Przykład

$A=B \iff \bigwedge_{x\in X} x\in A \iff x\in B$

Wykazać, że dla dowolnych zbiorów A, B i C zachodzi równość

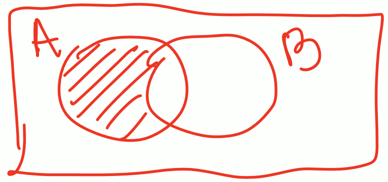
$$(A\cup B)\setminus C = (A\setminus C)\cup(B\setminus C).$$

Musimy pokazać, że

$$\bigwedge_{x\in X} x\in(A\cup B)\setminus C \iff x\in(A\setminus C)\cup(B\setminus C)$$

Niech $x\in X$. Wtedy

$$\begin{aligned}
x\in(A\cup B)\setminus C &\iff x\in A\cup B \wedge x\notin C \iff \\
&\iff (\underbrace{x\in A}_{p} \vee \underbrace{x\in B}_{q}) \wedge \underbrace{x\notin C}_{r} \iff \\
&\iff (x\in A \wedge x\notin C) \vee (x\in B \wedge x\notin C) \iff \\
&\iff x\in A\setminus C \vee x\in B\setminus C \iff \\
&\iff x\in(A\setminus C)\cup(B\setminus C)
\end{aligned}$$



Przykład

$$\{\; A\setminus B = A\cap B^c$$

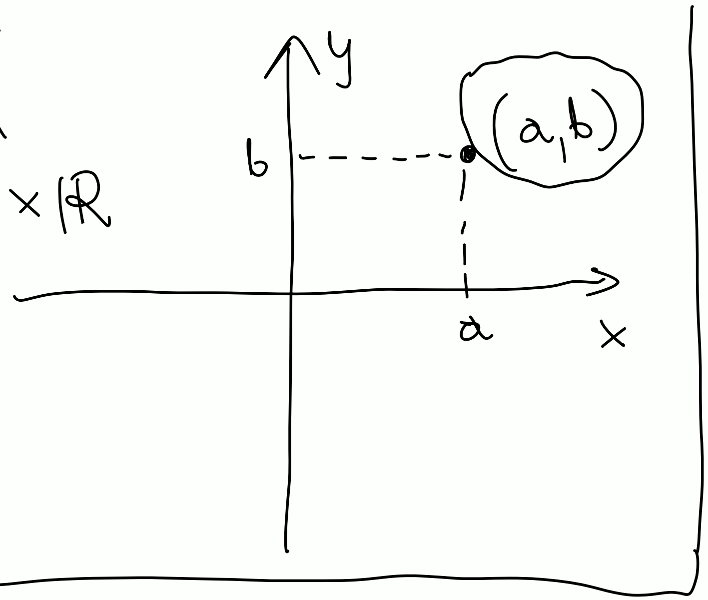
Wykazać, że dla dowolnych zbiorów A, B i C zachodzi równość

$$(A\cup B)\setminus C = (A\setminus C)\cup(B\setminus C).$$

$$(A\cup B)\setminus C = (A\cup B)\cap C^c = (A\cap C^c)\cup(B\cap C^c) =$$
$$= (A\setminus C)\cup(B\setminus C)$$

Iloczyn kartezjański

$\mathbb{R}^2 = \mathbb{R}\times\mathbb{R}$



$A, B \subset X$

$a\in A, \quad b\in B$

$\boxed{(a,b)}$ ← para uporządkowana

pierwszy el. — drugi el.

$(a,b) \overset{def.}{=} \{\{a\},\{a,b\}\}$ (ćw.)

$A\times B = \{(a,b) : a\in A, \ b\in B\}$

iloczyn kartezjański

$A_1\times A_2\times\dots\times A_n = \{\underline{(a_1,a_2,\dots,a_n)} : a_i\in A_i, \ i\in\{1,\dots,n\}\}$

Paradoks kłamcy

Jeśli kłamca mówi, że kłamie, to kłamie, czy mówi prawdę?

