28.08.2022

PROBLEM DECYZYJNY

- Σ – skończony alfabet
- Σ^* – zbiór skończonych słów nad alfabetem Σ
- $\Sigma^* \supseteq L$ – język / problem

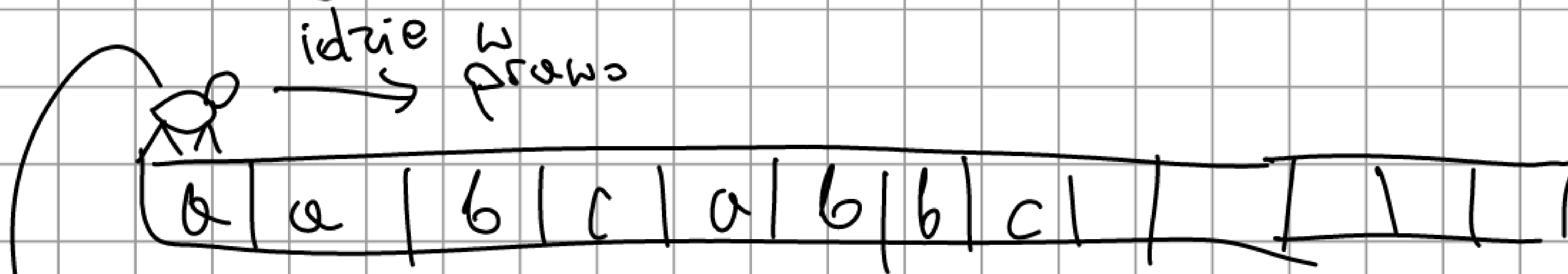
- pytamy o "zasoby obliczeniowe" potrzebne do rozstrzygania tych problemów.

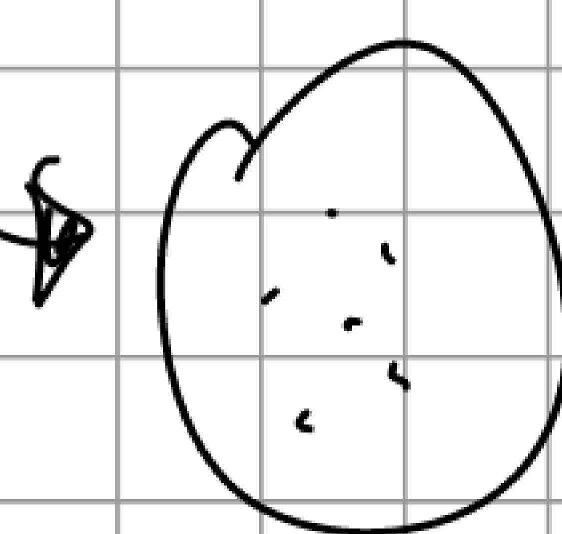
- klasyfikujemy problemy ze względu na te zasoby

Bla bla...

Część I

Automaty skończone





skończony zbiór stanów Q, Funkcja przejścia $\delta : Q \times \Sigma \to Q$

- Stan początkowy $q_0 \in Q$
- zbiór stanów akceptujących $F \subseteq Q$

Ćw. Skonstruuj δ, Q dla $\Sigma = \{0,1\}$,

$L = \{ w \in \{0,1\}^* : |w|_1 \text{ jest parzysta} \}$

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

się nie da!

D-d. (A.a.) Niech Zenon będzie żuczkiem rozstrzygającym L. Zał. że $|Q| = k$.

$$w_0 = \varepsilon \rightsquigarrow s_0 \in Q$$
$$w_1 = 0 \rightsquigarrow s_1 \in Q$$
$$\vdots$$
$$w_i = 0^i \rightsquigarrow s_i \in Q$$
$$\vdots$$
$$w_k = 0^k \rightsquigarrow s_k \in Q$$

Jest i, j t. że $s_i = s_j$

Spójrzmy na

$$a = w_i 1^i \rightsquigarrow s \in F$$
$$b = w_j 1^i \rightsquigarrow s \notin F \; \lightning$$

<u>Deterministyczny automat skończony</u> (DFA) to krotka $\langle \Sigma, Q, q_0, \delta, F \rangle$.

Mamy $\delta : Q \times \Sigma \to Q$, definiujemy $\hat{\delta} : Q \times \Sigma^* \to Q$:

$$\begin{cases} \hat{\delta}(q, \varepsilon) = q \\ \hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a) \end{cases}$$

LUB

równoważne ćw.: $\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$

Dla DFA $A = \langle \Sigma, Q, q_0, \delta, F \rangle$ przez L_A oznaczamy $\{w \in \Sigma^* : \hat{\delta}(q_0, w) \in F\}$.

__Def.__ $L \subseteq \Sigma^*$ nazywamy regularnym gdy istnieje DFA A t. że $L = L_A$.

__Lemat__ (o pompowaniu dla języków regularnych)

Dla każdego j. reg. L istnieje $n \in \mathbb{N}$ t. że dla każdego $w \in L$ t. że $|w| \geq n$ istnieją słowa x, y, z t. że $xyz = w$

oraz $y \neq \varepsilon$, $|xy| \leq n$ 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 $|xz|_0 < |xz|_1$ ↯

Dowód lematu

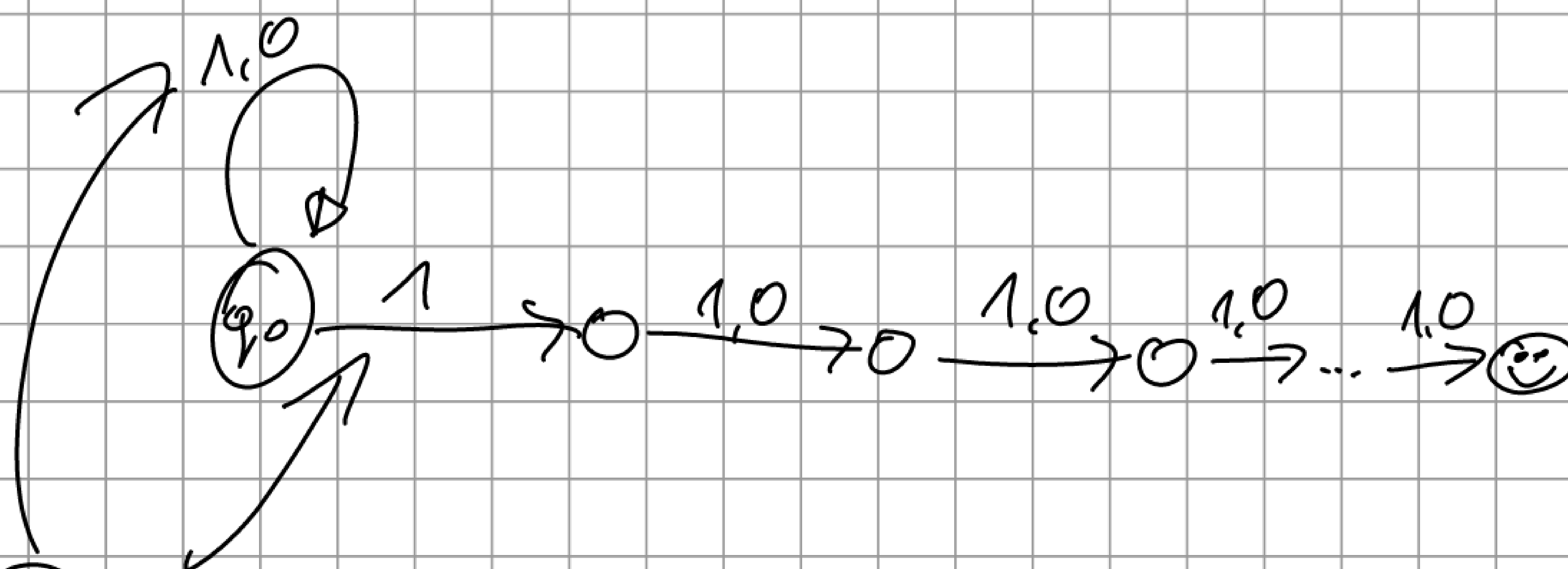
Weźmy $L = L_A$ regularny $(A = \langle \Sigma, Q, q_0, \delta, F \rangle)$.
Niech $n = |Q| + 1$. Weźmy $w \in L$ t.że $|w| \geq n$.
Niech $w = a_1 a_2 \ldots a_l$, niech $s_i = \hat{\delta}(q_0, a_1 \ldots a_i)$.
Wtedy $s_i = s_j$ dla pewnych $i < j \leq n$.
Niech $x = a_1 \ldots a_i$, $y = a_{i+1} \ldots a_j$,
$z = a_{j+1} \ldots a_l$. Wtedy dla $k \in \mathbb{N}$ ...

1.03.2022

NIEDETERMINISTYCZNE AUTOMATY SKOŃCZONE (NFA)

Przykład

$L = \{ w1v \in \{0,1\}^* : |v| = 9 \}$



DWA PRZEJŚCIA Z 1 → ŻUCZEK PYTA NIEBIOS

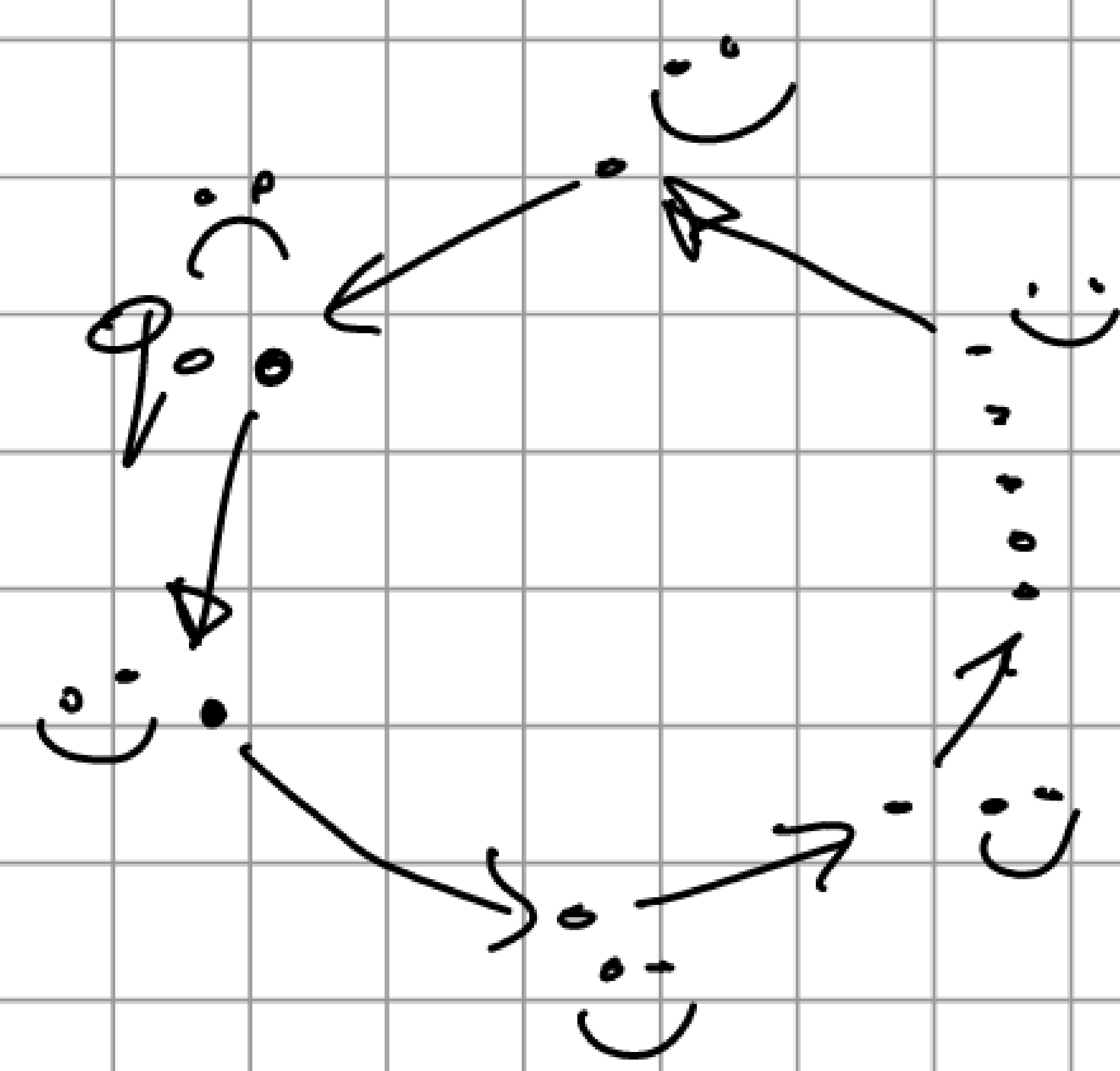
"CO ROBIĆ? JAK ŻYĆ"

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")

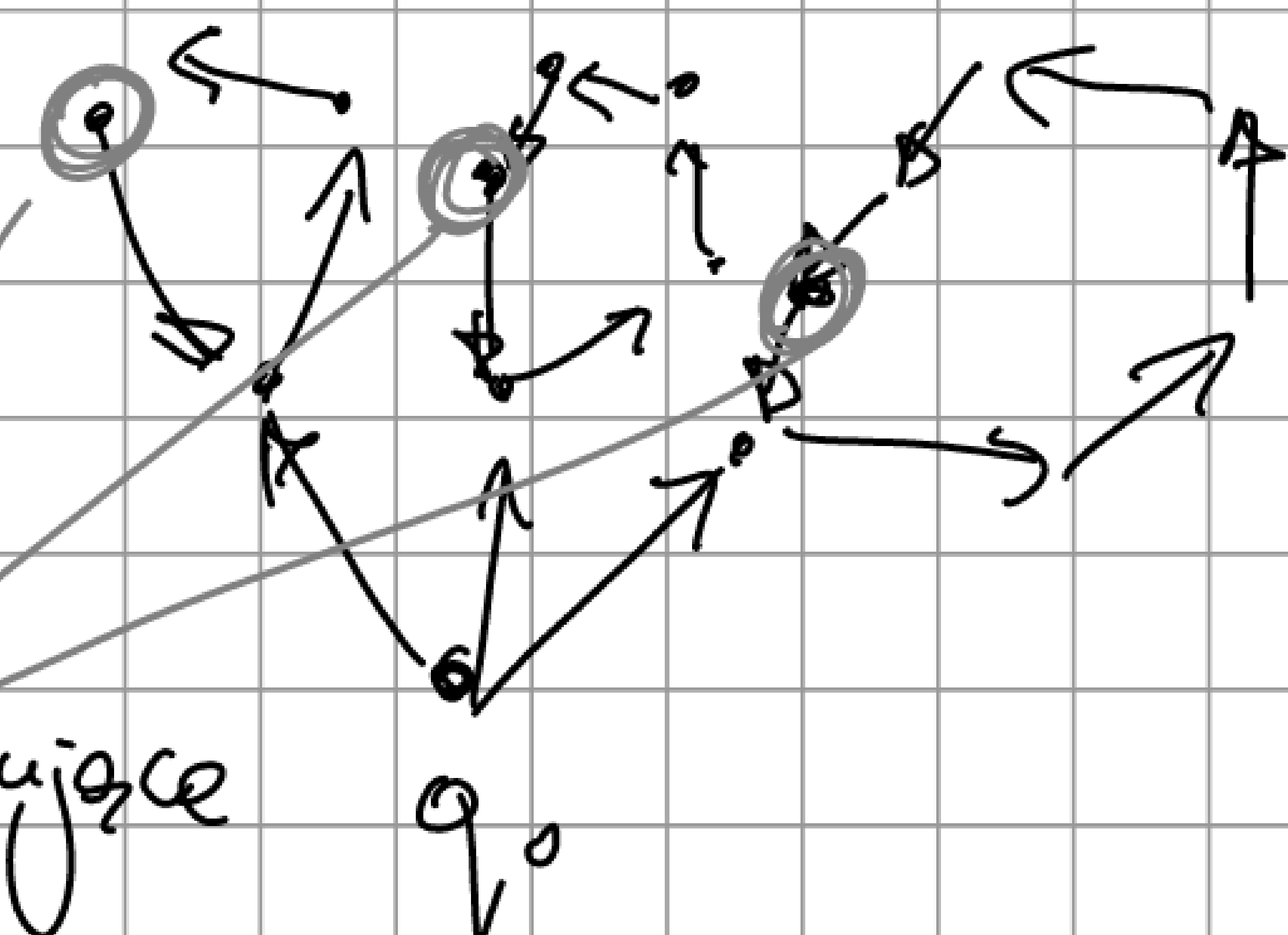
Przykład 2 $\quad L = \{0^i : 105 \nmid i\}$

Det. aut.:



105 stanów

Niedet.:



Stany nieakceptujące

Znowu: 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 δ, gdzie

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

$\delta(q, a, q')$ ~ z q do q' jest strzałka z etykietą a.

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

$$\begin{cases} \hat{\delta}(q, \varepsilon, q') \iff q = q' \\ \hat{\delta}(q, wa, q') \iff \exists p \in Q \ \hat{\delta}(q, w, p) \wedge \delta(p, a, q') \end{cases}$$

Alternatywna wersja wg JMa.

