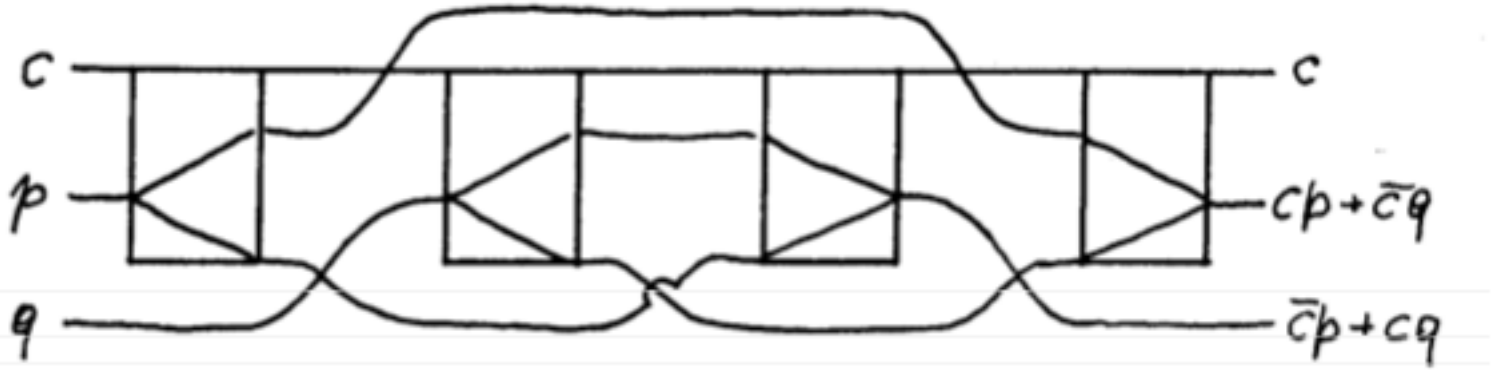


http://jameslin.name/bball/


c
x
c
cx
c̄x
(a)


c̄x
c
cx
x
c


(a)
(b)
(c)
(d)


c
p
q
c
cp+c̄q
c̄p+cq