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.

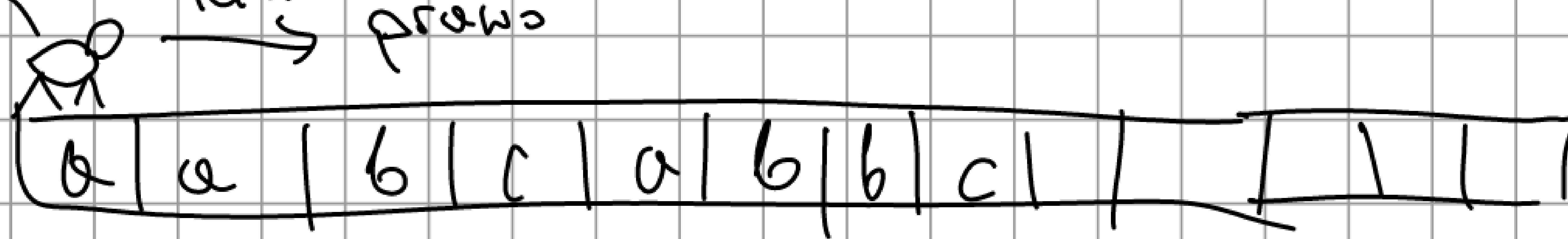
- klasyfikujemy problemy ze względu na te zasoby

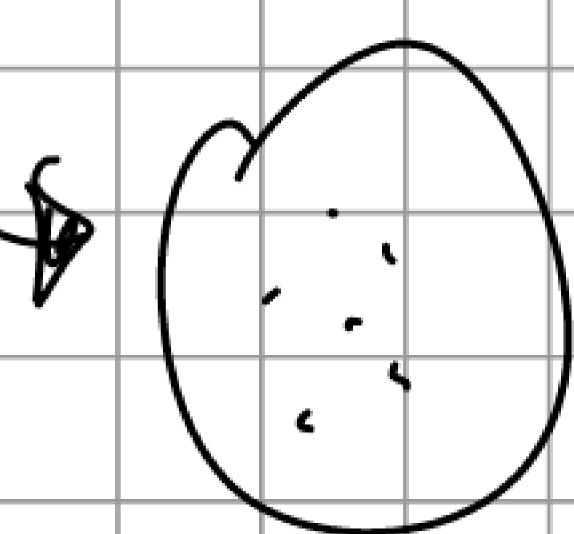
Bla bla...

CZĘŚĆ I

Automaty skończone

idzie w prawo





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 automatem 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 $\{w \in \Sigma^* : \hat{\delta}(q_0, w) \in F\}$.

<u>Def.</u> $L \subseteq \Sigma^*$ nazywamy regularnym gdy istnieje DFA A t.że $L = L_A$.

<u>Lemat</u> (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 $k \in \mathbb{N}$ $xy^kz \in L$.

<u>Przykład</u> 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 $|xz|_0 < |xz|_1$ ↯

<u>Dowód lematu</u>

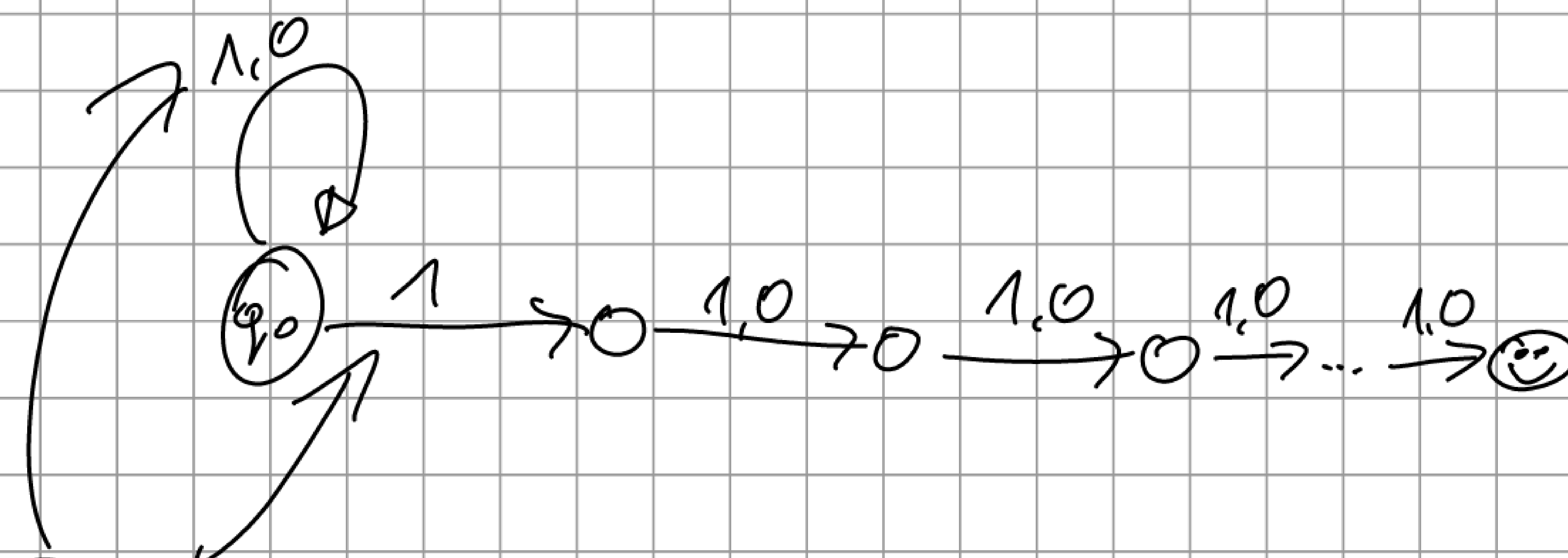
Weźmy $L = L_A$ regularny $(A = \langle \Sigma, Q, q_0, \delta, F \rangle)$.
Niech $n = |Q| + 1$. Weźmy $w \in L$ t.że $|w| \geq n$.
Niech $w = a_1 a_2 \dots a_l$, niech $s_i = \hat{\delta}(q_0, a_1 \dots a_i)$.
Wtedy $s_i = s_j$ dla pewnych $i < j \leq n$.
Niech $x = a_1 \dots a_i$, $y = a_{i+1} \dots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

1.03.2022

NIEDETERMINISTYCZNE AUTOMATY SKOŃCZONE (NFA)