Dla każdego $w \in \Sigma^*$ definiujemy $\delta_w \subseteq Q \times Q$:

$$\begin{cases} \delta_\varepsilon = id_Q \\ \text{jeśli } a \in \Sigma \text{ to } \delta_a(q, q') \iff \delta(q, a, q') \\ \delta_{wa} = \delta_w \circ \delta_a \end{cases}$$

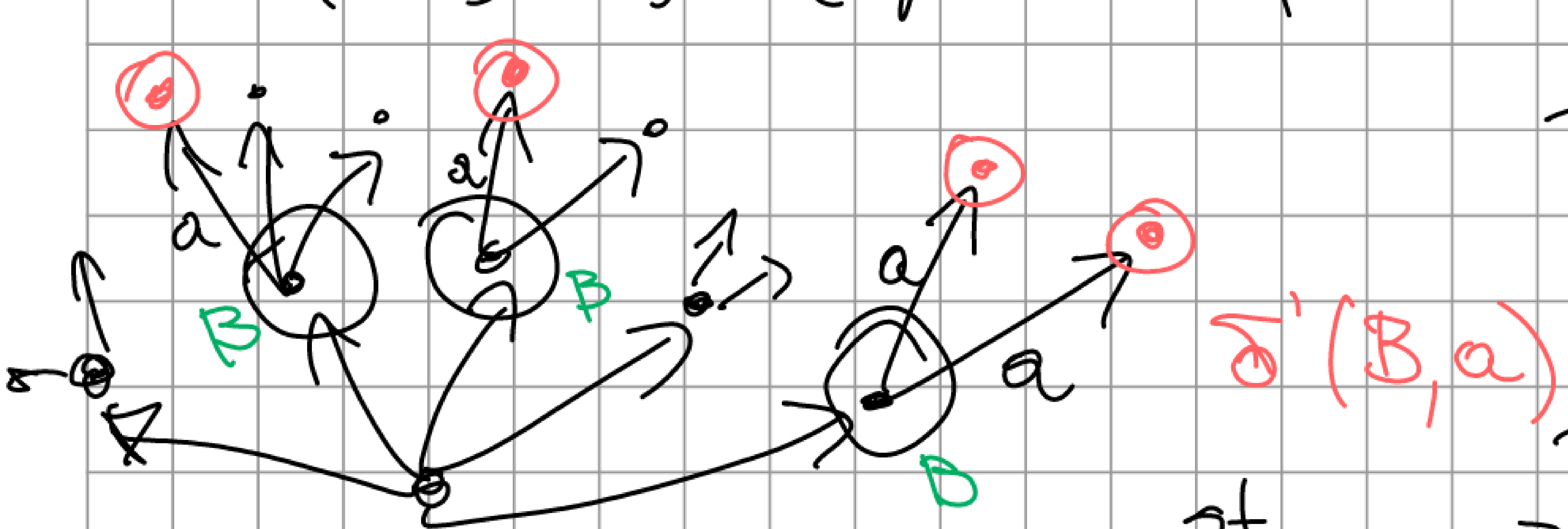
Wtedy $\hat{\delta}(q, w, q') \iff \delta_w(q, q')$.

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

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 z B po kraw. z etykietą a.

NFA Z ε-PRZEJŚCIAMI

Takie coś, że możemy czasem sobie przejść ze stanu do stanu bez wczytania znaków. Je też można zdeterminizować.

7.03.2022

WYRAŻENIA REGULARNE (nad Σ)

- $\emptyset$ jest wyrażeniem regularnym i $L_\emptyset = \emptyset$
- ε jest wyr. reg. i $L_\varepsilon = \{\varepsilon\}$
- jeśli $a \in \Sigma$ to a jest wyr. reg. oraz $L_a = \{a\}$
- jeśli φ, ψ są wyr. reg. to $\varphi + \psi$ jest wyrażeniem regularnym i $L_{\varphi+\psi} = L_\varphi \cup L_\psi$, $\varphi\psi$ jest wyrażeniem regularnym i $L_{\varphi\psi} = L_\varphi L_\psi$ $= \{ w_1 w_2 : w_1 \in L_\varphi, w_2 \in L_\psi \}$.

$$L_1 L_2 = \{ w_1 w_2 : w_1 \in L_1, w_2 \in L_2 \}$$

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

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

- jeśli φ jest wyr. reg. to φ^* też jest oraz $L_{\varphi^*} = (L_\varphi)^*$

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

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

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

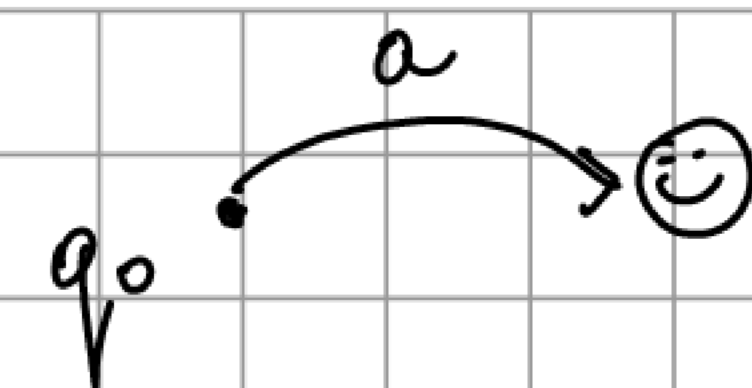
(2) Istnieje wyrażenie regularne φ t. że $L = L_\varphi$.

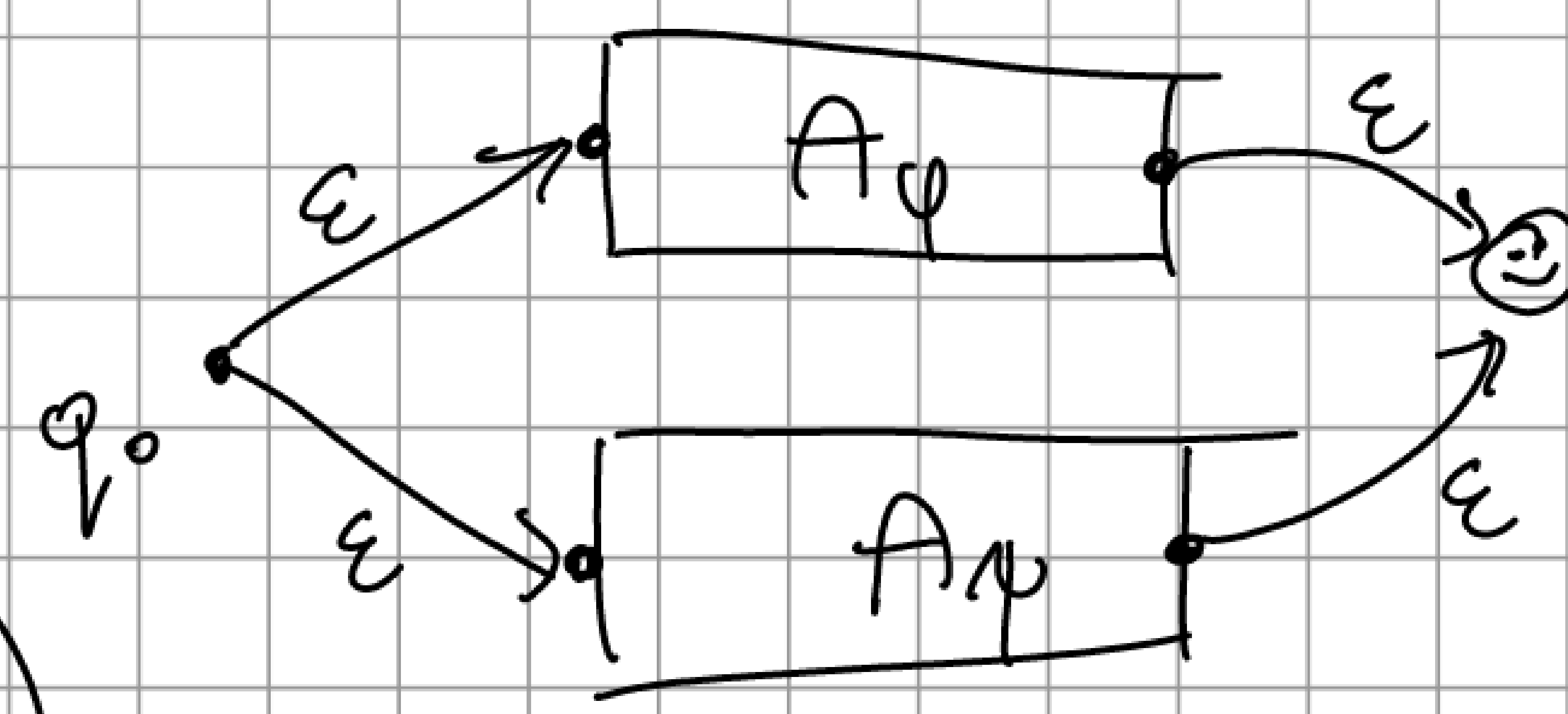
(3) Istnieje NFA z ε-przejściami A' t. że $L = L_{A'}$.

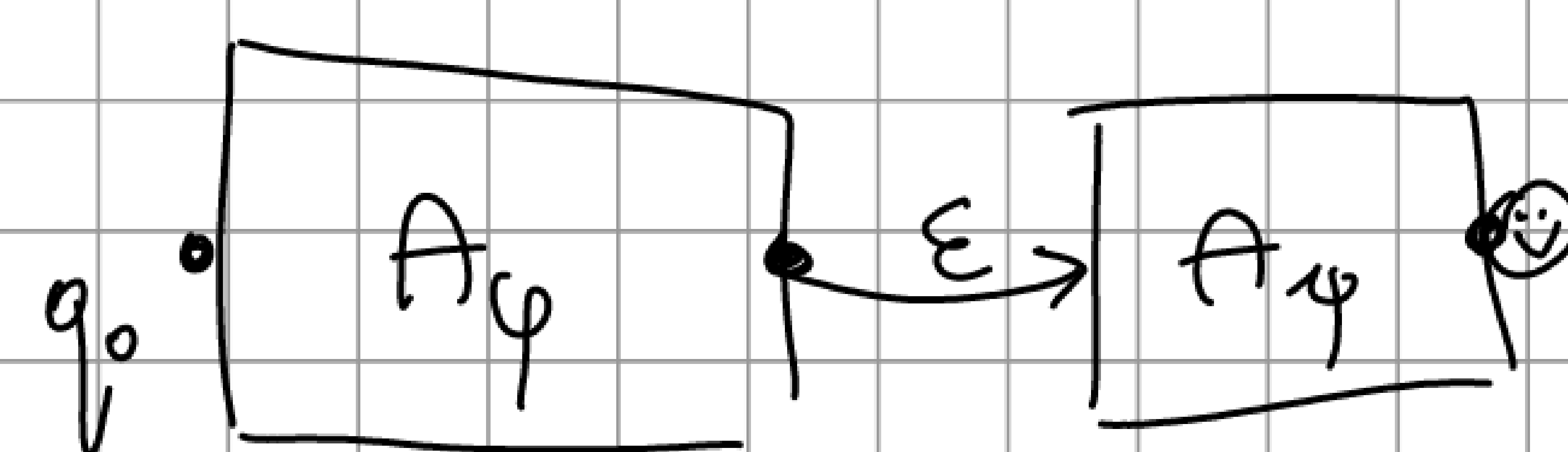
D-d. (3) $\Rightarrow$ (1) Było. $\blacksquare$

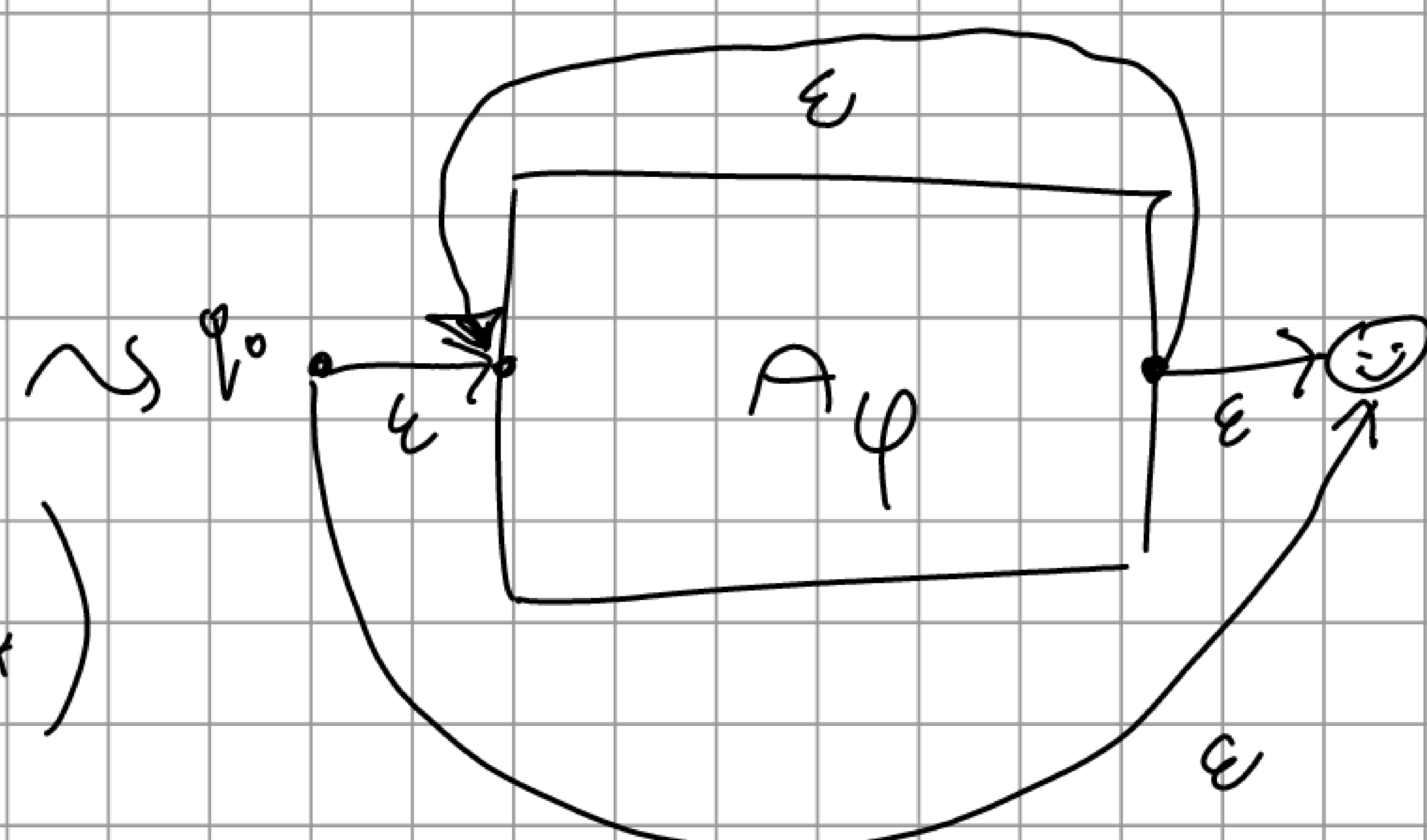
(2) $\Rightarrow$ (3) Indukcja względem długości wyrażenia. Każdy NFA który zbudujemy będzie miał jeden stan wejściowy niebędący akceptującym i jeden akceptujący.

- $\emptyset \rightsquigarrow q_0 \bullet \quad$ ☺
- $\varepsilon \rightsquigarrow q_0 \bullet \xrightarrow{\varepsilon}$ ☺

- $a \rightsquigarrow$ 

- $\varphi, \psi \rightsquigarrow$ 

  (automat dla $\varphi + \psi$)
- $\varphi, \psi \rightsquigarrow$ 

  (Automat dla $\varphi\psi$)
- $\varphi \rightsquigarrow$ 

  (Automat dla φ^*)

∎

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

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

Dla $1 \le i, j \le n, 0 \le k \le n$ napiszemy wyrażenie regularne $\varphi^k_{i,j}$ wyrażające język niepusty oraz $\neq v$

$$\{ w \in \Sigma^* : \hat{\delta}(q_i, w) = q_j \wedge \forall v \in \Sigma^* \big(\text{jeśli } v \text{ jest właściwym prefiksem } w \text{ oraz } \hat{\delta}(q_i, w) = q_l \text{, to } l < k \big) \}$$

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

Indukcje względem k:

- $$\varphi^0_{i,i} = \varepsilon + \bigoplus_{\substack{a \in \Sigma \\ \delta(q_i, a) = q_i}} a$$ (Suma po literach spełniających warunek)

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

- $$\varphi^{k+1}_{i,j} = \varphi^k_{i,j} + \varphi^k_{i,k+1} \left(\varphi^k_{k+1,k+1}\right)^* \varphi^k_{k+1,j}$$

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

$$\psi = \bigoplus_{q_i \in F} \varphi^n_{1,i}$$

■

UOGÓLNIENIA

Są dwa możliwe kierunki:

- słowa nieskończone
- drzewa

$\Big\}$ można iść w obu tych kierunkach jednocześnie

Automat skończony na słowach nieskończonych.

