

28.08.2022

# PROBLEM DECYZYJNY

- $\Sigma$  - skończony alfabet
- $\Sigma^*$  - zbiór skończonych słów nad alfabetem  $\Sigma$
- $\Sigma^* \supseteq L$  - język / problem
- Pytamy o "zasoby obliczeniowe" potrzebne do rozstrzygnięcia tych problemów.

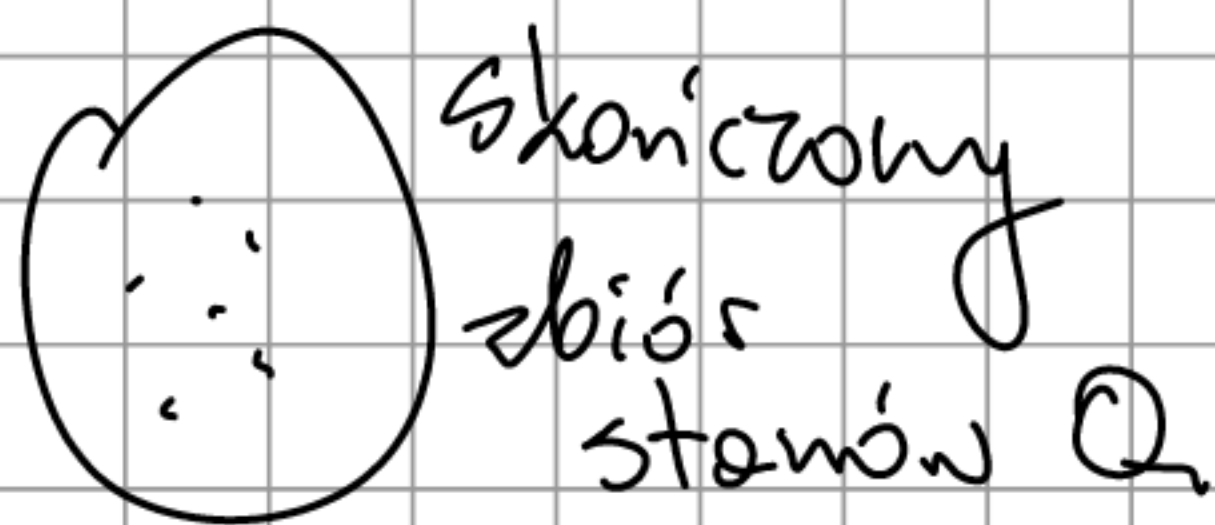
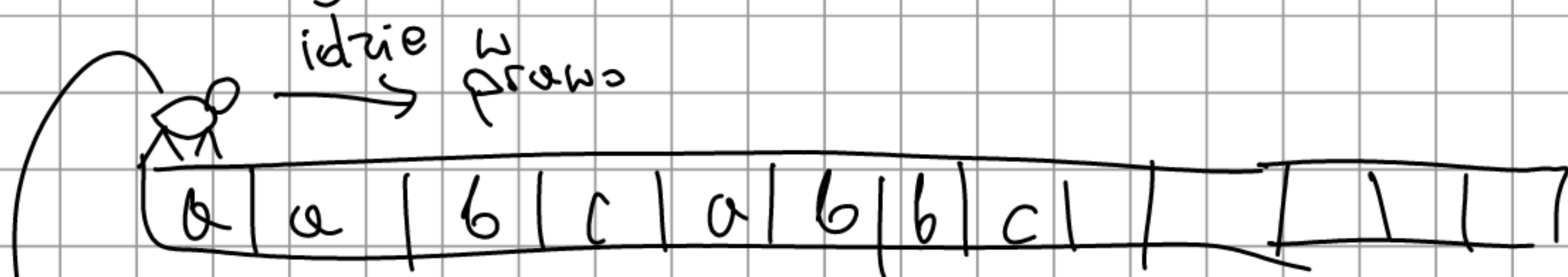


- klasyfikujemy problemy ze względu na te zasoby

Blę bla...

# CZĘŚĆ I

## Automaty skończone



skończony  
zbiór  
stanów  $Q$

Funkcja przejścia  
 $\delta: Q \times \Sigma \rightarrow Q$

• Stan początkowy  
 $q_0 \in Q$

• zbiór stanów akceptujących  $F \subseteq Q$

Ćw. Skonstruuj  $\delta, Q$  dla  $\Sigma = \{0,1\}$ ,  
 $L = \{w \in \{0,1\}^* : |w|_1 \text{ jest parzyste}\}$

Ale dla  $L = \{w \in \{0,1\}^* : |w|_1 = |w|_0\}$   
się nie da!

D-d. (A.a.) Niech Zenon będzie zuchwiałym

rozstrzygnięciem  $L$ . Zet. ie  $|Q| = k$ .

$$w_0 = \epsilon$$

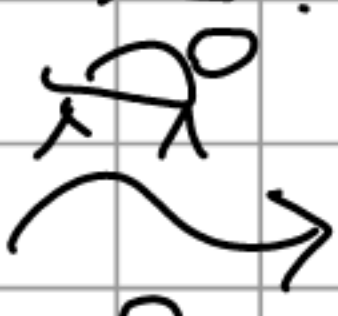
$$w_1 = 0$$

$$\vdots$$

$$w_i = 0^i$$

$$\vdots$$

$$w_k = 0^k$$



$$s_0 \in Q$$

$$s_1 \in Q$$

$$s_i \in Q$$

$$s_k \in Q$$

Jest  $i, j$  t. że

$$s_i = s_j$$

Spójrzmy na

$$a = w_i \perp^i \rightsquigarrow s \in A$$

$$b = w_j \perp^i \rightsquigarrow s \notin A$$

Deterministyczny automat skończony (DFA)  
to krotka  $(\Sigma, Q, q_0, \delta, F)$ .





oraz  $y \neq \epsilon$ ,  $|xy| \leq n$  takie że dla  
każdego  $k \in \mathbb{N}$   $xy^k z \in L$ .

Przykład Weźmy  $L = \{ w \in \{0,1\}^* : |w|_0 = |w|_1 \}$ .

Zał. że  $L$  regularny. Weźmy  $n$  jak z  
lematu. Niech  $w = 0^n 1^n \in L$ . Weźmy  $x, y, z$   
jak w lemacie. Ale  $|xy| \leq n$ .

więc  $|y|_1 = 0$ . Dla  $k=0$ :  $xz \in L$

(nie ma "1"  
w  $y$ ) ale  $|xz|_0 < |xz|_1$   $\downarrow$

Dowód lematu


Weźmy  $L = L_A$  regularny ( $A = \langle \Sigma, Q, q_0, \delta, F \rangle$ ).

Niech  $n = |Q| + 1$ . Weźmy  $w \in L$  t.ż.  $|w| \geq n$ .

Niech  $w = a_1 a_2 \dots a_l$ , niech  $s_i = \hat{\delta}(q_0, a_1 \dots a_i)$ .

Wtedy  $s_i = s_j$  dla pewnych  $i < j \leq n$ .

Niech  $x = a_1 \dots a_i$ ,  $y = a_{i+1} \dots a_j$ ,

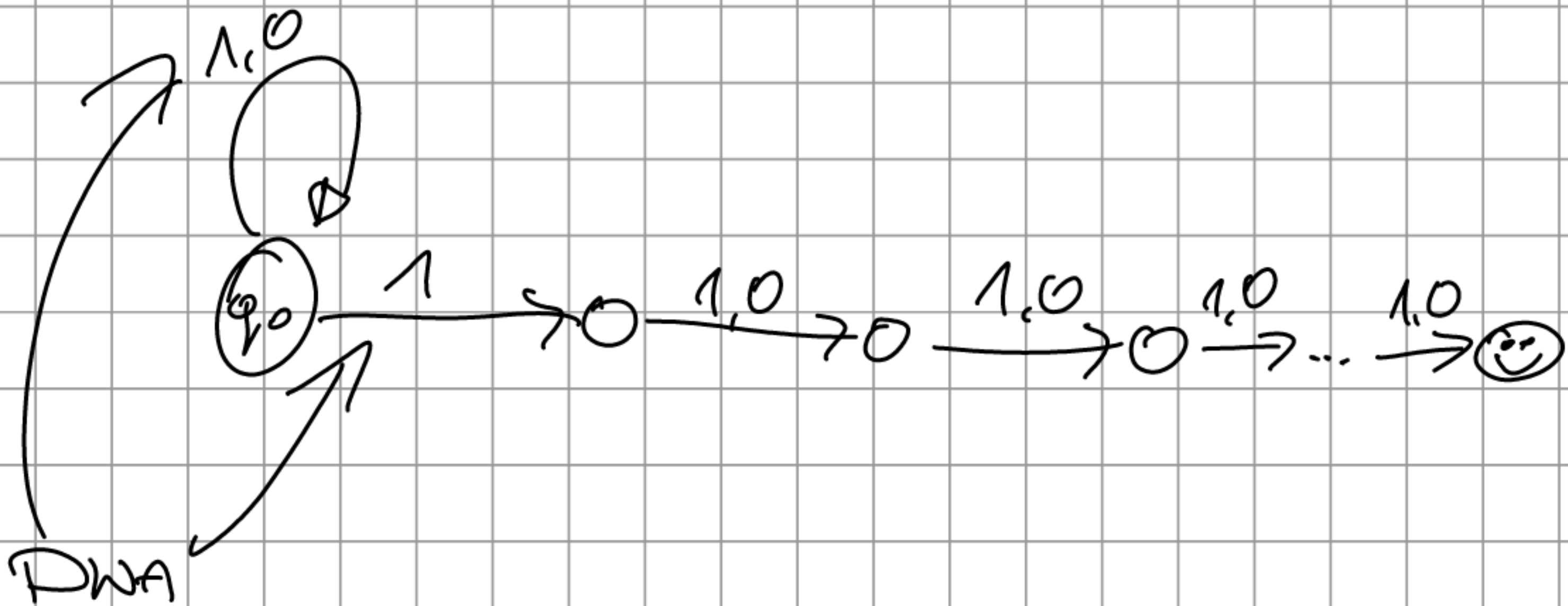
$z = a_{j+1} \dots a_l$ . Wtedy dla  $k \in \mathbb{N} \dots$  

1.03.2022

# NIEDETERMINISTYCZNE AUTOMATY SKOŃCZONE (NFA)

Projekt

$$L = \{ w \mid w \in \{0,1\}^* : |w| \geq 9 \}$$



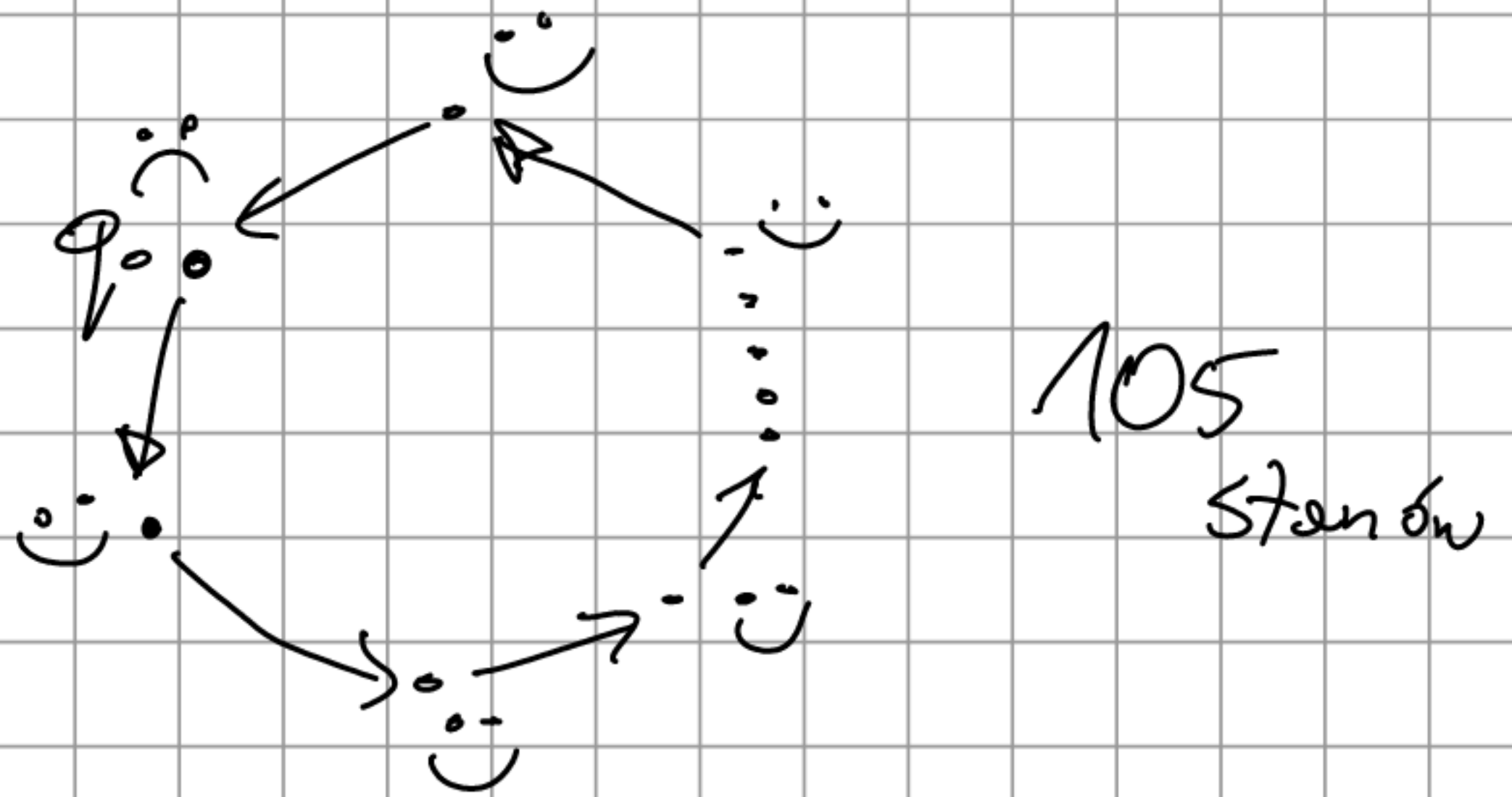
DWA PRZEJŚCIA z 1 → ZUCZEK PYTA NIEBIOS "CO ROBIĆ? JAK ŻYĆ"

A NIEBIOSA ODPOWIADAJĄ

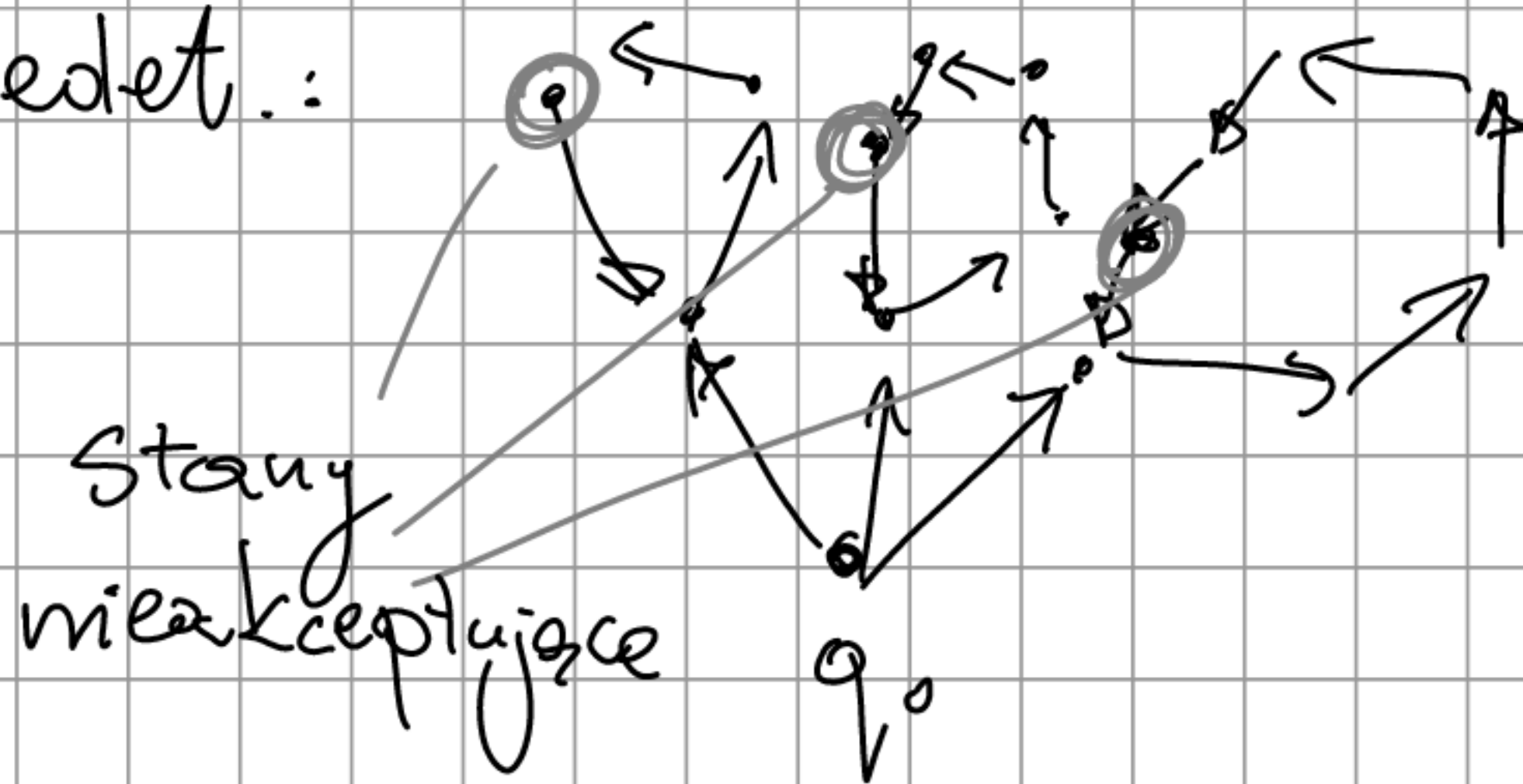
Taki automat daje gwarancję, że na pewno nie trafimy w stan akceptujący, jeśli słowo nie jest z języka (nie ma "false-positive" są "false-negative")

Przykład 2  $L = \{0^i : 105 \mid i\}$

Det. aut.:



Niedet.:



Znaczenie: na pewno jeśli  $105 \mid i$ ,  
to trafimy w stan nieakcept.

Def. Niedeterministyczny automat skończony

to krotka  $\langle \Sigma, Q, q_0, \delta, F \rangle$  jak w

DFA poza  $\delta$ , gdzie

$$\delta \subseteq Q \times \Sigma \times Q.$$

$\delta(q, a, q')$  oznacza  $q$  do  $q'$   
 jest stanem z etykietą  $a$ .

Teraz  $\hat{\delta} \subseteq Q \times \Sigma^* \times Q$ :

$$\left\{ \begin{array}{l} \hat{\delta}(q, \varepsilon, q') \Leftrightarrow q = q' \\ \hat{\delta}(q, wa, q') \Leftrightarrow \exists p \in Q \hat{\delta}(q, w, p) \wedge \delta(p, a, q') \end{array} \right.$$

Alternatywna wersja wg JMa.

Dla każdego  $w \in \Sigma^*$  definiujemy

$$\delta_w \subseteq Q \times Q:$$

$$\left. \begin{array}{l} \delta_\varepsilon = \text{id}_Q \\ \text{jeśli } a \in \Sigma \text{ to } \delta_a(q, q') \Leftrightarrow \delta(q, a, q') \\ \delta_{wa} = \delta_w \circ \delta_a \end{array} \right\}$$



Wtedy  $\hat{\delta}(q, w, q') \Leftrightarrow \delta_w(q, q')$

Def.  $A$ : NFA. Wtedy  
 $L_A = \{ w \in \Sigma^* : \exists q \in F \cup \hat{\delta}(q_0, w, q) \}$

Tw. Niech  $A$ : NFA. Wtedy  $\exists A'$ : DFA  
t.że  $L_A = L_{A'}$ .

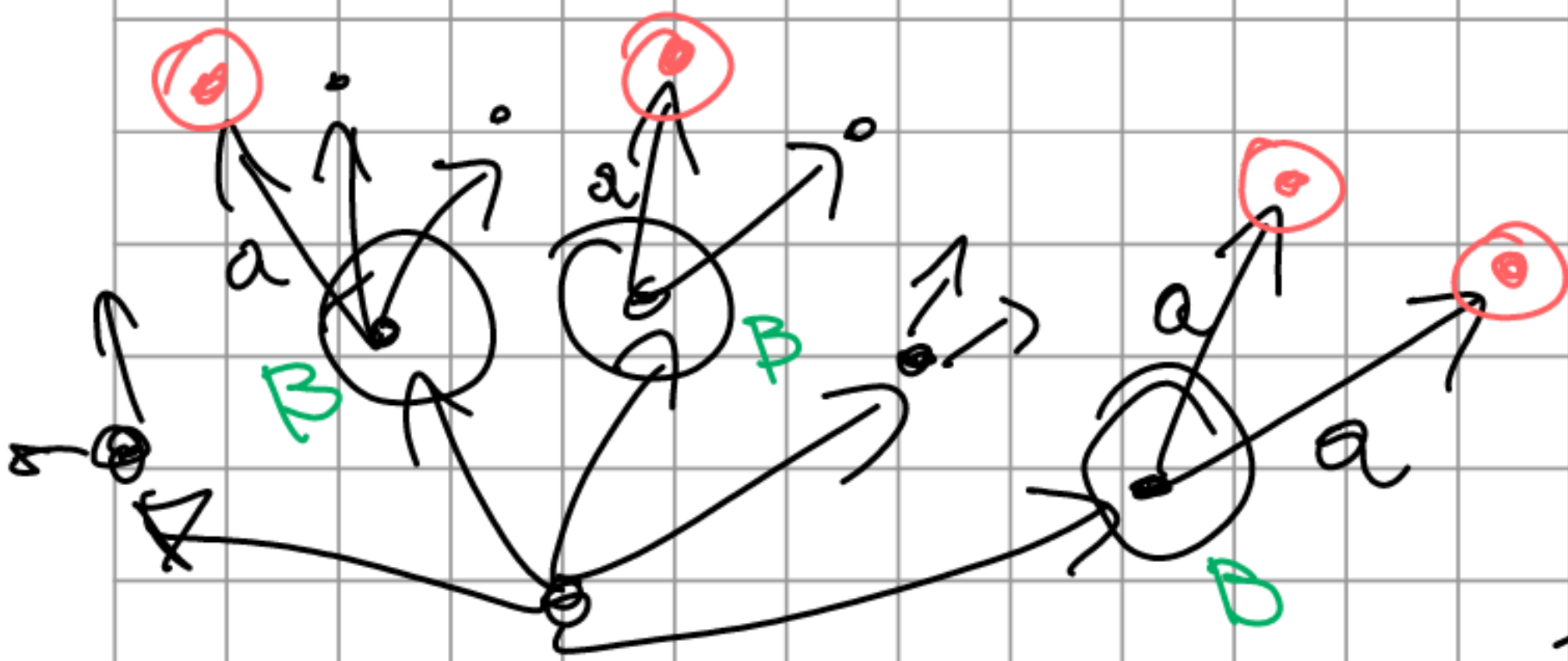
D-d. Weźmy dowolne NFA  $A = \langle \Sigma, Q, q_0, \delta, F \rangle$ .

Zbudujemy  $A' = \langle \Sigma, Q', q'_0, \delta', F' \rangle$ .

Niech  $Q' = \mathcal{P}(Q)$ ,  $q'_0 = \{ q_0 \}$ ,

$F' = \{ B \subseteq Q : B \cap F \neq \emptyset \}$ ,

$\delta'(B, a) = \{ q \in Q : \exists p \in B \delta(p, a, q) \}$ .



Te stany,  
do których  
można dojść  
z któregoś  
stanu  $\in B$  po kraw.  
z etykiety  $a$ .



## NFA Z $\epsilon$ -PRZEJŚCIAMI

Takie coś, że możemy czasem sobie

przejsć ze stanu do stanu bez

wczytania znaków. Jeździ też można

zdeteminować.

7.03.2022

## WYRAŻENIA REGULARNE (nad $\Sigma$ )

•  $\emptyset$  jest wyrażeniem regularnym i  $L_{\emptyset} = \emptyset$

•  $\varepsilon$  jest wyr. reg. i  $L_{\varepsilon} = \{\varepsilon\}$

• jeśli  $a \in \Sigma$  to  $a$  jest wyr. reg.

oraz  $L_a = \{a\}$

• jeśli  $\varphi, \psi$  są wyr. reg. to  $\varphi + \psi$

jest wyrażeniem regularnym i  $L_{\varphi + \psi} = L_{\varphi} \cup L_{\psi}$

$\varphi\psi$  jest wyrażeniem regularnym i  $L_{\varphi\psi} = L_{\varphi}L_{\psi}$ .

$= \{w_1w_2 : w_1 \in L_{\varphi}, w_2 \in L_{\psi}\}$ .

$\lceil L_1L_2 = \{w_1w_2 : w_1 \in L_1, w_2 \in L_2\} \rceil$

$L \in \Sigma^*$ , wtedy  $L^0 = L_{\varepsilon}$ ,  $L^1 = L$ ,  $L^{i+1} = L^iL$

$\lfloor L^* = \bigcup_{n=1}^{\infty} L^n \rfloor$

• jeśli  $\varphi$  jest wyr. reg. to  $\varphi^*$  też

jest oraz  $L_{\varphi^*} = (L_{\varphi})^*$

Przykład  $\Sigma = \{0,1\}$ ,  $O^*(10^*10^*)^*$

Tw. Niech  $L \subseteq \Sigma^*$ . Wtedy NWSR:

(1)  $L$  jest regularny, tj. istnieje DFA  $A$  t.ze  $L = L_A$ .

(2) Istnieje wyrażenie regularne  $\varphi$  t.ze  
 $L = L_\varphi$ .

(3) Istnieje NFA z  $\epsilon$ -przejściami  $A'$   
t.ze  $L = L_{A'}$ .