Przykład

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



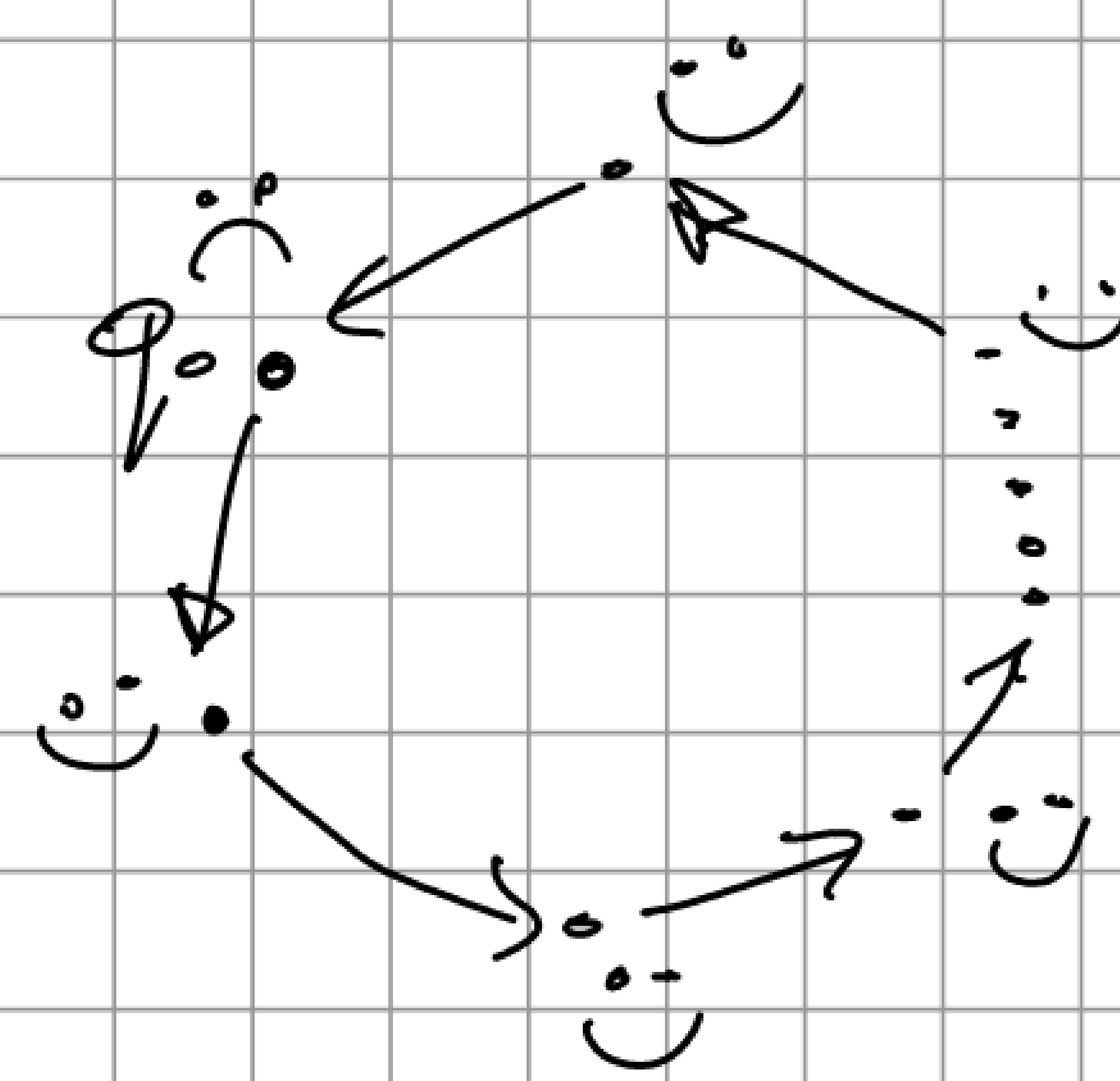
DWA PRZEJŚCIA Z 1 ⟶ ŻUCZEK PYTA NIEBIOS
"CO ROBIĆ?
JAK ŻYĆ"

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")

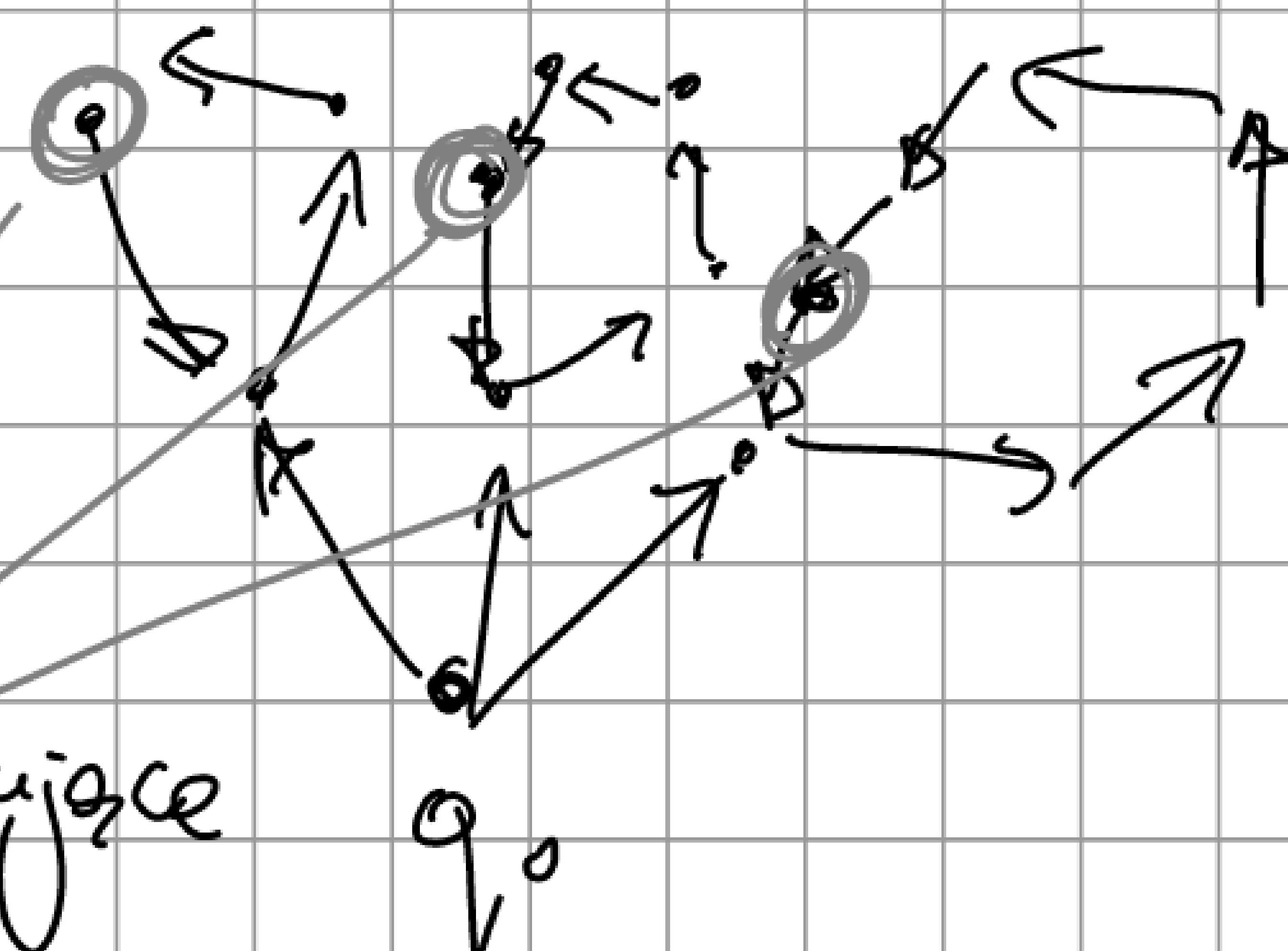
Przykład 2 $L = \{0^i : 105 \nmid i\}$

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 $\delta_w \subseteq Q \times Q$:

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

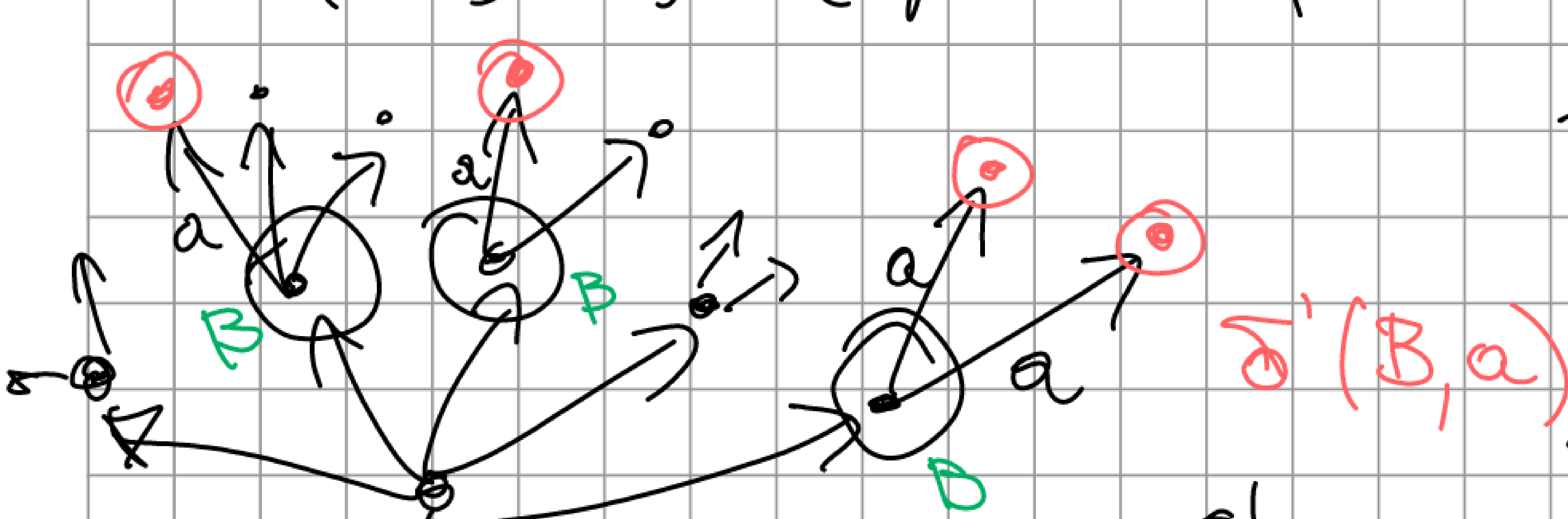
Wtedy $\hat{\delta}(q, w, q') \iff \delta_w(q, q')$.

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

$$\boxed{\begin{array}{l} L_1 L_2 = \{w_1 w_2 : w_1 \in L_1, w_2 \in L_2\} \\ L \subseteq \Sigma^*, \text{ wtedy } L^0 = L_\varepsilon, L^1 = L, L^{i+1} = L^i L \\ L^* = \bigcup_{n=1}^{\infty} L_n \end{array}}$$

- 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 $L = L_A$.

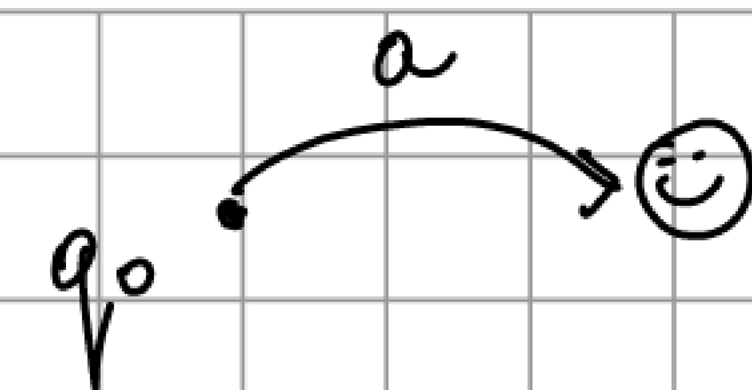
(2) Istnieje wyrażenie regularne φ t. że $L = L_\varphi$.

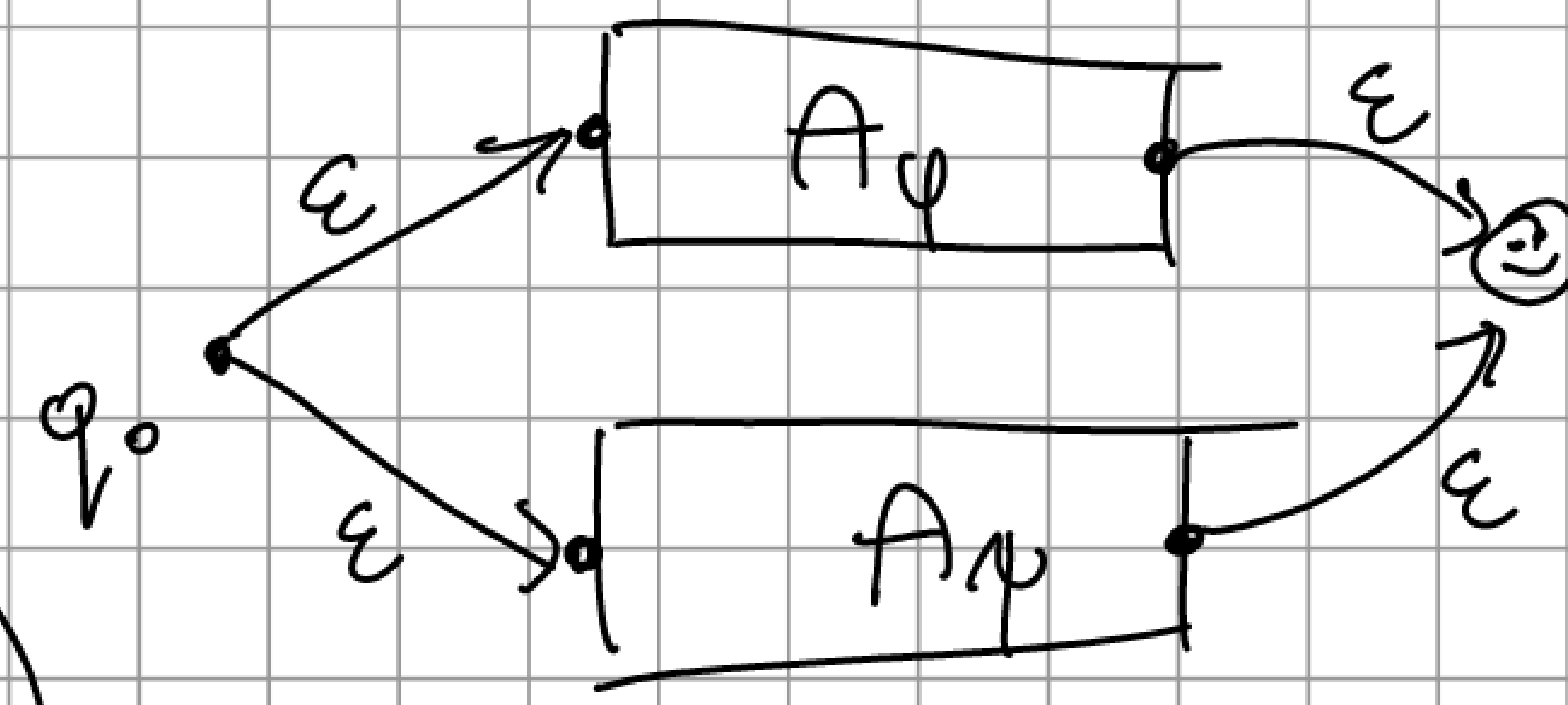
(3) Istnieje NFA z ε-przejściami A' t. że $L = L_{A'}$.