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

Σ – skończony

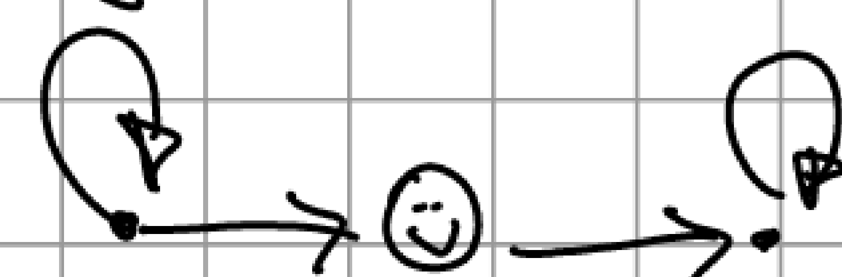
Dużo warunków akceptacji Büchiego, Rabina...

Interpretacja Büchiego: $F \subseteq Q$.

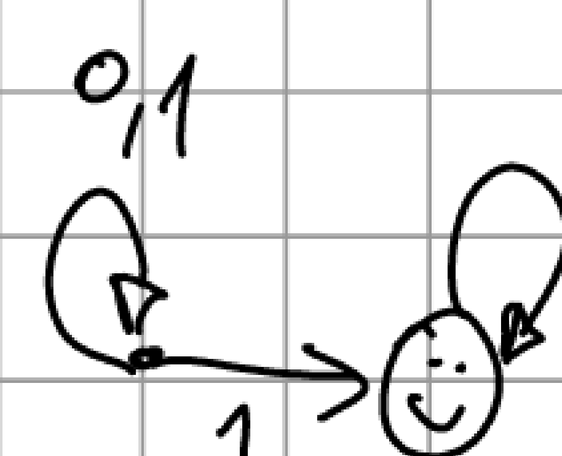
Słowo jest akceptowane, gdy automat nieskończenie wiele razy odwiedza stany z F.

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

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

NFA: 

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 Σ jest najmniejszą klasą języków nad Σ, która:

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

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

GRAMATYKI BEZKONTEKSTOWE

Def. **Gramatyka bezkontekstowa** (CFG) to krotka $\langle N, \Sigma, S, \Pi \rangle$, $S \in N$,

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

CFG – Context Free Grammar

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

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

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

Relacje $\xrightarrow{*}_\Pi$ i $\xleftrightarrow{*}_\Pi$ definiujemy odpowiednio jako tranzytywne i równoważnościowe domknięcie relacji $\longrightarrow_\Pi$.

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

KONIEC DYGRESJI

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

<u>Def.</u> $L \subseteq \Sigma^*$ jest **bezkontekstowy** (CFL), gdy istnieje CFG G t. że $L = L_G$

<u>Obserwacja</u> Każdy język regularny jest bezkontekstowy.

<u>Dowód</u> Klasa CFL spełnia warunki z uwagi (*), (i), (ii), (iii):

(*): proste, dodajemy reguły $S \to w$ dla w z języka

(i): bierzemy sumę rozłączną dwóch gramatyk 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 $G = \langle N' \sqcup N'' \sqcup \{S\}, \Sigma, S, \Pi' \sqcup \Pi'' \sqcup \langle S, S'S'' \rangle \rangle$

(iii) Podobnie gwiazdka.

Przykład • Notacja: $S \to aSa \mid bSb \mid \varepsilon$

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

(to konkretnie daje język palindromów parzystej długości).

• $S \to SS \mid \varepsilon \mid aSb \mid bSa$

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

• $S \to (S) \mid [S] \mid SS \mid \varepsilon$

poprawne nawiasowanie z (,), [,].

Konwencja notacyjna (niemoralna Marcinkowskiego):

Dla języków L, L' piszemy $L = L'$ gdy $L \div L' \subseteq \{\varepsilon\}$

różnica symetryczna

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

Tw. (Chomskiego o postaci normalnej)
Dla każdego CFL L istnieje CFG G w postaci Chomsky'ego t. że $L = L_G$.

Dowód: Nudny :(

Lemat (o pompowaniu dla CFG)
Dla każdego CFL L istnieje $n \in \mathbb{N}$ t. że dla każdego $w \in L$, $|w| \geqslant n$, istnieją słowa

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

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

Odp.: TAK (proste)

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

Odp.: TAK (to samo)

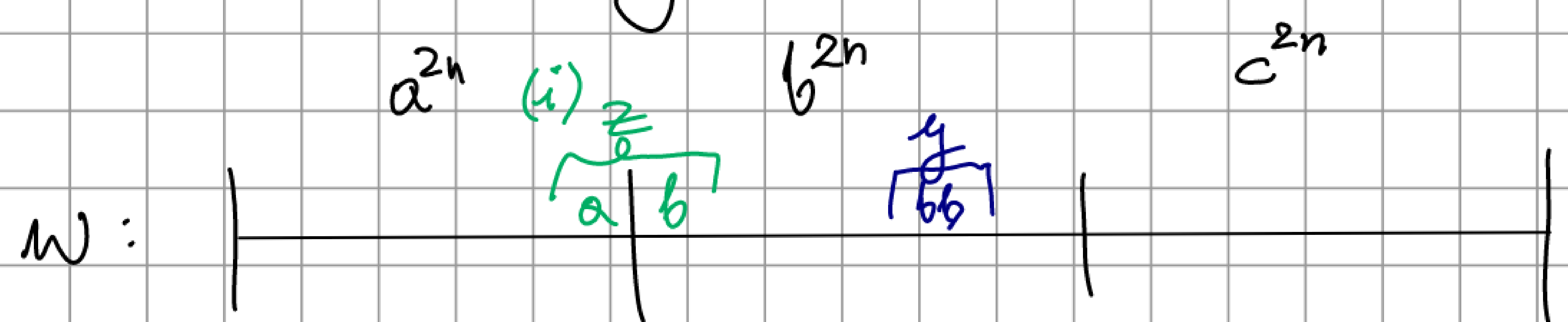
A czym jest przekrój tych języków?

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

Ten język jednak nie jest CFL!

<u>Dowód</u> Załóżmy, że L jest CF. Niech n: stała z lematu o pompowaniu.

Niech $w = a^{2n} b^{2n} c^{2n}$. Wtedy są stałe s, z, t, y, x z lematu.



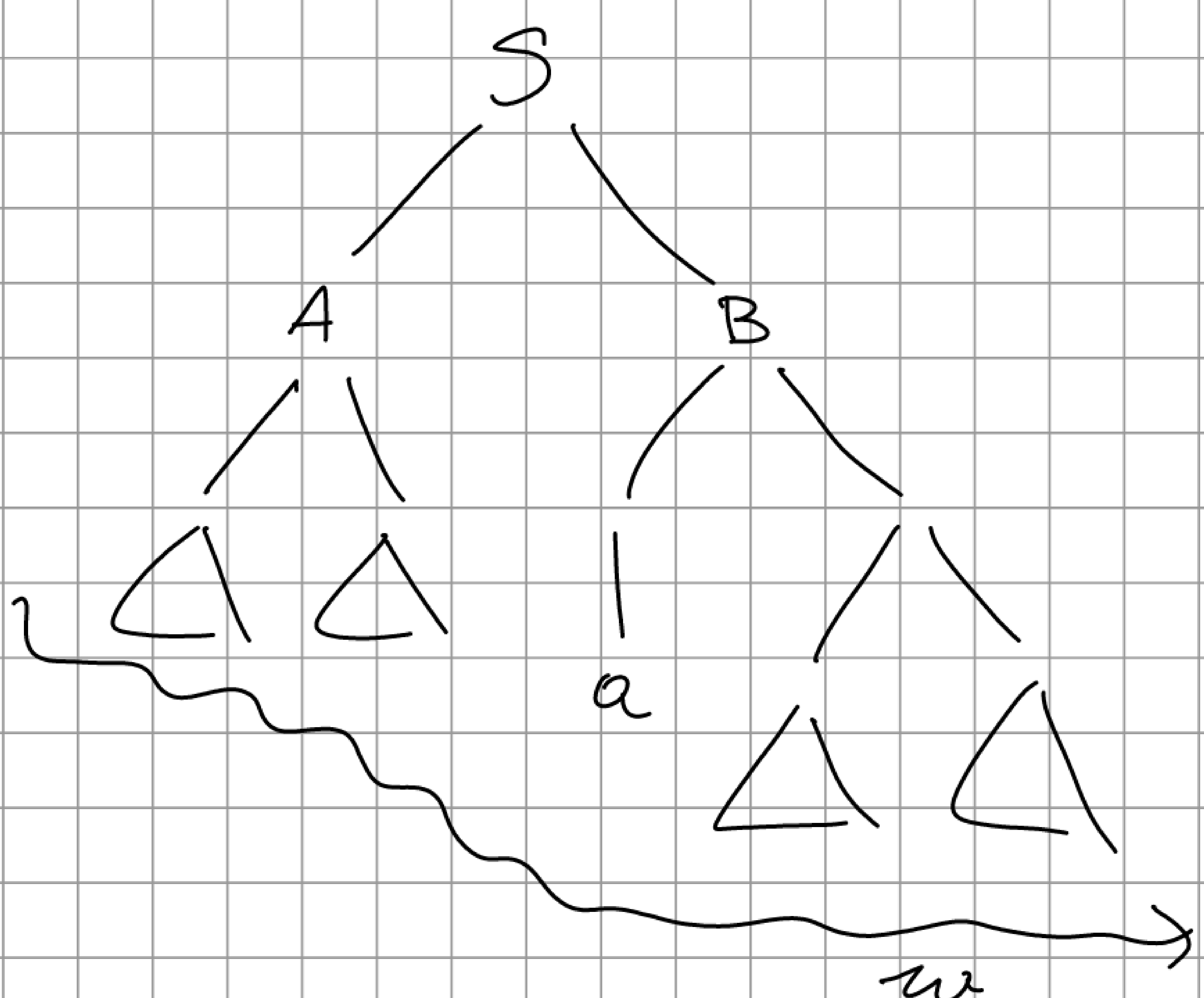
Przypadki: (i) z lub y zawiera dwie różne literki, to dla $k=2$ wygrywamy

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

21.03.2022

D-d. (lematu o pompowaniu)

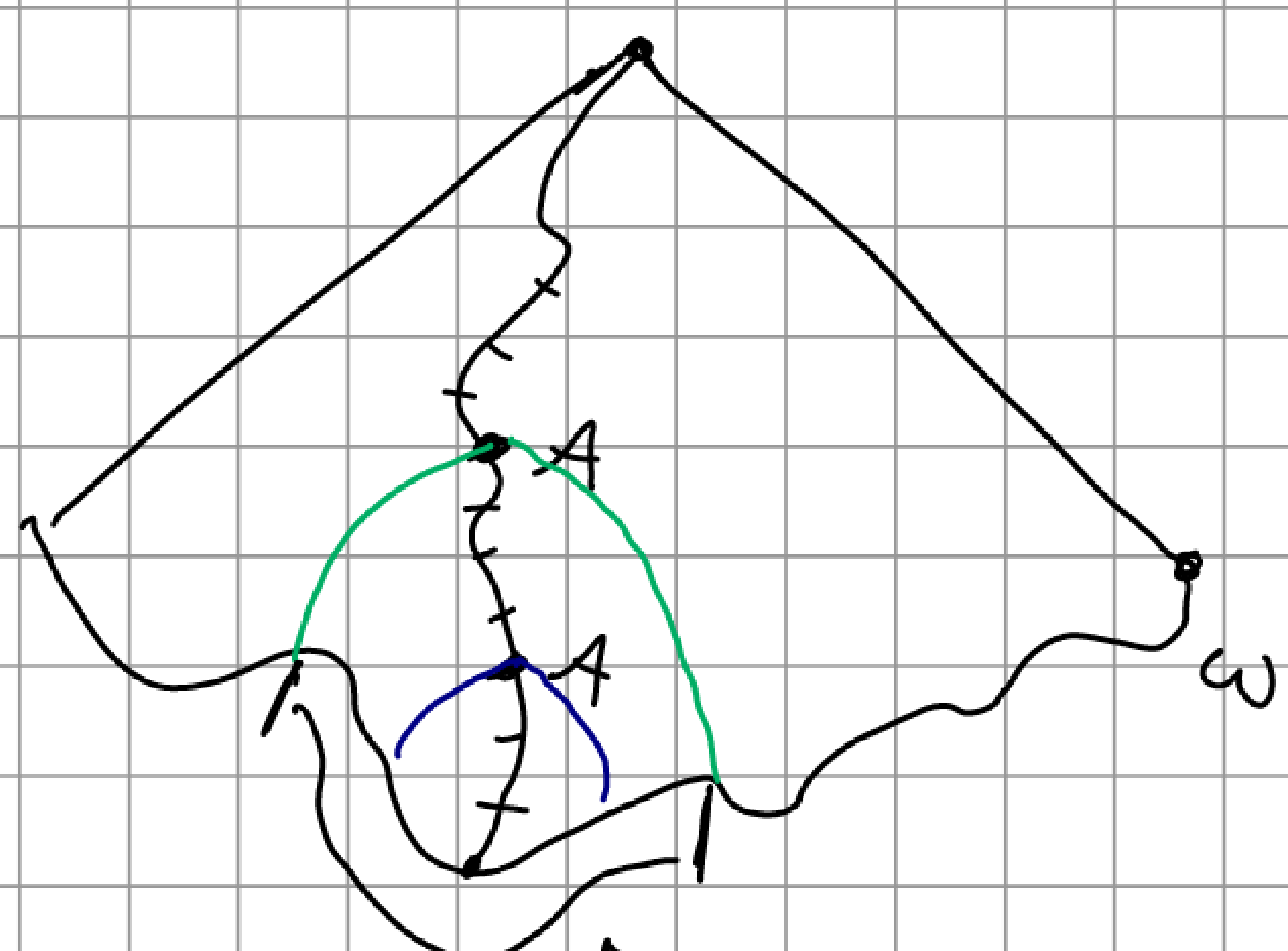
Weźmy $L \in CFL$ oraz CFG $G = \langle N, \Sigma, S, \Pi \rangle$ w postaci normalnej Chomsky'ego i niech $n = 2^{|N|+3}$. Niech $w \in L$, $|w| \geq n$.



Drzewo słowa w.

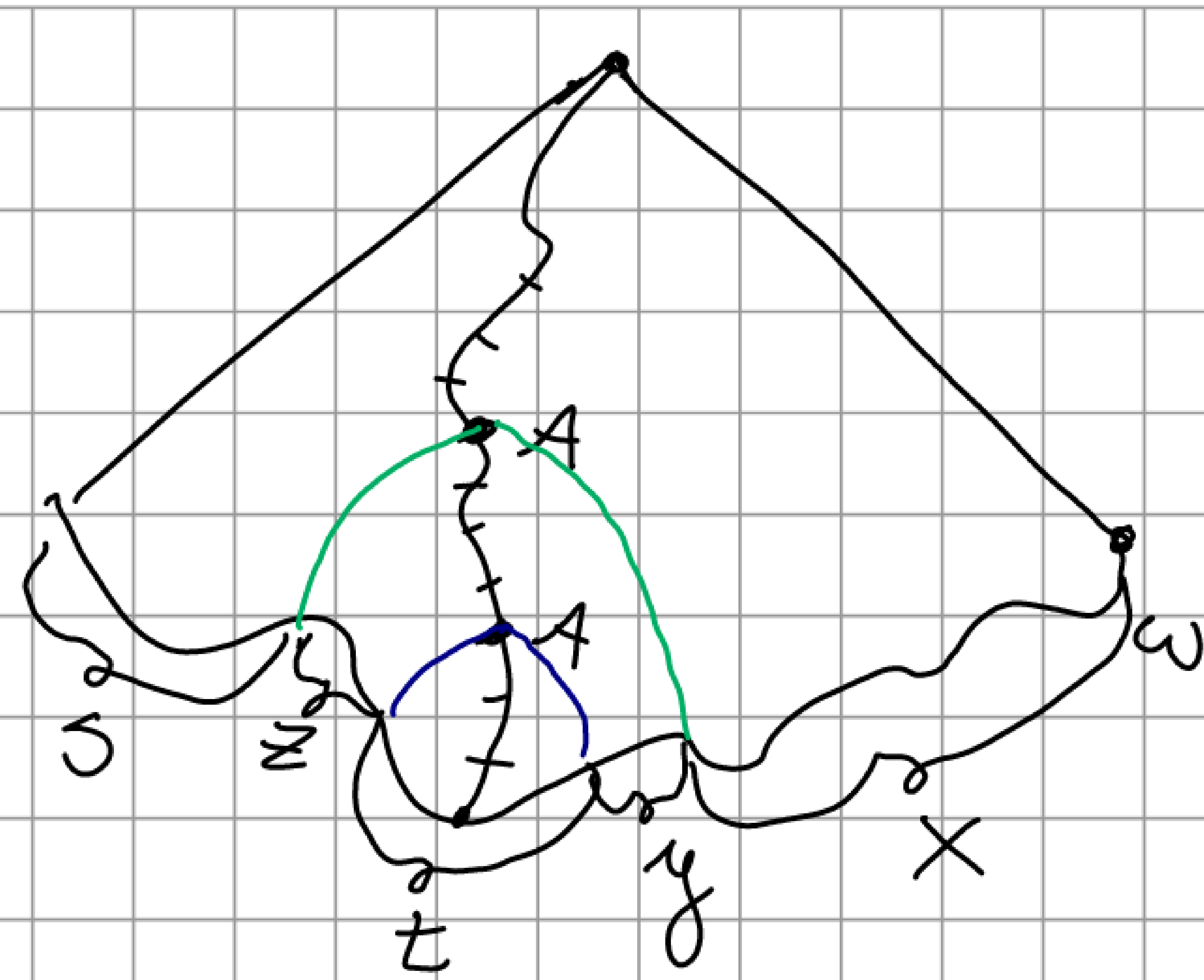
Jaka jest najdłuższa ścieżka od korzenia do liścia? Co najmniej ma długość $|N|+3$ ($\log n$).