D-d. (3)  $\Rightarrow$  (1) Było. ~~III~~

(2)  $\Rightarrow$  (3) Indukcja względem długości

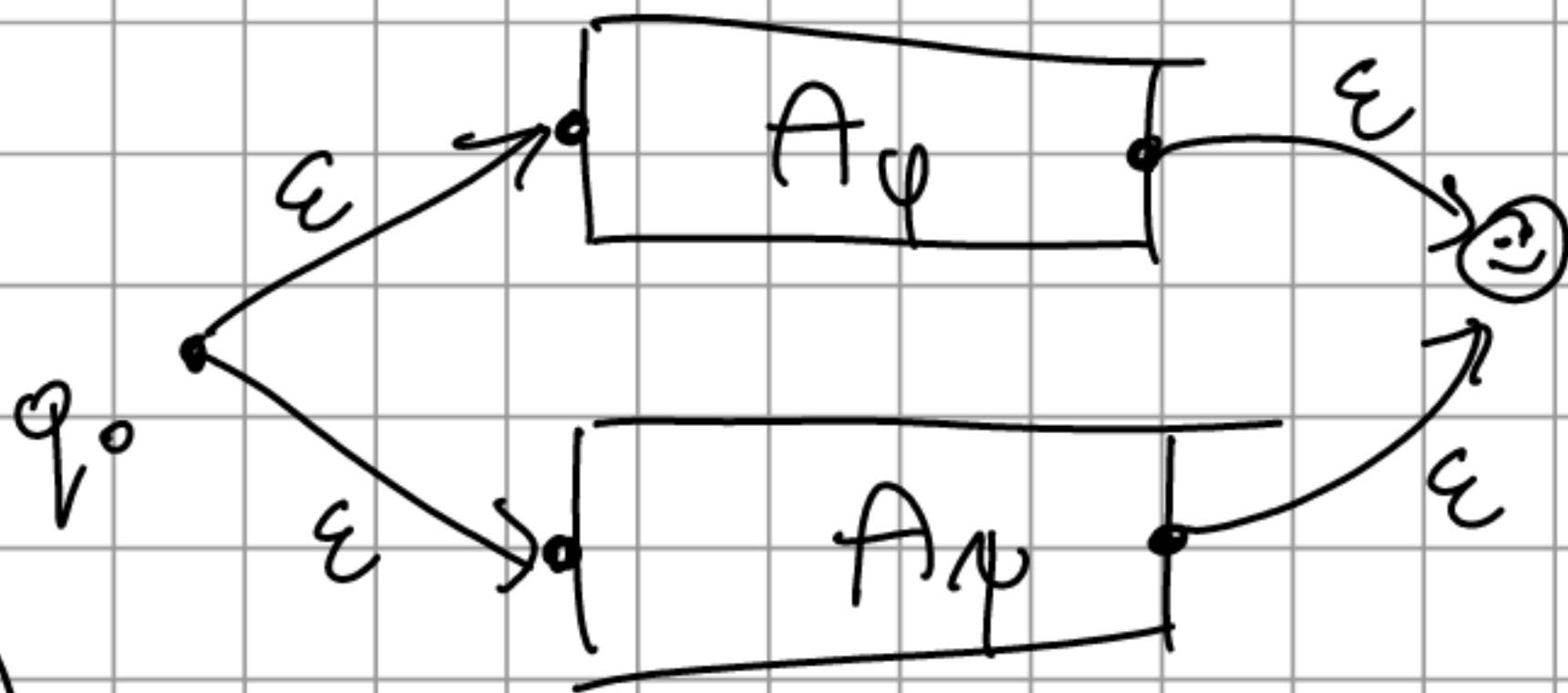
wyrażenia. Każdy NFA który zbudujemy  
będzie miał jeden stan wyjściowy niebędący  
akceptującym i jeden akceptujący.



- $a \sim q_0 \xrightarrow{a} \text{☺}$

- $\varphi, \psi \rightsquigarrow$

(Automat dla  $\varphi + \psi$ )

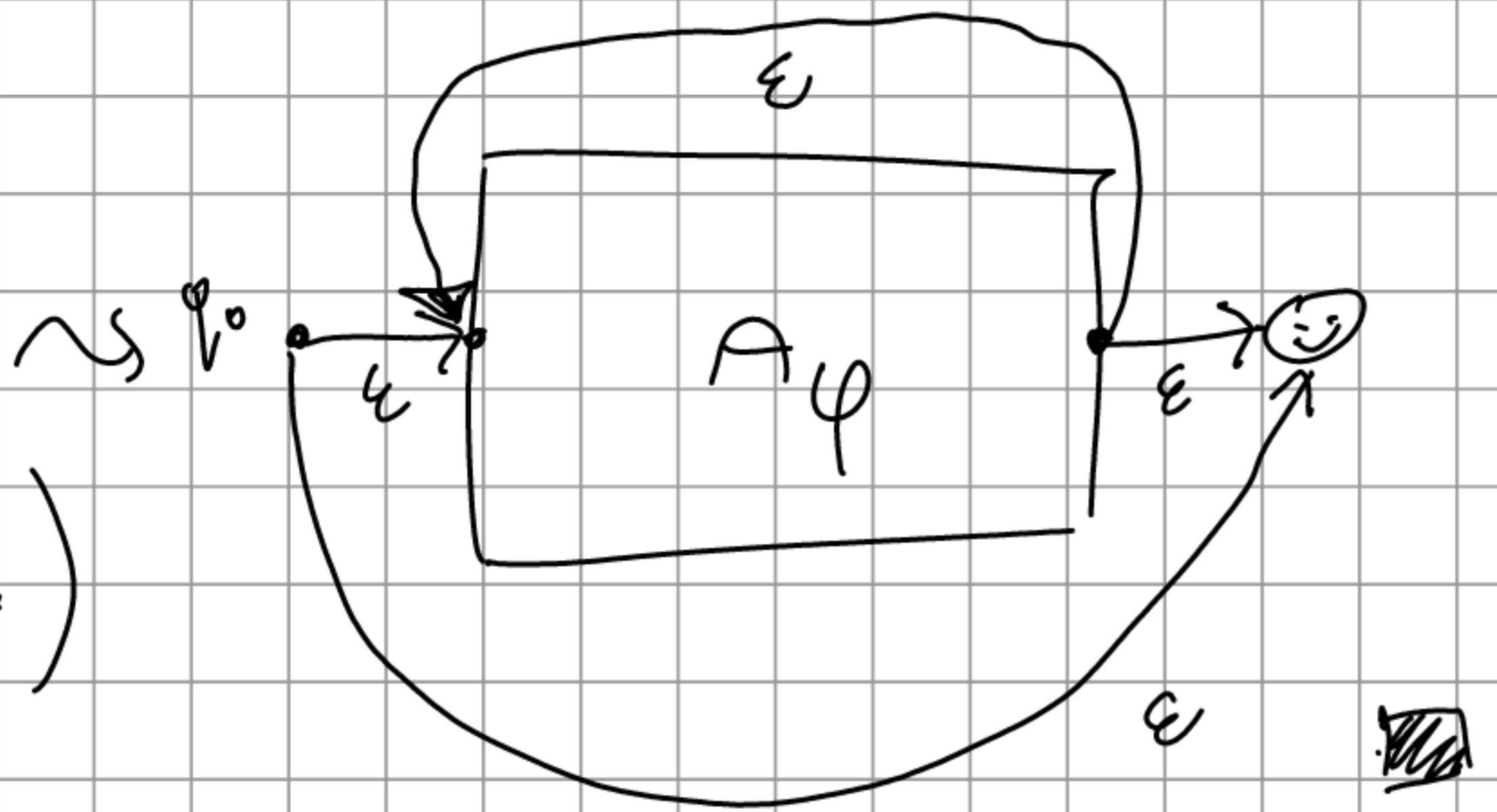


- $\varphi, \psi \rightsquigarrow$

(Automat dla  $\varphi\psi$ )



- $\varphi$   
(Automat dla  $\varphi^*$ )



(1)  $\Rightarrow$  (2) Mamy DFA  $A = (\Sigma, Q, q_0, \delta, F)$

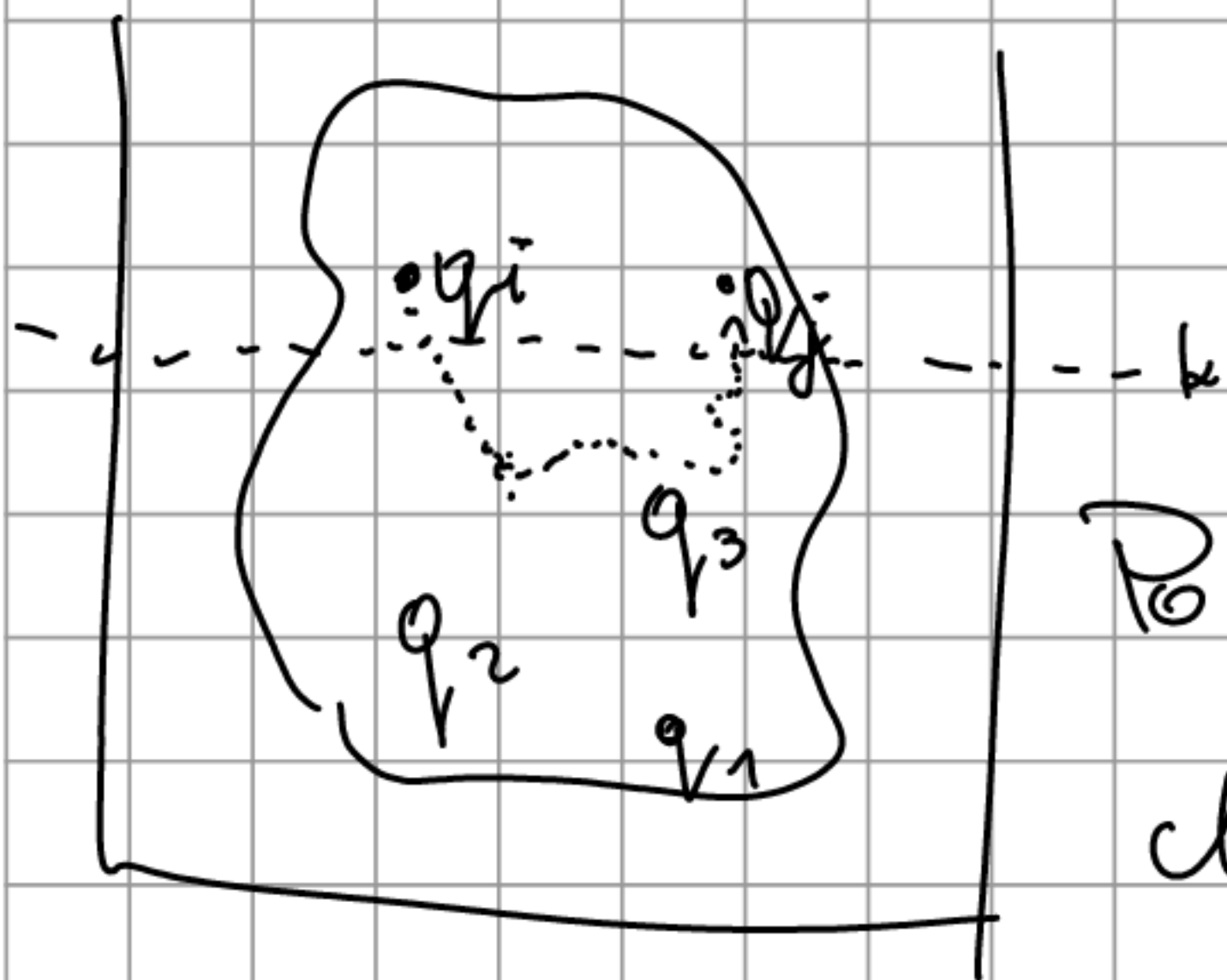
$Q = \{q_1, \dots, q_n\}$  (gdzie  $q_0 = q_1$ ).

Dla  $1 \leq i, j \leq n, 0 \leq k \leq n$  napiszemy wyrażenie regularne  $\varphi_{i,j}^k$  wyrażające język

$$\forall w \in \Sigma^* : \hat{\delta}(q_i, w) = q_j \wedge$$

niepusty  
oraz  $\neq \epsilon$

$\forall v \in \Sigma^*$  (jeśli  $v$  jest właściwym  
prefiksem  $w$  oraz  
 $\hat{\delta}(q_i, w) = q_k$ , to  $L < k$ )



o drodze z  $q_i$  do  $q_j$   
chodzimy tylko po stanach  
o indeksach  $< k$ .

Indukcja względem  $k$ :

- $\varphi_{i,i}^0 = \epsilon + \bigoplus_{a \in \Sigma} a$  (suma po literach spełniających warunek)  
 $\delta(q_i, a) = q_i$

- $\varphi_{i,j}^0 = \bigoplus_{a \in \Sigma} a$   
 $\delta(q_i, a) = q_j$

- $\varphi_{i,j}^{k+1} = \varphi_{i,j}^k + \varphi_{i,k+1}^k (\varphi_{k+1,k+1}^k)^* \varphi_{k+1,j}^k$

Mając te wyrażenia regularne piszemy  
 $\psi$  t. je  $L_A = L_\psi$ :



$$\psi = \sum_{q_i \in F} \varphi_{1,i}^n$$



## UOGÓLNIENIA

Są dwa możliwe kierunki:

- słowa nieskończone
  - drzewa
- } można iść w obu tych kierunkach jednocześnie

Automat skończony na słowach nieskończonych.

$\langle \Sigma, Q, q_0, \delta, \text{CO TUTAJ?} \rangle$

↑  
skończony

↑  
Duzo warunków akceptacji  
Büchig, Rabine...

Interpretacja Bückiego:  $F \subseteq Q$ .

Słowo jest akceptowane, gdy automat nieskończenie wiele razy odwiedza stany


z  $F$ .

Pytanie: czy deterministyczne automaty Büchiego robią to samo co niedeterministyczne?

Odpowiedź: nie w ten sposób co poprzednio. Przykład:  $|\Sigma| = 1$ ,



Lepsza odpowiedź:  $L$  - zbiór słów nieskończonych nad  $\Sigma = \{0, 1\}$ , w których jest tylko skończenie wiele zer.

$L$  jest rozstrzygany przez 

Deterministycznym się nie da: ćwiczenie.

14.03.2022

Uwaga Klasa języków regularnych nad  $\Sigma$  jest najmniejszą klasą języków nad  $\Sigma$ , która:

- zawiera wszystkie języki skończone (\*)
- jest zamknięta na sumę, konkatenację i gwiazdkę Kleene'go ( $L \cup L^*$ )

Istnienie: proste, Najmniejszość: pokazać, że dowolna klasa języków spełniająca powyższe warunki zawiera języki regularne.

## GRAMATYKI BEZKONTEKSTOWE

Def. Gramatyka bezkontekstowa (CFG) to

krotka  $\langle N, \Sigma, S, \Pi \rangle$ ,  $S \in N$ ,

$\Pi \subseteq N \times (N \cup \Sigma)^*$ ,  $N \cap \Sigma = \emptyset$   
skończone

CFG - Context Free Grammar

Dygresja  $A$ : alfabet,  $\Pi \subseteq A^* \times A^*$   
skończone

Dla  $w, v \in A^*$  definiujemy  $w \xrightarrow{\Pi} v$  gdy  
istnieją słowa  $x, y \in A^*$  i para  
 $\langle l, r \rangle \in \Pi$  t.ż.  $w = xly$  oraz  $v = xry$

Przykład: bierzemy  $w$ , znajdujemy  
w nim infiks  $l$ , zamieniamy go  
na  $r$  i dostajemy  $v$ .

Relacje  $\xrightarrow{*} \Pi$  i  $\overset{*}{\leftrightarrow} \Pi$  definiujemy <sup>odpowiednio</sup> jako  
transytywne i równoważnościowe domknięcie

relacji  $\xrightarrow{\Pi}$ .

- $\xrightarrow{*} \Pi$  osiągalność w grafie
- $\overset{*}{\leftrightarrow} \Pi$  bycie w jednej spójnej  
składowej (osiągalność w grafie  
bez skierowania)

KONIEC DYGRESJI



Dla danej CFG  $G = \langle N, \Sigma, S, \Pi \rangle$  przez  $\bar{L}_G$  oznaczamy  $\{w \in (N \cup \Sigma)^* : S \xrightarrow{*} \Pi w\}$ ,  
przez  $L_G = \bar{L}_G \cap \Sigma^*$

Def.  $L \subseteq \Sigma^*$  jest bezkontekstowy (CFL),  
gdy istnieje CFG  $G$  t.ż.  $L = L_G$

Obserwacja Każdy język regularny jest bezkontekstowy.

Dowód Klasa CFL spełnia warunki z uwagi (\*), (i), (ii), (iii):

(\*) = proste, dodajemy reguły  $S \rightarrow w$  dla  $w \in$  języka

(i): bierzemy sumę rozłączną dwóch grammatyk i dodajemy reguły  $\langle S, S' \rangle, \langle S, S'' \rangle$   
nowy  $S$



(ii):  $G' = \langle N', \Sigma, S', \Pi' \rangle$ ,  $G'' = \langle N'', \Sigma, S'', \Pi'' \rangle$ ,

konstruujemy  $G = \langle N' \cup N'' \cup \{S\}, \Sigma, S, \Pi' \cup \Pi'' \cup \langle S, S'S'' \rangle \rangle$

(iii) Podobnie gwiazdka.

Przykład • Notacja:  $S \rightarrow aSa \mid bSb \mid \epsilon$

oznacza, że  $\Pi = \{ \langle S, aSa \rangle, \langle S, bSb \rangle, \langle S, \epsilon \rangle \}$

(to konkretnie daje język palindromów parzystej długości).

•  $S \rightarrow SS \mid \epsilon \mid aSb \mid bSa$

$\leadsto$  język słów, które mają tyle samo liter a co b.

•  $S \rightarrow (S) \mid [S] \mid SS \mid \epsilon$

poprawne nawiasowanie =  $(, ), [, ]$ .

Konwencja notacyjna (nieformalna Marcinkowskiego):

Dla języków  $L, L'$  piszemy  $L = L'$

gdy  $L \stackrel{\cdot}{=} L' \subseteq \{ \epsilon \}$   
↑  
różnica symetryczna

Def CFG  $G = \langle N, \Sigma, S, \Pi \rangle$  jest postaci normalnej Chomskiego, gdy każda produkcja  $\in \Pi$  jest postaci  $A \rightarrow BC$  dla  $A, B, C \in N$  lub  $A \rightarrow a$  dla  $A \in N, a \in \Sigma$ .

Tw. (Chomskiego o postaci normalnej)

Dla każdego CFL  $L$  istnieje CFG  $G$  w postaci Chomskiego t.ż.  $L = L_G$ .

Dowód: Nudny :-)

Lemat (o pompowaniu dla CFG)

Dla każdego CFL  $L$  istnieje  $n \in \mathbb{N}$  t.ż. dla każdego  $w \in L, |w| \geq n$ , istnieją słowa



$s, z, t, y, x$  t.ze  $|zty| \leq n, |zy| > 0, w = sztyx$   
 dla każdego  $k \in \mathbb{N}$   $sz^k t y^k x \in L$ .

Uwaga Czy język  $\{a^i b^j c^k : i, j \in \mathbb{N}\}$  jest CFG?

Odp.: TAK (proste)

A co z  $\{a^i b^j c^k : i, j \in \mathbb{N}\}$ ?

Odp.: TAK (to samo)

A czym jest przekrój tych języków?

Odp.:  $\{a^i b^i c^i : i \in \mathbb{N}\}$

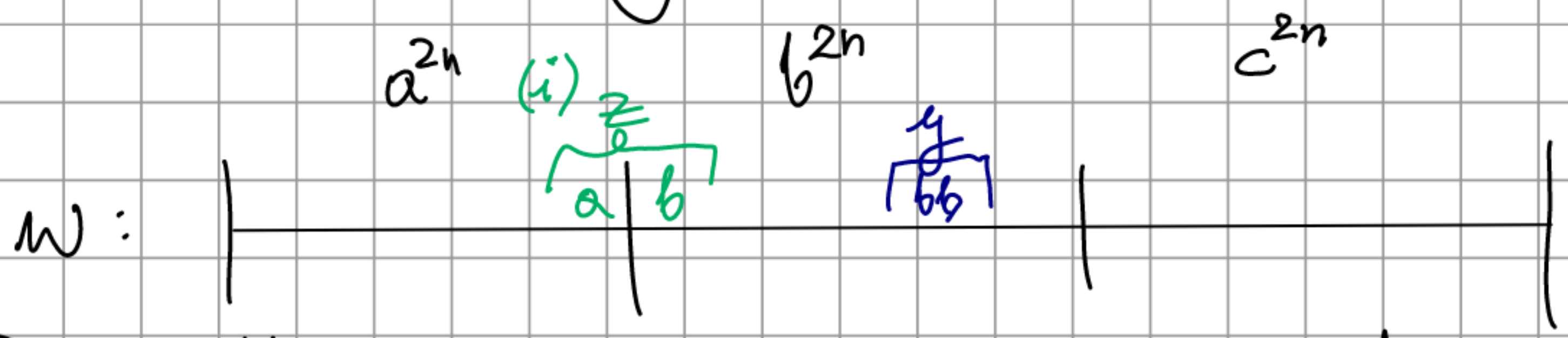
Ten język jednak nie jest CFL!

Dowód Założmy, że  $L$  jest CF. Niech

$n$ : stała z lematu o pomiarze.

Niech  $w = a^{2n} b^{2n} c^{2n}$ . Wtedy są

stałe  $s, z, t, y, x$  z lematu.



Przypadki: (i)  $z$  lub  $y$  zawiera dwie różne literki, to dla  $k=2$  wygrujemy

(ii) jeżeli  $x, y$  mają tylko po jednej literce, to  $k=0$  wygramy bo tej trzeciej literki jest więcej.

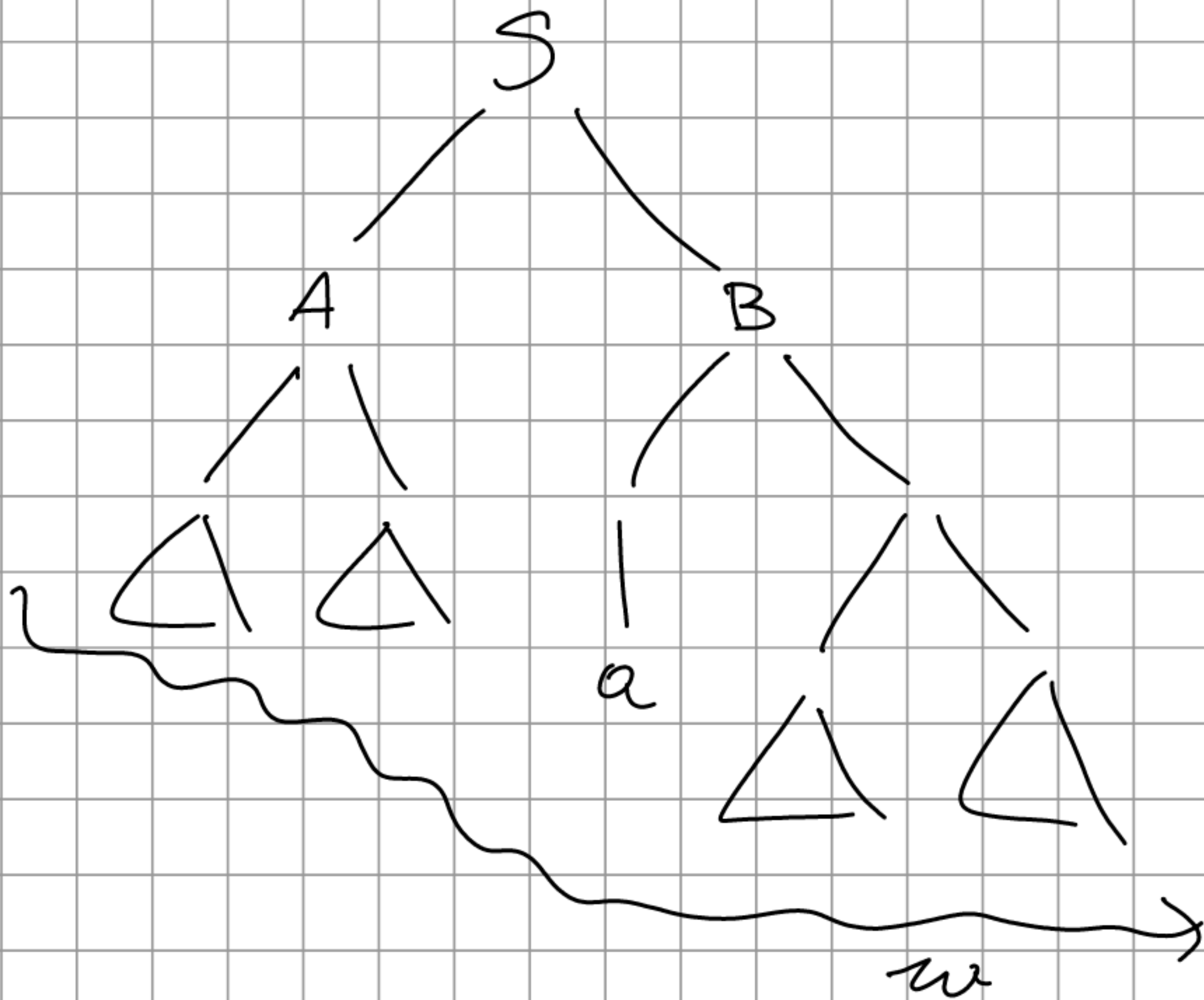
21.03.2022

D-d. (lematu o pompowaniu)

Weźmy  $L \in CFL$  oraz CFG  $G = \langle N, \Sigma, S, \Pi \rangle$

w postaci normalnej Chomsky'ego i niech

$n = 2^{|N|+3}$ . Niech  $w \in L$ ,  $|w| \geq n$ .