D-d. (3) $\Rightarrow$ (1) Było. ∎

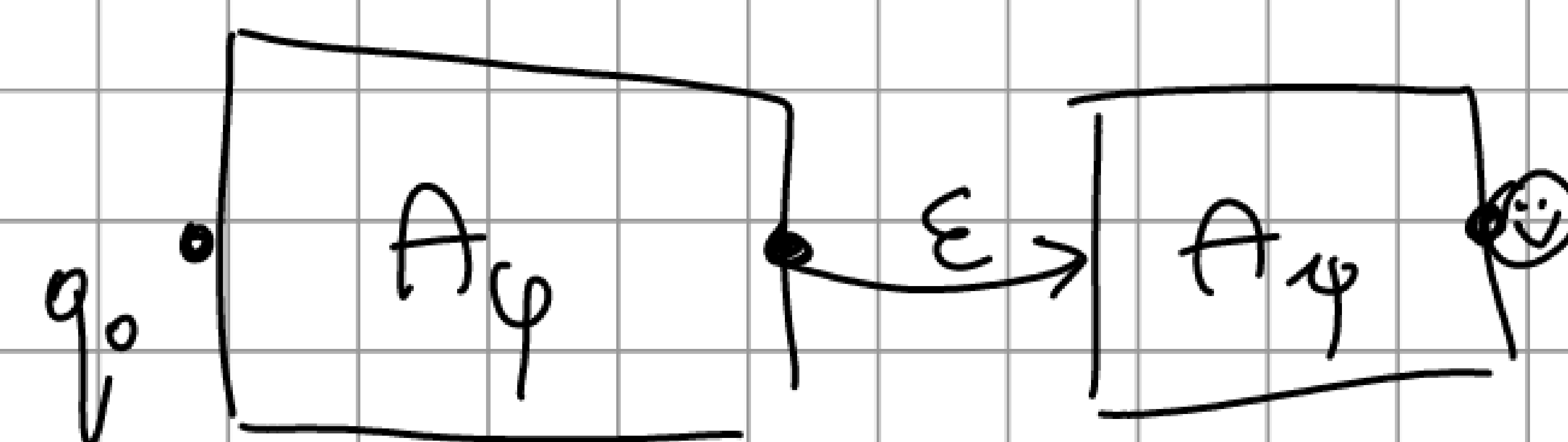
(2) $\Rightarrow$ (3) Indukcja względem długości wyrażenia. Każdy NFA który zbudujemy będzie miał jeden stan wejściowy niebędący akceptującym i jeden akceptujący.

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

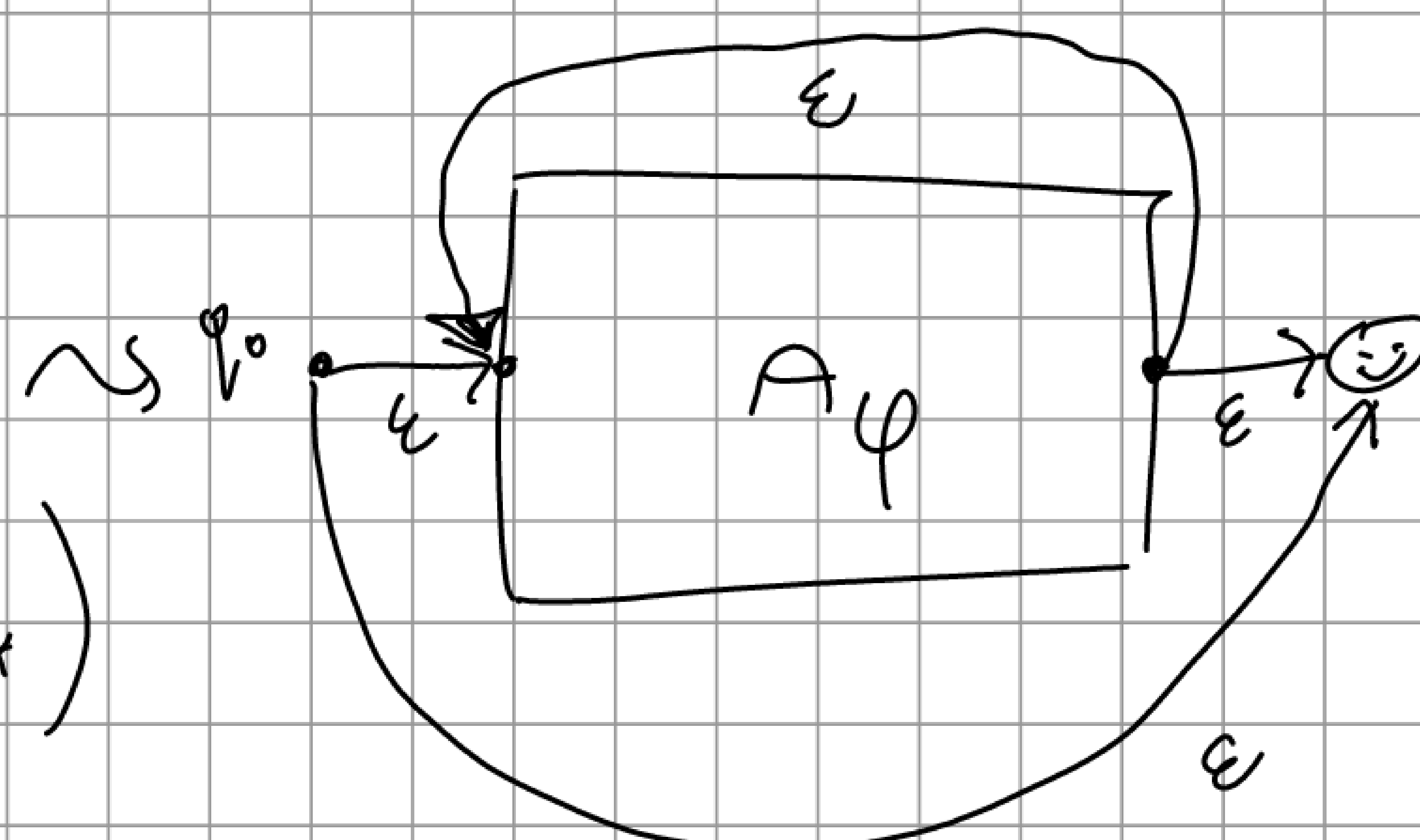
- $a \rightsquigarrow$ 

- $\varphi, \psi \rightsquigarrow$ 

(automat dla $\varphi + \psi$)

- $\varphi, \psi \rightsquigarrow$ 

(Automat dla $\varphi\psi$)

- $\varphi \rightsquigarrow$ 

(Automat dla φ^*)

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

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

Dla $1 \leq i, j \leq n, 0 \leq k \leq n$ 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 $< k$.

Indukcja względem k:

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

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

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

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

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

$\blacksquare$

UOGÓLNIENIA

Są dwa możliwe kierunki:

- słowa nieskończone
- drzewa

można iść w oba tych kierunkach jednocześnie

Automat skończony na słowach nieskończonych.

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

Σ — skończony

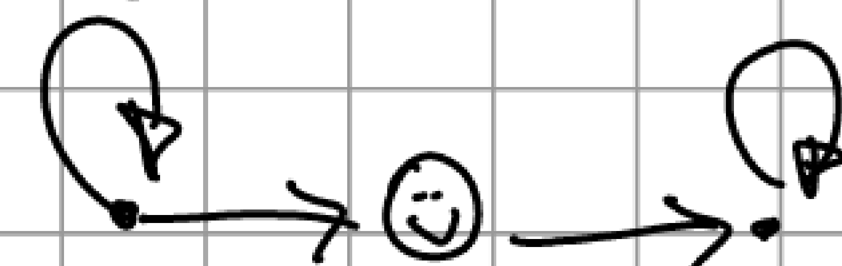
CO TUTAJ? — Dużo warunków akceptacji Büchiego, Rabina...