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



to ma dł. $\leq 2^{|N|+1}$
(no bo tylko A się
może powtórzyć na
tej ścieżce)

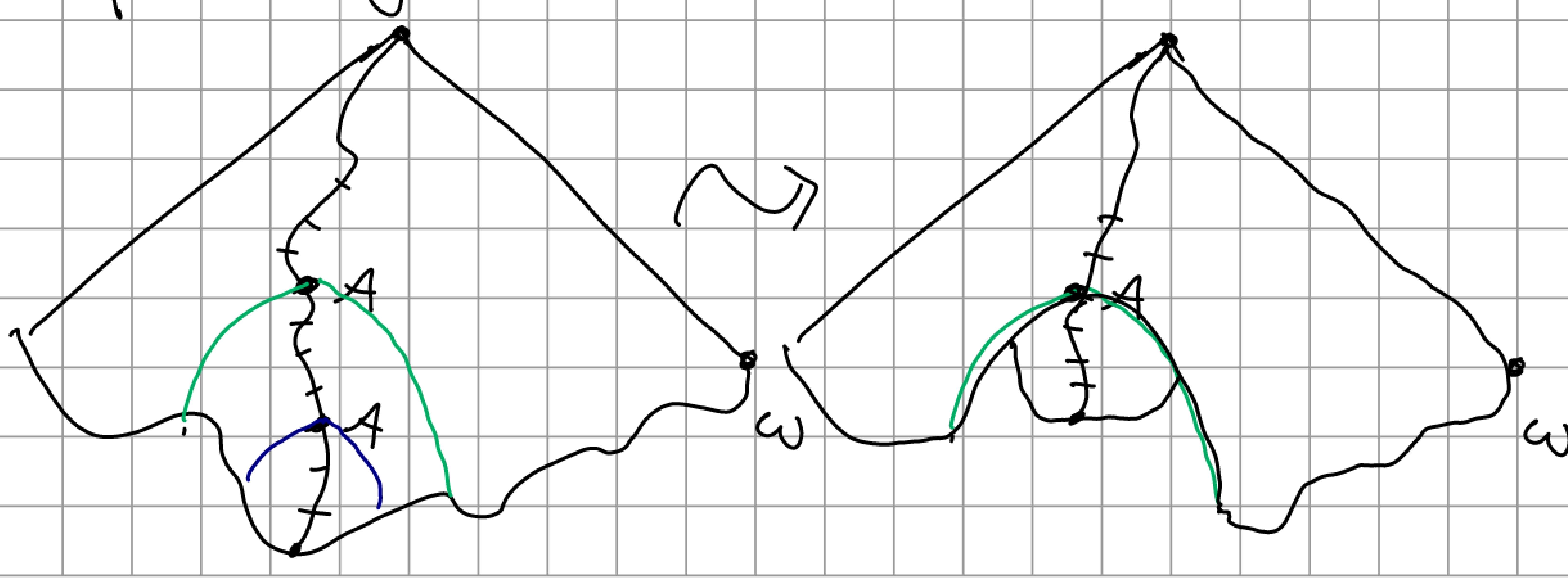
Poddrzewo jest naturalny:



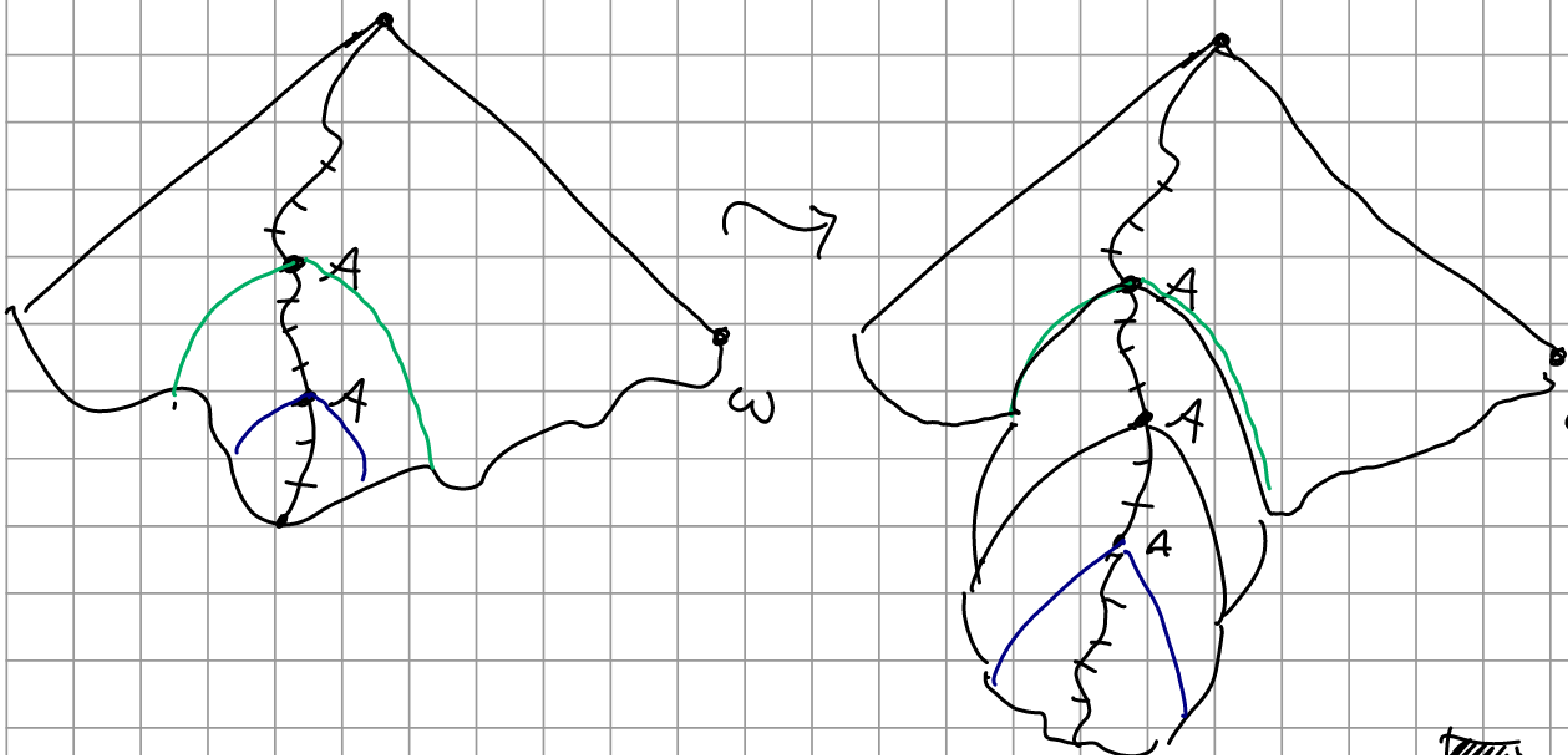
Wtedy $|zty| \leq 2^{|N|+1} < n$.

Ponadto $|zy| > 0$ (to niższe wystąpienie A jest potomkiem tylko jednego dziecka tego wystąpienia wyżej).

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



• gdy $k \geq 1$: "replikujemy" tę wyższą produkcję tyle ile trzeba.



∎

AUTOMATY ZE STOSEM

Def. **Automat ze stosem** (NPDA: Nondeterministic Push Down Automaton) to krotka

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

- Σ – alfabet skończony
- Q – zbiór stanów
- $q_0 \in Q$
- S – zbiór kolorów sweterków (skończony)
- $Z \in S$ – sweterek dna stosu
- δ – relacja przejścia

gdzie $\delta \subseteq (Q \times (\Sigma \cup \{\varepsilon\}) \times S) \times (Q \times S^*)$

oraz ma takie własności:

- $\delta(q, a, Z, q', w) \Rightarrow w = Zv$ 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:

- $\hat{\delta}(\varepsilon, q_0, Z)$
- $\hat{\delta}(w, q, VA) \wedge \delta(q, a, A, q', W) \Rightarrow \hat{\delta}(wa, q', VW) \quad (a \in \Sigma \cup \{\varepsilon\})$.

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

Przykład Palindromy parzyste:

$$\delta(q_1, B, A, q_1, AB)$$
$$\delta(q_1, \varepsilon, A, q_2, A)$$
$$\delta(q_2, A, A, q_2, \varepsilon)$$

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

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

Uwaga (bardzo ważna) Klasy rozstrzygnane przez maszyny deterministyczne są zamknięte na dopełnienia.

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

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

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

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

- Jeśli w Π jest produkcja $A \to a \underset{N^*}{w}$, to $\delta(q, a, A, q, w^R)$,
- $\delta(q_0, \varepsilon, Z, q, ZS)$,

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

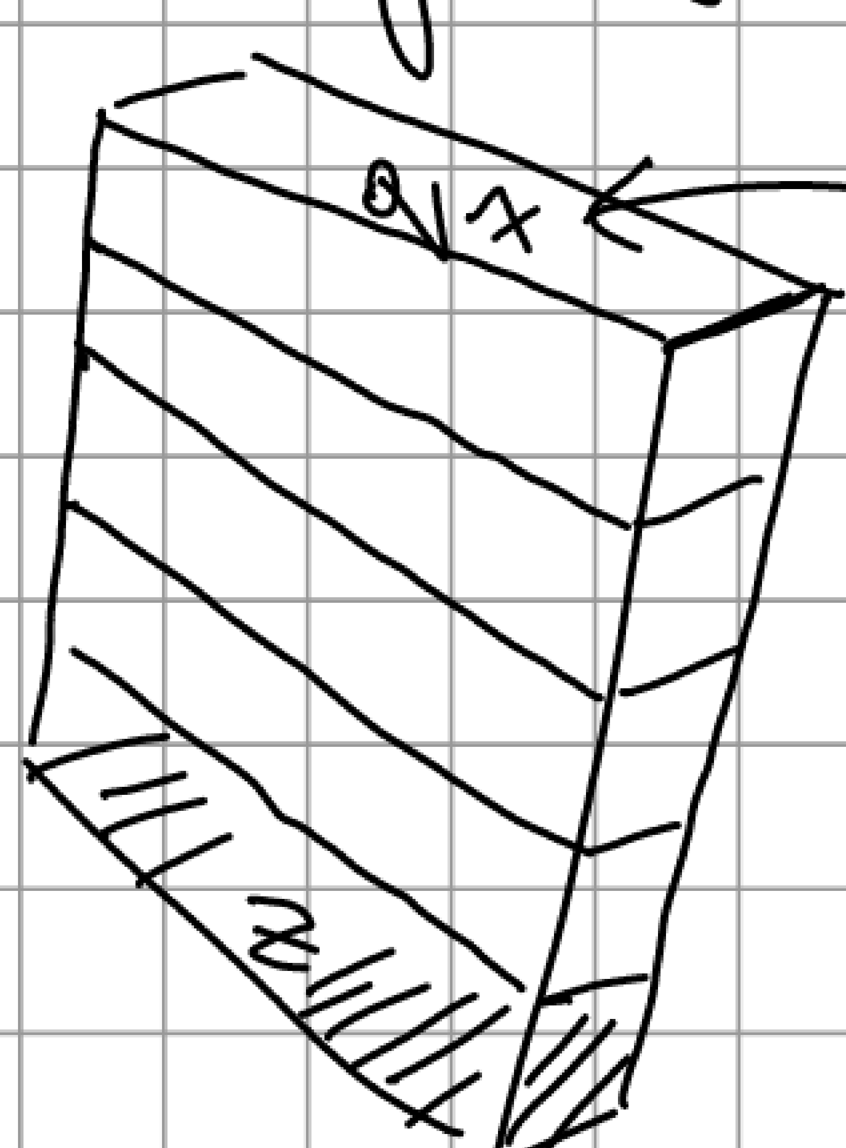
28.03.2022

Kontynuacje dowodu NPDA → CFG

NPDA → NPDA z JEDNYM STANEM → CFG

tą część już potrafimy, analogicznie do poprzedniej części dowodu.

Teraz stos to kolorowe klocki, na wierzchu których napisany jest jakiś stan



"wyobraźcie sobie, że oglądając ten klocek jesteś w stanie q_7"

Problem pojawia się przy ściąganiu klocków. Ten napisany stan może być już nieaktualny.

stary automat

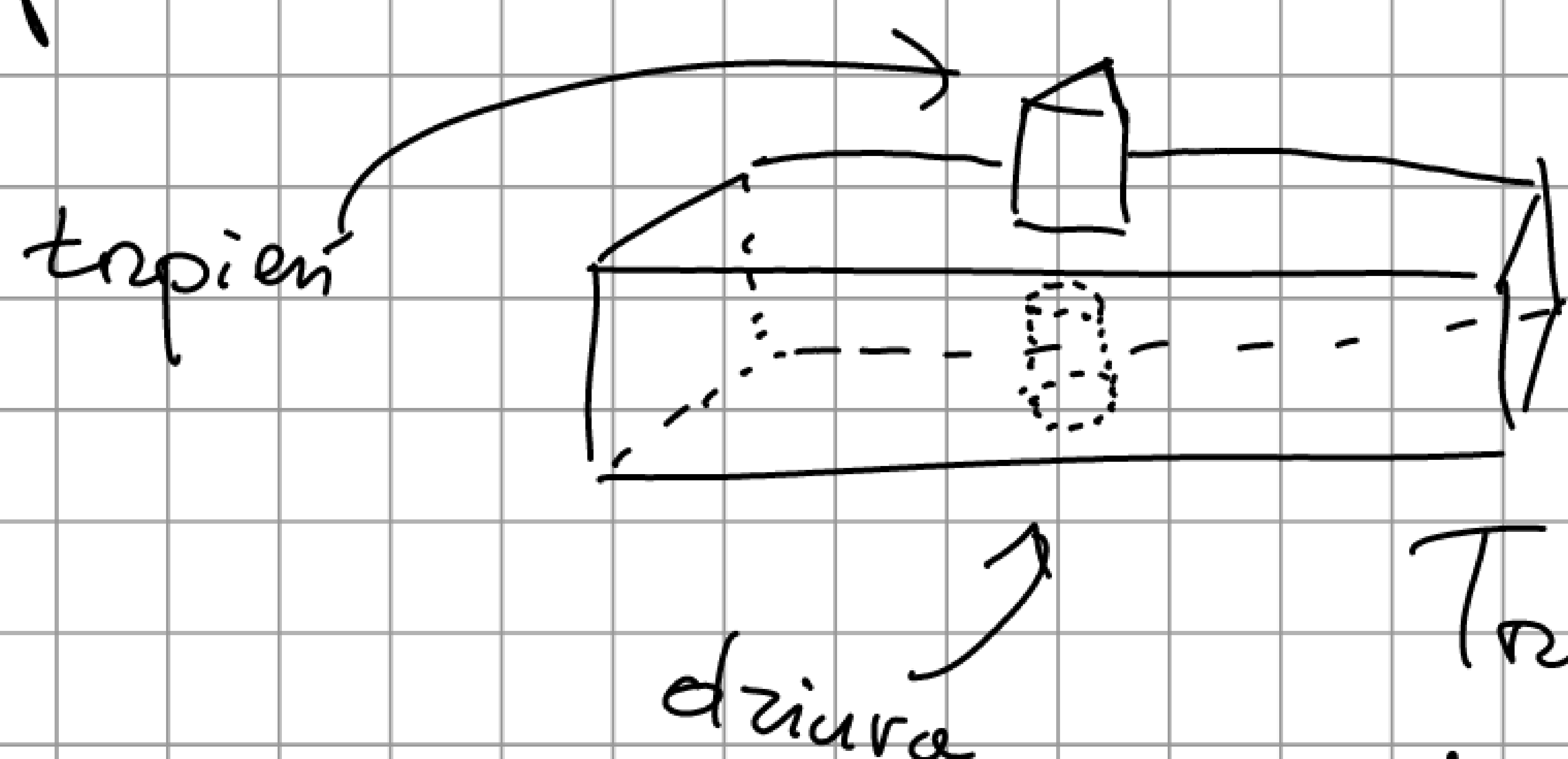
Np.: $\langle q_7$, czerwony, a, q_3, ziel-nieb-ziel-czerw$\rangle$

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:



Trzpieni jest tyle co stanów.