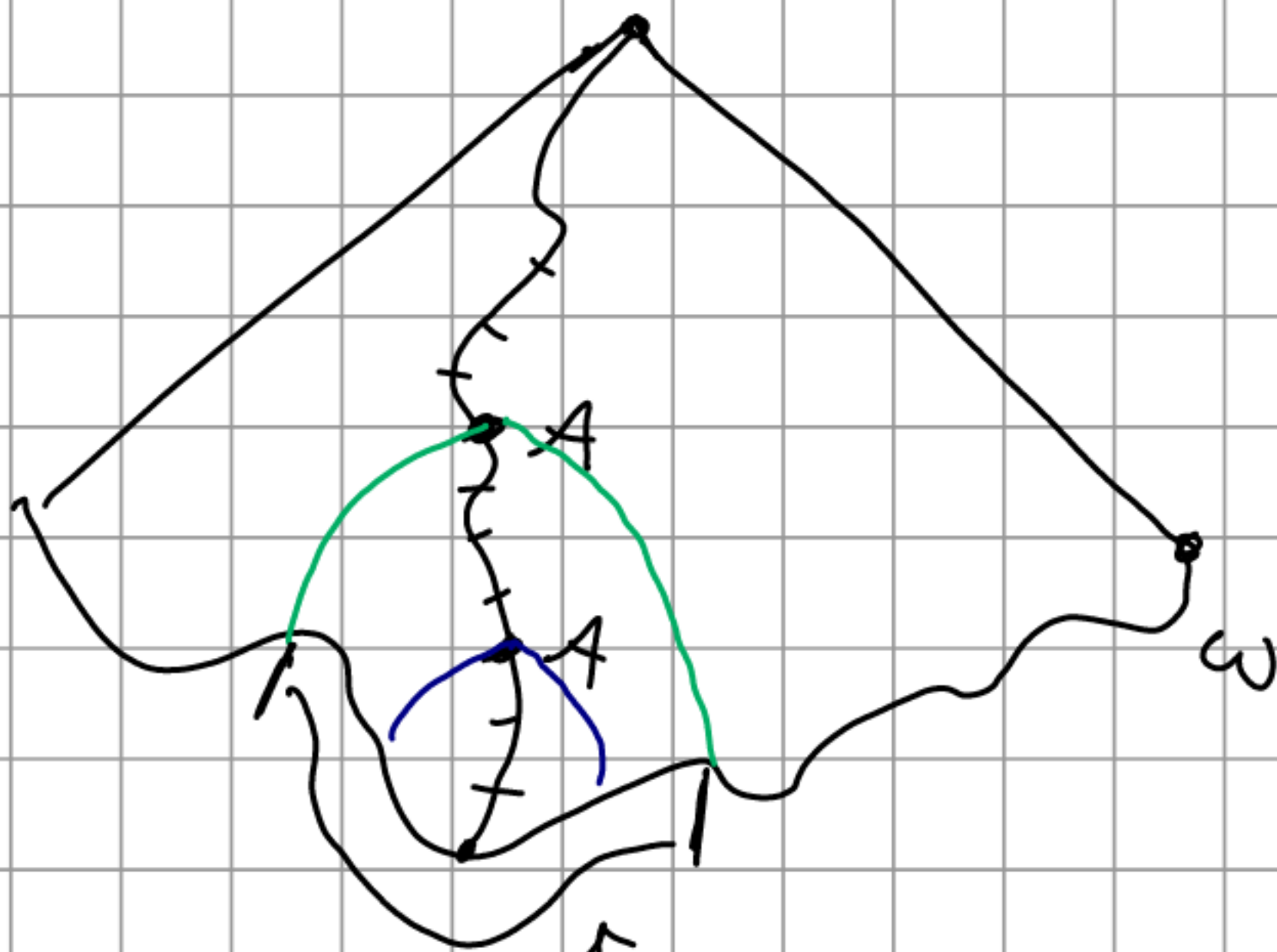


Drzewo słowa  $w$ .

Jaka jest najdłuższa ścieżka od  
korzenia do liścia? Co najmniej  
ma długość  $|N|+3$  ( $\log n$ ).



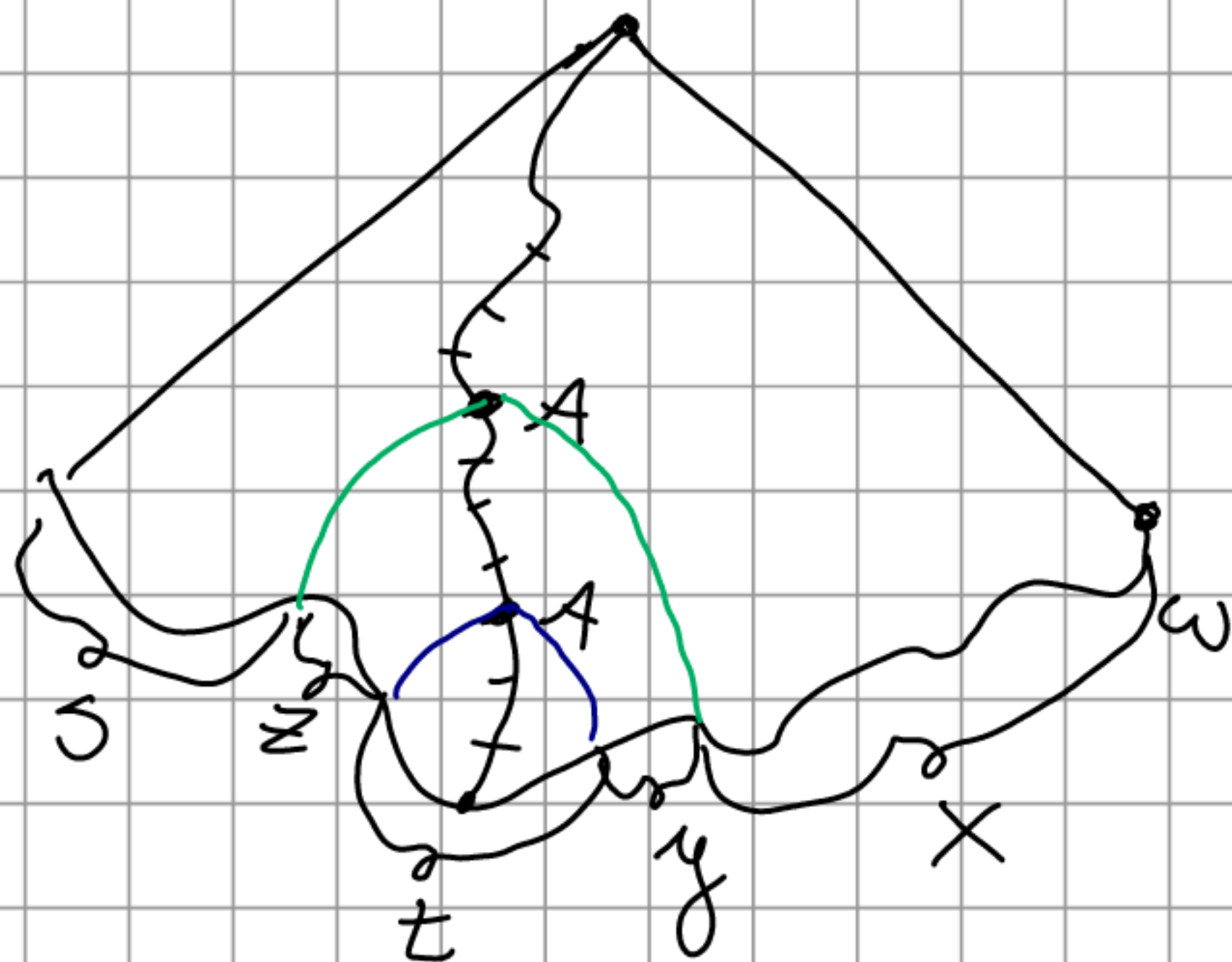
Na tej ścieżce są same nieterminale  
(poza liściem), zatem musi być  
jakiś nieterminal, który się powtórza.  
Niech  $A$  będzie takim nieterminalem  
który jest najniższy na tej ścieżce.



to ma  $dt. \leq 2^{|N|+1}$

(mo to tylko  $A$  się  
może powtórzyć na  
tej ścieżce)

Podział jest naturalny:

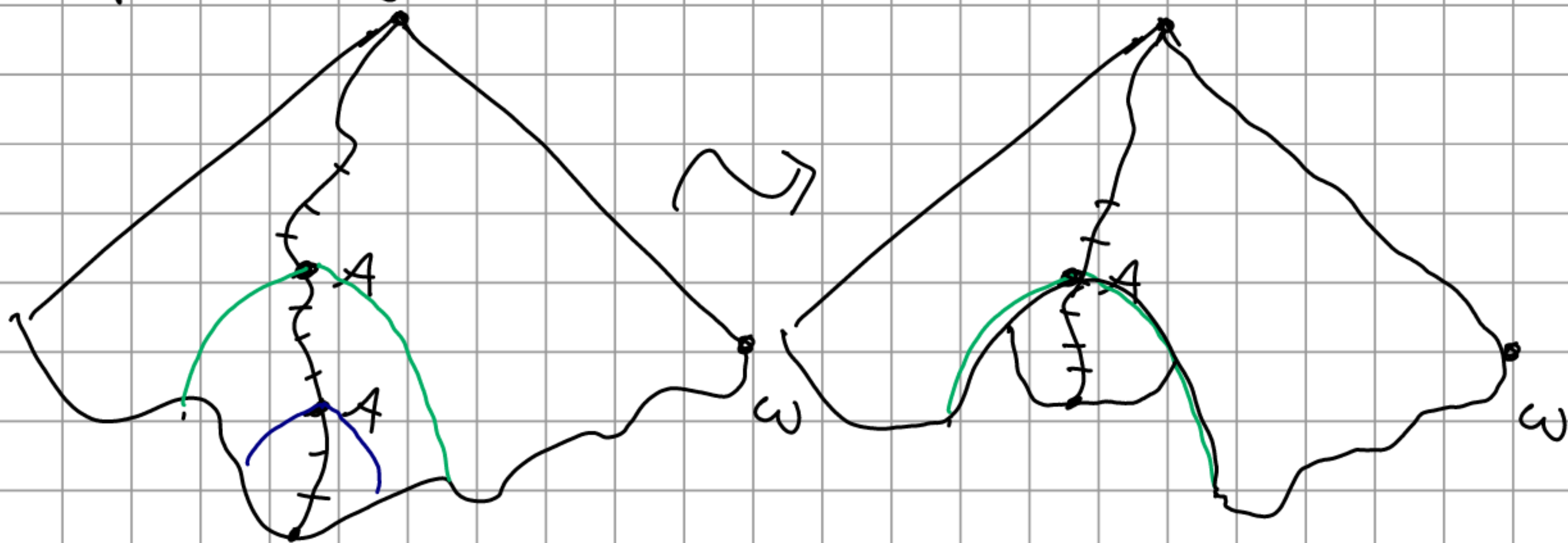


Wtedy  $|zty| \leq 2^{|\mathcal{N}|+1} \leq n$ .

Ponadto  $|zy| > 0$  (to niszcz wystąpienie

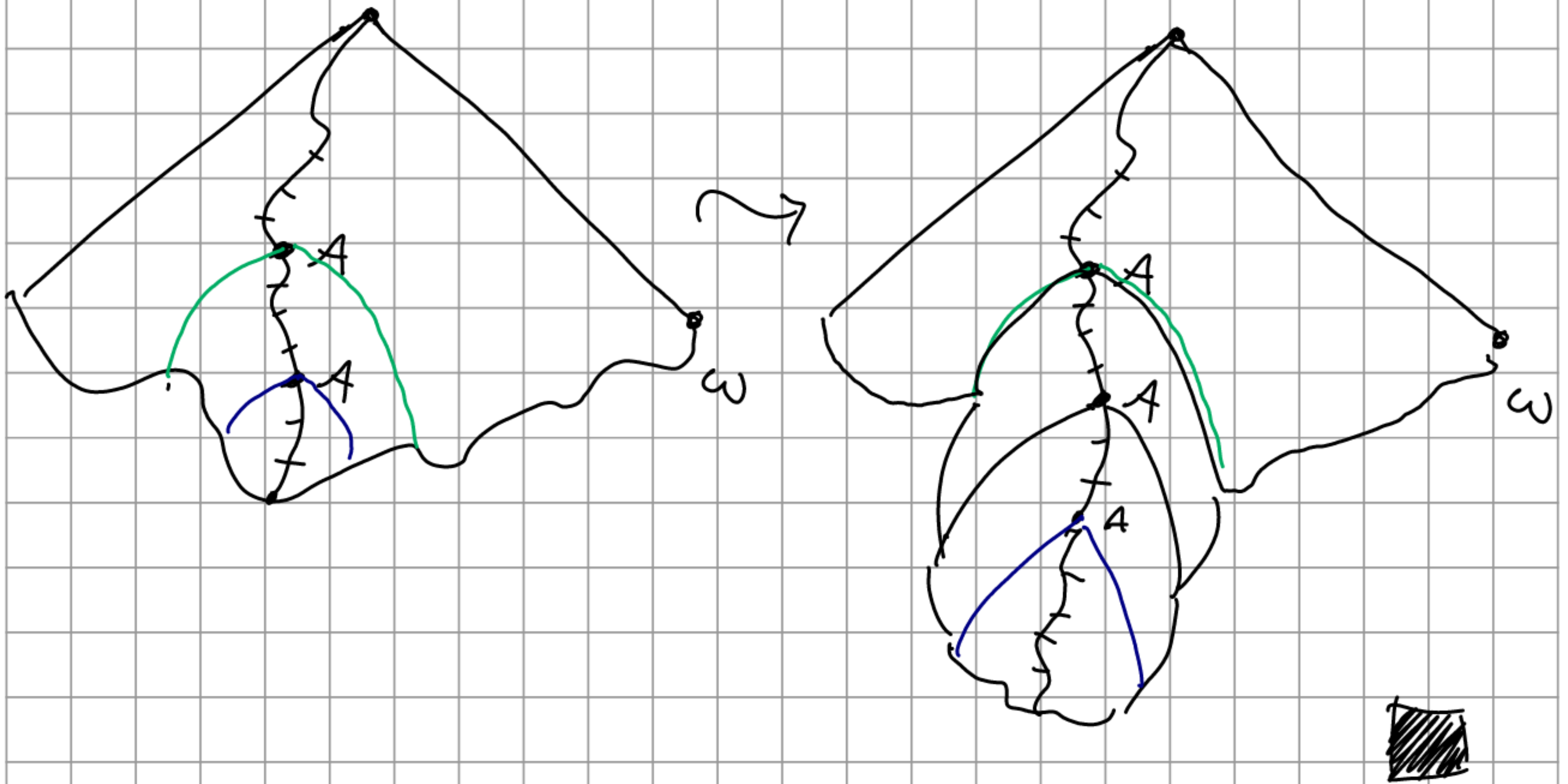
$A$  jest potomkiem tylko jednego  
dziecka tego wystąpienia wyżej).

- gdy  $k=0$ : "przeszczepiamy" tą niszcz  
produkcję  $A$  pod tą większą





- gdy  $k \geq 1$ : "replikujemy" tę wyzszą produkcję tyle ile trzeba.



## AUTOMATY ZE STOSEM

Def. Automat ze stosem (NPDA: Nondeterministic

Push Down Automaton) to krótka

$$A = \langle \Sigma, Q, q_0, S, Z, \delta \rangle,$$

$\Sigma$  alfabet skończony  
 $Q$  zbiór stanów  
 $q_0$  stan początkowy  
 $S$  zbiór kółek sweterków (skończony)  
 $Z$  sweterek dna stosu  
 $\delta$  relacja przejścia

gdzie  $\delta \subseteq (Q \times (\Sigma \cup \epsilon) \times S) \times (Q \times S^*)$

ora  $\delta$  ma takie własności:

- $\delta(q, a, Z, q', w) \Rightarrow w = Z^v$  gdzie  $v$  nie zawiera  $Z$ .
- $\delta(q, a, A, q', w) \wedge A \neq Z \Rightarrow w$  nie zawiera  $Z$ .

Wtedy  $\hat{\delta} \subseteq \Sigma^* \times (Q \times S^*)$  jest

najmniejsza relacja spełniająca:

- $\hat{\delta}(\epsilon, q_0, Z)$

- $\hat{\delta}(w, q, VA) \wedge \delta(q, a, A, q', w)$

$\Rightarrow \hat{\delta}(wa, q, VW)$  ( $a \in \Sigma \cup \{\epsilon\}$ ).

Wtedy  $L_A = \{w : \exists q \hat{\delta}(w, q, Z)\}$ .

Przykład Palindromy pangramy:

$$\delta(q_1, B, A, q_1, AB)$$

$$\delta(q_1, \epsilon, A, q_2, A)$$

$$\delta(q_2, A, A, q_2, \epsilon)$$

Przykład Słowa w t. z.  $|w|_0 = |w|_1$ .



Tw. Klasa języków bezkontekstowych jest  
równa klasie języków wyrażanych przez NPDA.

Uwaga (bardzo ważna) Klasy rozstrzygane  
przez maszyny deterministyczne są zamknięte  
na dopełnienie.

Pytanie Czy w sformułowaniu tw. nie  
można zmienić NPDA na PDA:

$\{a^i b^i c^i : i \in \mathbb{N}\}$  nie jest CFL

ale dopełnienie jest.

Def. Zanim to, wprowadzimy pojęcie postaci  
normalnej Greibach: CFG  $G$  jest w  
tej postaci gdy dla każdego  $\langle A, w \rangle \in TT$   
 $w \in \Sigma N^*$ .

Lemat Dla każdego  $L: CFL$  istnieje  $G$   
w postaci normalnej Greibach t.ż.  $L = L_G$

Dawód tw. " $\Rightarrow$ " Weźmy CFL  $L$  oraz CFG  $G$   
w postaci normalnej Greibach t.ze  $L = L_G$ .

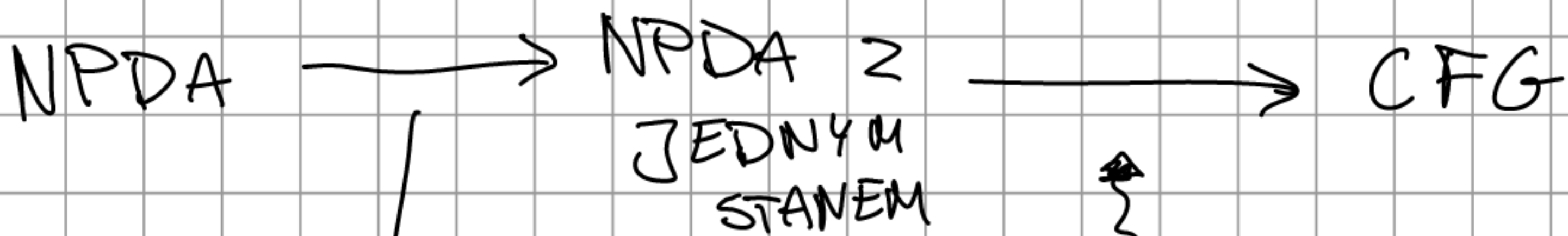
Zbudujemy NPDA  $\langle \Sigma, Q, q_0, S, Z, \delta \rangle$   
t.ze  $L = L_A$ .

- Jeśli w  $\Pi$  jest produkcja  $A \rightarrow a w$ ,  
to  $\delta(q, a, A, q, w^R)$ ,  
•  $\delta(q_0, \epsilon, Z, q, ZS)$ ,

gdzie  $Q = \{q_0, q\}$ ,  $S = N$

28.03.2022

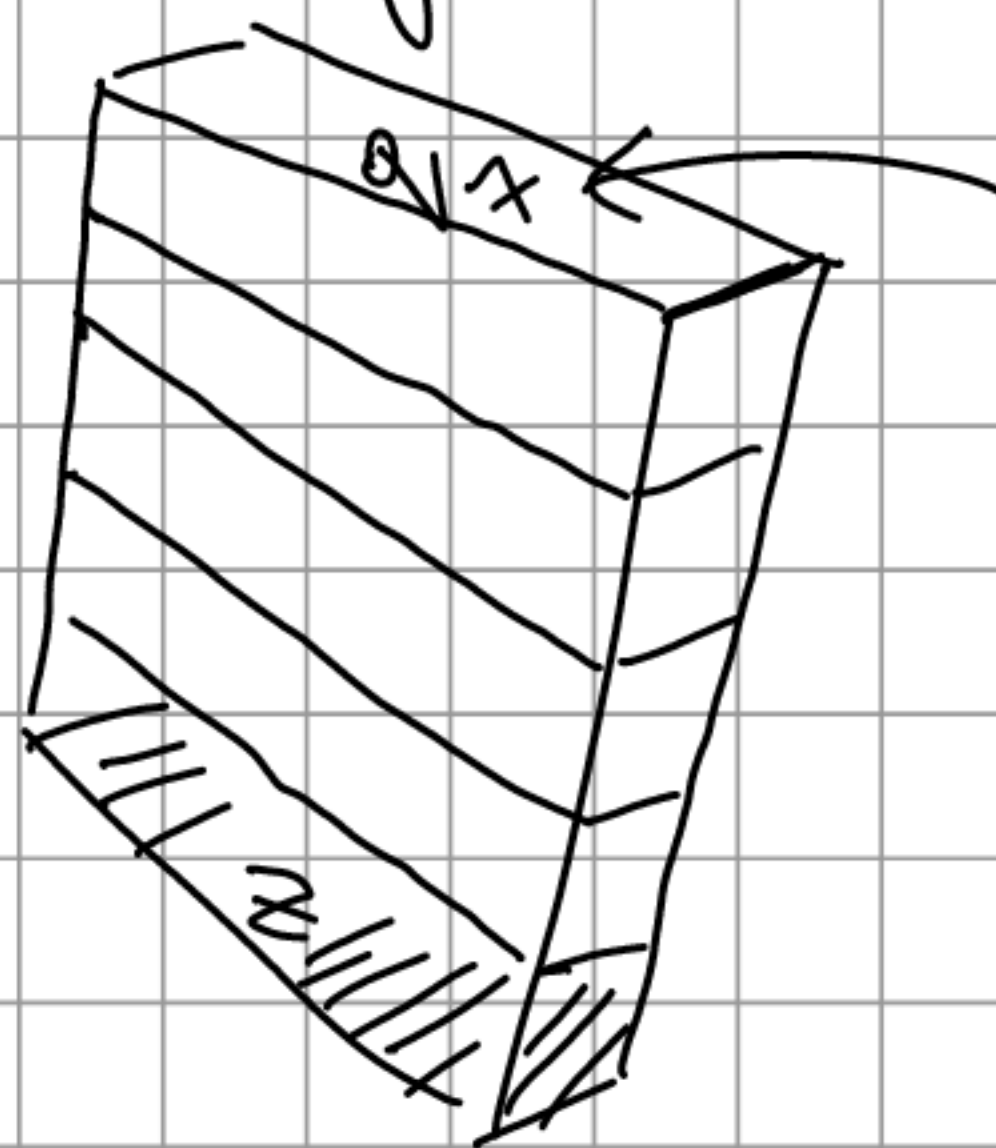
Kontynuacja dowodu NPDA  $\rightarrow$  CFG



ta część już potrafimy, analogicznie do poprzedniej części dowodu.

Teraz stos to kłobrowe

kłocki, na wierzchu których napisany jest jakiś stan



"Wyobraź sobie, że oglądając ten klocek jesteś w stanie  $q_7$ "

Problem pojawi się przy ślizganiu kłobków. Ten napisany stan może być już nieakceptowny.

stary automat

Np.:  $\langle q_7, \text{czerwony}, a, q_3, \text{ziel-nieb-ziel-czerw} \rangle$

nowy automat

$\langle -, \langle q_7, \text{czerw} \rangle, a, \langle q_3, \text{ziel} \rangle, \langle ?, \text{nieb} \rangle, \langle ?, \text{ziel} \rangle, \langle ?, \text{czerw} \rangle \rangle$

co tutaj?  
no nie wiadomo

Naprawa: teraz klocki będą takie:



dziura

Trypieni jest  
tyle co stanów.

Nasze stany:  $\langle \text{trypień}, \text{dziura}, \text{kolor} \rangle$

Instrukcje starego automatu:

$\langle q, k, a; q', A_1 A_2 \dots A_L \rangle, L \geq 1$

↑  
wierzch

→ zastępujemy zbiorem wszystkich  
instrukcji postaci



(pomijemy stan: jest tylko jeden)

$\langle [d_0, k, q], a; [d_0, A_1, d_1] [d_1, A_2, d_2] \dots [d_{L-1}, A_L, q'] \rangle$

↑ kolor ↑  
dziurka ↑  
topień

(kwantyfikujemy po  $d_0, d_1, \dots, d_{L-1}$ )

Instrukcje starego automatu:

$\langle q, k, a, q', \epsilon \rangle$

$\hookrightarrow \langle [q', k, q], a, \epsilon \rangle$

Poprawność Jasne jest, że każdy przebieg  
starego automatu można zesymulować nowym.

W drugą stronę: nie wprost w miarę

łatwo.

Kilka szczegółów o których nie chcemy  
rozmawiać: co z dnem stosu? Co

z akceptowaniem?

---

## DRUGA CZĘŚĆ KURSU

Wcześniej: Język  $\subseteq \Sigma^*$

Teraz: Problem  $\subseteq \mathbb{N}$

Będziemy pisać programy, które wczytują jedną liczbę naturalną, a jeśli zwrócają wynik, to on też będzie liczbą naturalną.

Def.  $A \subseteq \mathbb{N}$  nazywamy rekurencyjnym (obliczalnym, rozstrzygalnym) jeśli istnieje program  $\varphi$  t.że dla każdej  $n \in \mathbb{N}$ :

$$\begin{cases} \varphi(n) = 0 \Leftrightarrow n \notin A \\ \varphi(n) = 1 \Leftrightarrow n \in A \end{cases} .$$

Obserwacja każdy zbiór skonieczony jest rekurencyjny. Klasa zbiorów rekurencyjnych jest zamknięta na sumę, przecięcie i dopełnienie.

Obserwacja Istnieje nierekurencyjne podzbiory  $\mathbb{N}$ .

Def. Podzbiór  $A \subseteq \mathbb{N}$  jest rekurencyjnie przeliczalny (r.e.: recursively enumerable) gdy istnieje program  $\varphi$  t.ż. dla każdego  $n \in \mathbb{N}$ :

$$(i) \quad \varphi(n) = 1 \Leftrightarrow n \in A$$

$$(ii) \quad \varphi(n) \in \mathbb{N} \Leftrightarrow n \in A \quad \wedge$$

$$\varphi(n) = \perp \Leftrightarrow n \notin A$$

$\uparrow$   $\varphi$  się zapętla na  $n$ .

Obserwacja Klasa r.e. jest zamknięta na przecięcie i sumę.

Obserwacja Jeżeli  $A$  jest r.e. i  $\mathbb{N} \setminus A$  jest r.e., to  $A$  i  $\mathbb{N} \setminus A$  są rekurencyjne.

Numerujemy wszystkie programy **EFEKTYWNE**,  
 tzn. mamy inny program, który może  
 wczytać program i zwrócić jego  
 numer oraz dla numeru zwrócić program.