Antynomia Russella

$$A = \{ B : \underbrace{B \notin B} \}$$

Antynomia Russella

Niech zbiór A będzie złożony ze wszystkich zbiorów B, które nie są swoimi elementami, to znaczy

$$B \in A \iff B \notin B.$$

$$\boxed{A = \{B : B \notin B\}}$$

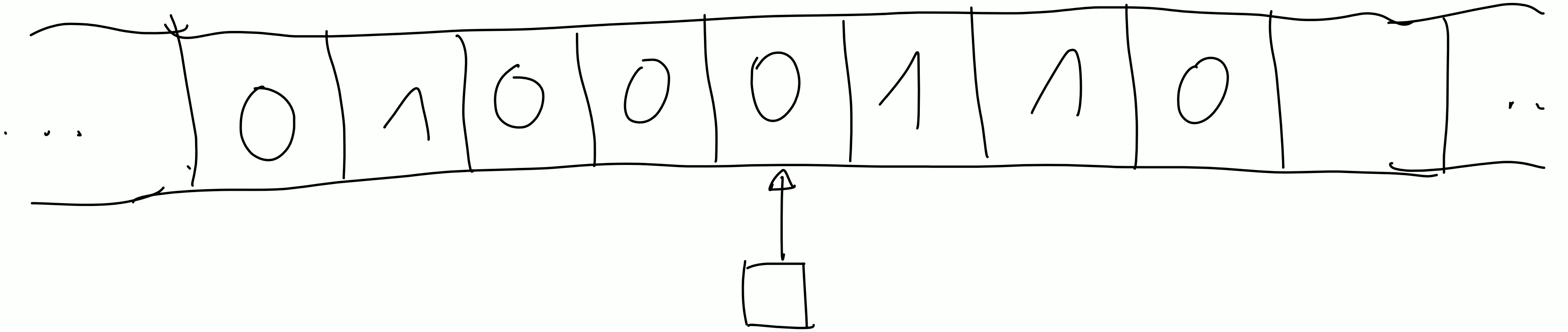
Czy $A \in A$?

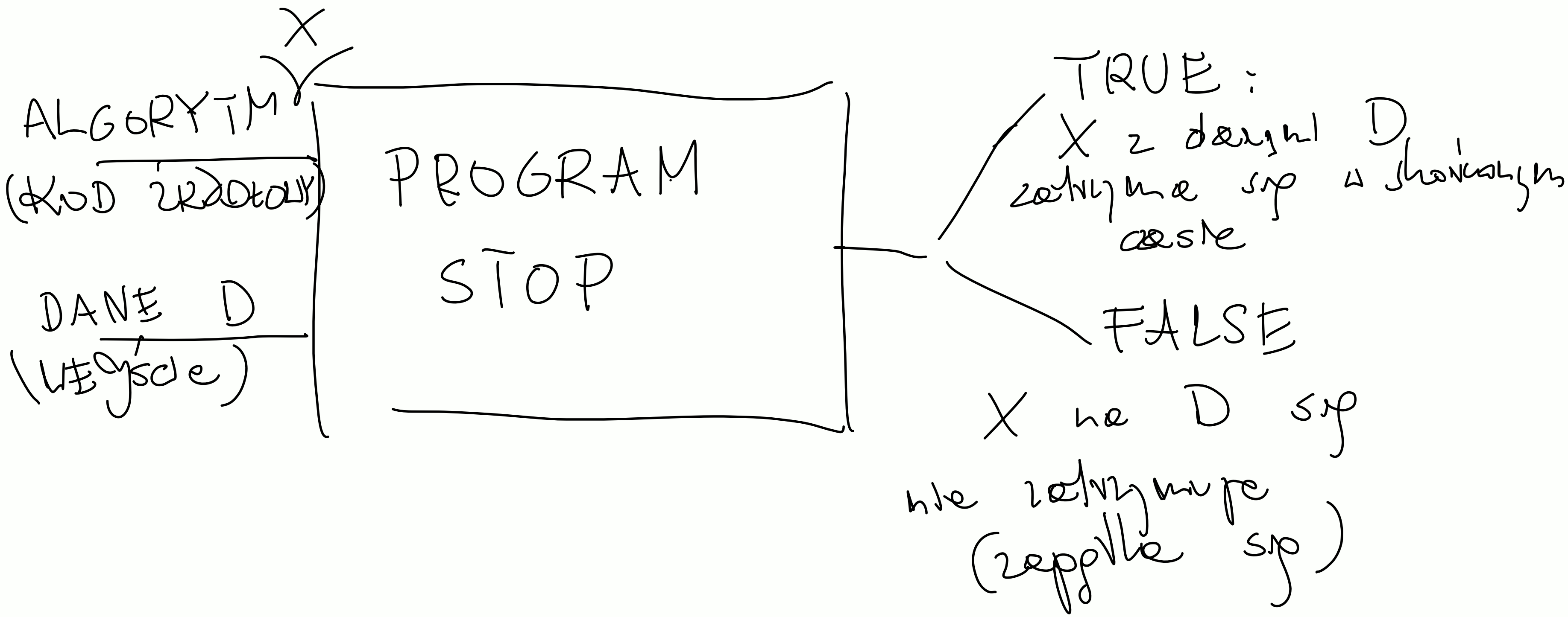
1) $A \in A \Rightarrow A \notin A$ sprzeczność

