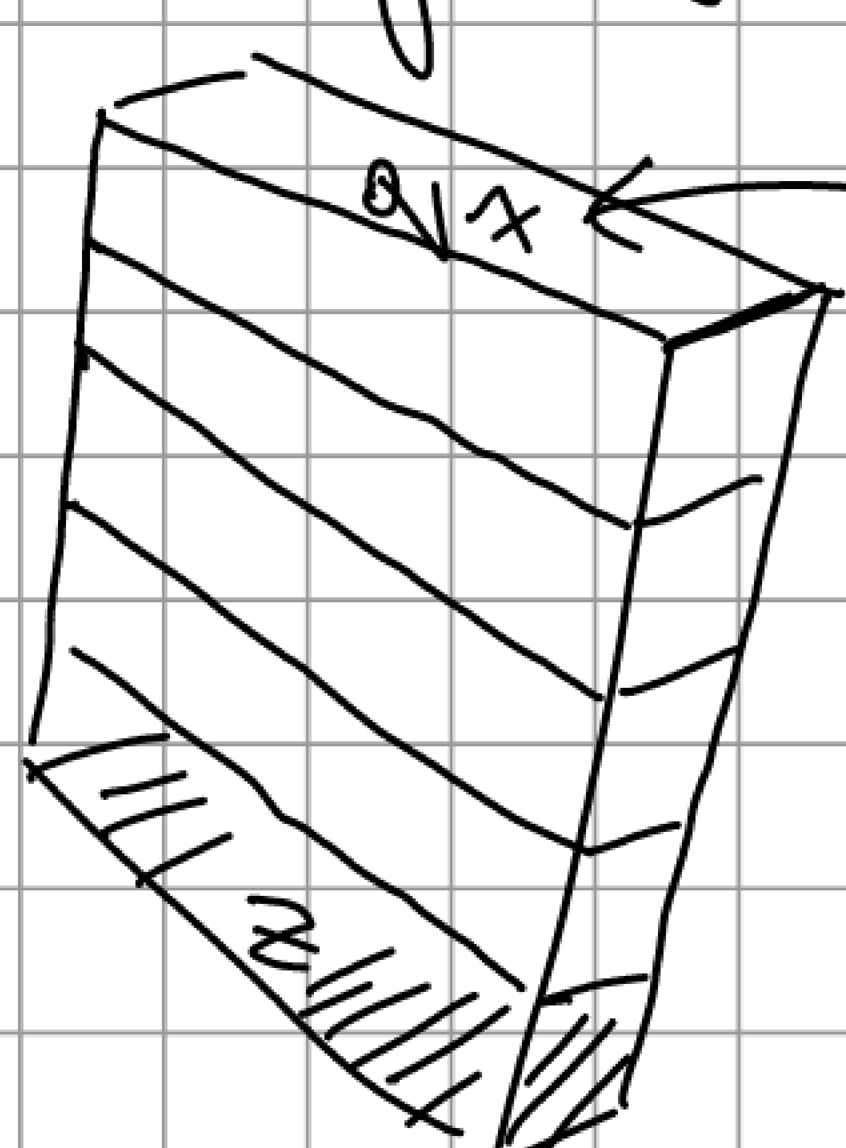
28.03.2022

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



"wyobraźmy 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

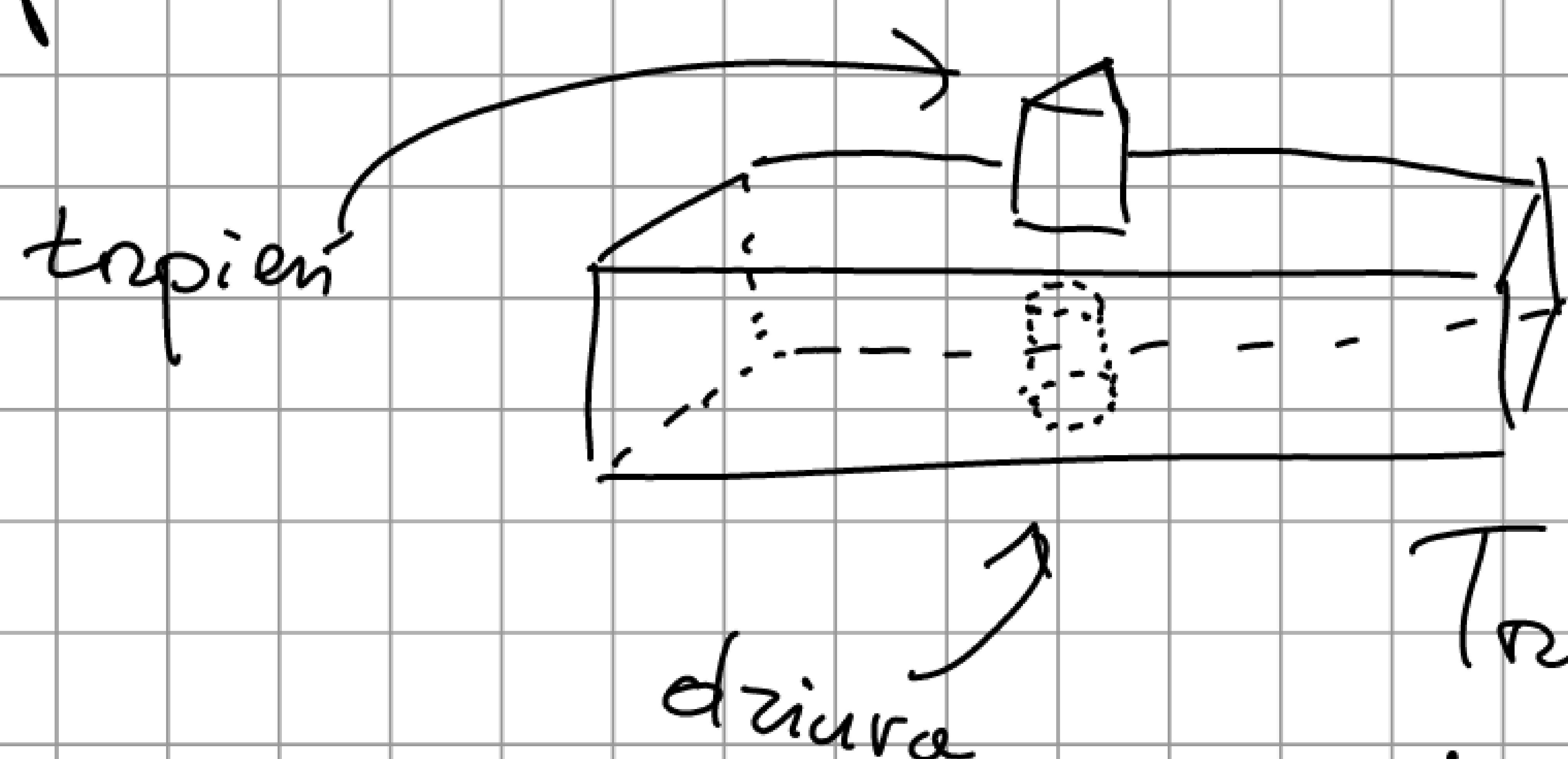
Np.: $\langle q_7, \text{czerwony}, a, q_3, \text{ziel-nieb-ziel-czerw} \rangle$

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:



Trzpieni jest tyle co stanów.

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

Instrukcje starego automatu:

$\langle q, K, a; q', A_1 A_2 \ldots \underset{\text{wierzch}}{A_l} \rangle, \quad l \geq 1$

zastępujemy zbiorem wszystkich instrukcji postaci

(pomijemy stan: jest tylko jeden)

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

dziurka

kolor

korzeń

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

Instrukcje starego automatu:

$\langle q, k, a; q', \varepsilon \rangle$

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

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 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.

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

$\lightning$