	1	2	3	4	5	...
$\varphi_1$	⊥	⊥	⊥			
$\varphi_2$	4	2137	28			
$\varphi_3$	1	⊥	⊥			
$\varphi_4$	0	7	⊥			
⋮						

$$K = \{ n : \varphi_n(n) \in \mathbb{N} \}$$

Obserwacja  $K$  jest r.e.

Umiemy policzyć  $\varphi_n$   
 i uruchomić go na  $n$ .

zbiór tych miejsc  
 na przekątnej  
 gdzie jest liczbę,  
 czyli te programy  
 które się zatrzymają  
 dla swojego numeru



T.W. (Turinga o nierozstrzygalności problemu stopa)  
 $K$  jest nierozstrzygalny.

D-d. Dowód nie wprost. Założymy że

$K$  jest rozstrzygalny przez pewien program  $\varphi$ .

Niech  $\varphi$  będzie programem, który:

- wczytuje  $n$ ,
- jeśli  $\varphi(n) = 1$ , to się zapętli,
- w p.p. zwróć 1.

Wtedy  $\varphi$  ma pewien numer  $n$ .

• Jeśli  $\varphi(n) = 1$ , to znaczy, że  $\varphi$  nie zapętla się na  $n$ , ale z definicji  $\varphi$  powinien się zapętlić na  $n$ .

• Jeśli  $\varphi(n) = 0$ , to znaczy, że  $\varphi$  nie powinien się zatrzymać, ale z jego definicji  $\varphi$  się zatrzyma na  $n$ .

$$n \in K \Leftrightarrow \varphi_n(n) \in \mathbb{N} \Leftrightarrow \varphi(n) = 0 \Leftrightarrow n \notin K$$

4.04.2022

Def. Funkcja rekurencyjna to relacja wejścia - wyjścia dla programu w MVFP.

Obserwacja Funkcje rekurencyjne to częściowe funkcje  $\varphi: \mathbb{N} \rightarrow \mathbb{N}$ . Ich wzim podklasa: funkcje rekurencyjne całkowite.

Uwaga Zbiór  $A \subseteq \mathbb{N}$  jest r.e.  $\Leftrightarrow$  istnieje  $f$  rek. t.ze  $A = \text{dom}(f)$

Def. Niech  $A, B \subseteq \mathbb{N}$ . Wtedy  $A \leq_{\text{rek}} B$  (czytaj "A jest nie trudniejszy od B ze względu na redukcje całkowite rekurencyjne") gdy istnieje całkowita funkcja rek.  $f$  (zwana redukcją) t.ze  $\forall n \in \mathbb{N} (n \in A \Leftrightarrow f(n) \in B)$



Obserwacje (1)  $A \leq_{\text{rek}} B$  i  $B \leq_{\text{rek}} C$ , to  $A \leq_{\text{rek}} C$

(2)  $A \leq_{\text{rek}} B$  i  $B$  jest rekurencyjny, to  $A$  też.

(3)  $A \leq_{\text{rek}} B$  i  $B$  jest r.e., to  $A$  też jest r.e.

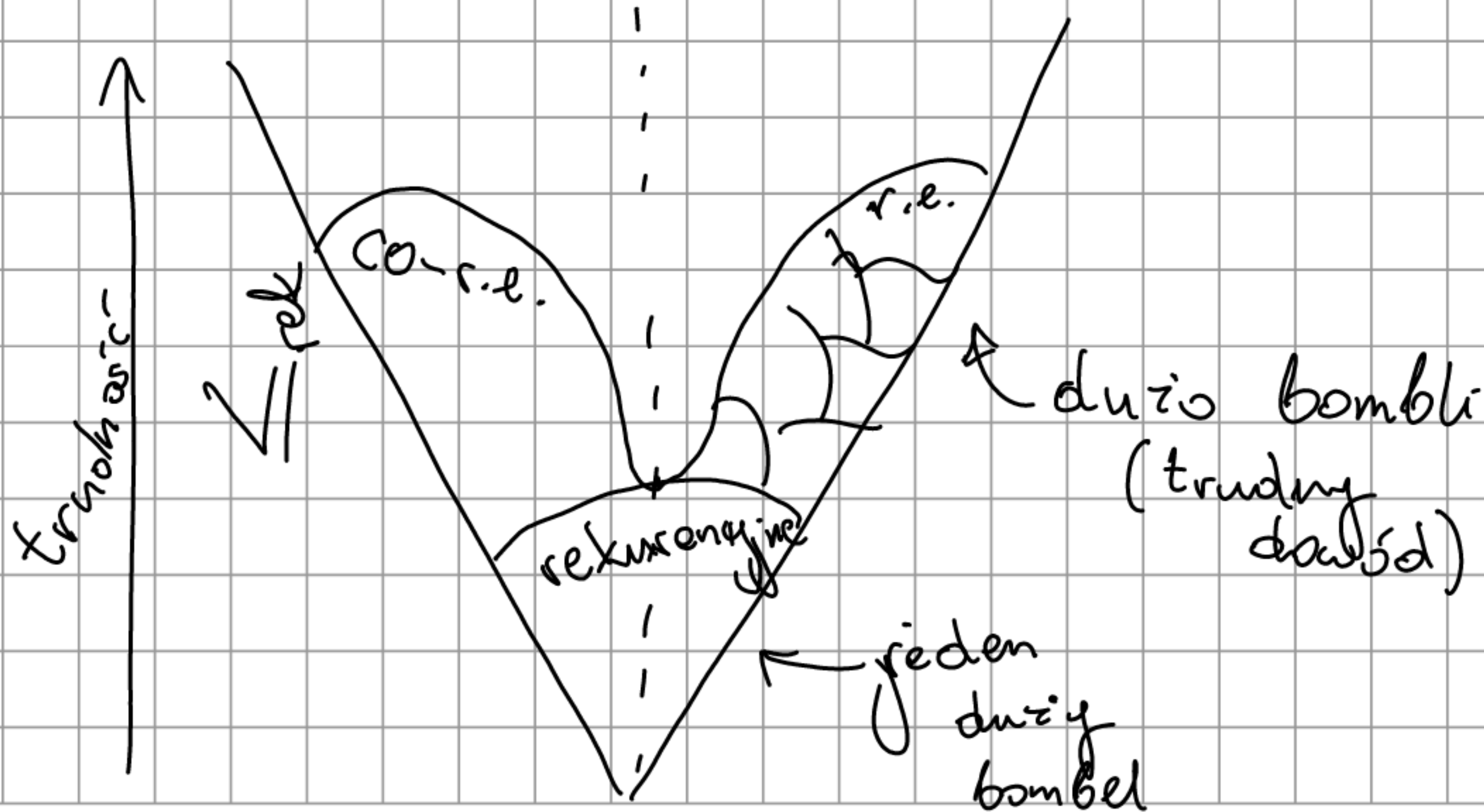
(4) Jeżeli  $A, B$  rekurencyjne i nietrywialne,  
( $\neq \emptyset, \mathbb{N}$ ), to  $A \leq_{\text{rek}} B$  i  $B \leq_{\text{rek}} A$

D-d. (4) Weźmy  $b \in B$ ,  $b' \in \mathbb{N} \setminus B$ . Niech

$f_A$ : rozstrzyga  $A$ . Zbudujemy redukcję  $f$ :

$f(n) =$  "wczytaj  $n$ , uruchom  
 $f_A(n)$ , jeśli wyszło  
0, to zwróć  $b'$ , jeśli  
wyszło 1, to  $b$ "

$\leq_{\text{rek}}$  dzięki  $P(\mathbb{N})$  ma "bomble"





Tw.  $K$  (z poprzedniego wykładu) jest zupełny w klasie r.e. ze względu na całkowite redukcje rekurencyjne, tj. dla każdego  $B \in r.e.$  zachodzi  $B \leq_{rek} K$

Wykład  $A_7 = \{n : \varphi_n(7) = 77\}$  jest r.e. Pokażemy, że  $K \leq_{rek} A_7$ . Budowa redukcji:

- przyjmujemy numer  $n$ ,
- piszemy program: 

	wczytaj $m$
	uruchom $\varphi_n(n)$
	zwróć 77

- zwracamy numer tego programu

To jest funkcja całkowite i do tego działa. Gdy  $n \in K$ , to  $\varphi_{f(n)}$  się skończy i zwraca 77 dla dowolnego wejścia, więc  $f(n) \in A$ . Gdy  $n \notin K$ , to  $\varphi_{f(n)}$  się nie skończy dla każdego



wejścia, więc  $f(n) \notin A_7$ .

Def  $A \subseteq \mathbb{N}$  jest ekstensjonalny jeśli

$$\forall i \in A, j \notin A \exists n \varphi_i(n) \neq \varphi_j(n)$$

↑  
może  
być  
pińeczka

Przykład  $A = \{n : \varphi_n \text{ jest całkowite}\}$

Alt. definicja  $A$  ekstensjonalny gdy  $\forall i, j \in \mathbb{N}$

jeśli  $\varphi_i = \varphi_j$  (jako funkcje), to  $i \in A \Leftrightarrow j \in A$

Tw. (Rice'a) Żaden nietrywialny zbiór

ekstensjonalny nie jest rekurencyjny.

D-d. Weźmy  $A \subseteq \mathbb{N}$ : nietrywialny i ekstensjo-

nalny. BSO przyjmijmy, że żaden

"numer funkcji pustej" nie należy do  $A$

(zawsze możemy wziąć  $A^c$ ).

Pokażemy, że  $K \leq_{rek} A$ . Niech  $k \in A$ .

Konstruujemy redukcję  $f$ :

• dają nam  $n$

• piszemy program:

wczytaj  $m$   
oblicz  $\varphi_n(n)$   
oblicz  $\varphi_k(m)$   
i zwróć wynik

• zwróć jako  $f(n)$  numer tego programu.

$\Sigma$  - alfabet skończony,  $\Pi \subseteq \Sigma^* \times \Sigma^*$ ,  
Skończony

$w \xrightarrow{\Pi} v$  : relacja na  $\Sigma^*$   
 $\Downarrow$  def  
 $w = w_1 l w_2, v = w_1 r w_2, \langle l, r \rangle \in \Pi,$

$w \xrightarrow{*} \Pi v$  : przechodnie domknięcie  $\xrightarrow{\Pi}$

$w \overset{*}{\longleftrightarrow} \Pi v$  : równoważnościowe domknięcie  $\xrightarrow{\Pi}$ .

Tw. (o nierozstrzygalności problemu słów Thuego)

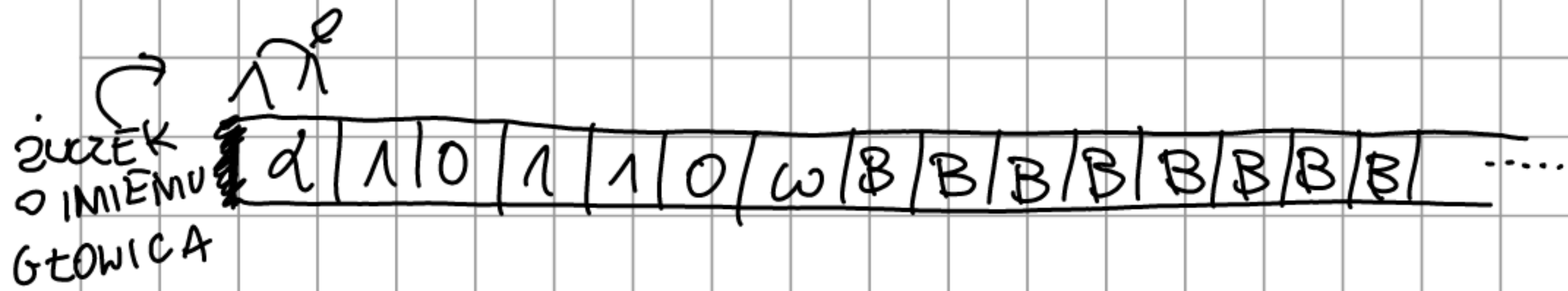
Problemy Semithue =  $\{ \langle w, v, \Pi \rangle : w \xrightarrow{*} \Pi v \}$ ,

Thue =  $\{ \langle w, v, \Pi \rangle : w \overset{*}{\longleftrightarrow} \Pi v \}$  są

nierozstrzygalne.

11.04.2022

# MASZYNA TURINGA



$$\mathcal{A} = \{a, \omega, 0, 1, B\}$$

↑  
blank, początkowo  
tym wypełnione  
taśmę

$Q$ : skończony zbiór stanów,  $q_0 \in Q$ : stan początkowy,  
 $q_f \in Q$ : stan końcowy

$$\delta: Q \times \mathcal{A} \rightarrow Q \times \mathcal{A} \times \{L, R\}$$

↑ aktualny stan      ↑ litarka pod głowicą      ↑ nowy stan      ↑ na co zmienić literę pod głowicą      ↑ przesunąć głowicę w lewo lub praw.

Własności:

- $a$  jest znakiem końca taśmy, nie można go napisać, zmaszczyć, ani przejść na lewo stojąc na  $a$
- widząc  $B$  należy coś napisać
- $\delta$  jest f. częściową



- wynikiem jest ciąg znaków na prawo od  $w$ .
- nie ma znaków z  $q_F$ .

Teraz maszyną Turinga to nasz MUZF, dalej mamy  $K$  "z tytułu glory".

Teza Churcha każdy algorytm daje się przedstawić jako maszyną Turinga.

D-d. (tw. o nierozstrzygalności słów SemiTrue)

(Daję  $\pi, v, w$  i pytają czy  $w \xrightarrow{\pi} v$ )

Pokażemy, że  $K \leq_{red} \text{SemiTrue}$ . n-ta maszyna turinga

Konstrukcja redukcji: daję nam  $M_n, n$

mamy zwrócić  $w, v, \pi$  takie, że  $n \in K \Leftrightarrow w \xrightarrow{\pi} v$

Niech  $Z = Q \cup A \cup \{ \text{półkula} \}$ ,

↑ pocięte

$w = a q_0 \text{---} n \text{---} w B$   
 ↑  
 $n$  zapisane  
 binarnie



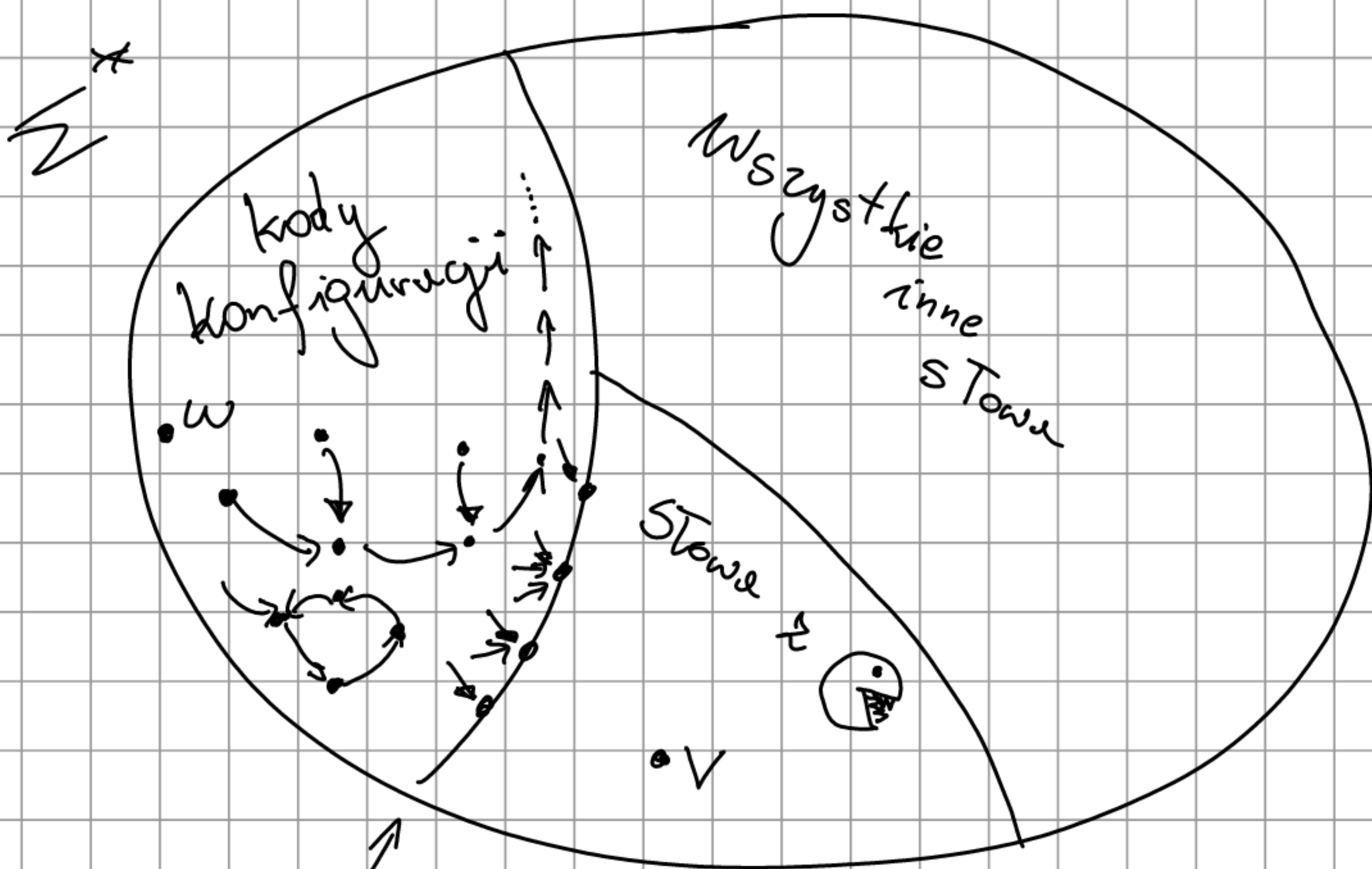
## Konstrukcja $\Pi$ :

- dla każdej "instrukcji" z  $\delta$ , która ma postać  $\delta(q, a) = \langle q', a', L \rangle$ , gdzie  $a \neq B$ , dajemy do  $\Pi$  parę  $\langle aq, q'a' \rangle$
- dla każdej instrukcji z  $\delta$ , która ma postać  $\delta(q, B) = \langle q', a', L \rangle$ , do  $\Pi$  dajemy  $\langle Bq, q'a'B \rangle$  (powiększenie taśmy o białką)
- dla każdej instrukcji z  $\delta$ , która ma postać  $\delta(q, a) = \langle q', a', R \rangle$ ,  $a \neq B$  do  $\Pi$  dajemy  $\langle aqb, a'bq \rangle$  dla każdego  $b \in A$ .
- dla każdej instrukcji z  $\delta$ , która ma postać  $\delta(q, B) = \langle q', a, R \rangle$ , do  $\Pi$  dajemy  $\langle Bq, aBq' \rangle$
- dla każdego  $a \in A$  w  $\Pi$  są takie pary:  $\langle a\text{☹}, \text{☹} \rangle, \langle \text{☹}a, \text{☹} \rangle$

• dodajemy do  $\Pi$  parę  $\langle q_F, \text{☹} \rangle$

Teraz  $V = \text{☹}$ .

kod: konfiguracje  $M_n \rightarrow$  słowo nad  $\Sigma$



tu są słowa z  $q_F$ , z nich wpada wszystko w porządku.

D-d. (tw. o nierozstrzygalności problemu Thue)  
 $\{ \langle w, v, \Pi \rangle : w \stackrel{*}{\leftrightarrow}_{\Pi} v \}$

W zasadzie to samo, co poprzednio.

Trzeba tylko pokazać, że gdy  $n \notin K$ , to  $w \not\stackrel{*}{\leftrightarrow}_{\Pi} v$   
 Widać z tego obrazka wyżej!

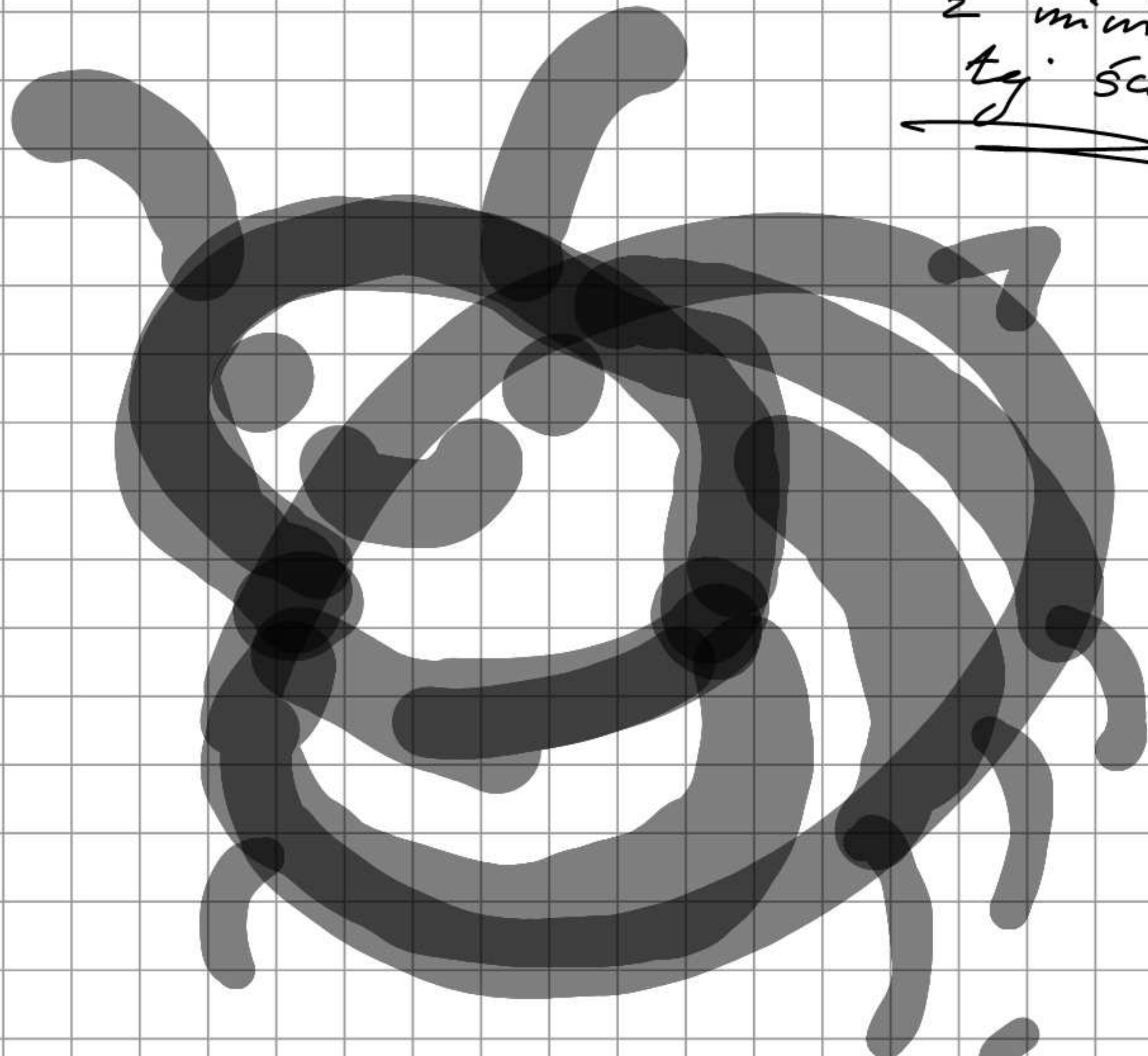
W°



gdyby tak było, to to musi być ten

sen wiechotek

⇓  
sprzeczność z minimalnością tej ścieżki.