Nasze stany: $\langle$trzpień, dziura, kolor$\rangle$

Instrukcje starego automatu:

$\langle q, K, a; q', A_1 A_2 \ldots A_l \rangle$, $l \geq 1$

wierzch

zastępujemy zbiorem wszystkich instrukcji postaci

(pomijamy stan: jest tylko jeden)

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

dziurka, kolor, korzeń

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

Instrukcje starego automatu:

$$\langle q, K, a; q', \varepsilon \rangle \longrightarrow \langle [q', K, q], a; \varepsilon \rangle$$

Poprawność: Jasne jest, że każdy przebieg starego automatu można zasymulować 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 zwrócą wynik, to on też będzie liczbą naturalną.

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

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

Obserwacja: Każdy zbiór skończony jest rekurencyjny. Klasa zbiorów rekurencyjnych jest zamknięta na sumę, przecięcie i dopełnienie.

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

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

(i) $\varphi(n) = 1 \iff n \in A$ (równoważne definicje)

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

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

φ się zapętla na n.

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

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

Numerujemy wszystkie programy EFEKTYWNIE, tzn. mamy inny program, który może wczytać program i zwrócić jego numer oraz dla numera zwraca program.

	1	2	3	4	5	...
φ_1	$\perp$	$\perp$	$\perp$			
φ_2	4	2137	28			
φ_3	1	$\perp$	$\perp$			
φ_4	0	7	$\perp$			
$\vdots$						

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

(zbiór tych miejsc na przekątnej, gdzie jest liczba, czyli te programy które się zatrzymują dla swojego numeru)

<u>Obserwacja</u> K jest r.e.

Umiemy policzyć φ_n i uruchamiamy go na n.

TW. (Turinga o nierozstrzygalności problemu stopu)
K jest nierozstrzygalny.

D-d. Dowód nie wprost. Załóżmy że
K jest rozstrzygany przez pewien program ψ.
Niech φ będzie programem, który:

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

Wtedy φ ma pewien numer n.

- Jeśli $\psi(n) = 1$, to znaczy, że φ nie zapętla się na n, ale z definicji φ powinien się zapętlić na n.
- Jeśli $\psi(n) = 0$, to znaczy, że φ nie powinien się zatrzymać, ale z jego definicji φ się zatrzyma na n.

$$n \in K \iff \varphi_n(n) \in \mathbb{N} \iff \psi(n) = 0 \iff n \notin K$$

$\lightning$

4.04.2022

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

Obserwacja Funkcje rekurencyjne to częściowe funkcje $\varphi: \mathbb{N} \rightharpoonup \mathbb{N}$. Ich ważną podklasą: funkcje rekurencyjne całkowite.

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

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

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

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

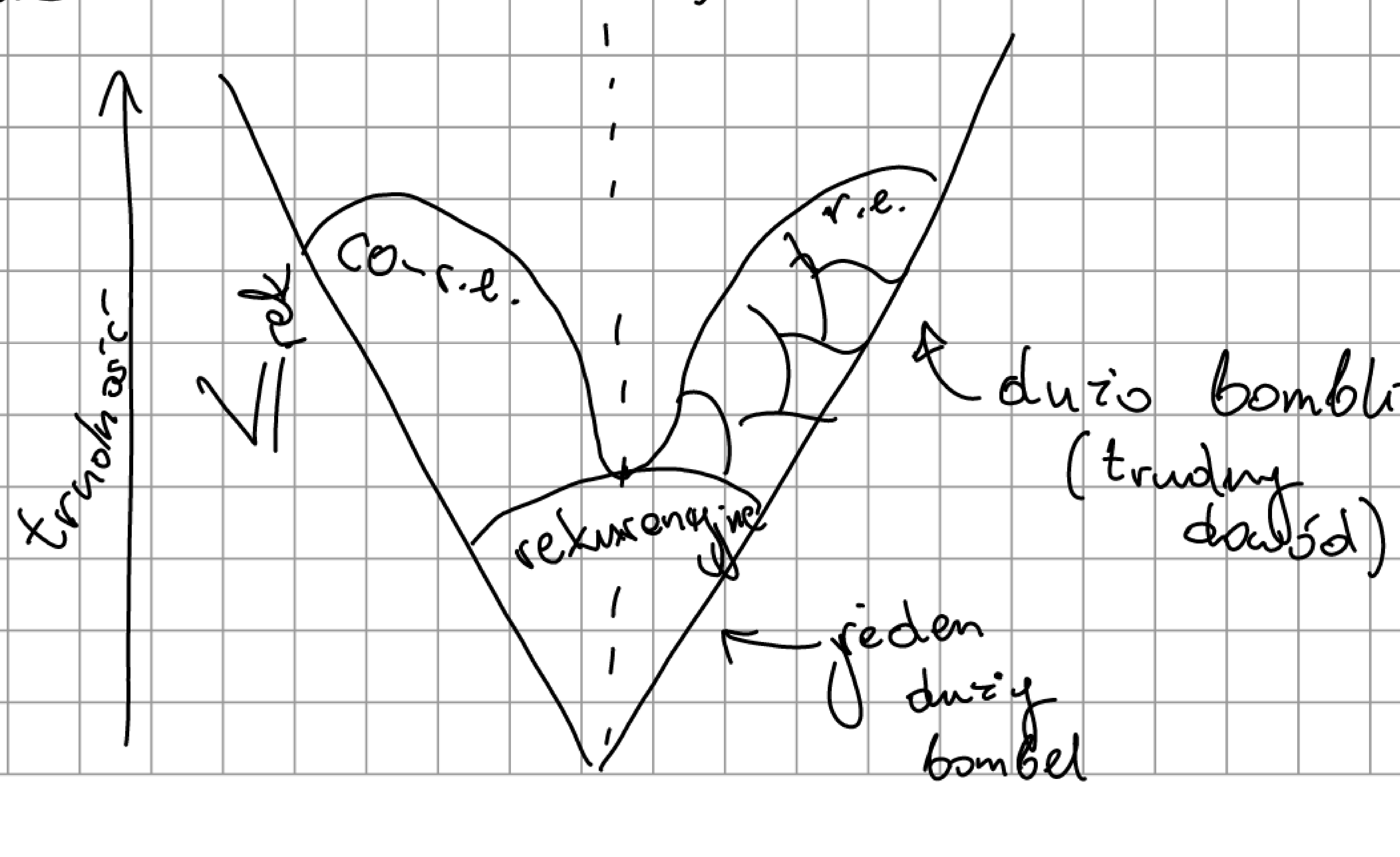
(3) $A \leq_{rek} B$ i B jest r.e., to A też jest r.e.

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

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"

$\leq_{rek}$ dzieli $P(\mathbb{N})$ na "bomble"



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

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

- przynoszą nam n,
- piszemy program: | wczytaj m
uruchom $\varphi_n(n)$
zwróć 77
- zwracamy numer tego programu

To jest funkcja całkowita i do tego dzieła. Gdy $n \in K$, to $\varphi_{f(n)}$ się kończy i zwraca 77 dla dowolnego wejścia, więc $f(n) \in A$. Gdy $n \notin K$, to $\varphi_{f(n)}$ się nie skończy dla każdego wejściu, więc $f(n) \notin A_7$.

<u>Def.</u> $A \subseteq \mathbb{N}$ jest **ekstensjonalny** jeśli $\forall i \in A, j \notin A \ \exists n \ \varphi_i(n) \neq \varphi_j(n)$

może być pinezka

<u>Przykład</u> $A = \{n : \varphi_n$ jest całkowity$\}$

<u>Alt. definicja</u> A ekstensjonalny gdy $\forall i, j \in \mathbb{N}$ jeśli $\varphi_i = \varphi_j$ (jako funkcje), to $i \in A \Leftrightarrow j \in A$

<u>Tw. (Rice'a)</u> Żaden nietrywialny zbiór ekstensjonalny nie jest rekurencyjny.

<u>D-d.</u> Weźmy $A \subseteq \mathbb{N}$: nietrywialny i ekstensjonalny. BSO przyjmijmy, że żaden "numer funkcji pustej" (same pinezki) 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.

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

$w \to_\Pi v$: relacja na Σ^*

$\Updownarrow$ 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 $\to_\Pi$

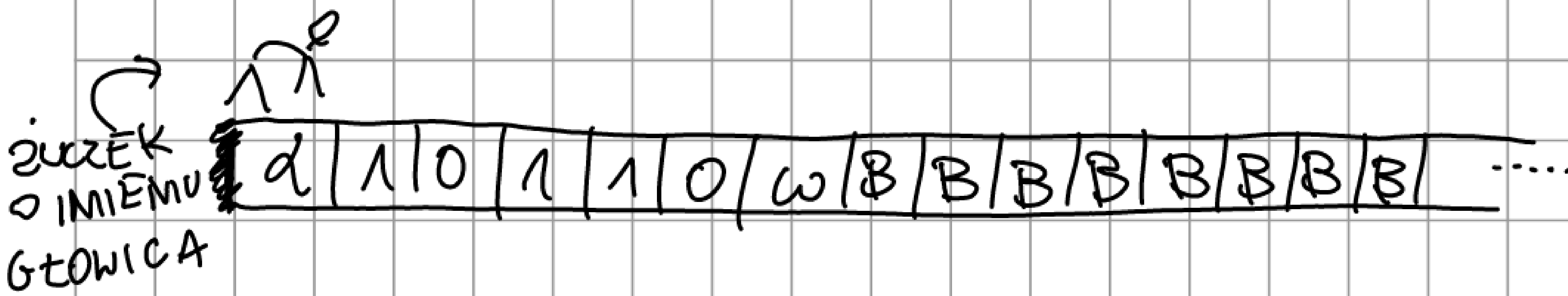
$w \xleftrightarrow{*}_\Pi v$: równoważnościowe domknięcie $\to_\Pi$.

Tw. (o nierozstrzygalności problemu słów Thuego)

Problemy $\text{SemiThue} = \{\langle w, v, \Pi\rangle : w \xrightarrow[\Pi]{*} v\}$,

$\text{Thue} = \{\langle w, v, \Pi\rangle : w \xleftrightarrow{*}_\Pi v\}$ są nierozstrzygalne.

11.04.2022 MASZYNA TURINGA



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

blank, początkowo tym wypełniona taśma

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

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

aktualny stan | literka pod głowicą | nowy stan | na co zmienić literę pod głowicą | przesunąć głowicę w lewo lub prawo.

Własności:

- α jest znakiem końca taśmy, nie można go napisać, zmazać, ani pójść na lewo stojąc na α
- widząc B należy coś napisać
- δ jest f. częściową

- wynikiem jest ciąg znaków na prawo od ω.
- nie ma ruchów z q_F.

Teraz maszyna Turinga to nasz MUJP, dalej mamy K „z tyłu głowy”.

<u>Teza Churcha</u> Każdy algorytm daje się przedstawić jako maszyna Turinga.

<u>D-d.</u> (tw. o nierozstrzygalności słów SemiThue)
(Dają Π, v, w i pytają czy $w \xrightarrow{*}_{\Pi} v$)

Pokażemy, że $K \leq_{rek}$ SemiThue. n-ta maszyna turinga

Konstrukcja redukcji: dają nam M_n, n
mamy zwrócić w, v, Π takie, że $n \in K \iff w \xrightarrow[\Pi]{*} v$

Niech $Z = Q \cup A \cup \{$  $\}$,
← pożeracz

$w = A q_0 \llcorner \underline{\quad n \quad} \lrcorner \omega B$

n zapisane binarnie

Konstrukcja Π:

- dla każdej "instrukcji" z δ, która ma postać
  $\delta(q, a) = \langle q', a', L\rangle$, gdzie $a \neq B$,
  dajemy do Π parę $\langle aq, q'a'\rangle$

- dla każdej instrukcji z δ, która ma postać $\delta(q, B) = \langle q', a', L\rangle$, do Π dajemy $\langle Bq, q'a'B\rangle$ (powiększenie taśmy o blanka)

- dla każdej instrukcji z δ, która ma postać $\delta(q, a) = \langle q', a', R\rangle$, $a \neq B$ do Π dajemy $\langle aqb, a'bq\rangle$ dla każdego $b \in \mathcal{A}$.

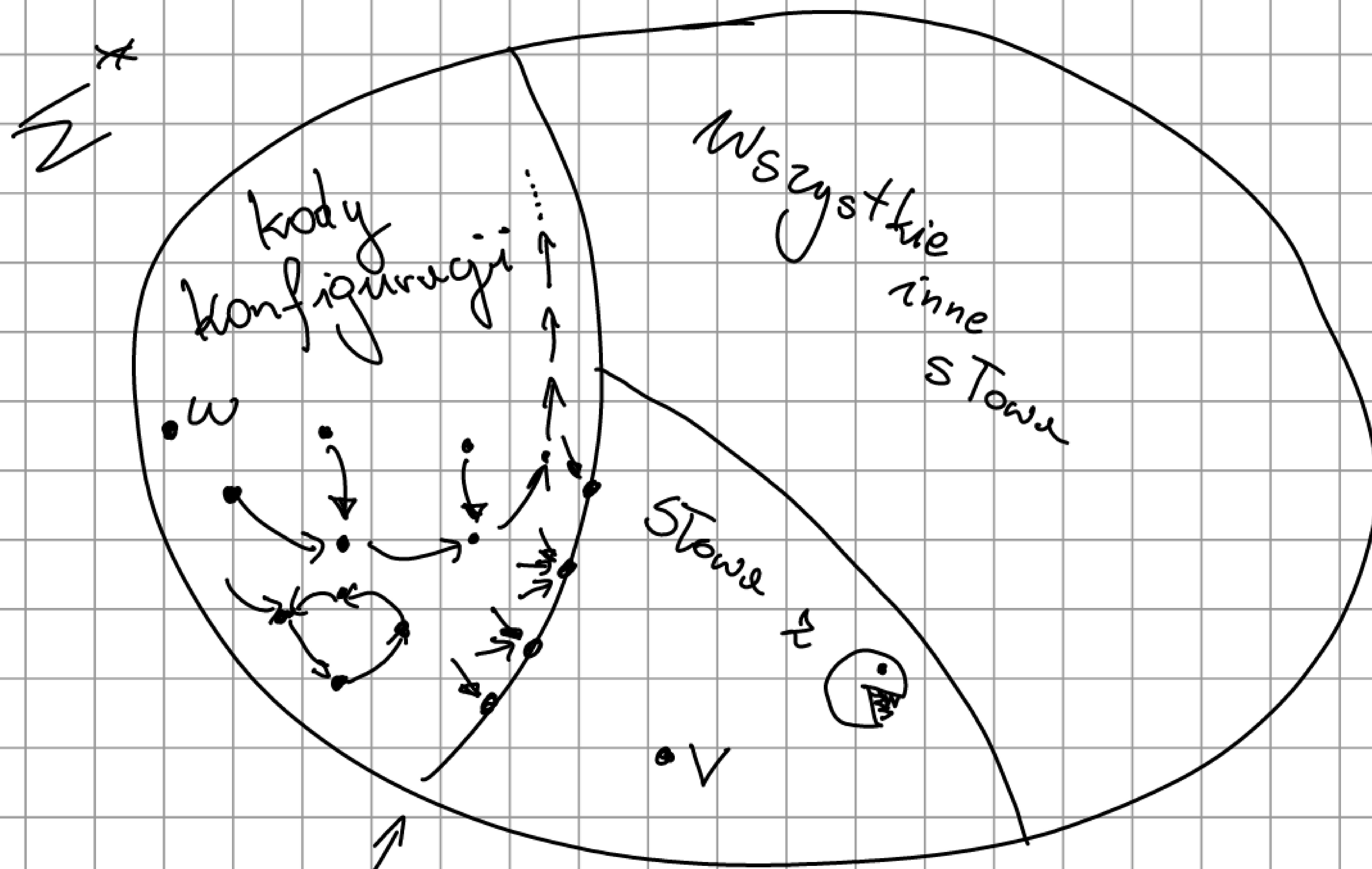
- dla każdej instrukcji z δ, która ma postać $\delta(q, B) = \langle q', a, R\rangle$, do Π dajemy $\langle Bq, aBq'\rangle$

- dla każdego $a \in \mathcal{A}$ w Π są takie pary: $\langle a$ ,  $\rangle$, $\langle$  a,  $\rangle$

• dodajemy do Π parę $\langle q_F , \ \rangle$

Teraz $v =$.

Kod: konfiguracje $M_n \longrightarrow$ słowo nad Σ



tu są słowa z q_F, z nich wszystko wpada w pożeracza. ∎

D-d. (tw. o nierozstrzygalności problemu Thue)

$\{\langle w, v, \Pi \rangle : w \overset{*}{\longleftrightarrow}_{\Pi} v\}$

W zasadzie to samo, co poprzednio.

Trzeba tylko pokazać, że gdy $n \notin K$, to $w \not\overset{*}{\longleftrightarrow}_{\Pi} v$.