Interpretacja Büchiego: $F \subseteq Q$.

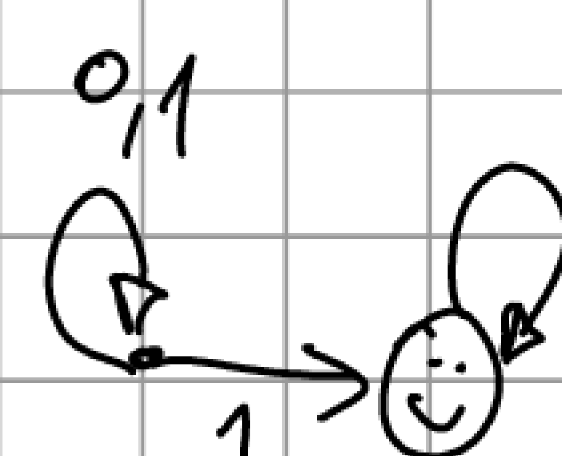
Słowo jest akceptowane, gdy automat nieskończenie wiele razy odwiedza stany z F.

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 (*)
- jest zamknięta na sumę, (i) 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 $\langle N, \Sigma, S, \Pi \rangle$, $S \in N$, $\Pi \subseteq_{\text{skończone}} N \times (N \cup \Sigma)^*$, $N \cap \Sigma = \emptyset$

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.

<u>Przykład</u> • 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 $A \rightarrow BC$ 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 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 $|zty| \leq n$, $|zy| > 0$, $w = sztyx$ dla każdego $k \in \mathbb{N}$ $sz^k t y^k x \in L$.

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

Odp.: TAK (proste)

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

Odp.: TAK (to samo)

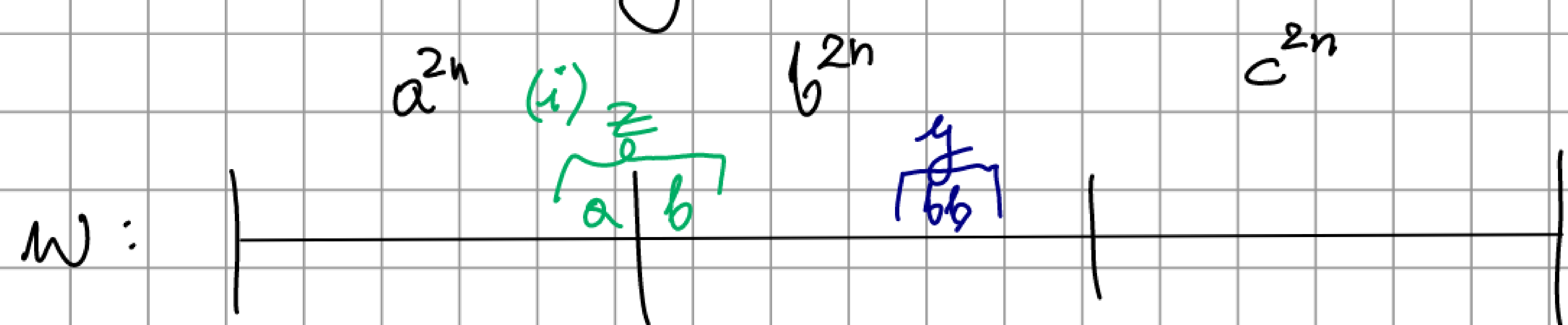
A czym jest przekrój tych języków?

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

Ten język jednak nie jest CFL!

__Dowód__ 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.



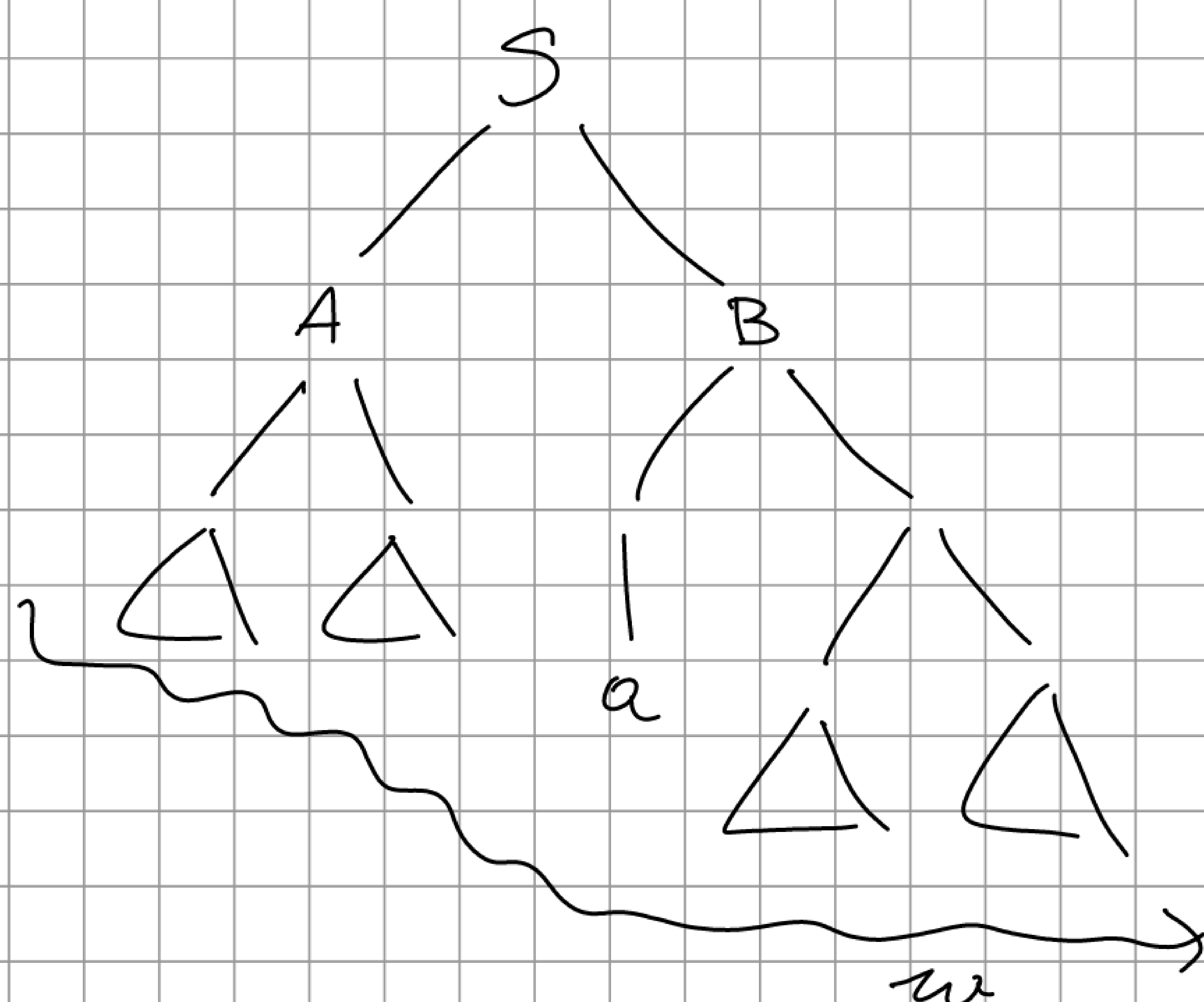
Przypadki: (i) z lub y zawiera dwie różne literki, to dla $k=2$ wygrywamy