PSZCZÓŁKA



# DWUKIERUNKOWE AUTOMATY SKOŃCZONE



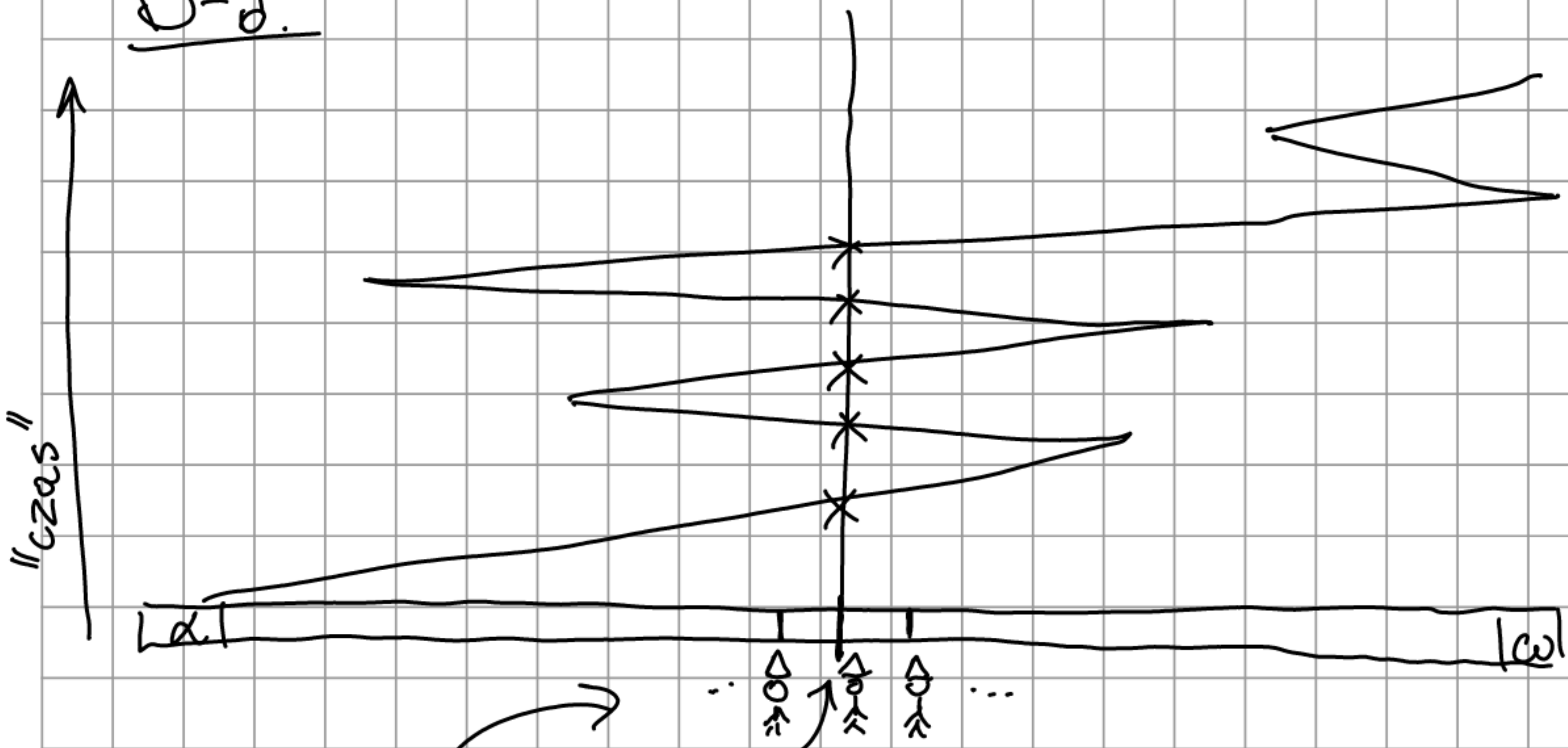
$Q$ : zbiór stanów,  $q_0, q_f \in Q$

$\Sigma' = \Sigma \cup \{\alpha, \omega\}$

$\delta: Q \times \Sigma' \rightarrow Q \times \{L, R\}$ , musi kończyć w  $\omega$

T.W. To model wyraża tylko języki regularne  
(Chcemy z pszczołki zrobić zuzczka)  
(NFA)

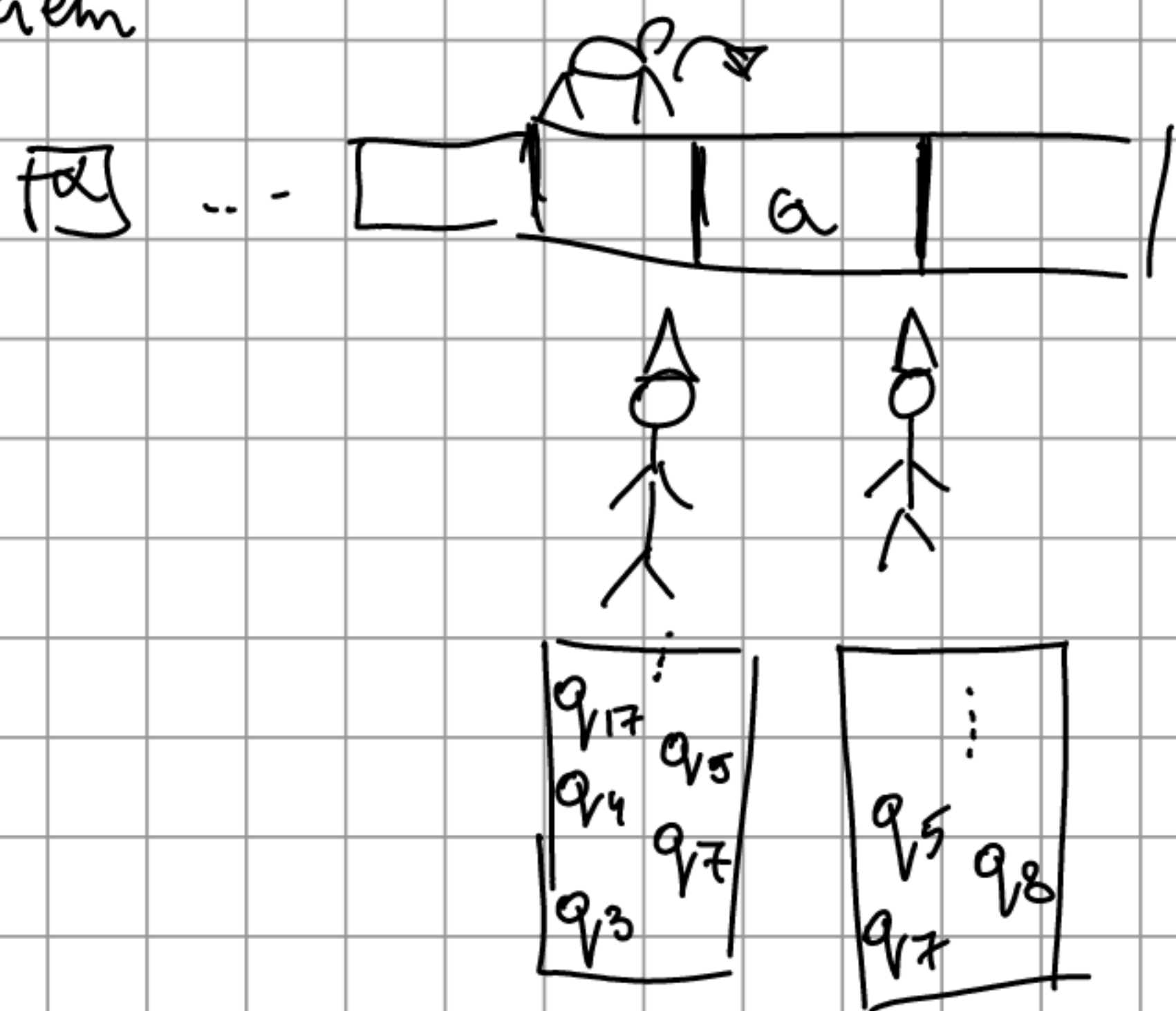
D-d.



w przejściach między komórkami jest krasno linie  
punktów precyzji jest  $\max. 2/|Q|$



Każdy krasnoludek sporządza listę stanów,  
z których pszczołka przeletywała nad krasnolud-  
kiem



sprawdza,  
czy te listy są kompatybilne

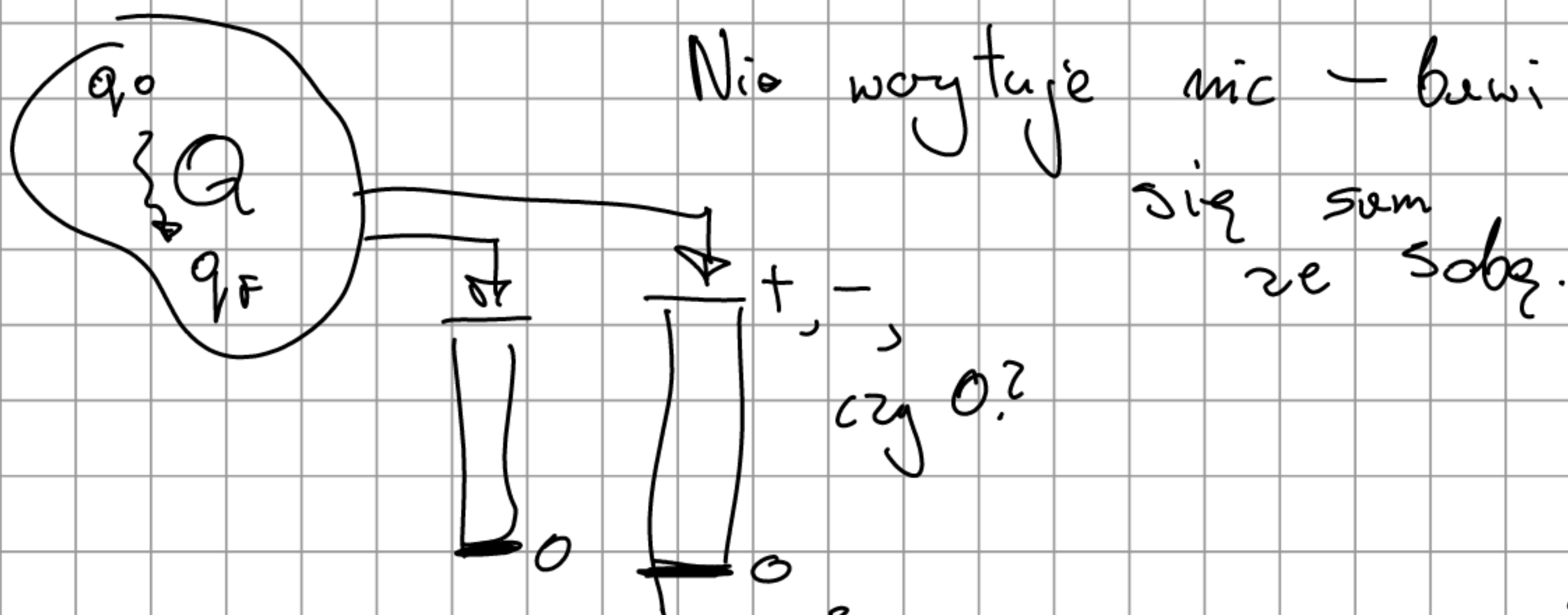
Fakt Pszczołka nie istnieje: są tylko  
krasnoludki powiadające bajki

Obserwacja Niekonsekwentna pszczołka jest

także sama.

9.05.2022 NIEROZSTRZYGALNOŚĆ ARYTMETYKI

Maszyny Minsky'ego :  $\langle Q, q_0, q_f, \delta \rangle$



$$\delta: Q \times \{\text{zero}, \text{nie-zero}\}^2 \rightarrow Q \times \{+1, 0, -1\}^2$$

TW Problem rozstrzygnięcia czy dane MM zakończy działanie jest nierozstrzygalny. (Problem MM)

• Logika I rzędu w  $+$ ,  $\cdot$ ,  $\uparrow$

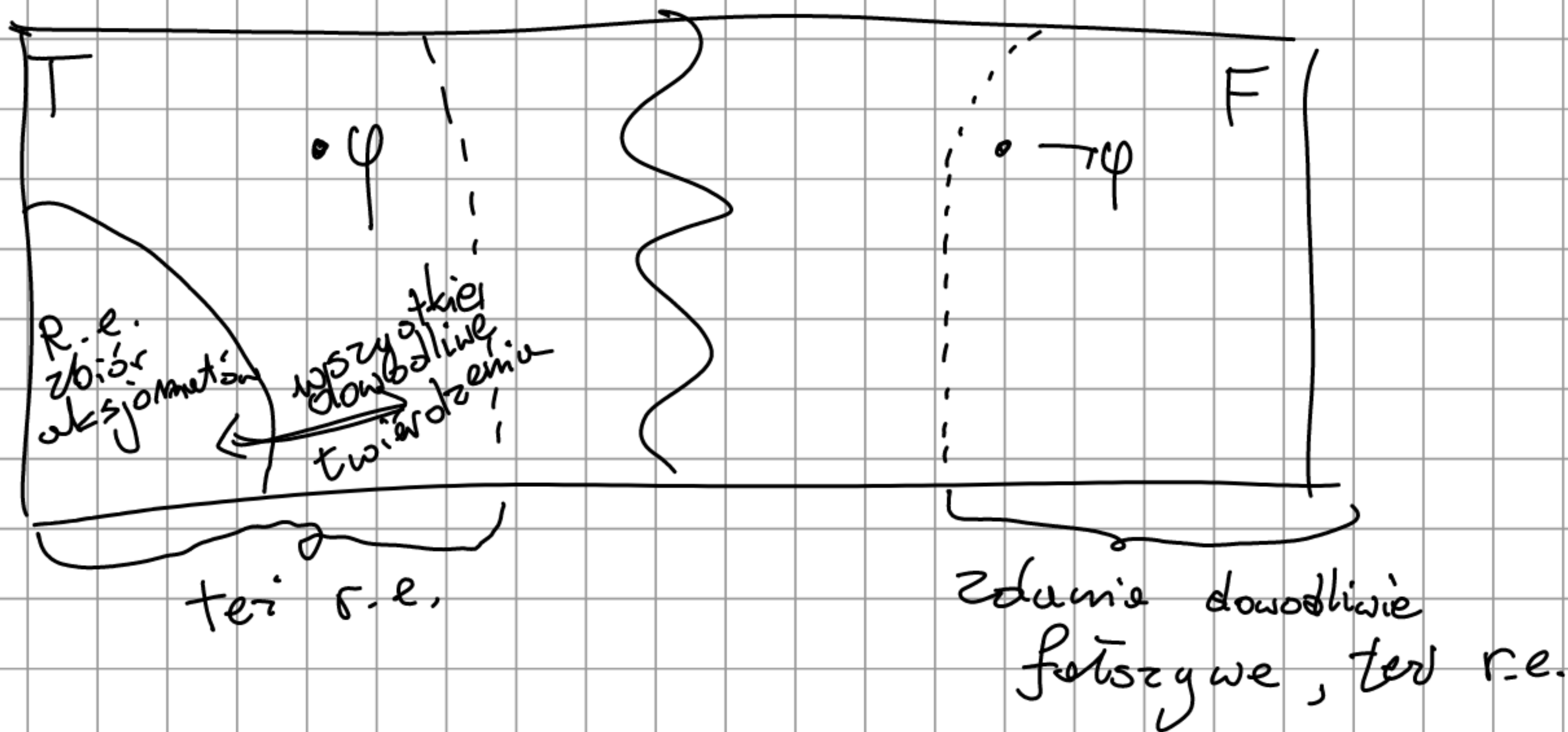
$$\forall m \exists n \forall k, l (m > n \wedge (k \cdot l = m \rightarrow (k=1 \vee l=1)))$$

• Aksjomaty arytmetyki Peano  $\langle \mathbb{N}, +, \cdot \rangle$

Pytanie: czy każde prawdziwe zdanie  $\text{Th}(\mathbb{N}, +, \cdot)$  jest dowodliwe w arytmetyce Peano?

ODP: NIE.

# WSZYSTKIE ZDANIA ARYTMETYKI



Gdyby wszystkie zdania arytmetyki  
 dało się udowodnić, to  $\text{Th}(\mathbb{N}, +, \cdot)$

byłoby rozstrzygalny.

Tw.  $\text{Th}(\mathbb{N}, +, \cdot, \uparrow)$  nie jest rozstrzygalny.

D-d. Pokażemy, że  $\text{MM} \leq_{\text{rek}} \text{Th}(\mathbb{N}, +, \cdot, \uparrow)$

Czyli chcemy nam program  $M$  dla  $\text{MM}$ .

Mamy napisać zdanie  $\Phi_M$  t.że

$$M \text{ się zatrzymuje} \Leftrightarrow (\mathbb{N}, +, \cdot, \uparrow) \models \Phi_M$$

Potrzebujemy zapamiętać konfigurację  $M$   
 jako trójkę liczb.

Niech  $k = |Q|$ ,  $Q = \{q_1, \dots, q_k\}$  gdzie

$q_1 = q_0$ ,  $q_2 = q_k$ . Konfiguracja  $C$ : Maszyna  
jest w stanie  $q_i$ , a na licznikach  
jest  $m_1$  oraz  $m_2$ , będzie oznaczona  
przez trójkę  $t(C) = \langle i, k+1+m_1, k+1+m_2 \rangle$

Def. Fajna szóstka to szóstka liczb

$\langle a_1, a_2, a_3, a_4, a_5, a_6 \rangle$  która:

•  $a_1 > k$  lub

•  $a_1 \leq k$ ;  $a_2, a_3 > k$ ;  $a_4 \leq k$ ;  $a_5, a_6 > k$

oraz istnieją konfiguracje  $C$  oraz  $C'$   
maszyny  $M$  t.ż.  $C'$  wynika z

$C$  w jednym kroku obrotu  $M$

oraz że  $t(C) = \langle a_1, a_2, a_3 \rangle$  i

$t(C') = \langle a_4, a_5, a_6 \rangle$



Def Ciąg  $a_1, \dots, a_N$  jest fajny gdy

•  $\langle a_1, a_2, a_3 \rangle = t(C_0)$ , początkowa konfiguracja  $M$

•  $\langle a_{n-2}, a_{n-1}, a_n \rangle = t(C_F)$ , końcowa konfiguracja  $M$

•  $\langle a_l, a_{l+1}, \dots, a_{l+5} \rangle$  jest fajny szóstki

dla  $1 \leq l \leq N-5$

Obserwacja Istnieje formuła arytmetyki

$\psi$  z 6 zm. wolnymi tak, że

$(\mathbb{N}, +, \cdot, \uparrow) \models \psi(x_1, \dots, x_6)$  w.t.w. gdy

$x_1, \dots, x_6$  są fajne.

sprawdź czy  $x_2$  jest w opisanym stanie przez  $j$

$\psi(x_1, \dots, x_6) : x_1 > k \vee$

$$\bigwedge_{\substack{i \in \{1, \dots, k\} \\ j, j' \in \{0, 1\}}} \left[ (x_1 = i \wedge \chi(x_2, j) \wedge \chi(x_3, j)) \rightarrow \right. \\ \left. x_4 = \Pi_1(\delta(q_{i,j,j'})) \wedge x_5 = x_2 + \Pi_2(\delta(q_{i,j,j'})) \right. \\ \left. \wedge x_6 = x_3 + \Pi_3(\delta(q_{i,j,j'})) \right]$$

gdzie te  $\Pi_1, \Pi_2, \Pi_3$  działają jak trzeba

Teraz chcemy formułę, która sprawdza czy ciąg jest fajny. Musimy kodować ciąg jakoby:  $a_1, \dots, a_n \rightarrow 2^{a_1} 3^{a_2} \dots p_n^{a_n}$

"Makra":

- pierwsza ( $x$ ):  $\forall y, z (yz = x \rightarrow y = 1 \vee z = 1) \wedge x > 1$

- kolejne\_pierwsze ( $x, y$ ):  $\forall z (x < z < y \rightarrow \neg \text{pierwsza}(z)) \wedge \text{pierwsza}(x) \wedge \text{pierwsza}(y) \wedge x < y$

- w\_rozkladzie ( $x, y, z$ ):  $\exists m x^y m = z \wedge \forall m x^{y+1} m \neq z \wedge \text{pierwsza}(x)$

- największa\_pierwsza ( $x, y$ ):  $\exists s \forall z, t \text{ pierwsza}(x) \wedge sx = y \wedge (\text{pierwsza}(z) \wedge z > x \rightarrow zt \neq y)$

$$\Phi_m: \exists m \forall x, x', y, y' (\text{kolejne\_pierwsze}(x, x') \wedge \text{w\_rozkladzie}(x, y, m) \wedge y' > 0 \wedge \text{w\_rozkladzie}(x', y', m) \rightarrow y > 0)$$

$\wedge$  w-rozkładzie  $(2, 1, m)$   $\wedge$  w-rozkładzie  $(3, k+1, m)$

$\wedge$  w-rozkładzie  $(5, k+1, m)$

$\wedge \forall p, p', p''$  (najm. pierwsza  $(p'', m)$   $\wedge$  kolejne pierwsze  $(p, p')$ )

$\wedge$  kolejne pierwsze  $(p', p'') \rightarrow$  w-rozkładzie  $(p, 2, m)$

$\wedge$  w-rozkładzie  $(p', k+1, m)$   $\wedge$  w-rozkładzie  $(p'', k+1, m)$

$\wedge \forall x_1, x_2, \dots, x_6, x$   $\left[ \begin{array}{l} \bigwedge_{i=1}^5 \text{kolejne pierwsze } (x_i, x_{i+1}) \wedge \text{ najm. pierwsza } (x, m) \\ \bigwedge_{i=1}^6 \text{ w-rozkładzie } (x_i, y_i, m) \rightarrow \psi(y_1, \dots, y_6) \end{array} \right]$



## X PROBLEM MILBERTA

Czy jest algorytm rozwiązujący układ  
równań diofantycznych?

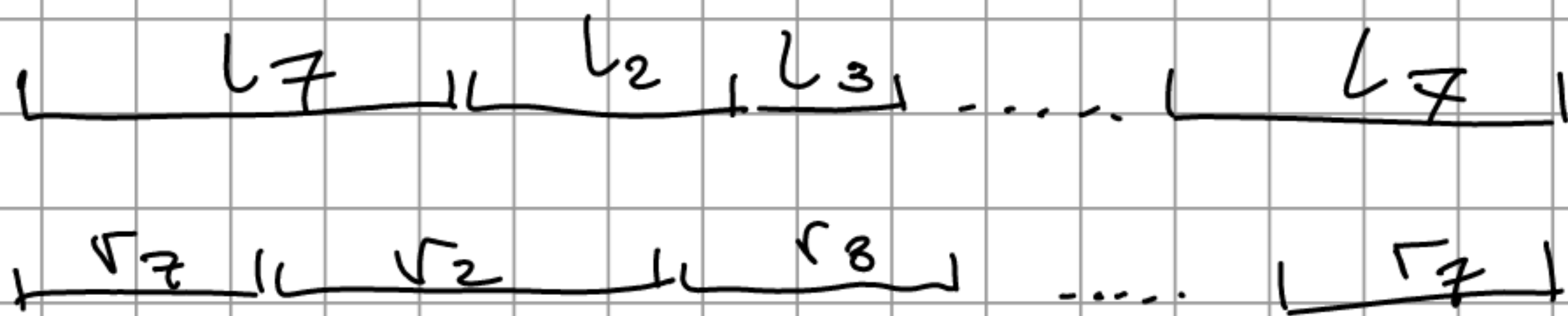
ODP: Nie, to jest nierozstrzygalne (trudne)



# 16.05.2022 PROBLEM ODPOWIEDNOŚCI POSTA (PCP)

Instancją jest skończony zbiór par słów  $\{ \langle l_i, r_i \rangle, i \in \{1, \dots, k\}, l_i, r_i \in \Sigma^* \}$  nad pewnym alfabetem  $\Sigma$ .

Czy istnieje  $s \in \{1, \dots, k\}^*$  t.j.e  $s \neq \epsilon$  oraz  $l_{s_1} l_{s_2} \dots l_{s_{|s|}} = r_{s_1} r_{s_2} \dots r_{s_{|s|}}$  ?



Tw. Problem odpowiedności Posta jest nierozstrzygalny.

D-d. I przymiarka dowodu:  $\text{SemiThuc}_{\text{red}} \leq_{\text{red}} \text{PCP}$ .

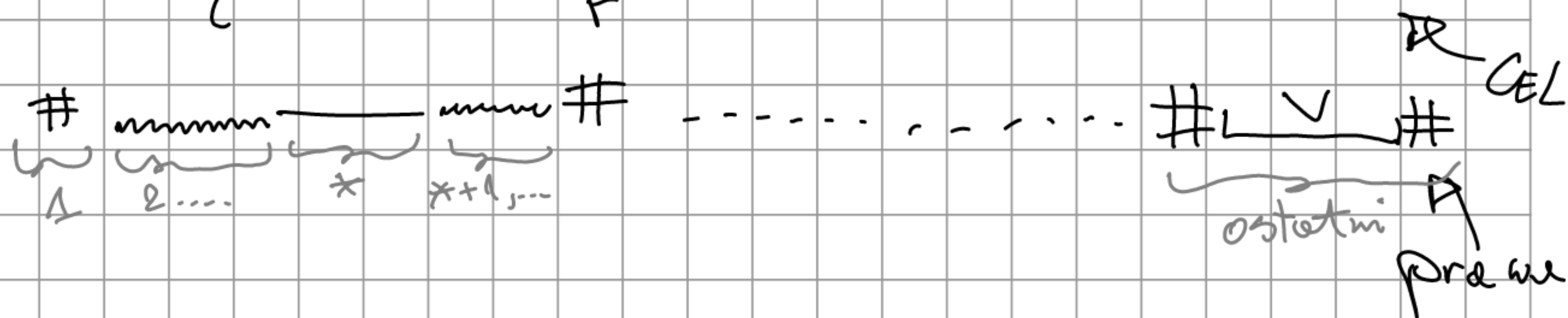
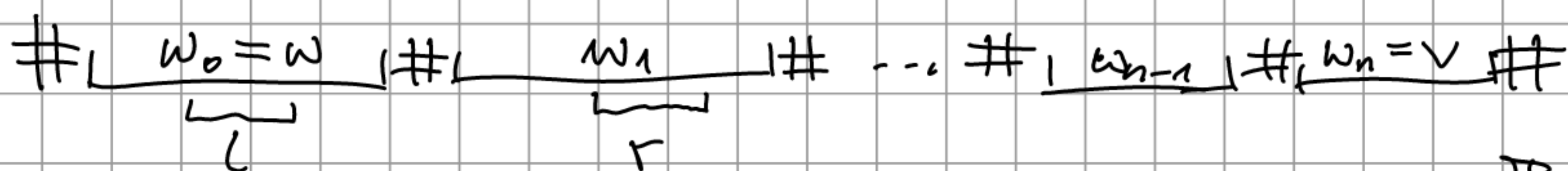
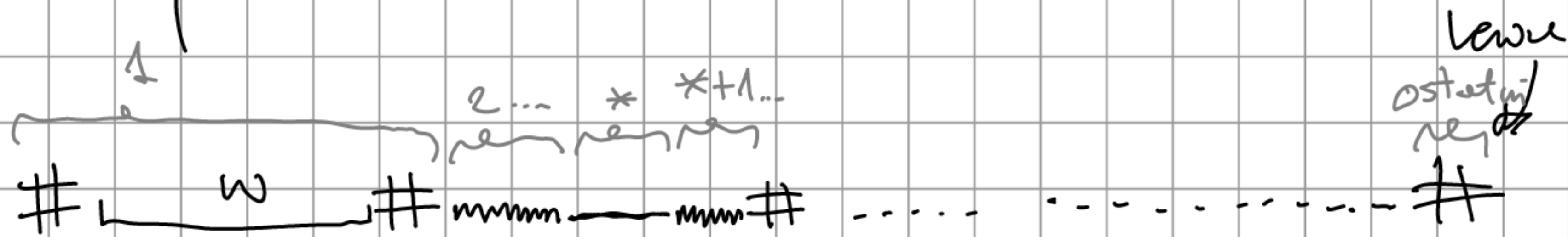
Daję nam  $(w, v, \Pi)$ . Mamy wyprodukować instancję PCP  $P$  t.j.e  $w \xrightarrow{*} v \iff P$  ma rozwiązanie.

Niech  $A$ : alfabet  $\Pi$ . Wtedy  $\Sigma = A \cup \{ \# \}$ .

$P = \{ \langle \# w \#, \# \rangle, \langle \#, \# v \# \rangle \} \cup \Pi^R \cup \{ \langle a, a \rangle : a \in \Sigma \}$



Takie Prawsze działa, ale pokazany  
tylko  $\Rightarrow$  w "fajny i zadowolający"  
sposób.