2) $A \notin A \Rightarrow A \in A$ sprzeczność

$$\boxed{\{x \in X : \phi(x)\}}$$

Maszyna Turinga, problem stopu





Maszyna Turinga, problem stopu

Problem stopu

Napisz program, który dla algorytmu X oraz danych wejściowych D zwraca:

- $\rightsquigarrow$ **true** wtedy i tylko wtedy, gdy X z danymi D zatrzymuje się w skończonym czasie,
- $\rightsquigarrow$ **false** wtedy i tylko wtedy, gdy X z danymi D nie zatrzymuje się w skończonym czasie (zapętla się).

Problem stopu

Tw. (A. Turing, 1936 r.)

Załóżmy, że program STOP(X, D) istnieje.

$\rightarrow T(X)$:

nowy program

algorytm

if STOP(X, X)

while true

Tw. Program STOP(X, D) nie istnieje!

T(T)? Czy T(T) się zatrzymuje?

1) T(T) zatrzymuje się w skończonym czasie.

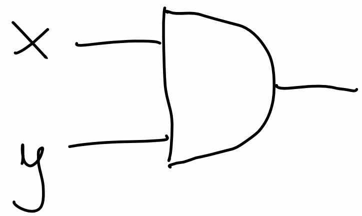
$\Rightarrow$ STOP(T, T) jest fałszem $\Rightarrow$ T(T) się zapętla

2) T(T) się zapętla $\Rightarrow$ STOP(T, T) jest prawdą

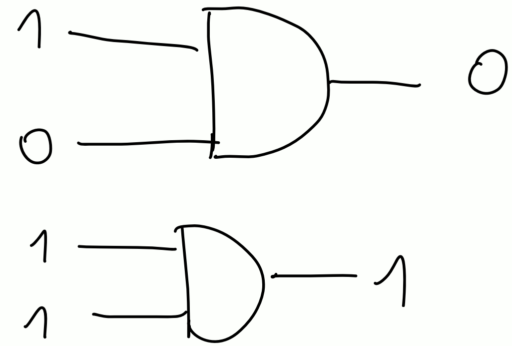
$\Rightarrow$ T(T) się zatrzymuje

Bramki logiczne

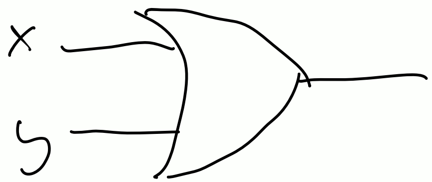
$\wedge$



$x \wedge y$

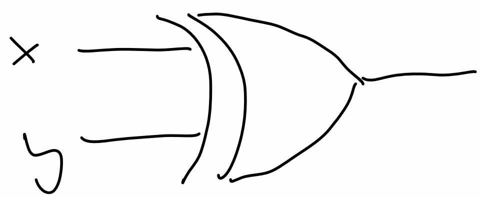


$\vee$



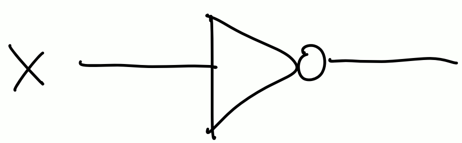
$x \vee y$

$\oplus$



$x \oplus y$

$\neg$



$\neg X$

$$
\begin{array}{cccccc}
 & 1 & 1 & 1 & & \\
 & 1 & 0 & 1 & 1 & 0 \\
+ & 0 & 1 & 1 & 1 & 1 \\
\hline
1 & 0 & 0 & 1 & 0 & 1
\end{array}
$$

x	0	1	0	1	
y	0	0	1	1	
$x \oplus y$	0	1	1	0	(1)

$x \oplus y$ $\qquad$ $x \wedge y$

Sumatory