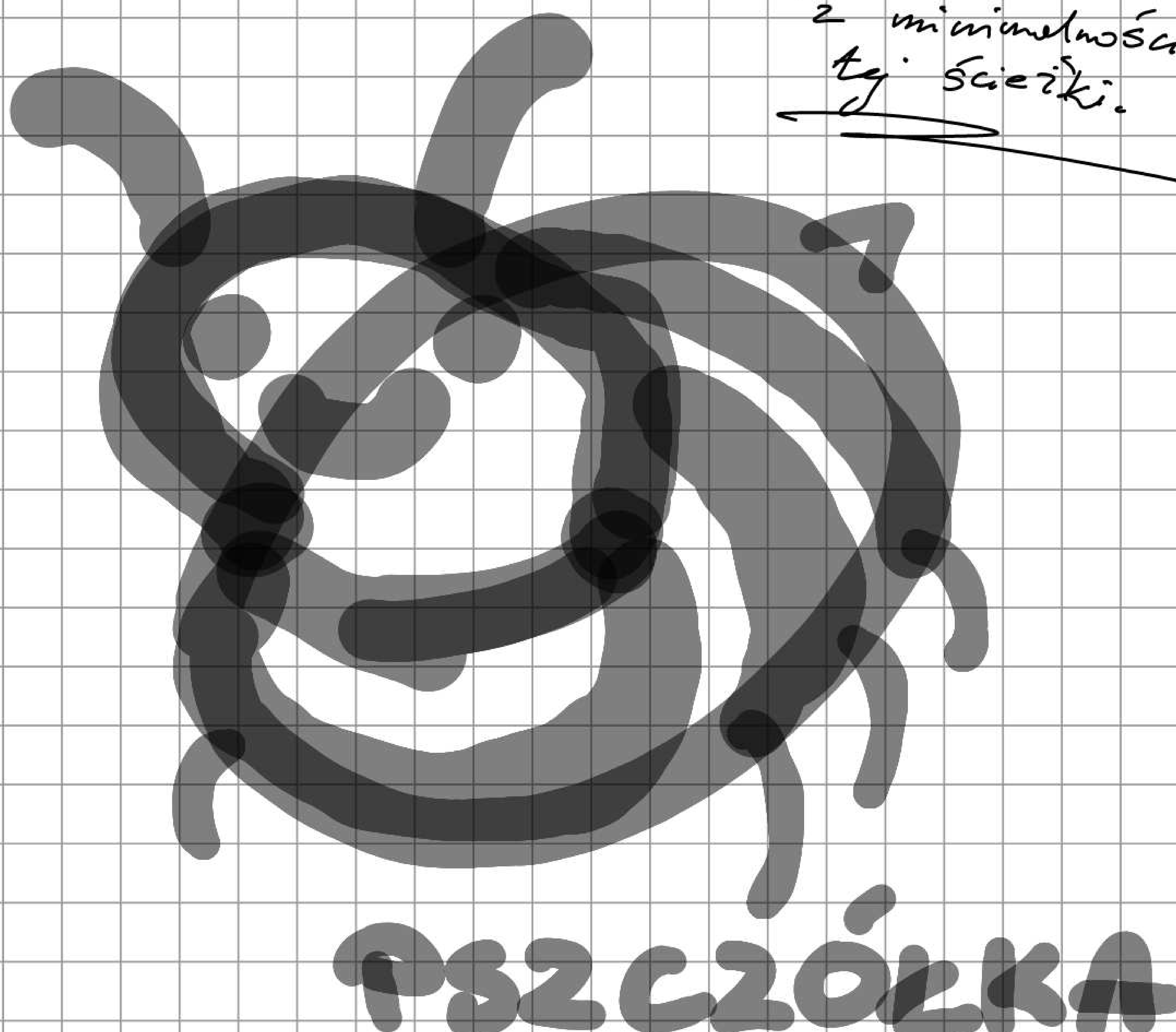
Widać z tego obrazka wyżej!

W°

gdyby tak było, to to musi być ten sam wierzchołek

$\Downarrow$

sprzeczność z minimalnością tej ścieżki.



Dwukierunkowe Automaty Skończone

Q: zbiór stanów, $q_0, q_F \in Q$

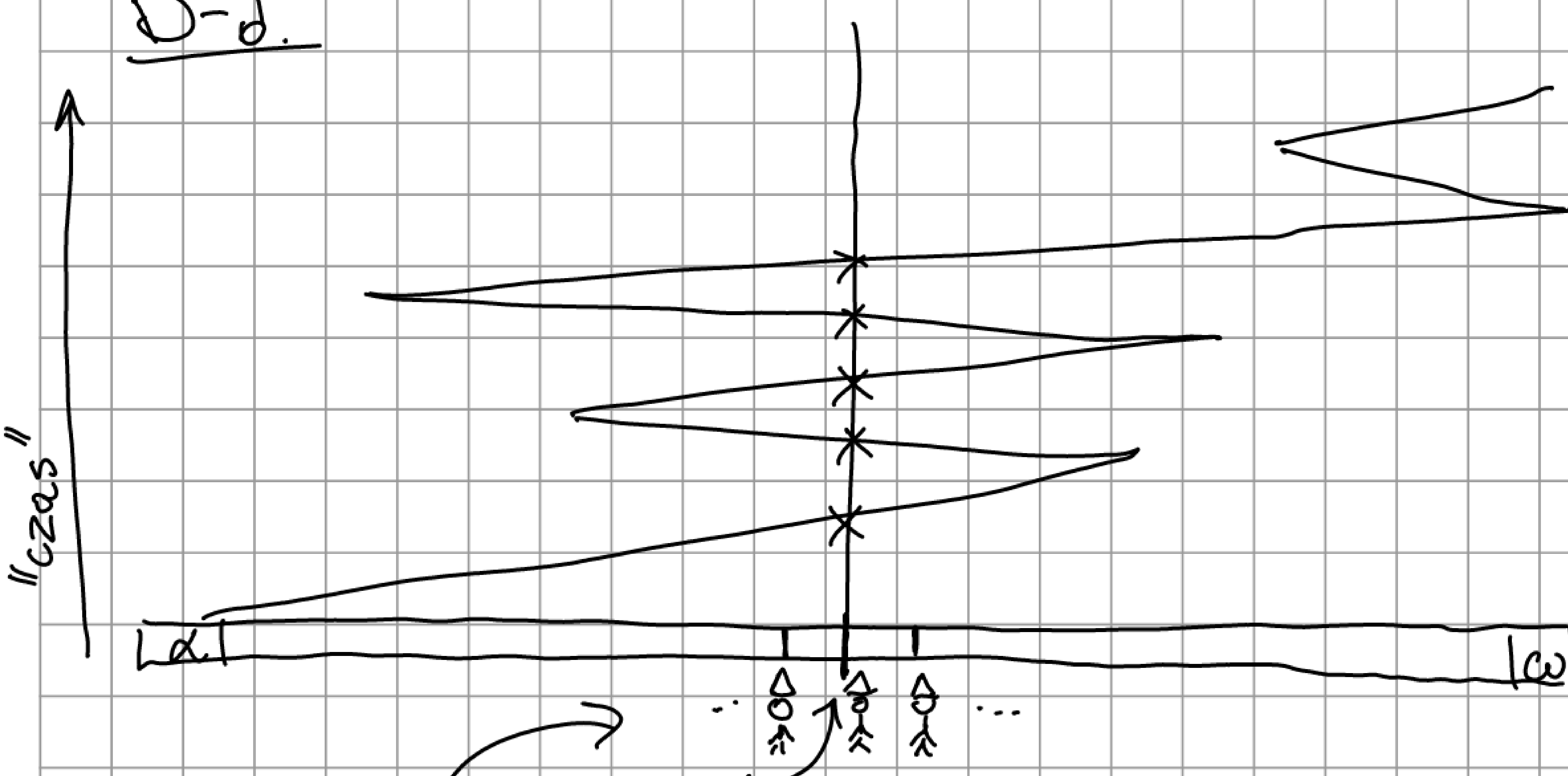
$\Sigma' = \Sigma \cup \{\alpha, \omega\}$

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

Tw. To model wyraża tylko języki regularne

(Chcemy z pszczółki zrobić żuczka
(NFA))

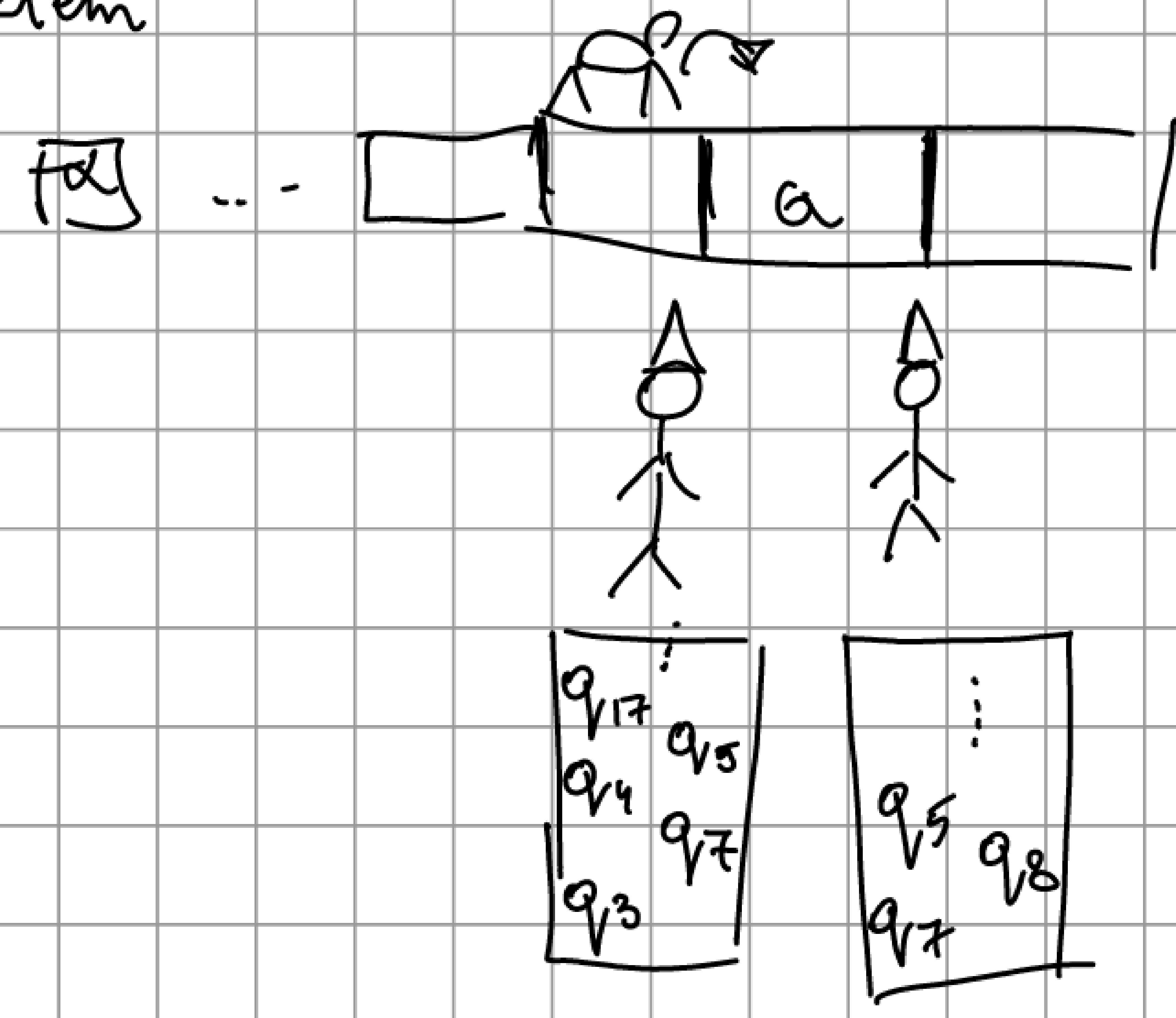
D-d.



punktów przecięcia jest max. $2|Q|$

w przejściach między komórkami jest krasnoludek

Każdy krasnoludek sporządza listę stanów, z których pszczółka przelatywała nad krasnoludkiem



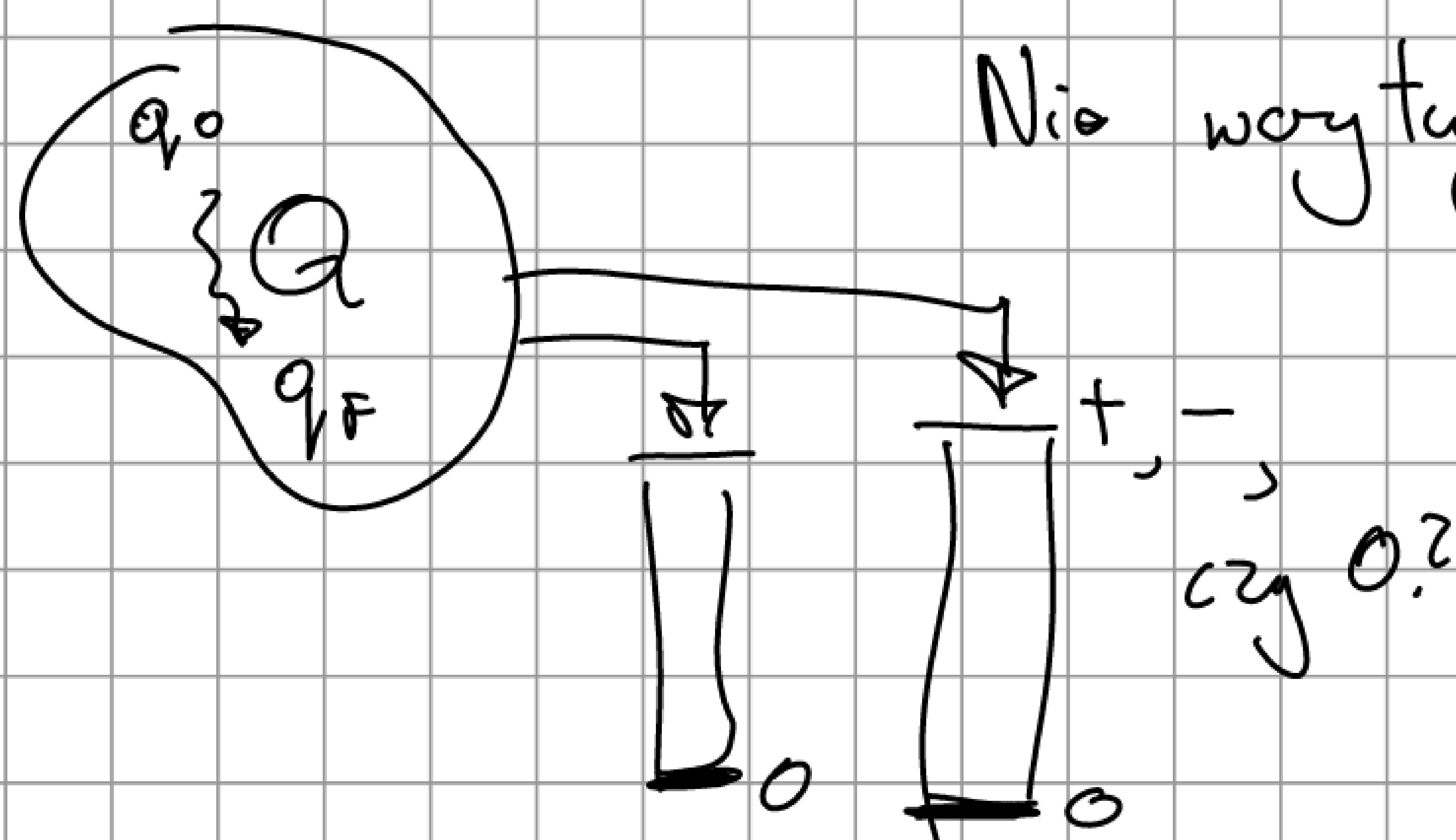
sprawdza, czy te listy są kompatybilne

Fakt Pszczółka nie istnieje: są tylko krasnoludki opowiadające bajki

Obserwacja Niedeterministyczna pszczółka jest taka sama.

9.05.2022 NIEROZSTRZYGALNOŚĆ ARYTMETYKI

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



Nie wczytuje nic − bawi się sam ze sobą.