II podejście:  $\Sigma = \{\#, \bar{\#}\} \cup \{a, \bar{a} : a \in A\}$ .

$$P = \{ \langle \#, \# w \bar{\#} \rangle, \langle \# v \bar{\#}, \bar{\#} \rangle \}$$

$$\cup \{ \langle \bar{l}, r \rangle, \langle l, \bar{r} \rangle : \langle l, r \rangle \in \Pi \}$$

$$\cup \{ \langle \#, \bar{\#} \rangle, \langle \bar{\#}, \# \rangle \} \cup \{ \langle a, \bar{a} \rangle, \langle \bar{a}, a \rangle : a \in A \}.$$

Wtedy:

$\# \underbrace{w_1}_{1} \underbrace{w_2}_{2 \dots} \underbrace{w_3}_{*} \underbrace{w_4}_{*+1 \dots} \# \dots \dots \dots \underbrace{\# w_n \#}_{\text{ostatni}}$

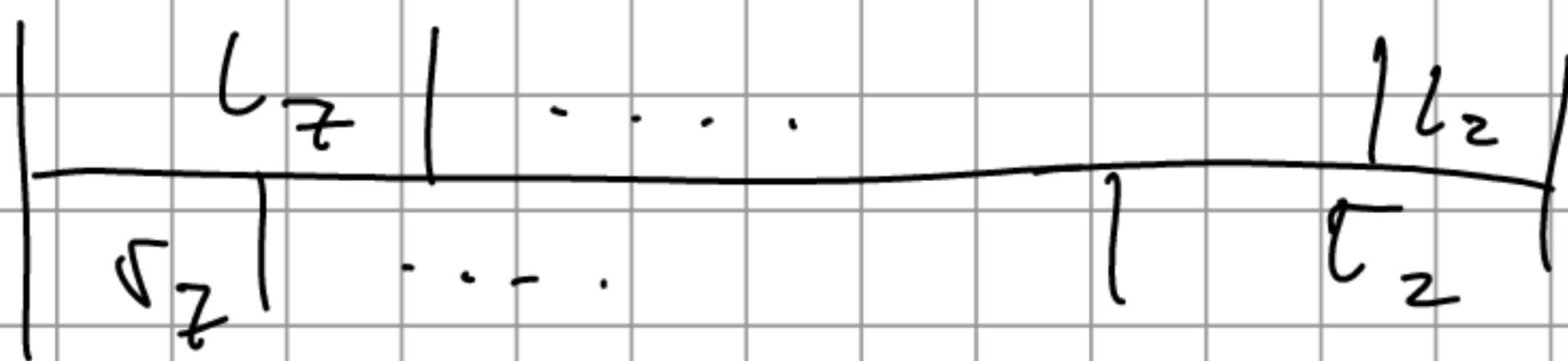
$\# \underbrace{w_0 = w}_1 \# \underbrace{w_1}_2 \# \underbrace{w_2}_3 \# \dots \# \underbrace{w_{n-1}}_{n-1} \# \underbrace{w_n = v}_n \#$

$\# \underbrace{w}_1 \# \underbrace{w_1}_{2 \dots} \underbrace{w_2}_{*} \underbrace{w_3}_{*+1 \dots} \# \dots \dots \dots \# \underbrace{w_n}_{\text{ostatni}}$

(jeśli parzystość się nie zgadza, to może  
 jakieś słowo zdublować)

wtedy  $\Rightarrow$  prosto. Teraz  $\Leftarrow$ :

$\{ \langle l_i, r_i \rangle : i \in \{1, \dots, k\} \}$  : pewna instancja PCP.



$r_1$  jest prefiksem  $l_2$   
 $l_1$  jest sufixem  $r_2$

Zatem musimy zacząć od  $\langle \#, \#w\# \rangle$   
 i skończyć na  $\langle \#v\#, \# \rangle$ .

$n_1, \dots, n_k, \dots, n_l$

$\# \underbrace{w}_{\substack{\text{---} \\ \text{---} \\ \text{---}}} \# \overset{\text{---}}{\underset{\text{---}}{\alpha}} \# \quad \hookrightarrow \quad \# \dots \# \overset{\text{---}}{\underset{\text{---}}{v}} \#$

Lemat  $\alpha \xrightarrow{*} \Pi \gamma$

D-d. Na powrót:  $w \xrightarrow{*} \Pi x$ . Łatwe.

Dalej indukcyjne i elo!

z dobrymi założeniami.

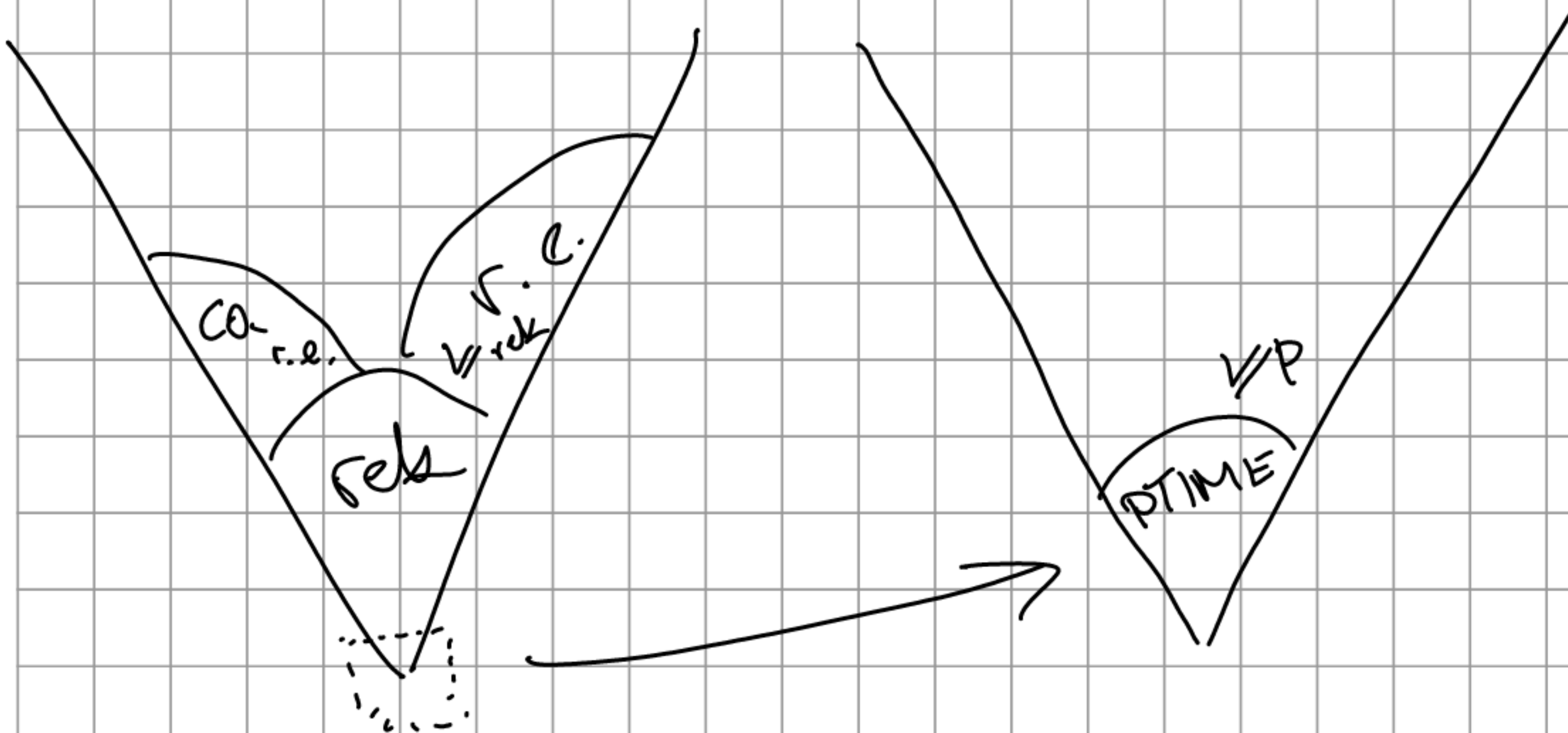


Koniec teorii 2.

# III CZĘŚĆ KURSU

Def.  $A \subseteq \Sigma^*$  jest w **P**TIME (dla przyjęcia w "P") gdy istnieje maszyna Turinga  $M$  oraz wielomian  $p$  takie, że:

- $M$  rozstrzyga  $A$
- dla każdego  $x \in \Sigma^*$   $M$  uruchomiona na  $x$  zatrzymuje się po najwyżej  $p(|x|)$  krokach.





$\Sigma_A^*$        $\Sigma_B^*$   
 $U$              $U$

Def.  $A \leq_p B$  (czytamy: „A jest mierniejsze od B w sensie redukcji wielomierowych”)

gdzie istnieje  $f: \Sigma_A^* \rightarrow \Sigma_B^*$  o następujących własnościach:

zwana redukcją wielomierową

- istnieje wielomian  $p$  oraz MT  $M$  t.z.e
   
 $\forall a \in \Sigma_A^* \quad M(a)$  zatrzymuje się po
   
 $\leq p(|a|)$  krokach i zwraca
   
 wartość  $f(a)$
- $\forall a \in \Sigma_A^* \quad (a \in A \iff f(a) \in B)$

Obserwacja Jeśli  $A \leq_p B$  i  $B \in \text{PTIME}$ ,  
 to  $A \in \text{PTIME}$ .

Obserwacja Jeśli  $A, B \in \text{PTIME}$  i obu są mierniejsze, to  $A \leq_p B$  oraz  $B \leq_p A$ .

Problem 3-SAT Instancją jest formuła rachunku zdań postaci 3CNF (koniekcje klanul o co najwyżej 3 literałach).

Pytanie czy formuła jest spełnialna.

Problem 3-kolorowania Graf nieskierowany.

Pytanie czy da się pokolorować wierzchołki grafu tak żeby żadne dwa sąsiednie były różnego koloru.

Tw.  $3\text{COL} \leq_p 3\text{SAT}$ .


D-d. Daję mi graf  $G = \langle V, E \rangle$ .

Mamy SZYBKO napisać formułę  $\psi_G$  w postaci 3CNF t.ż.  $G$  da się pokolorować  $\iff \psi_G$  jest spełnialna.

Konstrukcja  $\psi_G$ : zmiennymi będą

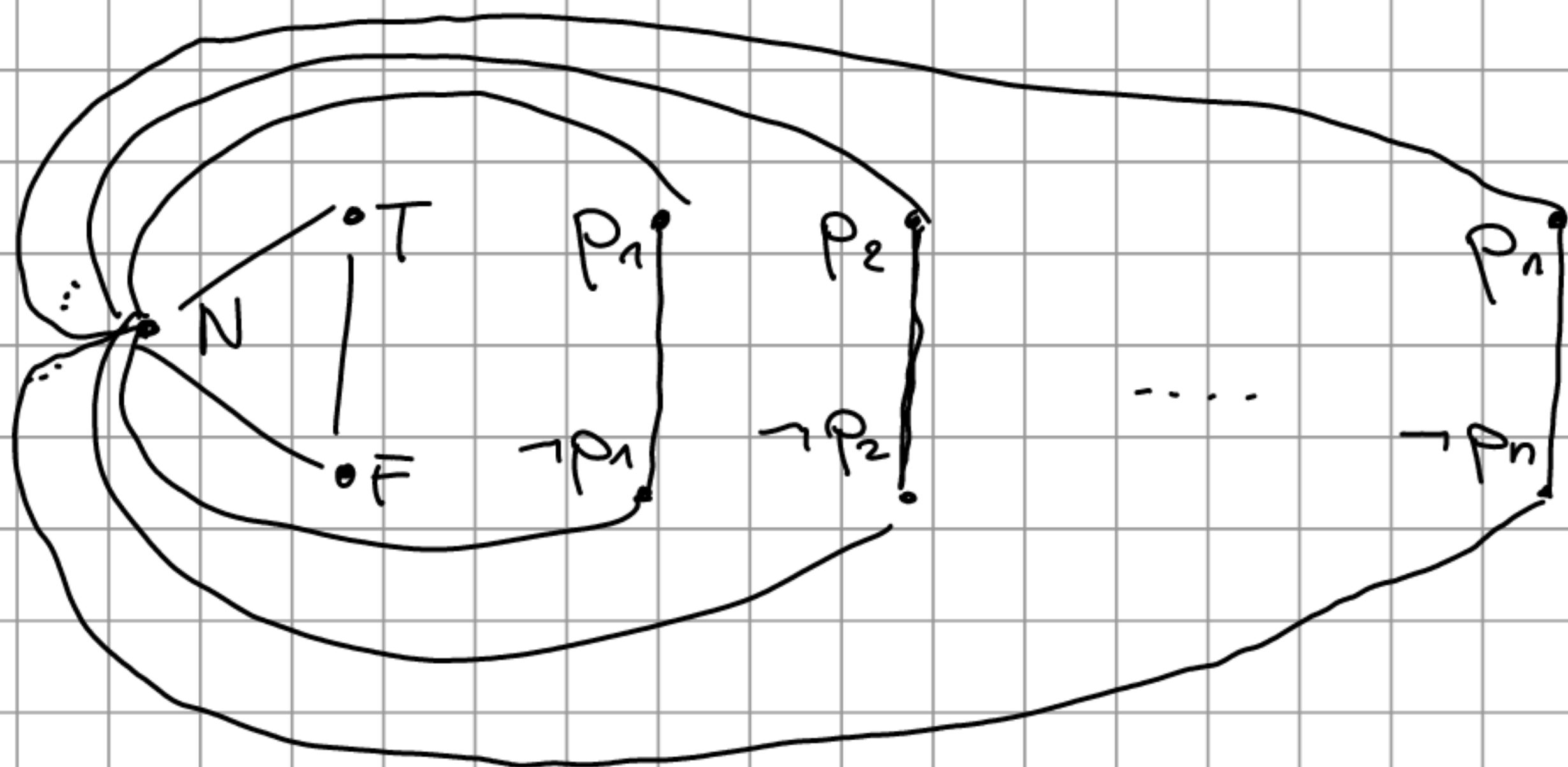
$r_u, g_u, b_u$  dla  $u \in V$ . Klauzule:

•  $\forall u \in V (r_u \vee g_u \vee b_u)$

•  $\forall (u, w) \in E (\neg r_u \vee \neg r_w), (\neg g_u \vee \neg g_w),$   
 $(\neg b_u \vee \neg b_w)$  

Tw  $3SAT \leq_p 3COL$ .

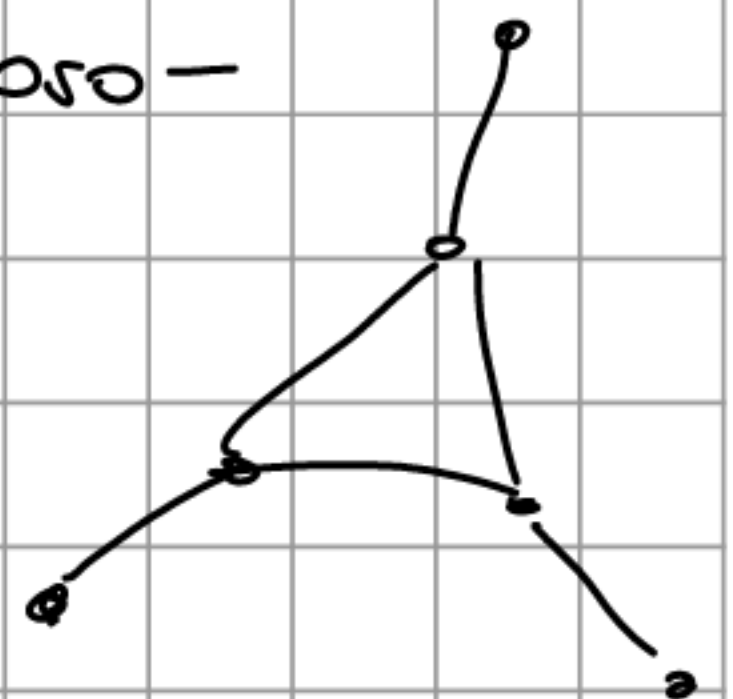
D-d. Daję nam formułę  $\varphi$  w postaci 3CNF o zmiennych  $p_1, \dots, p_n$ . Mamy szybko zbudować  $G_\varphi = (V, E)$  t.j.e  $G_\varphi$  jest 3-kolorowalny  $\iff \varphi$  jest spełniony.



Gadzet 3 wierzchołki zewnętrzne i 3 wewnętrzne.

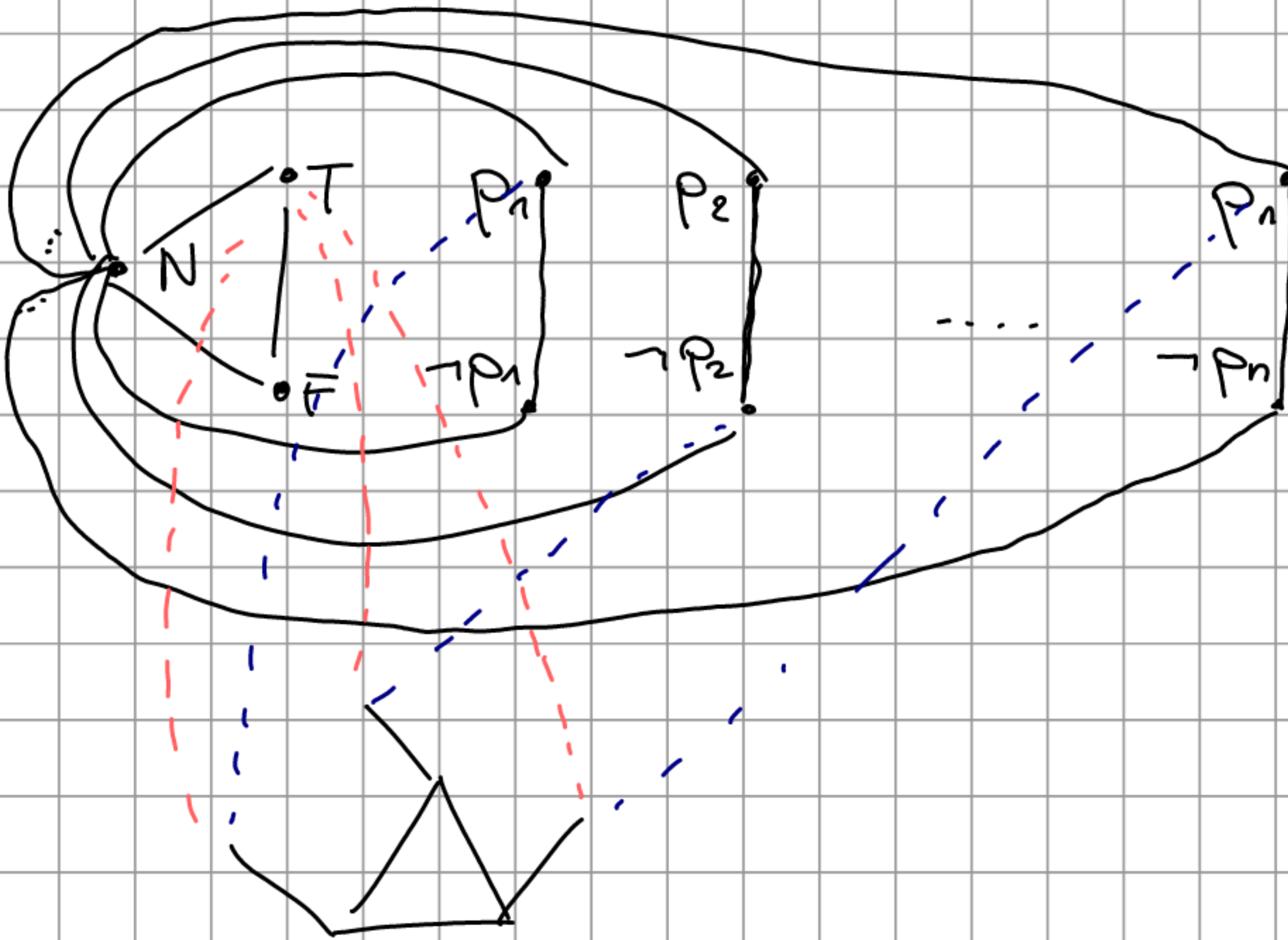
Obserwacja: kol: Zewn  $\rightarrow \{N, F, T\}$

rozszerz się do poprawnego koloro-  
waniem całości, gdy nie jest  
stata.



D-d. prosty.

Podanie grafu dla klanu  $p_1 \vee \dots \vee p_n$ :



No i to tyle ...



23.05.2022

## KLASA NP I TWIERDZENIE COOKA

Rozważamy <sup>wielomianową</sup> niedeterministyczną maszynę Turinga.

Teraz  $\delta \subseteq (\Sigma \times Q) \times (\Sigma \times Q \times \Gamma \cup \{L, R\})$

"Wielomianowość" oznacza, że maszynę przychodzi z wielomianem  $p$  oraz gwarantuje, że jeżeli istnieje ścieżka akceptująca dla słowa  $w$ , to istnieje ścieżka akceptująca długości  $p(|w|)$ .

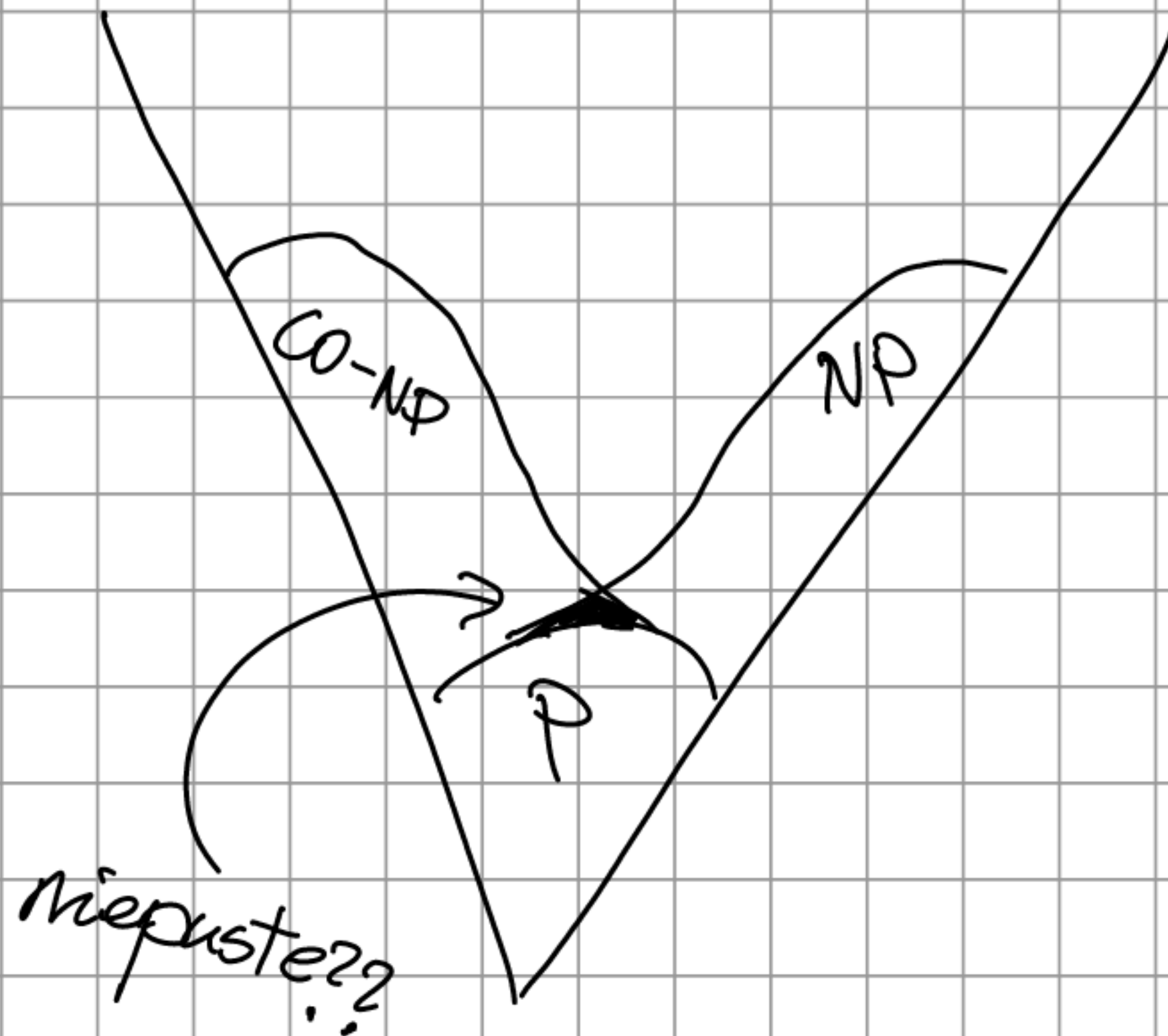
Def  $A \in NP$  gdy istnieje niedeterministyczna wielomianowa maszyna Turinga rozstrzygająca  $A$ .

Przykład  $3COL \in NP$ . Można zgadywać dobry wierzchołek i sprawdzić, czy jest poprawne.