(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)

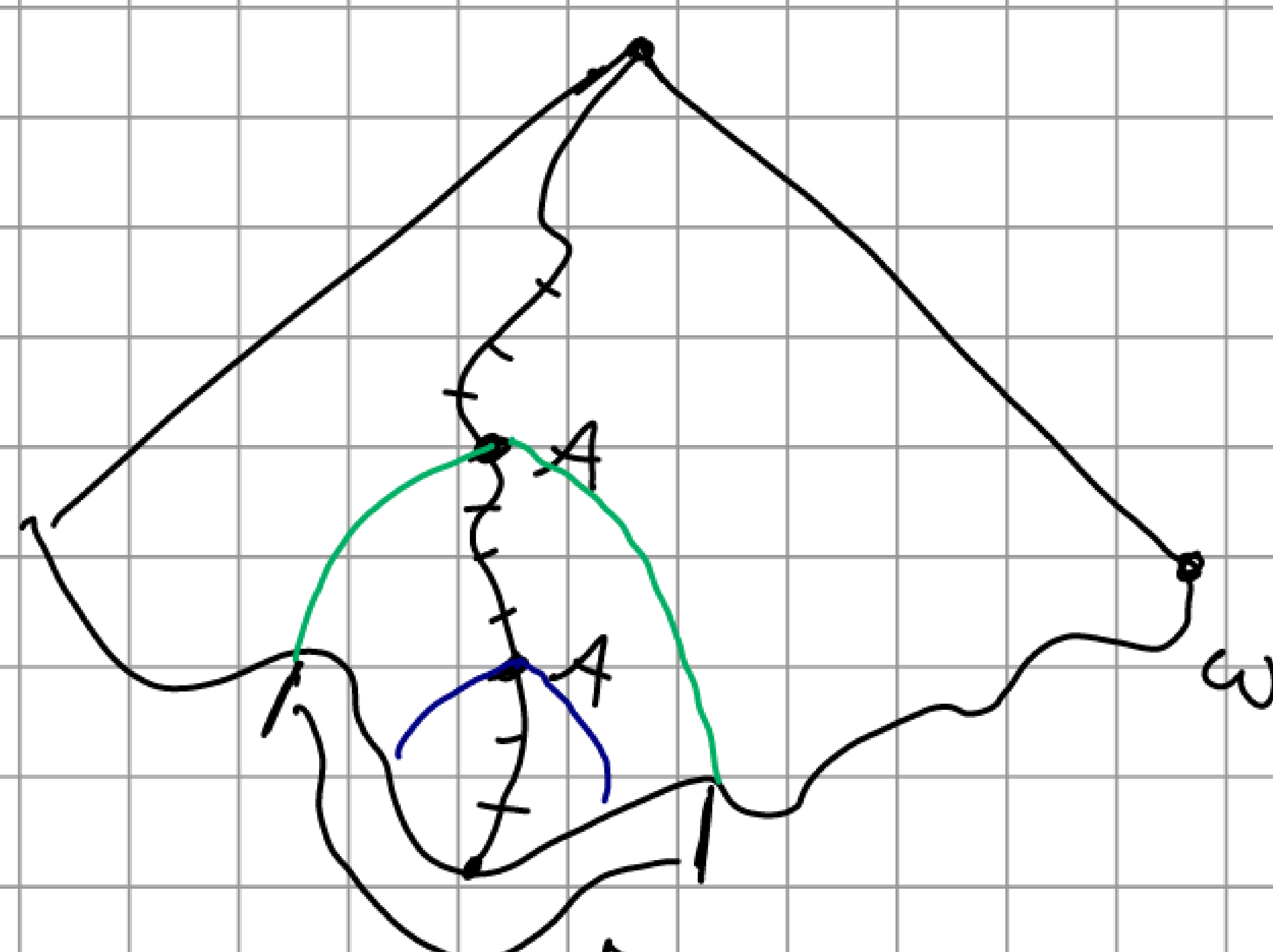
Weźmy $L \in CFL$ oraz CFG $G = \langle N, \Sigma, S, \Pi \rangle$ w postaci normalnej Chomsky'ego i niech $n = 2^{|N|+3}$. Niech $w \in L$, $|w| \geq n$.



Drzewo słowa w.

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

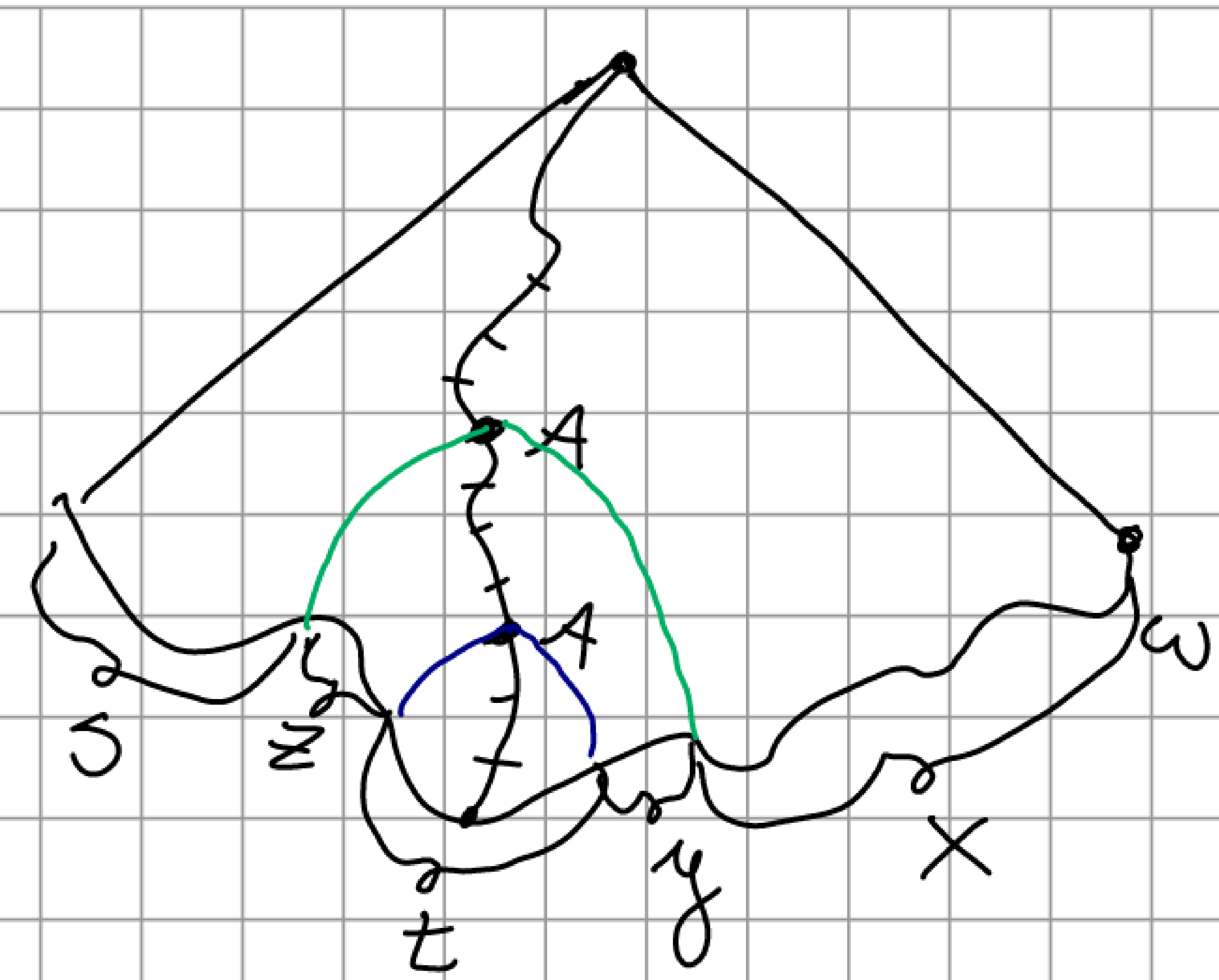
Na tej ścieżce są same nieterminale (poza liściem), zatem musi być jakiś nieterminal, który się powtarza. Niech A będzie takim nieterminalem który jest <u>najniżej</u> na tej ścieżce.