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

TW Problem rozstrzygania czy dana MM zakończy działanie jest nierozstrzygalny. (Problem MM)

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

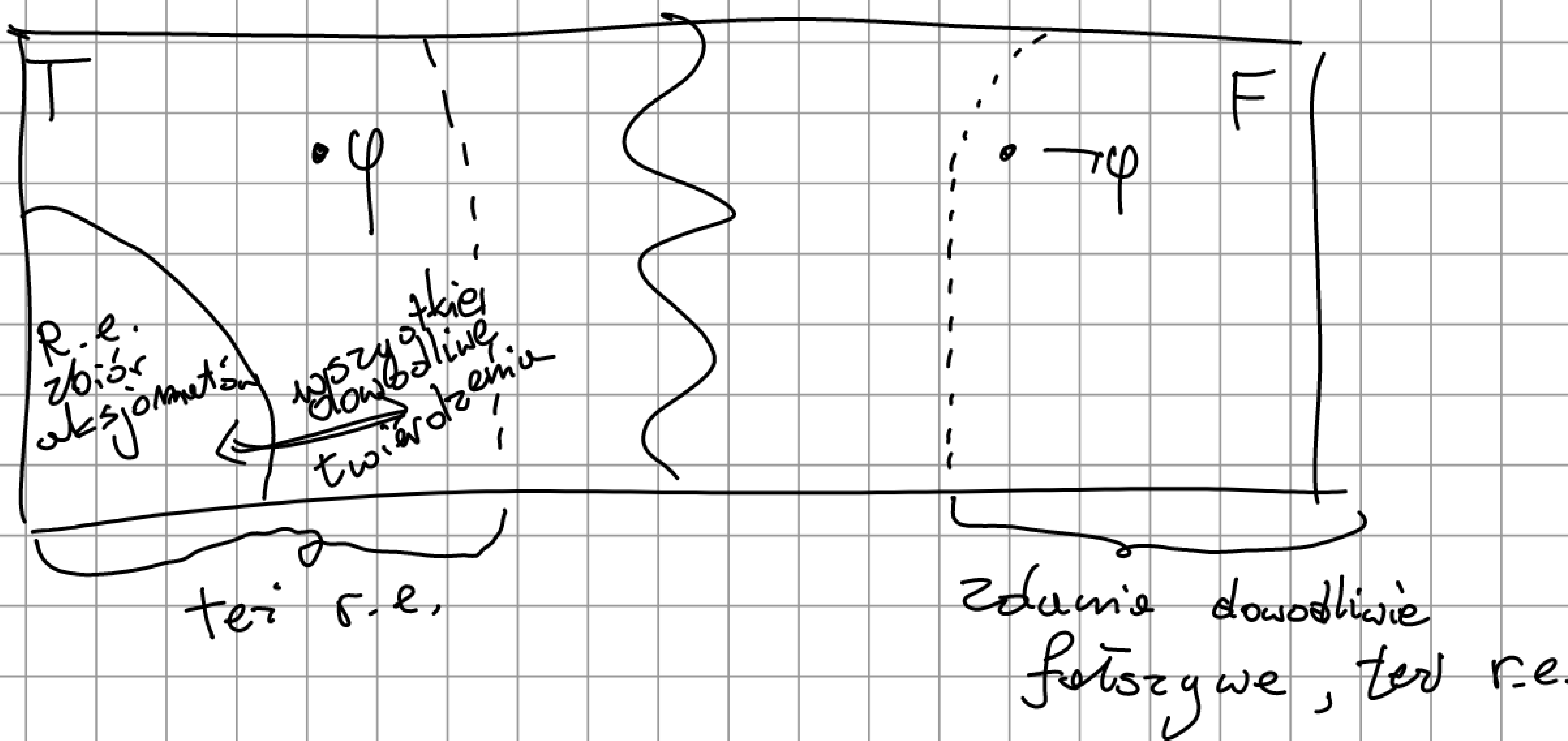
$$\forall n \, \exists m \, \forall k, l \, (m > n \wedge (k \cdot l = m \rightarrow (k = 1 \vee l = 1)))$$

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

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

ODP: NIE.

WSZYSTKIE ZDANIA ARYTMETYKI



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

<u>Tw.</u> $Th(\mathbb{N},+,\cdot,\uparrow)$ nie jest rozstrzygalny.

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

Czyli dają nam program M dla MM.

Mamy napisać zdanie Φ_M t.że

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

Potrzebujemy zapamiętać konfigurację M jako trójkę liczb.

Niech $k = |Q|$, $Q = \{q_1, \dots, q_k\}$ gdzie $q_1 = q_0$, $q_2 = q_F$. 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.że C' wynika z C w jednym kroku obliczenia 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 fajną szóstką dla $1 \leq l \leq N-5$

Obserwacja Istnieje formuła arytmetyki ψ z 6 zm. wolnymi taka, że

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

$x_1, \dots, x_6$ są fajne.

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

sprawdza czy X_2 jest w stanie opisanym przez j

$$\bigwedge_{\substack{i \in \{1 \dots k\} \\ j, j' \in \{0,1\}}} \Big[\big(x_1 = i \wedge \chi(x_2, j) \wedge \chi(x_3, j) \big) \to$$

$$X_4 = \Pi_1(\delta(q_i, j, j')) \wedge X_5 = x_2 + \Pi_2(\delta(q_i, j, j'))$$

$$\wedge\ X_6 = X_3 + \Pi_3(\delta(q_i, j, j')) \Big]$$

gdzie te Π_1, Π_2, Π_3 działają jak trzeba

Teraz chcemy formułę, która sprawdza czy ciąg jest fajny. Musimy kodować ciągi jako liczby: $a_1, \dots, a_N \longrightarrow 2^{a_1} 3^{a_2} \dots p_N^{a_N}$

"Makra":

- pierwsza(x): $\forall y,z \ (yz = x \rightarrow y=1 \lor z=1) \land x > 1$
- kolejne_pierwsze(x,y): $\forall z \ (x<z<y \rightarrow \neg \text{pierwsza}(z)) \land \text{pierwsza}(x) \land \text{pierwsza}(y) \land x<y$
- w_rozkładzie(x,y,z):

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

  $\exists s \forall z,t \ \text{pierwsza}(x) \land sx = y \land (\text{pierwsza}(z) \land z > x \rightarrow zt \neq y)$

Φ_M: $\exists m \ \forall x, x', y, y' \big(\text{kolejne\_pierwsze}(x,x') \land \text{w\_rozkładzie}(x,y,m) \land y' > 0 \land \text{w\_rozkładzie}(x',y',m) \longrightarrow y > 0\big)$

$$\wedge\ \text{w\_rozkładzie}(2,1,m) \wedge \text{w\_rozkładzie}(3,k+1,m)$$

$$\wedge\ \text{w\_rozkładzie}(5,k+1,m)$$

$$\wedge\ \forall_{p,p',p''} \Big(\text{najw\_pierwsza}(p'',m) \wedge \text{kolejne\_pierwsze}(p,p')$$

$$\wedge\ \text{kolejne\_pierwsze}(p',p'') \to \text{w\_rozkładzie}(p,2,m)$$

$$\wedge\ \text{w\_rozkładzie}(p',k+1,m) \wedge \text{w\_rozkładzie}(p'',k+1,m) \Big)$$

$$\wedge\ \forall_{\substack{x_1,x_2,\dots,x_6,x \\ y_1,\dots,y_6}} \Big[\bigwedge_{i=1}^{5} \text{kolejne\_pierwsze}(x_i,x_{i+1}) \wedge \text{najw\_pierwsza}(x,m)$$

$$\wedge\ x_6 \leqslant x \wedge \bigwedge_{i=1}^{6} \text{w\_rozkładzie}(x_i,y_i,m) \to \varphi(y_1,\dots,y_6) \Big]$$

$\blacksquare$

X PROBLEM HILBERTA

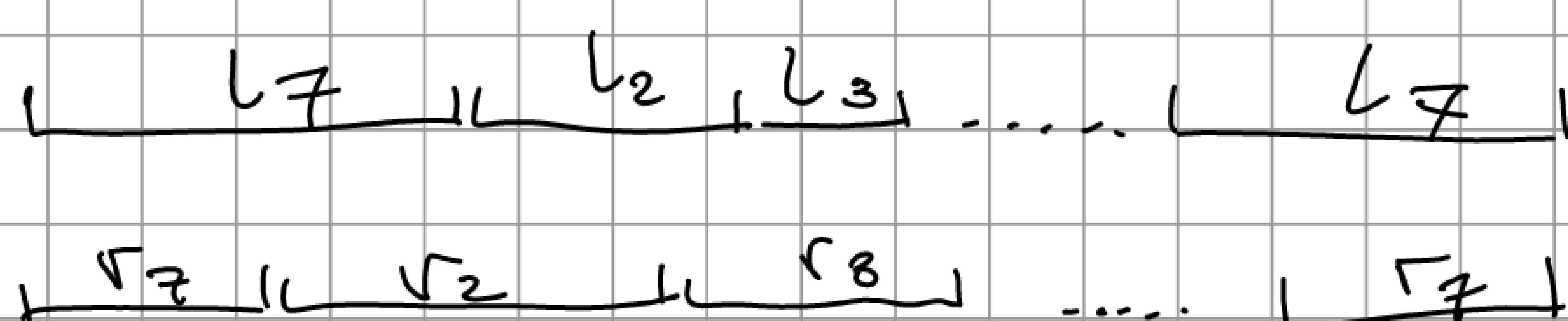
Czy jest algorytm rozwiązujący układ równań diofantycznych?

ODP: Nie, to jest nierozstrzygalne (trudne)

16.05.2022 Problem Odpowiedniości 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 Σ.

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



Tw. Problem odpowiedniości Posta jest nierozstrzygalny.

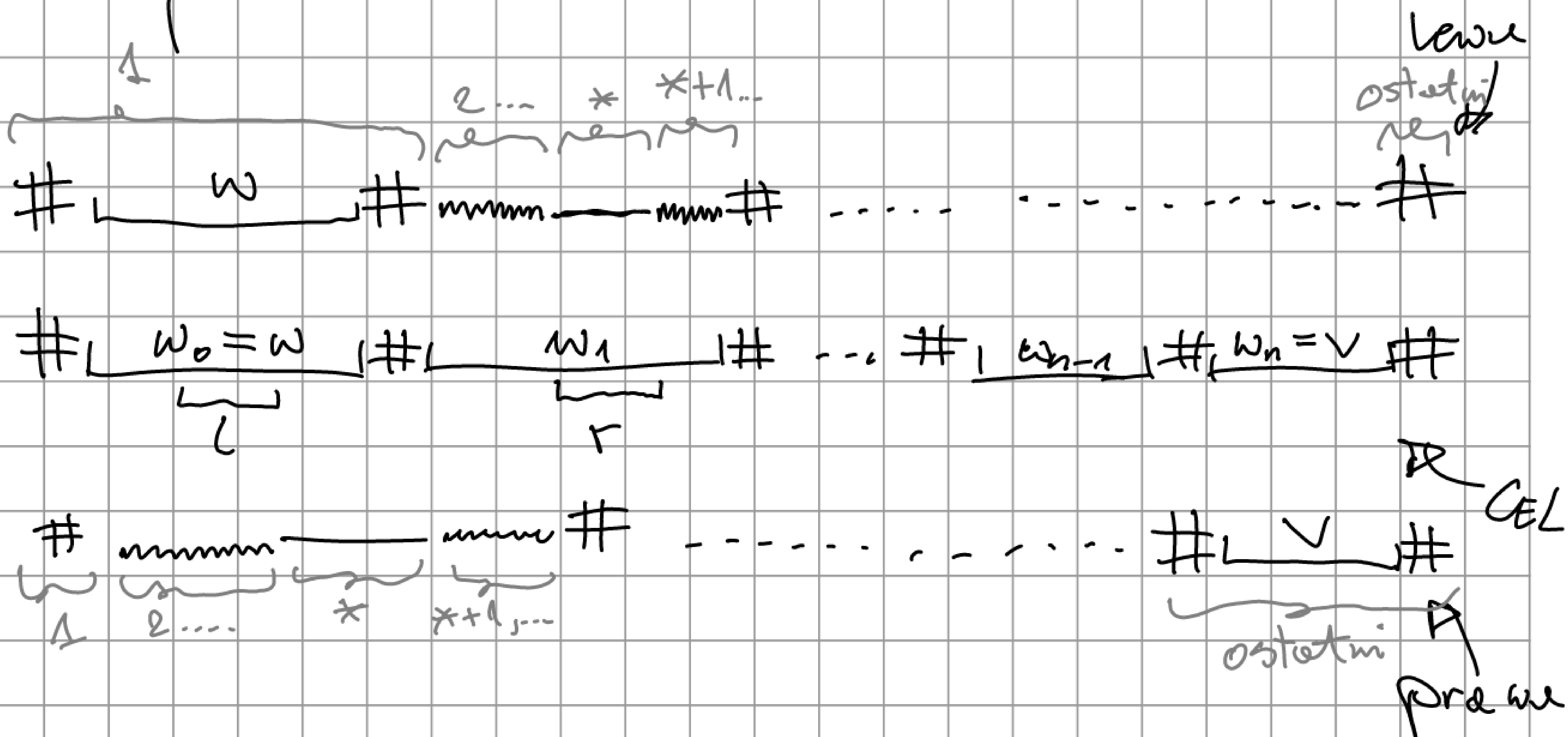
D-d. I przymiarka dowodu: SemiThue $\leq_{rek}$ PCP.

Dają nam $\langle w, v, \Pi\rangle$. Mamy wyprodukować instancję PCP P t.że $w \xrightarrow{*}_{\Pi} v \iff P$ ma rozwiązanie.

Niech $\mathcal{A}$: alfabet Π. Wtedy $\Sigma = \mathcal{A} \sqcup \{\#\}$.

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

Takie P zawsze działa, ale pokażemy tylko $\Rightarrow$ w "fajny i zadowalający" sposób.

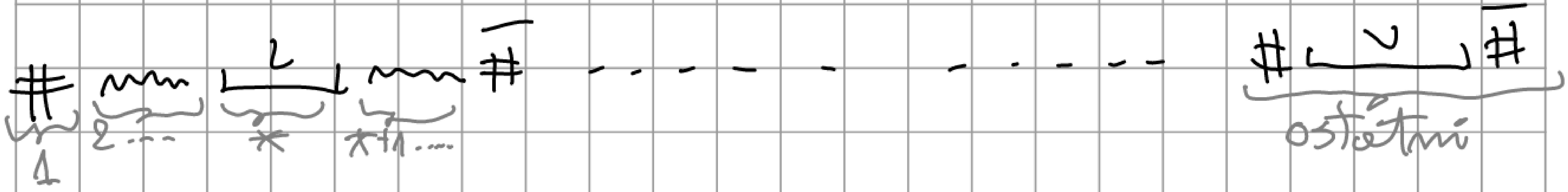


II podejście: $\Sigma = \{\#, \overline{\#}\} \sqcup \{a, \overline{a} : a \in A\}$.

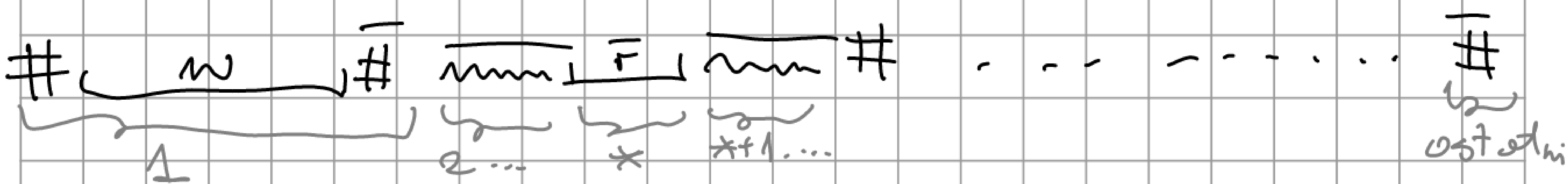
$\overline{a} \notin A$

$$P = \{\langle \#, \# w \overline{\#} \rangle, \langle \# v \overline{\#}, \overline{\#} \rangle\}$$
$$\cup \{\langle \overline{l}, r\rangle, \langle l, \overline{r}\rangle : \langle l, r\rangle \in \Pi\}$$
$$\cup \{\langle \#, \overline{\#}\rangle, \langle \overline{\#}, \#\rangle\} \cup \{\langle a, \overline{a}\rangle, \langle \overline{a}, a\rangle : a \in A\}.$$

Wtedy:



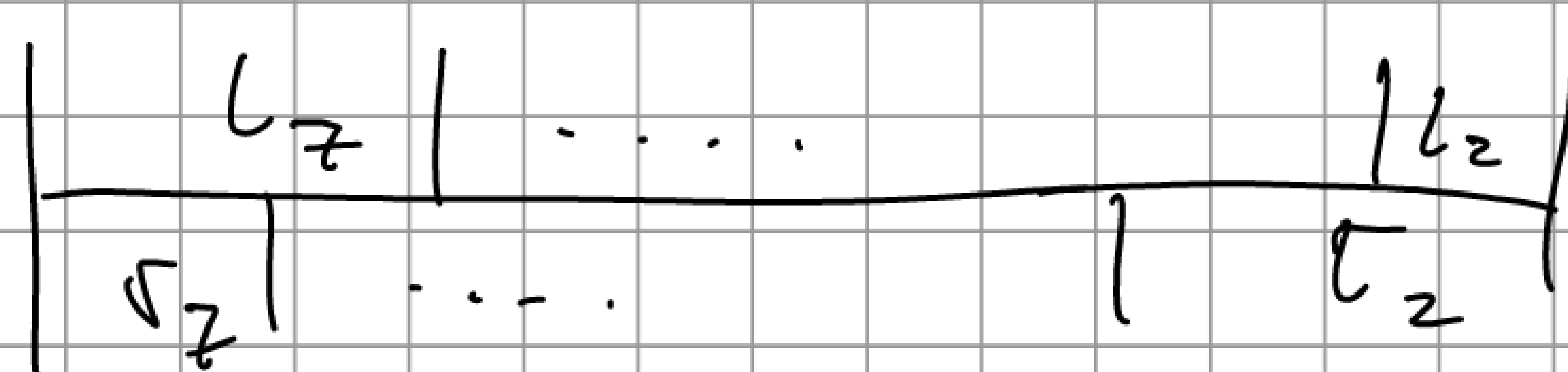
$$\#\ w_0 = w\ \overline{\#}\ \overline{w_1}\ \#\ w_2\ \overline{\#}\ \ldots\ \overline{\#}\ \overline{w_{n-1}}\ \#\ w_n = v\ \overline{\#}$$



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

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

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



r_z jest prefiksem l_z.
l_2 jest sufiksem r_2

Zatem musimy zacząć od $\langle \#, \#w\overline{\#}\rangle$

i skończyć na $\langle \#v\overline{\#}, \overline{\#}\rangle$.