MICHAŁ TO PAŁA

Observacja  $A \leq_p B \wedge B \in NP \Rightarrow A \in NP$

Pytanie Sprawdzenie czy formuła zdaniowa  
jest tautologią jest CO-NP

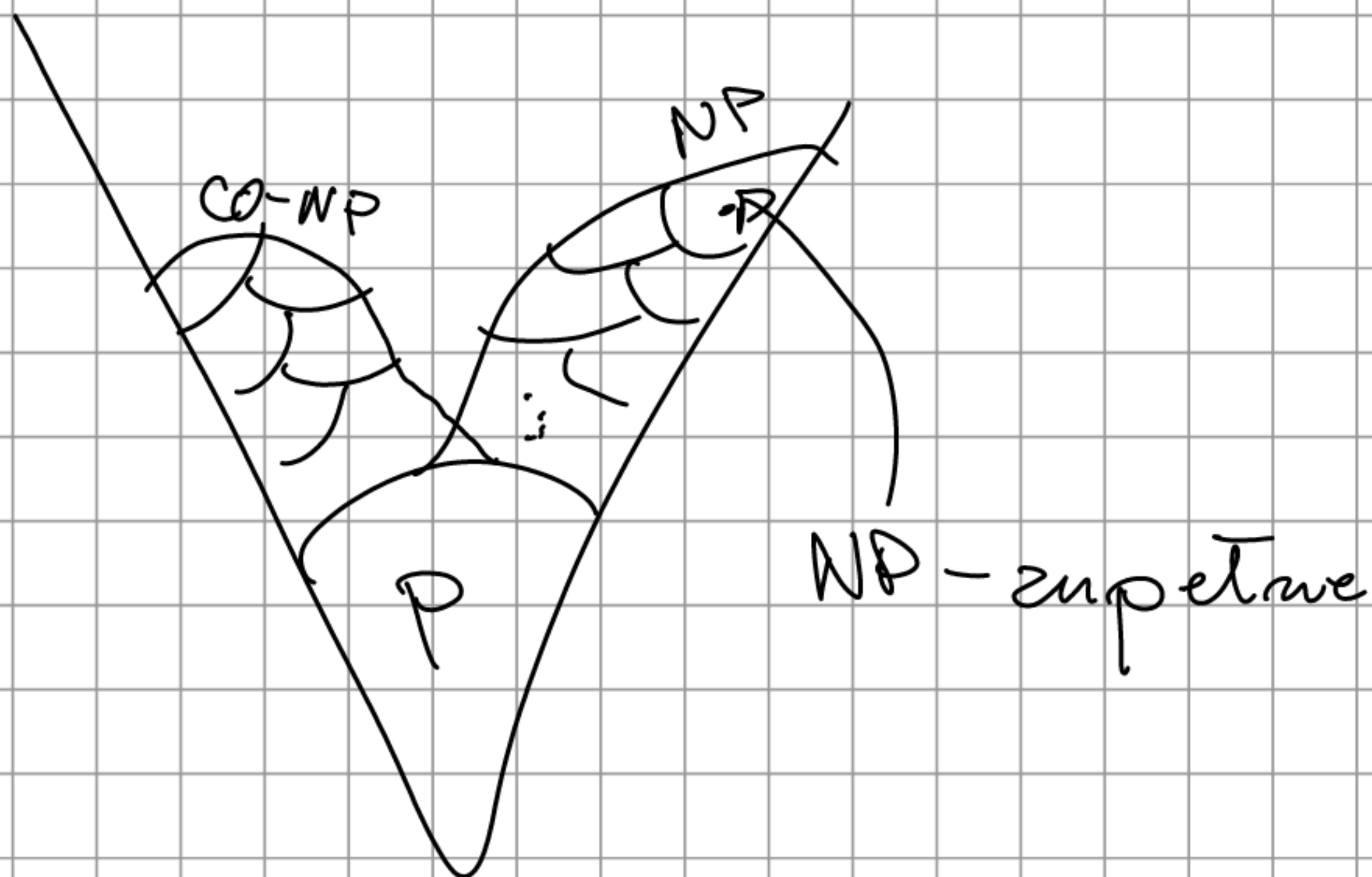


Nie wiemy, czy  $P = NP$  (ludzie raczej  
myślą, że  $NP \neq P$ )

Nie wiemy, czy  $NP \cap CO-NP = P$   
(nawet, jeśli założymy, że  $P \neq NP$ )

Od teraz zakładamy, że  $P \neq NP$ .

Wtedy w NP są "baniki":



Def.  $A$  jest NP-trudny jeżeli dla każdego  $B \in NP$   $B \leq_p A$ .

$A$  jest NP-zupełny gdy  $A \in NP$  oraz  $A$  jest NP-trudny.

Obserwacja "NP to aut P ma pierwszą oś".

Dokładniej:  $\forall A \in NP \exists B \in P \exists p(x)$

$A = \{ x : \exists y (|y| \leq p(|x|) \wedge [x, y] \in B) \}$

TW (Cooka) 3SAT jest NP-zupełny.

D-d. Niech  $A \in NP$ . Istnieją wielomiany  $p, q$  oraz deterministyczne MT  $M_B$  działające

w czasie  $q(|w|)$  t.że  $\ast(w)$

$$\forall w \ w \in A \Leftrightarrow \exists y (|y| \leq p(|w|) \wedge M_B(w, y) \text{ akceptuje})$$

Konstrukcja redukcji  $A \leq_p 3SAT$ :

- dając  $w$ : instancję problemu  $A$ . Mamy szybko zwrócić formułę  $\varphi_w \in 3CNF$  t.że

$$\ast(w) \Leftrightarrow \varphi_w \in 3SAT$$

- W każdym polu  $i, j$  poniżej tabelki mieszczą zmienne  $d_{i,j}, w_{i,j}, 0_{i,j}, 1_{i,j}, B_{i,j}$  oraz  $\forall q \in Q_{M_B}$  zmienna  $q_{i,j}$  oraz  $L_{i,j}, R_{i,j}$

- Piszemy formułę  $Conf(i)$  będzie koniunkcją mnóstwa klauzul:

- dla  $\beta \neq \gamma \in \Sigma$  oraz dla  $0 \leq j \leq q(|w|)$

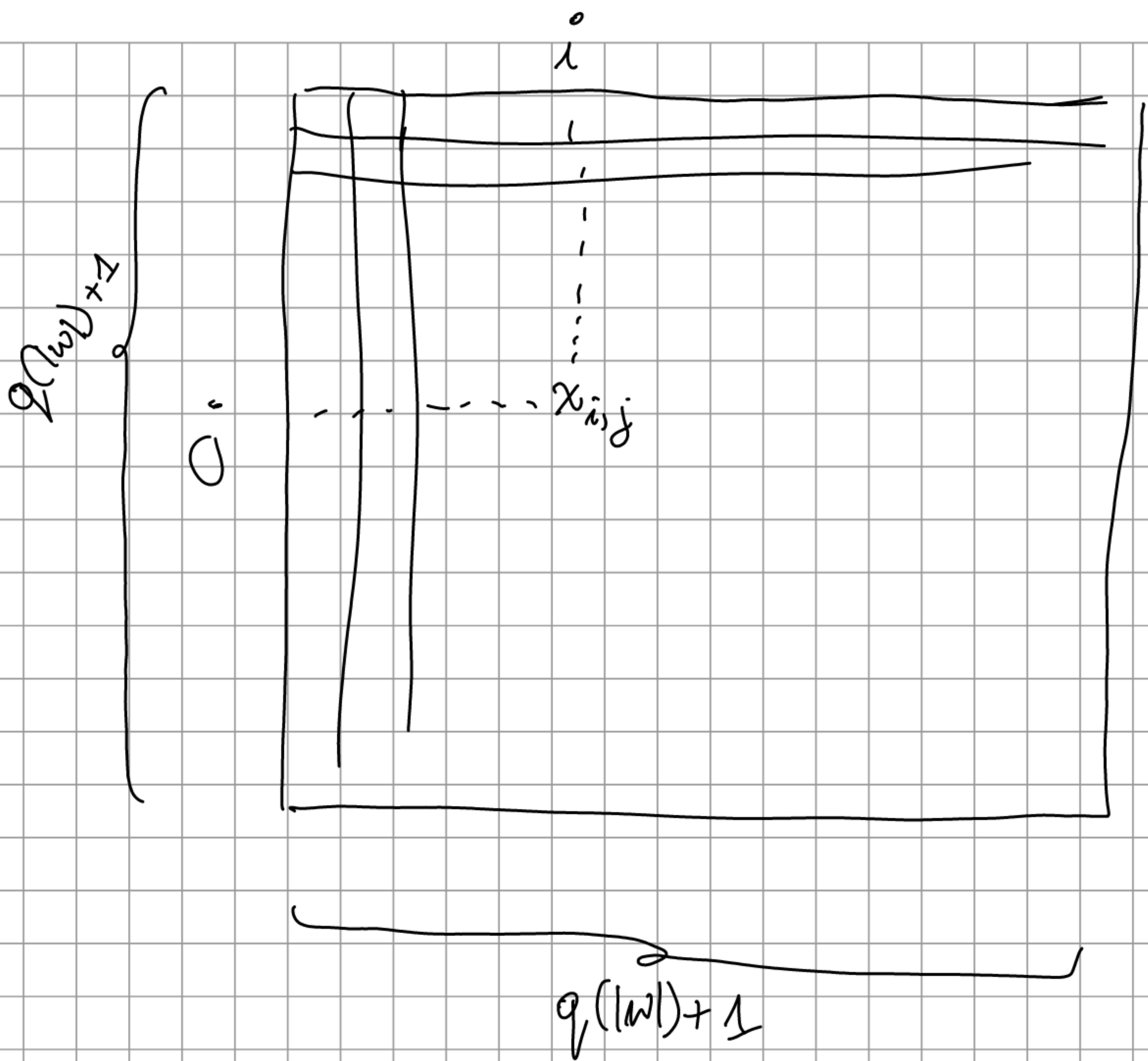
w  $Conf(i)$  będzie klauzula  $(\neg \beta_{i,j} \vee \neg \gamma_{i,j})$

- dla każdego  $0 \leq j \leq q(|w|)$  i każdych

$\beta \neq \gamma \in Q_{M_B} \cup \{L, R\}$  w  $Conf(i)$

będzie  $(\neg \beta_{i,j} \vee \neg \gamma_{i,j})$





- dla każdego  $j$  oraz  $q \in \mathcal{Q}_{M_B} \hookrightarrow \text{Conf}(i)$   
 będącej krawędzią

$$\begin{aligned}
 & L_{i,j} \rightarrow L_{i,j-1}, \quad R_{i,j} \rightarrow R_{i,j+1}, \\
 & q_{i,j} \rightarrow R_{i,j+1}, \quad q_{i,j} \rightarrow L_{i,j-1}
 \end{aligned}
 \left( \begin{array}{l} a \rightarrow b \\ \Leftrightarrow \\ \neg a \vee b \end{array} \right)$$

• Formuła Pierwszy( $i$ ) składa się z

klauzul  $q_{i,0}^0, \alpha_{i,0}$ , dla każdego  $1 \leq j \leq |w|$   $\underbrace{w[j]}_{0 \text{ albo } 1} i_{i,j}$ ,  $\omega_{i,|w|+1}$

dla każdego  $|w|+2 \leq j \leq |w|+1+p(|w|)$

$(0_{i,j} \vee 1_{i,j})$ , dla każdego  $|w|+1+p(|w|) < j \leq q(|w|)$   
 $B_{i,j}$

• Formuła Ostatni( $i$ ) składa się

z klauzuli  $q_{i,0}^F$

• Formuła Krok( $i$ ) składa się z

klauzul:

- dla każdego  $j$ , każdego  $p \in \Sigma$

mamy klauzulę  $L_{i,j} \wedge \beta_{i,j} \rightarrow \beta_{i+1,j}$

$R_{i,j} \wedge \beta_{i,j} \rightarrow \beta_{i+1,j}$

- dla każdej instrukcji w  $\delta_{M_B}$  postaci

$\langle q, \beta, i, q', \beta', L' \rangle$  i dla każdego  $j$

$q_{i,j} \wedge \beta_{i,j} \rightarrow \beta'_{i+1,j} \wedge q'_{i+1,j-1}$

- analogicznie dla instrukcji w prawo

• Wtedy  $\varphi_w = \bigwedge_i \text{Conf}(i) \wedge \text{Pierwszy}(0) \wedge \text{Ostatni}(q(iw))$   
 $\wedge \bigwedge_i \text{Krok}(i)$



30.05.2022

## ○ NIEROZSTRZYGAŁNOŚĆ RACHUNKU PREDYKATÓW

$$\forall x \exists y (E(x, y) \wedge \dots \wedge \dots)$$

formuła logiki I rzędu

Pytania:

a) czy jest spełniona?

b) czy jest tautologią?

1) dopuszczamy dowolne struktury

2) tylko skończone formuły

2a)  $\leadsto$  R.E.

2b)  $\leadsto$  CO-R.E.

Przykład

$$(\exists x \forall z \neg E(z, x)) \wedge (\forall x \exists y E(x, y)) \wedge$$

$$(\forall x, y, y' E(y, x) \wedge E(y', x) \rightarrow y = y')$$

↑ Ma model nieskończony, nie ma skończonego.



Tw. Problem 1b jest nierozstrzygalny

D-a. Pokażemy  $MM \leq_{red} \forall \text{AUT-FOL}$   
↑  
problem stopu maszyny  
Turinga

Dając nam maszynę  $M$ , ona ma stany  
 $Q_M$ , w tym  $q_0, q_F$  i  $\delta_M$ .

Napiżemy formułę  $\varphi_M$  t.ze  $\varphi_M$  jest tautologią  
 $\Leftrightarrow M$  się zatrzymuje.

Będę miał 3 predykaty binarne  $E, P, D$   
oraz tyle predykatów unarnych ile jest  
stanów w  $Q_M$  i może jeszcze jeden:  $Z$ .

$\varphi_M$  będzie koniunkcją następujących formuł:

$$\bullet \forall x \exists y \left[ \left( \bigvee_{R \in Q_M \setminus \{q_F\}} R(x) \right) \rightarrow E(x, y) \right]$$

$$\bullet \exists x \exists z \quad Q_0(x) \wedge P(x, z) \wedge D(x, z) \wedge Z(z)$$

$$\bullet \exists z \forall t \quad Z(z) \wedge \neg P(z, t) \wedge \neg D(z, t) \\ \wedge (Z(t) \rightarrow t = z)$$

• Dla każdej instrukcji  $\tau \in \mathcal{T}_M$  definiujemy formaty. Np. jeżeli  $\mathcal{T}_M$  ma instrukcję postaci  $\langle Q, z, NZ; Q', +1, -1 \rangle$ , to

dodajemy formaty:

$$\forall x, y, t, t', t'' \exists r \left[ (Q(x) \wedge P(x, t) \wedge D(x, t') \wedge Z(t) \wedge \neg Z(t') \wedge D(t', t'')) \right]$$

$$\rightarrow (Q'(y) \wedge D(y, t'') \wedge P(y, r) \wedge P(r, t))$$

Lemat Jeżeli po  $n$  krokach obliczeń  $M$  jest w konfiguracji  $\langle Q, m, m' \rangle$  i struktura  $S$  spełnia  $\varphi_M$ , to istnieje w niej wierzchołek  $x$  t.ze  $Q(x)$  i z  $x$  jest  $P$ -ścieżka do  $Z$  dł.  $m$  i  $D$ -ścieżka długości  $m'$ .

Teraz konstruujemy  $\varphi_M$ .

$$\varphi_M \rightarrow \exists x Q_P(x)$$



## KLASA PSPACE

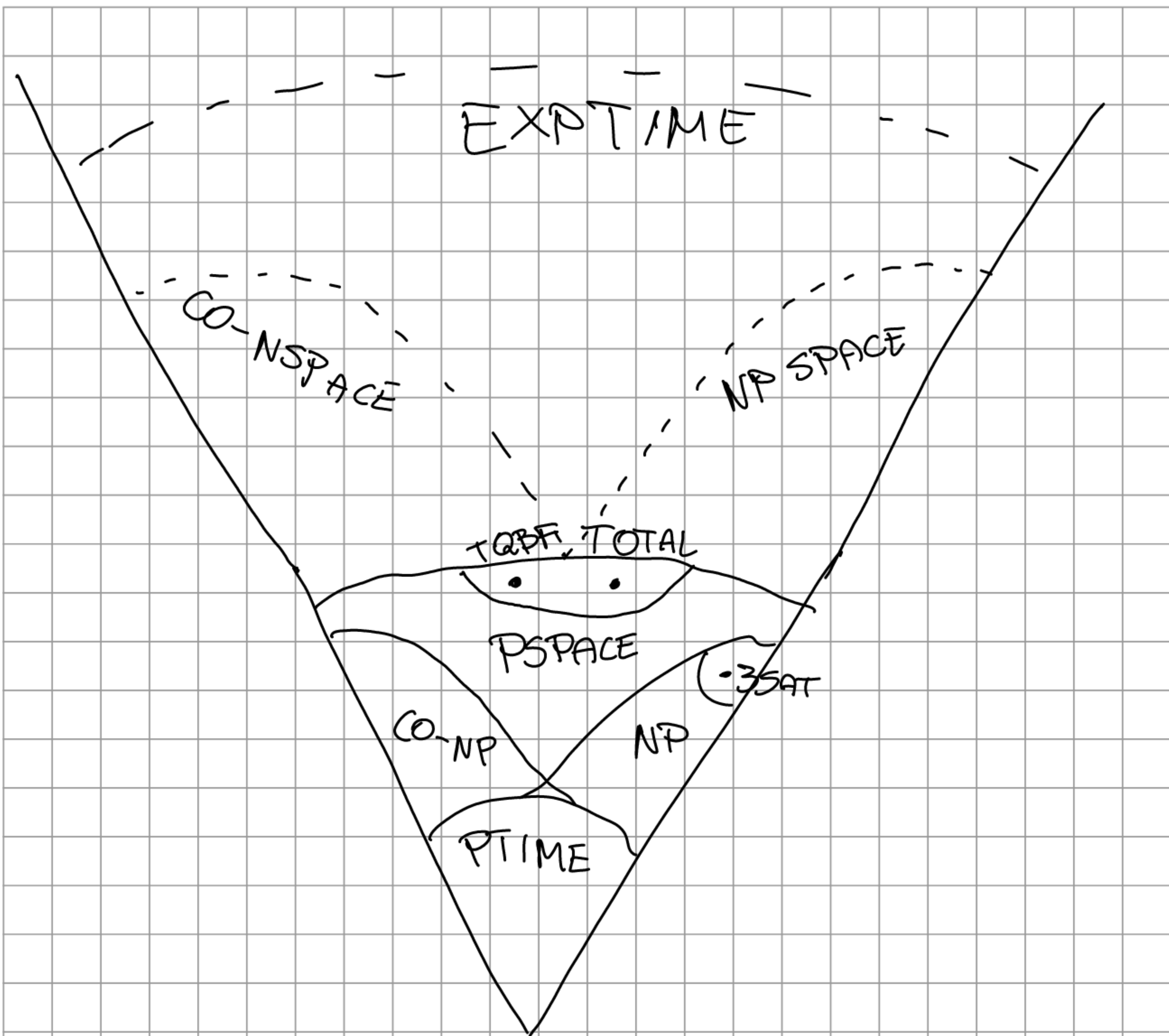
Def  $A \in PSPACE$ , gdy  $A$  daje się rozstrzygnąć przy pomocy Maszyny Turinga z wielomianową pamięcią.

Obserwacja •  $PSPACE \supseteq PTIME$

•  $NP \subseteq PSPACE$

•  $CO-NP \subseteq PSPACE$  ( $PSPACE$  jest zamknięty na dopełnienia)

Nie umiemy udowodnić NIC. Nawet nie umiemy powiedzieć, czy  $PTIME \neq PSPACE$ .



TW.  $PSPACE = NPSPACE$

Dając nam niedeterministyczną MT  $M$   
i wielomian  $p(x)$  t. z. dla każdego  $n$   
wejścia w maszynie  $M$  na każdej  
"ścieżce obliczeń" bzdzi  $\leq p(|w|)$ .



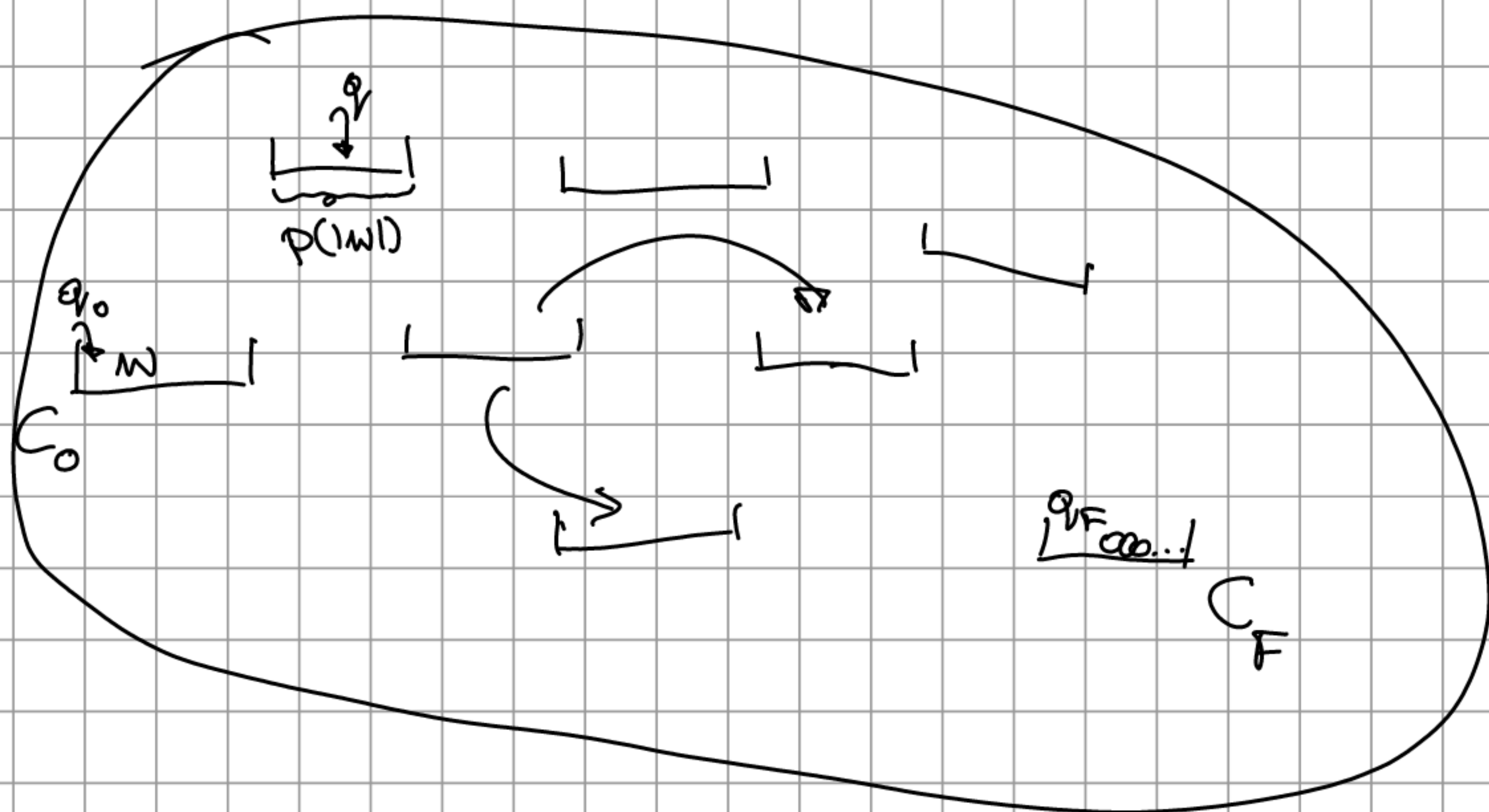
Mamy skonstruować deterministyczny algorytm  
 zużywający wielomianowo dużo pamięci  
 rozstrzygający czy  $M$  akceptuje dane  
 wejście.

(BSP  $M$  zanim się zakończy "zeruje" taśmę i wraca nad  $d$ )

$M$  ma zbiór stanów  $Q$  t.je  $|Q|=k$  i 5

symboli taśmowych. Jesteśmy teraz tym  
 wielomianowym algorytmem. Daję nam

słowo  $w$ . Wyobrażamy sobie graf konfigu-  
 racji  $M$  dla  $w$ .



Chcemy się dowiedzieć czy istnieje ścieżka z  $C_0$  do  $C_F$ .

W tym grafie jest  $5^{p(|N|)} \cdot |Q| \cdot p(|N|) \leq 8^{p(|N|)} = 2^{3p(|N|)}$   
↑  
jakiś stan      ↑  
gdzie jest głowica

Procedura<sub>i</sub> odpowie czy dla danych dwóch wierzchołków  $S, t$  grafu istnieje ścieżka z  $S$  do  $t$  dt.  $\leq 2^i$

Procedura<sub>0</sub> - sprawdza czy jest krawędź

Procedura<sub>i+1</sub> - dla każdego  $x$

- sprawdzi czy Procedura<sub>i</sub>( $S, x$ ),  
jeśli tak, to posprzątaj i sprawdzi

Procedura<sub>i</sub>( $x, t$ )

Teraz uruchamiam Procedura<sub>3p(|N|)</sub>( $C_0, C_F$ )

i zwróć wynik.

Pamięci użyje  $\sim 10 p^2 (|w|)$







W każdym węźle wewnętrznym mamy komunikację, alternatywę lub negację, a w liściach 0 lub 1.



$\exists p \forall q \ p \leftrightarrow q$  (kwantyfikujemy po  $\{T, F\}$ )

quantified boolean formula (QBF)

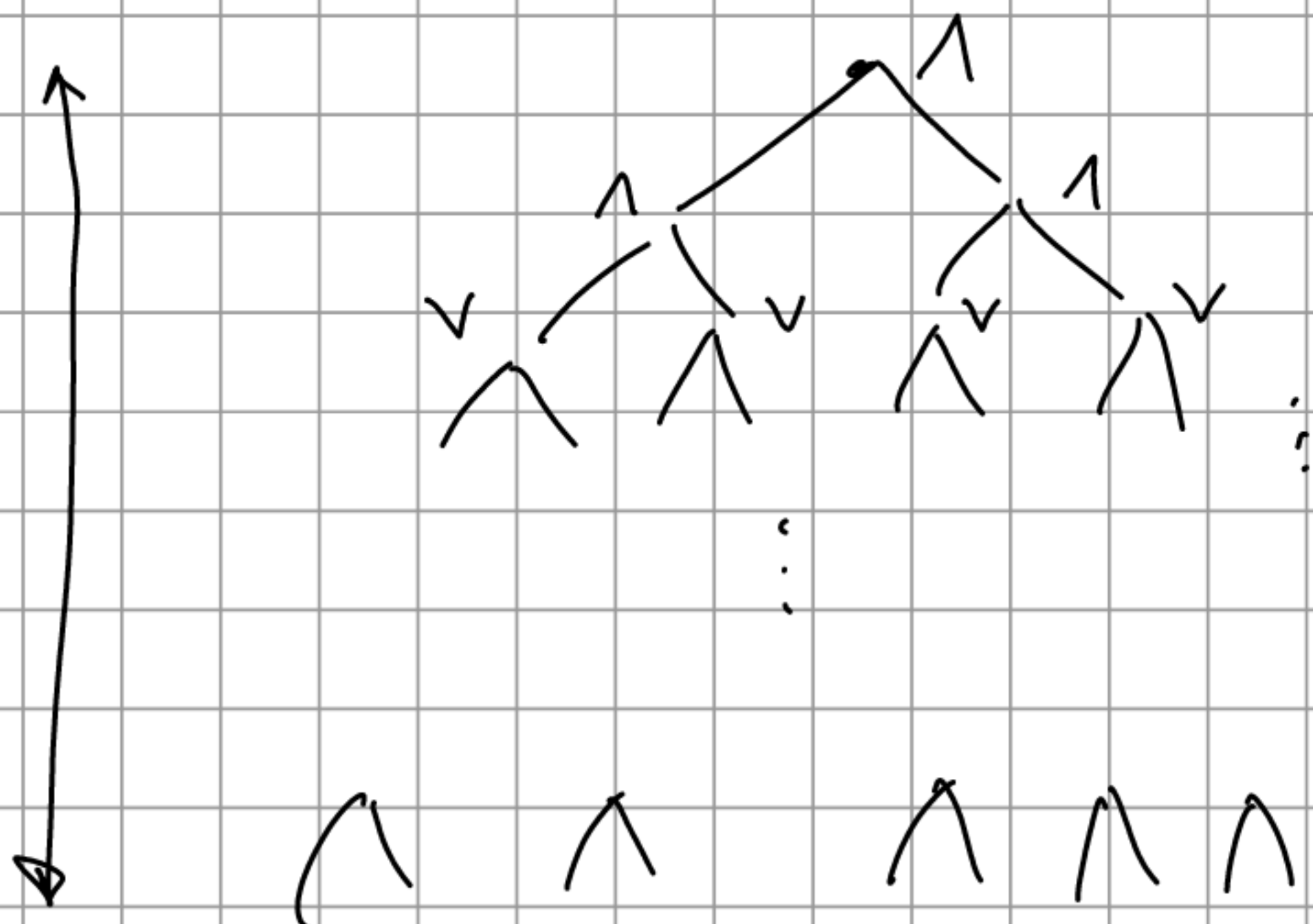
Tw.  $\uparrow$  TQBF jest problemem PSPACE-zupełnym.  
true

Dzd. Po pierwsze pokazujemy, że  $TQBF \in PSPACE$

Rysujemy drzewo:

AEEAA

ile jest zm.