to ma dł. $\leq 2^{|N|+1}$

(no bo tylko A się może powtórzyć na tej ścieżce)

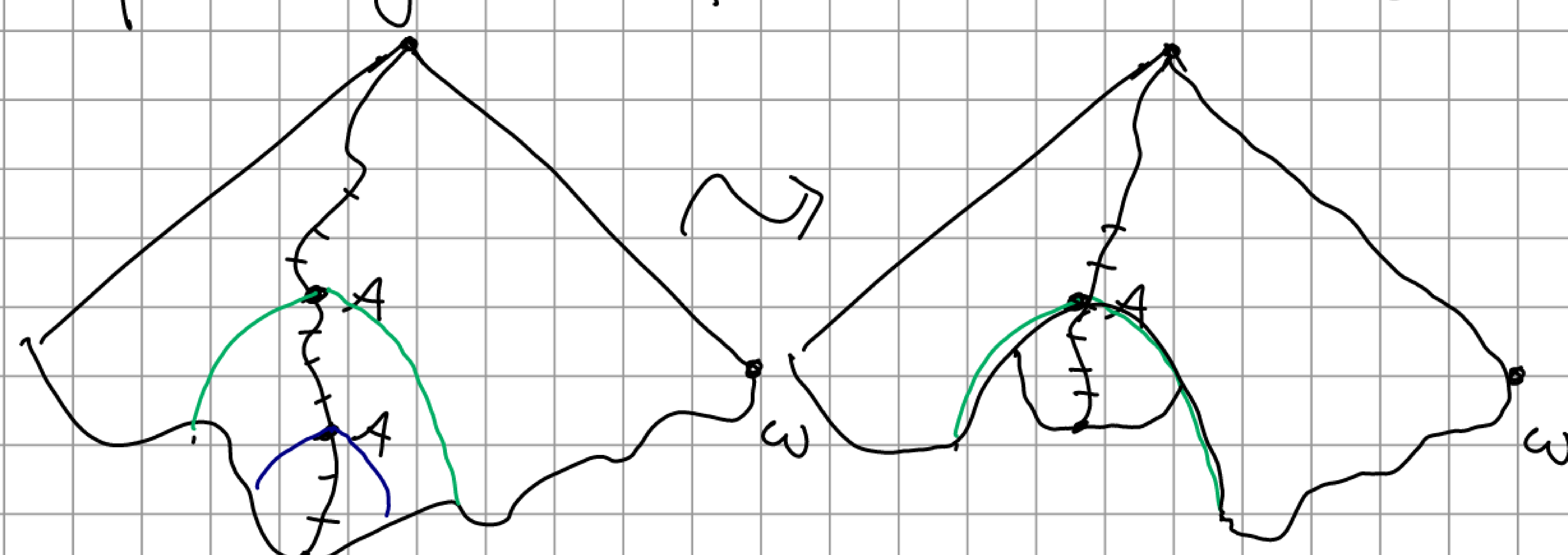
Poddrzewo jest naturalny:



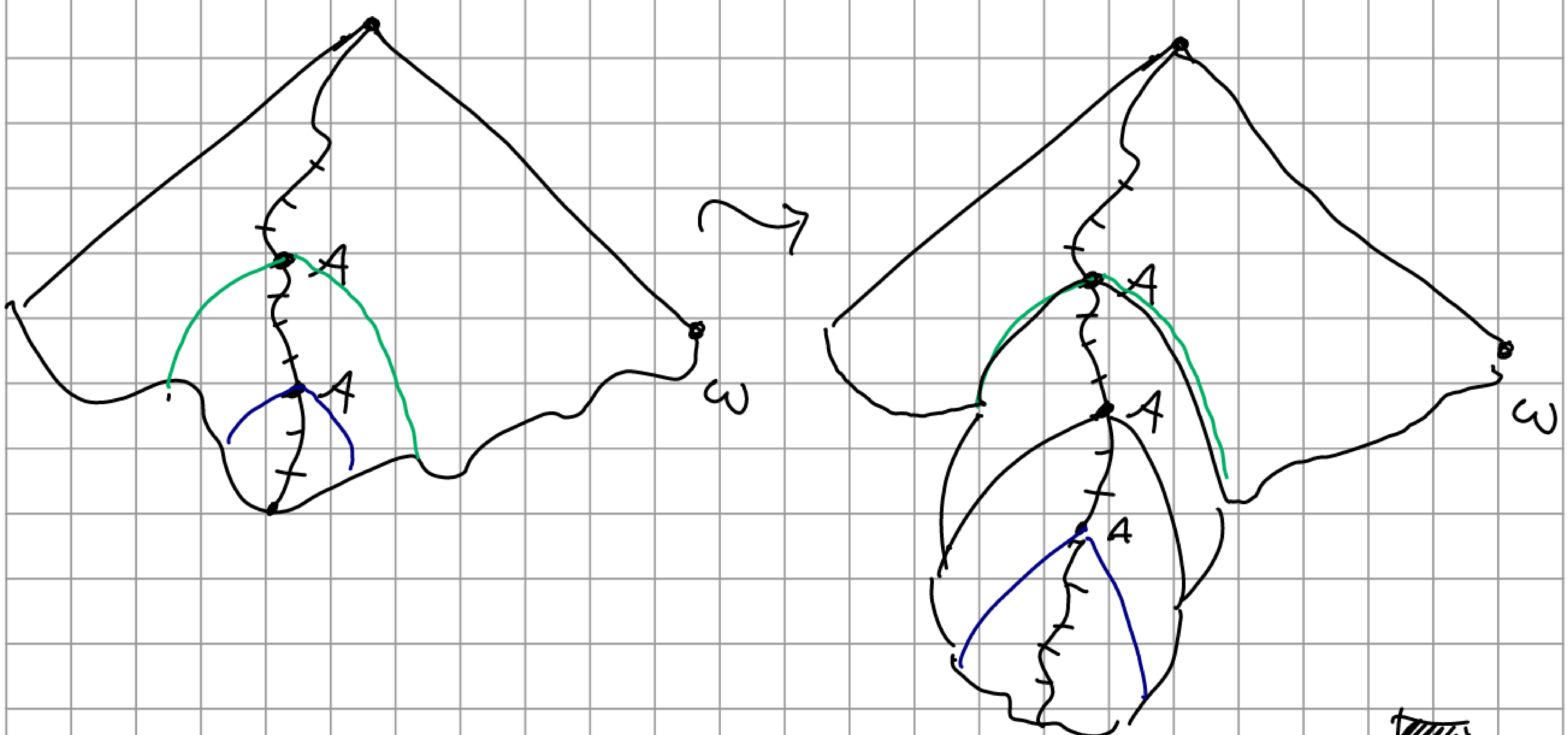
Wtedy $|zty| \leq 2^{|N|+1} < n$.