$\# \; w \; \overline{\#} \; x \; \# \; y \; \overline{\#} \; \ldots\ldots \; \overline{\#} \; v \; \#$

Lemat $\quad x \xrightarrow{*}_{\Pi} y$

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

Dalej indukcja i elo!

z dobrymi założeniami.

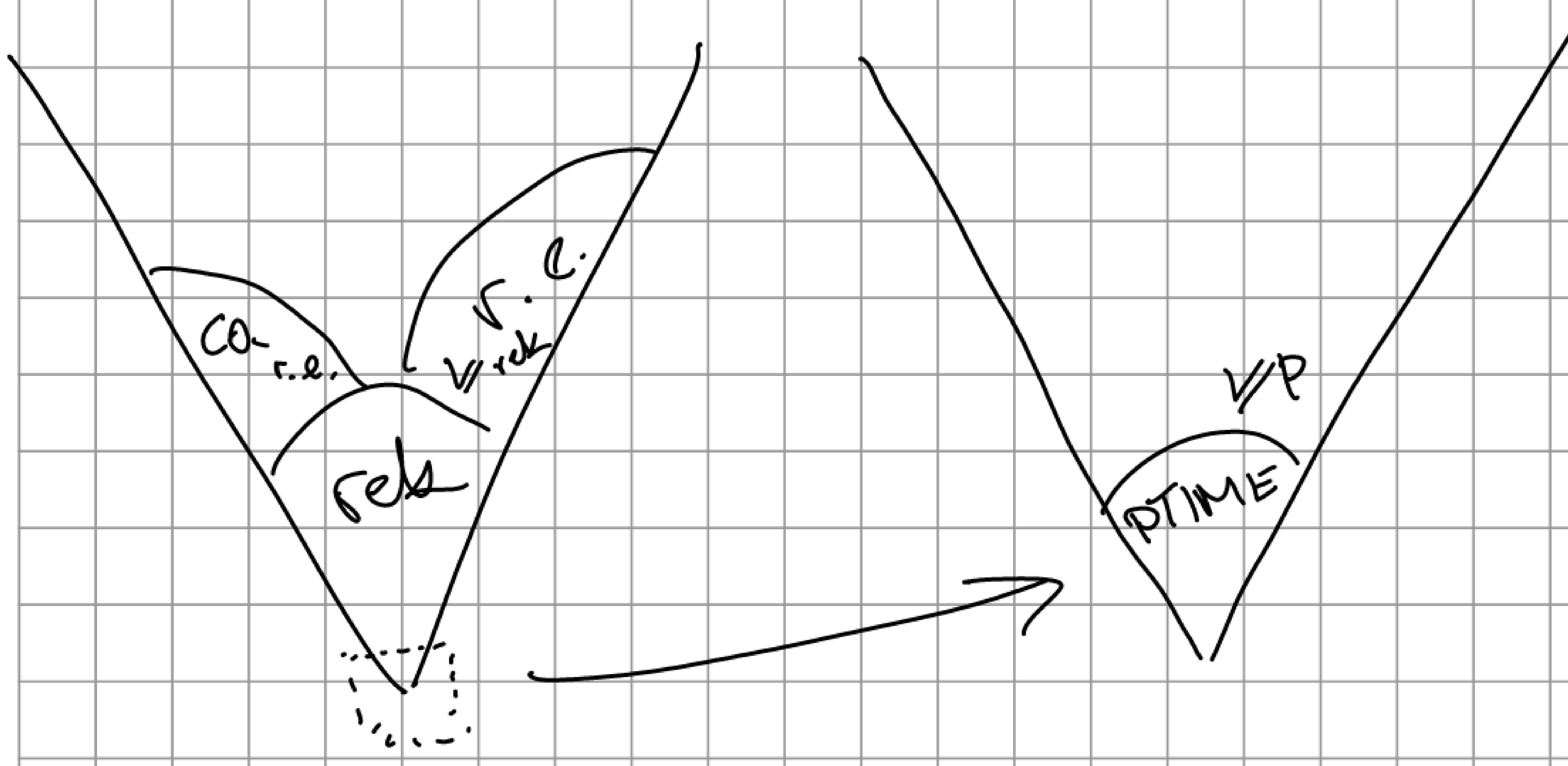
$\blacksquare$

Koniec teorii 2.

III CZĘŚĆ KURSU ....

Def. $A \subseteq \Sigma^*$ jest w **PTIME** (dla przyjaciół 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.



$$\begin{array}{ccc} \Sigma_A^* & & \Sigma_B^* \\ \cup\!\!\!\!| & & \cup\!\!\!\!| \end{array}$$

Def. $A \leqslant_p B$ (czytamy: „A jest nietrudniejsze od B w sensie redukcji wielomianowych) gdy istnieje $\underbrace{f : \Sigma_A^* \to \Sigma_B^*}_{\text{zwana redukcją wielomianową}}$ o następujących własnościach:

- istnieje wielomian p oraz MT M t.że $\forall a \in \Sigma_A^*$ $M(a)$ zatrzymuje się po $\leqslant p(|a|)$ krokach i zwraca wartoś $f(a)$
- $\forall a \in \Sigma_A^* \left(a \in A \iff f(a) \in B \right)$

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

Obserwacja Jeśli $A, B \in PTIME$ i oba są nietrywialne, to $A \leq_p B$ oraz $B \leq_p A$.

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

Pytają czy formuła jest spełnialna.

Problem 3-kolorowania Graf nieskierowany.

Pytają czy da się pokolorować wierzchołki grafu tak żeby żadne dwa sąsiednie były różnych kolorów.

Tw. $3COL \leq_p 3SAT$.

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

Mamy SZYBKO napisać formułę φ_G w postaci 3CNF t.że G da się pokolorować $\iff \varphi_G$ jest spełnialna.

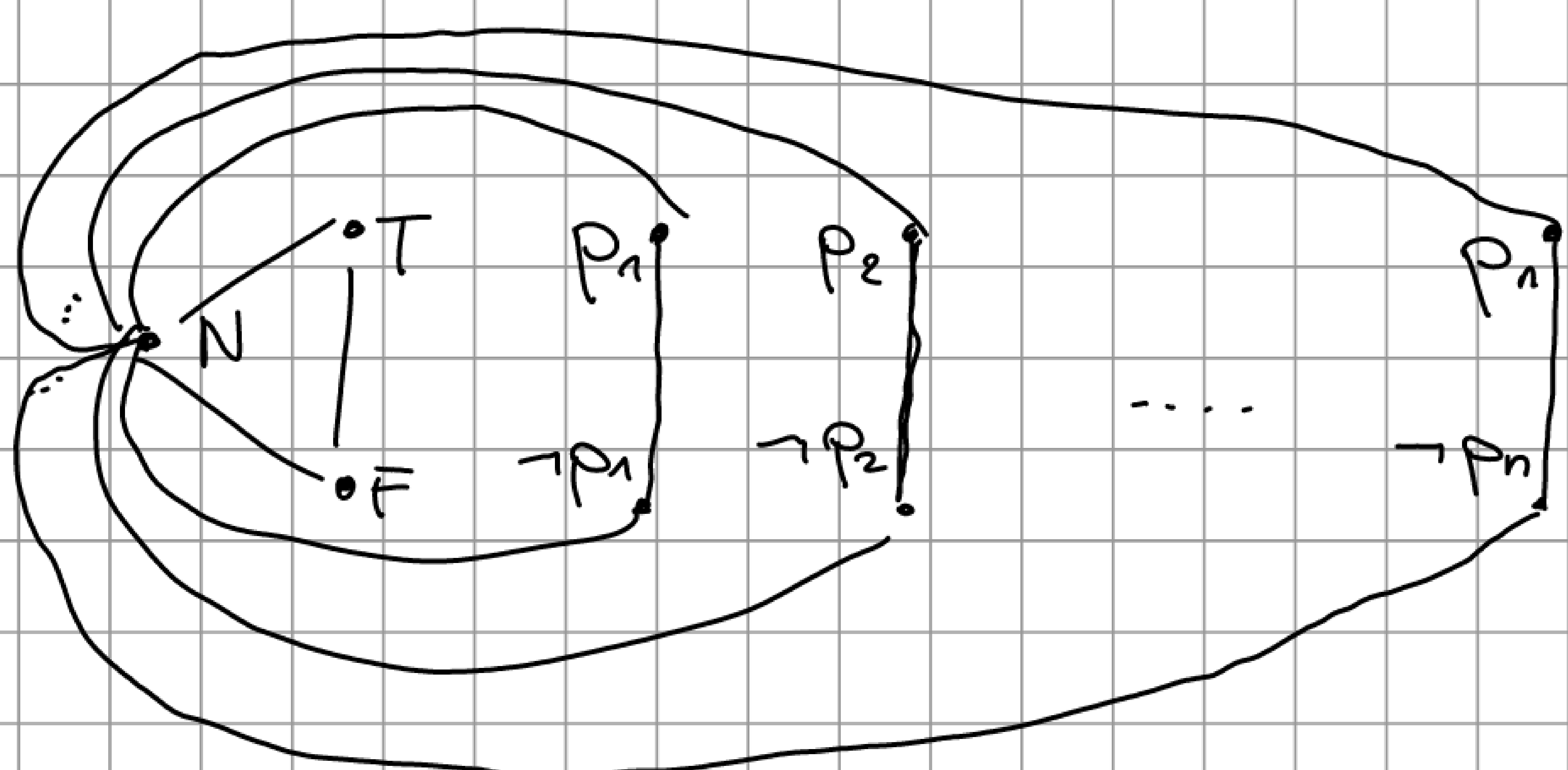
Konstrukcja φ_G: zmiennymi będą r_u, g_u, b_u dla $u \in V$. Klauzule:

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

$\blacksquare$

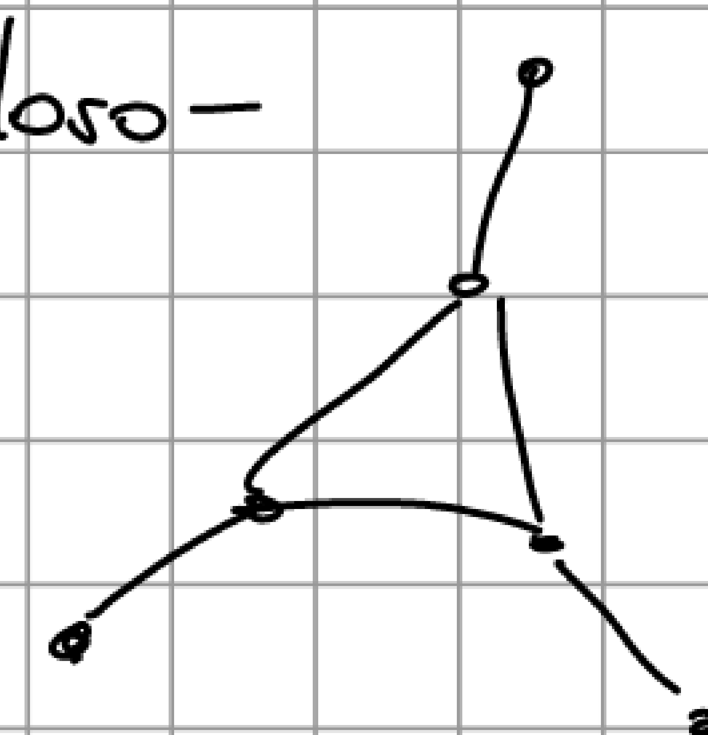
Tw 3SAT $\leqslant_p$ 3COL.

D-d. Dają nam formułę φ w postaci 3CNF o zmiennych $p_1, \dots, p_n$. Mamy szybko zbudować $G_\varphi = (V, E)$ t.że G_φ jest 3-kolorowalny $\iff$ φ jest spełnialny.



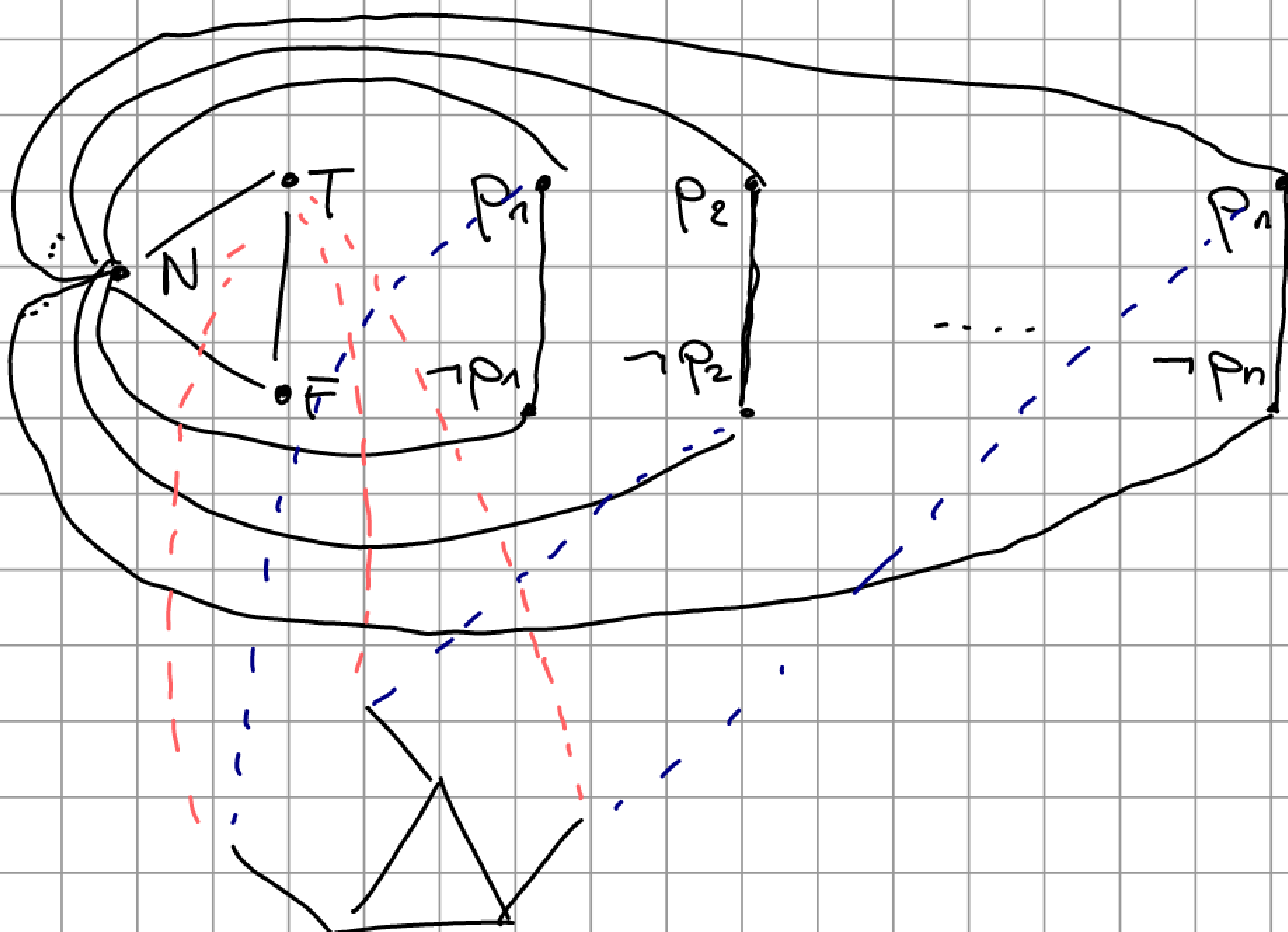
Gadżet 3 wierzchołki zewnętrzne i 3 wewnętrzne.

Obserwacja: $kol: Zewn \to \{N, F, T\}$ rozszerza się do poprawnego kolorowania całości, gdy nie jest stała.



D-d. prosty.

Podobnie gadżetu dla klauzuli $p_1 \vee \neg p_2 \vee p_n$:



No i to tyle ...