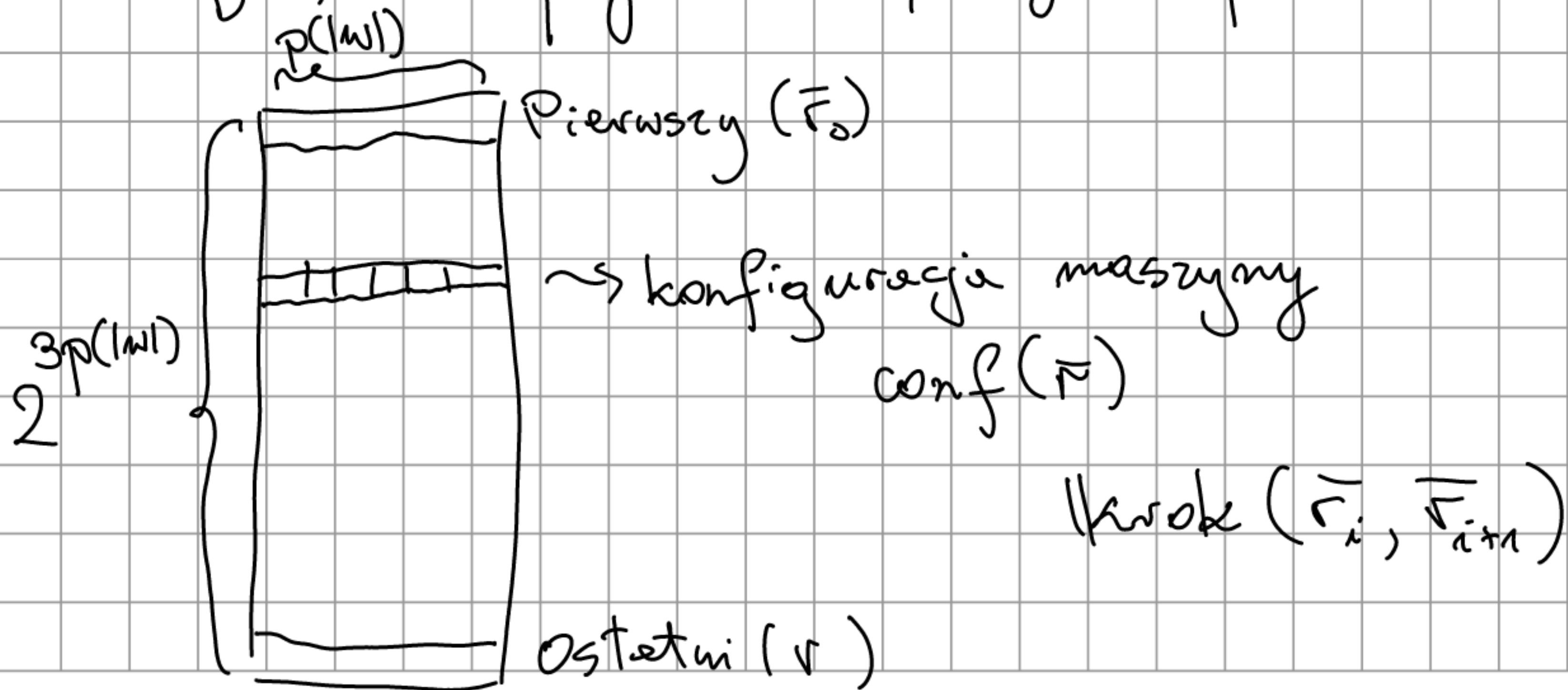
Znowu robimy zuzczka, ale teraz ma wielomianowo wiele pamizgi.

Teraz pokażemy, że dla każdego  $B \in PSPACE$   
 $B \leq_p TQBF$ .

Weźmy  $B \in PSPACE$  (czyli jest MT  $M_B$   
i wielomian t.ze  $M_B$  rozstrzyga  $B$  działając  
w przestrzeni  $p$  (wielkość wejścia).

Od teraz jesteśmy redukcją. Daję nam  
w: instancję  $B$ , mamy SZYBKO wyprodukować  
wec' formułę  $\psi_w$  w postaci QBF t.ze

$M_B(w)$  akceptuje  $\Leftrightarrow \psi_w$  jest prawdziwe



Problem: nie możemy mieć zmiennych na każdej komórce!

Najbardziej by nam pasowało napisać.

Pierwszy  $(\bar{r}_0) \wedge$

Ostatni  $(\bar{r}_{2^{3p(|M|)}}) \wedge$

Krok  $(\bar{r}_0, \bar{r}_1) \wedge$

Krok  $(\bar{r}_1, \bar{r}_2) \wedge$

$\vdots$

$\text{Conf}(\bar{t}) \wedge$

Napiszemy  $\text{Krok}_{i+1}(\bar{r}, \bar{s}) = \exists \bar{t} (\text{Krok}_i(\bar{r}, \bar{t}) \wedge \text{Krok}_i(\bar{t}, \bar{s}))$ .  $\text{Krok}_{i+1}(\bar{r}, \bar{s})$ : z  $\bar{r}$  do  $\bar{s}$  da się dojść w  $2^{i+1}$  kroków.

Ostateczna formuła:

Pierwszy  $(\bar{r}) \wedge$  Ostatni  $(\bar{s}) \wedge \text{Krok}_{3p(|M|)}(\bar{r}, \bar{s})$

Ale nadal coś jest nie tak z Krok: jest za duża. Naprawa:

$$\text{Krok}_{in}(\bar{F}, \bar{S}) = \exists \bar{t} \forall \bar{x}, \bar{y} \left[ (\bar{x} = \bar{F} \wedge \bar{y} = \bar{t}) \vee (\bar{x} = \bar{t} \vee \bar{y} = \bar{S}) \right]$$

$\rightarrow \text{Krok}_i(\bar{x}, \bar{y})$





# GRA W GEOGRAFIE

Instancją jest graf skierowany  $(V, E, v_0)$ , gdzie  $v_0$  jest wierzchołkiem początkowym.

Dwóch graczy: Stas i Nel. Mamy pieczętkę, która stoi w  $v_0$ . Następnie Stas bierze pieczętkę i przesuną po jakiejś krawędzi do jeszcze niedwiedzianego wierzchołka, potem Nel.

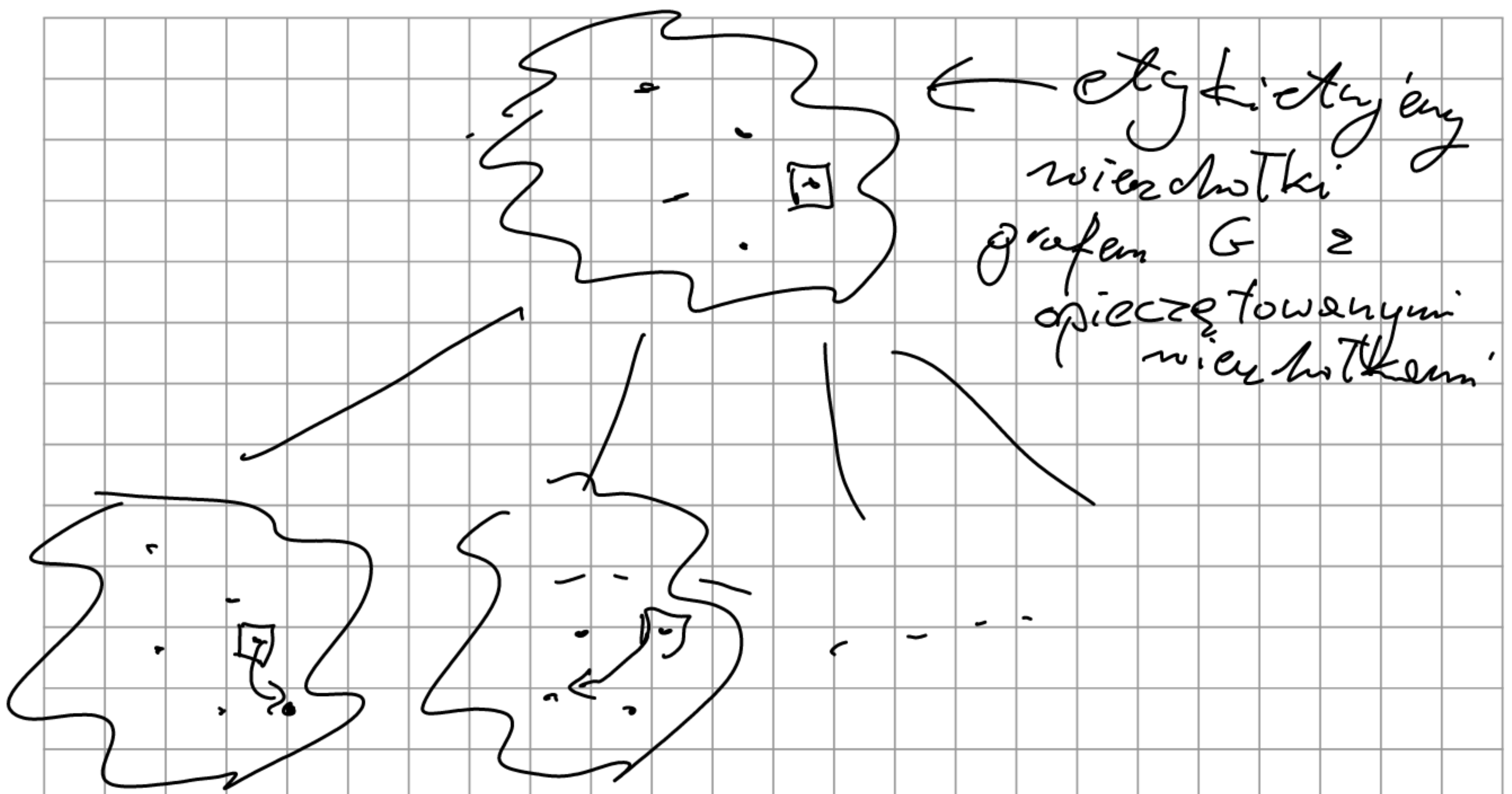
Przegrywa ten, kto nie może zrobić ruchu.

Pytanie czy Stas ma strategię wygrywającą.

Tw.  $\forall QBF \leq_p$  Gra w geografii

D-d. Na początku pokazujemy, że ta gra jest w ogóle w PSPACE





Rezult to samo co w TWA.

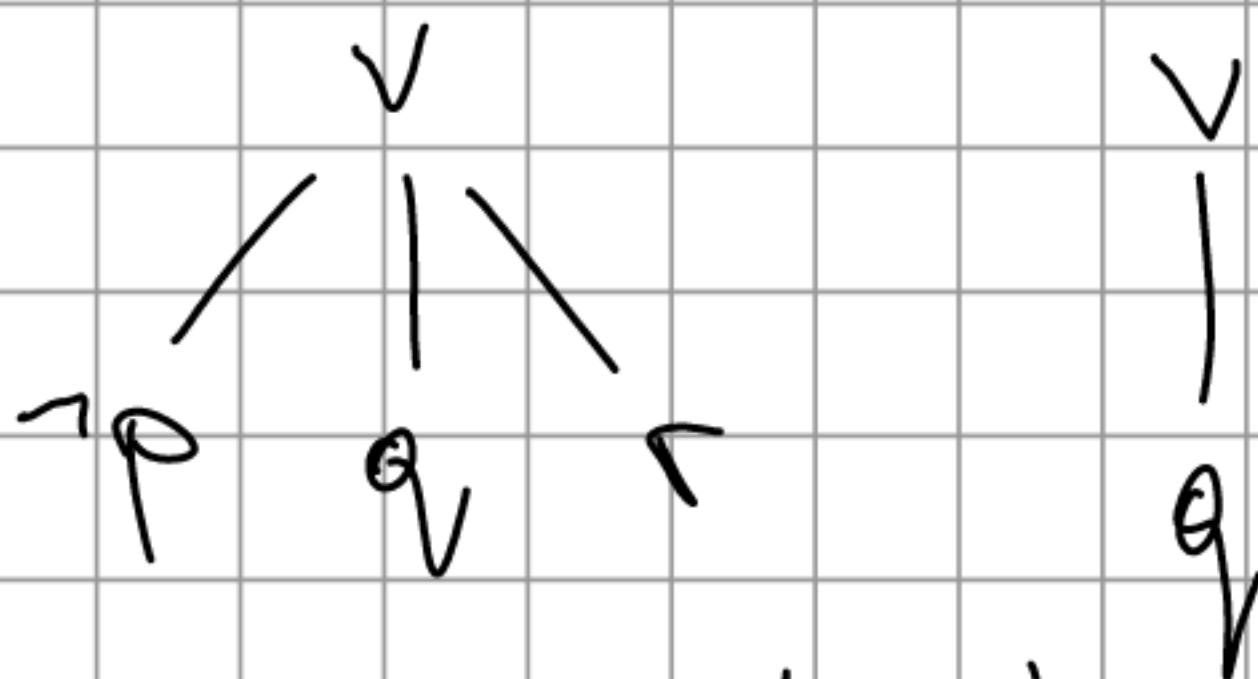
Teraz daję nam formułę  $\varphi$  w postaci QBF.

BSO  $\varphi$  ma prefiks kwantyfikatorsowy

$$\exists p_1 \forall q_1 \exists p_2 \forall q_2 \dots \exists p_k \forall q_k$$

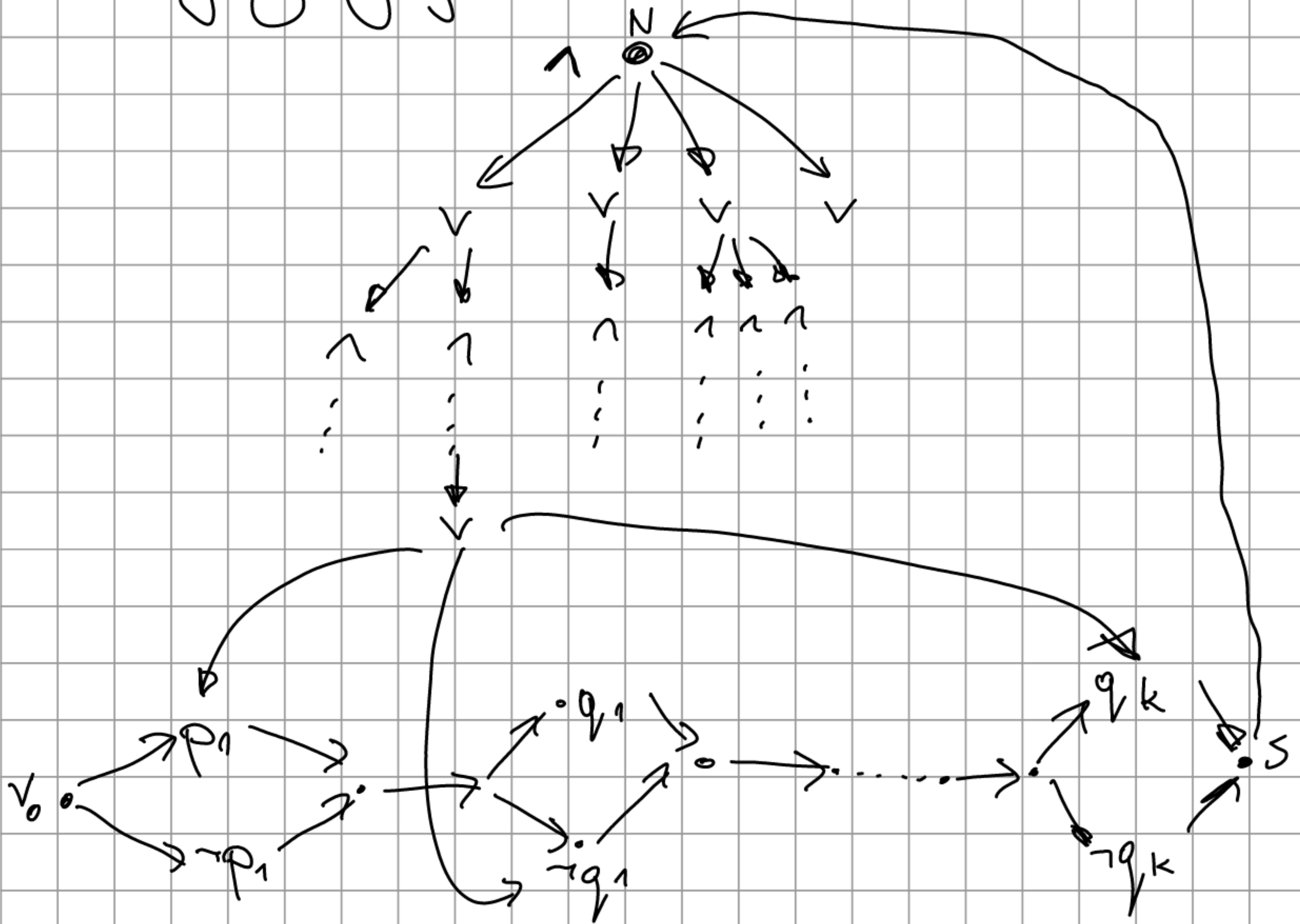
negacja jest tylko w liściach.

Każdy litera  $e_i$  ma bezpośrednio nad sobą alternatywę,



$\alpha$  w korzeniu jest komunikacja

Budujemy graf:



Nieostemplowane literały to te wybrane.

Nela chce pokazać, że formuła nie jest prawdziwa.

13.06.2022

Rozszerzamy wyrażenie regularne z

• dodatkowym symbolem podnoszenia

do kwadratu:  $e^2$ . Wtedy  $|e^2| = |e| + 1$ ,

$$L(e^2) = L(e) \cdot L(e).$$

• dodatkowym symbolem dopetnienia:  $e^c$ .

$$\text{Wtedy } L(e^c) = \sum^* L(e),$$

$$|e^c| = |e| + 1.$$

## PROBLEM TOTALNOŚCI REX

Okazuje się, że problem totalności:

1) dla RE jest PSPACE-zupełny

2) dla RE(2) jest EXPSPACE-zupełny

3) dla RE( $c$ ) jest nieelementarny.

Rozszerzamy rodziny funkcji:

• EXP:  $f = \Theta(2^g)$ ,  $g$ : wielomian

• k-EXP:  $f = \Theta(2^g)$ ,  $g \in (k-1)$ -EXP



Klasa funkcji elementarnych:  $\bigcup_{k>0}^{\infty} k\text{-EXPTIME}$

bo  $k\text{-EXPSPACE} \xrightarrow{\quad} \bigcup_{k>0}^{\infty} k\text{-EXPSPACE}$

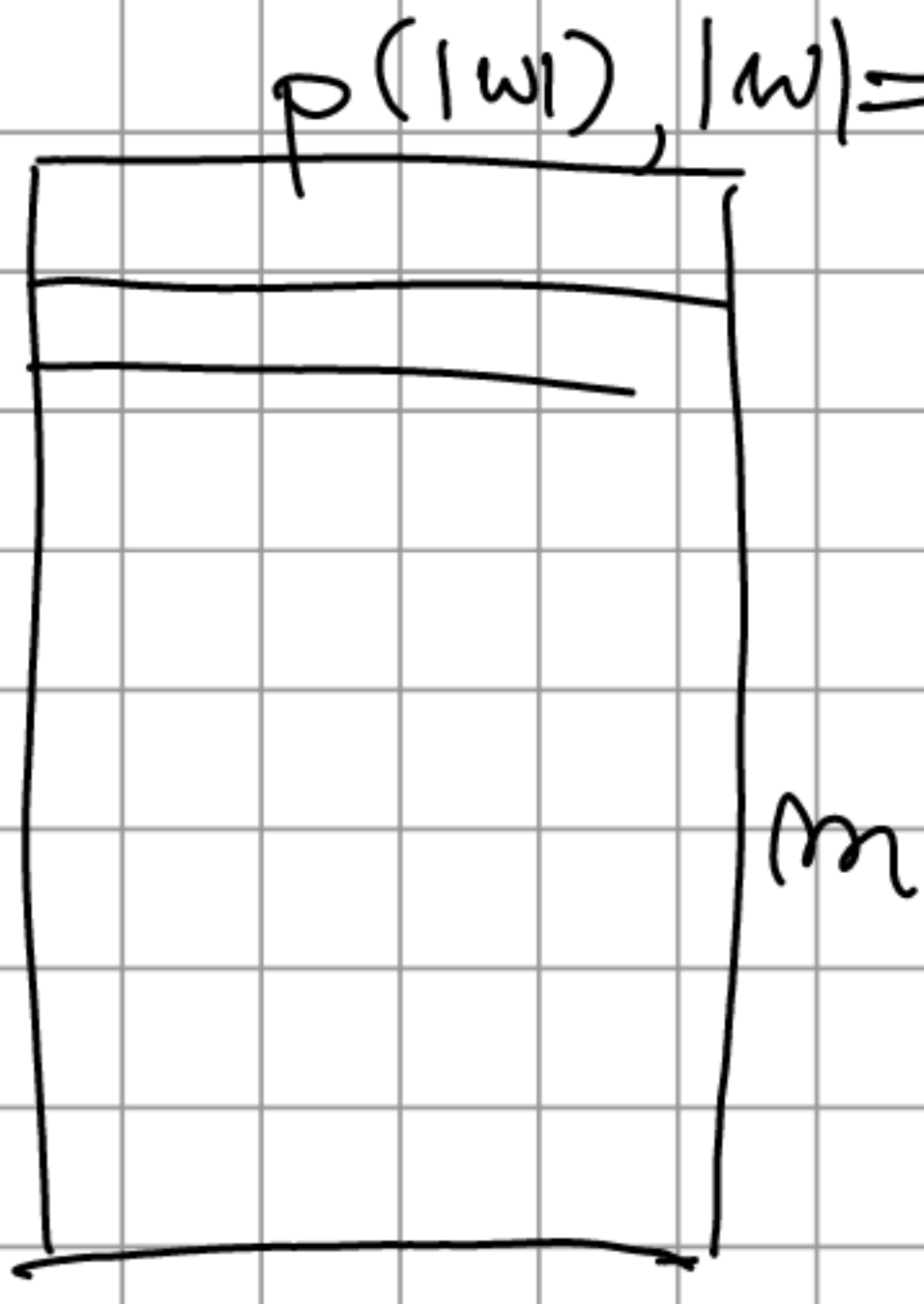
Rodzina  $\mathcal{F}$ . Konwniczny problem  $\mathcal{F}\text{-TIME}$   
(SPACE)-zupdy

Daję MT  $M, w \in \Sigma^*, p \in \mathcal{F}$  i  
pytają czy  $M$  akceptuje  $w$  w osie  
(pamięci)  $p(|w|)$ ?

D-d.(1) Daję MT  $M, w \in \Sigma^*, p \in \text{POLY}$ .

Najpierw zmieniamy (redukujemy)  $w$  w  $M', \epsilon, p$ .

To zmieniamy  
w problem kafet-  
kowania dla



Kaf 1) ka zedy  
kafelek  $2 \times 2$   
malerij do E  
Kaf 2) Pierwszy  
i ostatni wzd  
są postaci  
 $N \subset B^* N$

tej maszyny:  
 $K, E, p(|w|)$   
 $k^4$

Kaf 3) wymiary są  $m \times p(|w|)$

$$(k^2)^*$$

$$\Sigma = k^2 \cup \{ \# \}$$



$$\binom{w_1}{w_2} \# \binom{w_c}{w_3} \# \dots \# \binom{w_{m-1}}{w_m}$$

kodowanie kolorowanie

Napiszemy  $e$  dopasowujące się

do słów  $u$ , które NIE

kodują poprawnych kolorowań

K1)  $\Sigma^* \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \Sigma^*$ , gdzie  $\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \notin E$

K2) Początek i koniec NIE są postaci

$$NCB^*N$$

K3)  $\Sigma^* \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \Sigma^{p(n)} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \Sigma^*$ ,  $a_2 \neq b_1$

$$\Sigma^* \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \Sigma^{p(n)} \# \Sigma^*$$
 i symetrycznie

K4)  $(\neg \#)^{p(n)+1} \Sigma^* + (\neg \#)^{p(n)-1} \# \Sigma^*$

$e$ : alternatywna wyrażenie K1-4

Wniosek 1) Dla danej MT  $M$ , słów  $w \in \Sigma^*$ ,  $P(x) \in POL \cup$  istnieje  $e_{M,w}$  t.ż.  $\overline{L}(e_{M,w}) = \{H_{M,w}^P\}$

Problem EXPTIME-zupełności możemy

rozwiązać tak samo, ale w  $K3$  i  $K4$

wyrażemy  $\sum P(n) = \sum 2^{q(n)} = \left( \sum 2^{q(n)} \right) \cdot 2^{q(n)}$ ,

gdzie  $p(n) = 2^{q(n)}$ ,  $q \in \text{POLY}$

Dla problemu z dopelnieniem pokazujemy,  
ze problem RE(c) jest  $k$ -EXPSPACE  
trudny dla  $k \in \mathbb{N}$ .

Pokazujemy, ze jest 1-EXPSPACE-trudny,  
2-EXPSPACE-trudny, a reszta przez  
indukcję.

Wybacz czytelniku, ale nie dam rady  
tego sensownie zanotować.