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.



$\blacksquare$

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

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:

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

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

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

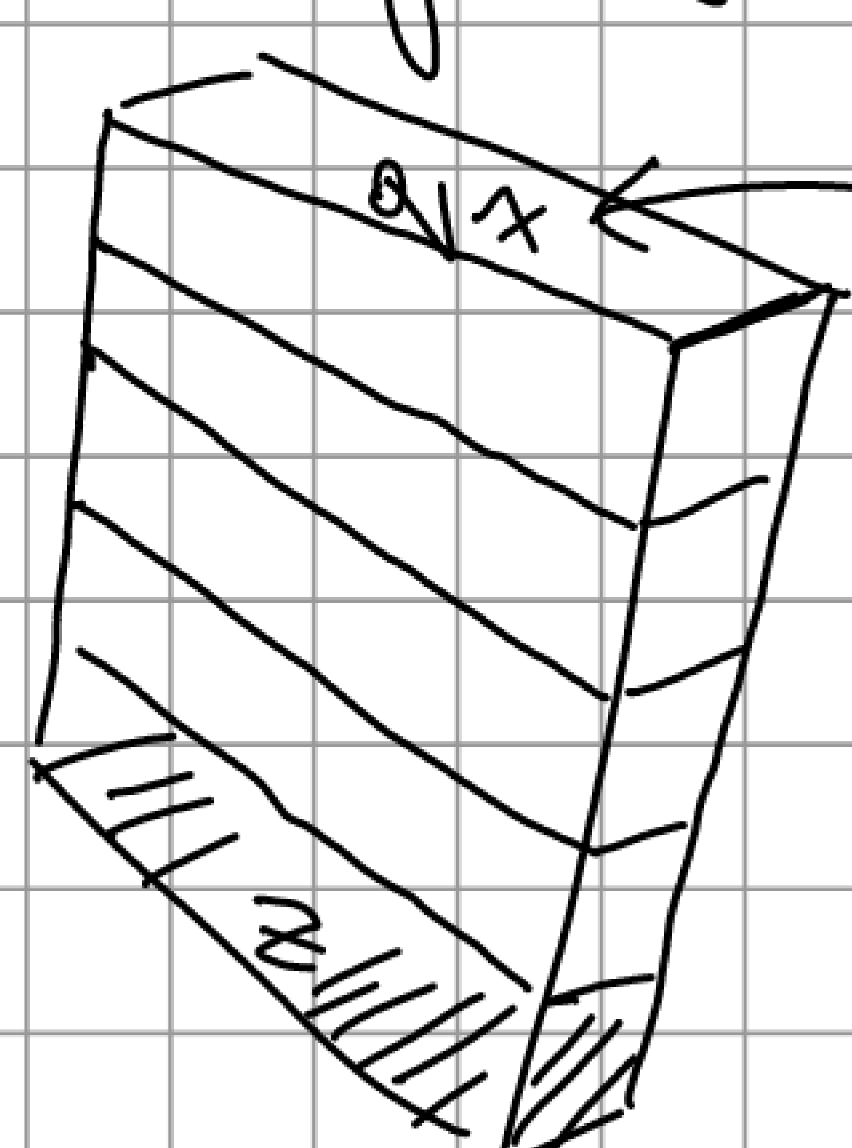
28.03.2022

Kontynuacja dowodu NPDA → CFG

NPDA → NPDA z JEDNYM STANEM → CFG

tę część już potrafimy, analogicznie do poprzedniej części dowodu.

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



"wyobraź 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

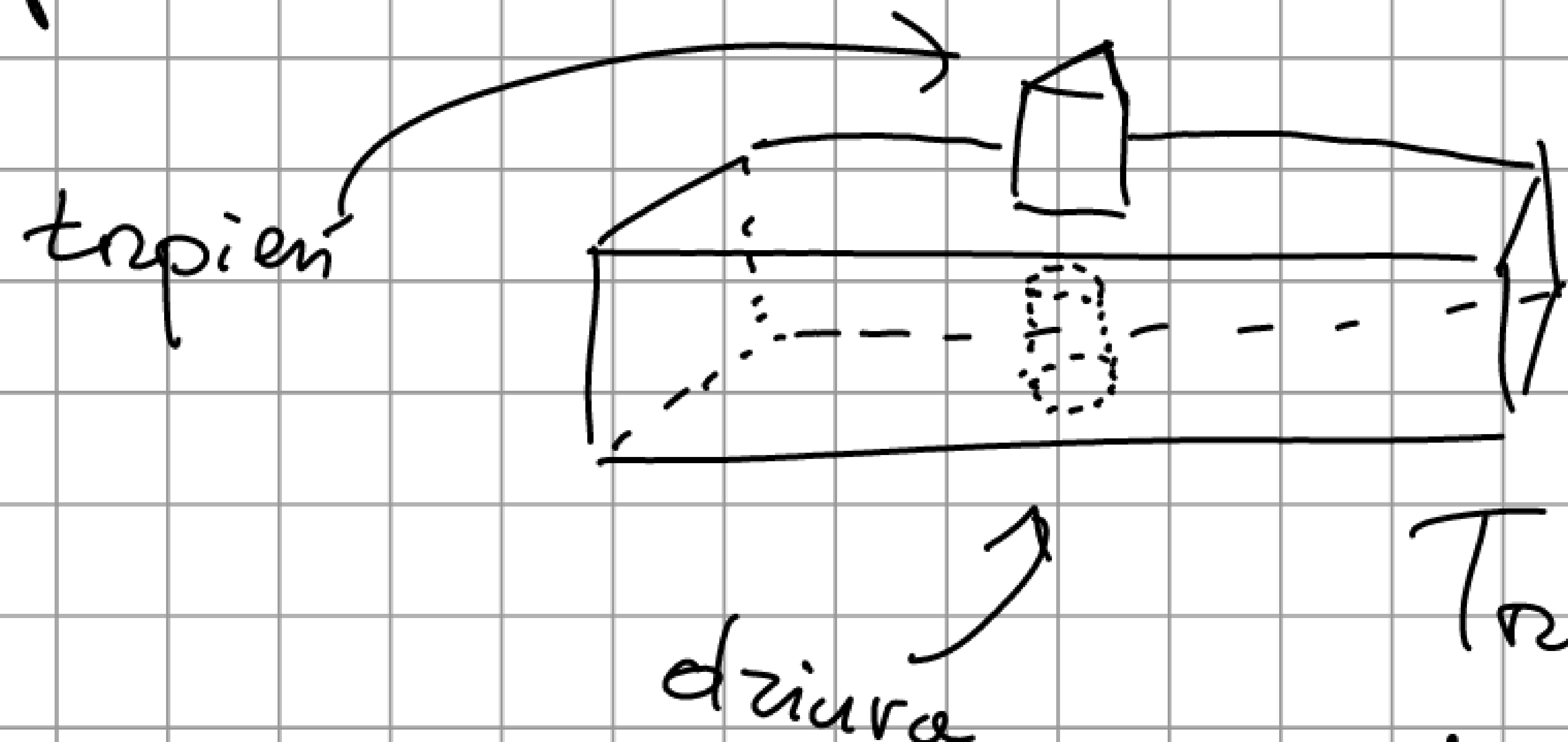
Np.: $\langle q_7$, czerwony, a, q_3, ziel-nieb-ziel-czerw$\rangle$

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, topień

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

Instrukcje starego automatu:

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

<u>Poprawność</u> 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 każdego $n \in \mathbb{N}$:

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

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 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 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.

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