ZLOŻONOŚĆ OBLICZENIOWA - WYK. 1

1. O czym jest ten przedmiot?
   - problemy obliczeniowe jako obiekt badań
   - Należy sobie przypomnieć JAiO: Języki, automaty, obliczenia
   - podstawowe pytanie: jak szybko można rozwiązać dany problem?
   - na JAiO: są problemy roztrzygalne i nierozstrzygalne.
   - tu: są (rozstrzygalne) problemy łatwe i trudne (w różnych sensach)
2. Maszyna Turinga: definicja
   - skończony alfabet roboczy Γ, gdzie $\sharp, \triangleright \in \Gamma$
   - alfabet wejściowy $\Sigma \subseteq \Gamma \setminus \{\sharp, \triangleright\}$
   - skończony zbiór stanów Q
   - stany: początkowy, akceptujący, odrzucający $q_I, q_A, q_R \in Q$
   - funkcja przejścia $\delta : (Q \setminus \{q_A, q_R\}) \times \Sigma \to Q \times \Sigma \times \{L, R, Z\}$

   Konfiguracja: słowo $w \in \Gamma^* Q \Gamma^*$, zaczynające się od $\triangleright$, w którym nie ma innych wystąpień $\triangleright$. Myślimy, że konfiguracja to:
   - zawartość *taśmy*, nieskończonej w prawo
   - położenie *głowicy* (konwencja: głowica jest na następnej literze)
   - stan maszyny.

   Identyfikujemy konfiguracje powstałe przez dopisanie $\sharp$'ów ma końcu (innymi słowy, konfiguracja to słowo nieskończone kończące się samymi $\sharp$'ami.)

   Maszyna M zadaje funkcję (częściową) $\to_M$ między konfiguracjami. Maszyna *akceptuje* słowo $w \in \Sigma^*$ jeśli z konfiguracji $\triangleright q_I w$ wpada w stan q_A, *odrzuca* jeśli wpada w q_R; w przeciwnym razie zadaje obliczenie nieskończone (maszyna się *pętli*).

   M ma *własność stopu* jeśli zatrzymuje się na każdym słowie.

   Język $L \subseteq \Sigma^*$ jest *rekurencyjnie przeliczalny* albo *częściowo obliczalny* jeśli istnieje maszyna, która akceptuje słowa dokładnie z L. Jeśli ta maszyna ma własność stopu, to L jest *obliczalny*, albo *rozstrzygalny*.

   Ćwiczenie: Jeśli L i dopełnienie L są częściowo obliczalne, to są rozstrzygalne.

   Funkcja częściowa $f : \Sigma^* \to \Sigma^*$ jest *obliczalna* jeśli istnieje maszyna M która akceptuje jej dziedzinę i dla każdego słowa w z tej dziedziny, M wpada w konfigurację akceptującą $\triangleright q_A v$, gdzie $v = f(w)$.
3. Warianty maszyn Turinga.
   - taśma nieskończona w obie strony (usuwamy znacznik $\triangleright$)

- wiele stanów akceptujących / odrzucających
- maszyny k-taśmowe:

$$\delta : (Q \times \{q_A, q_R\}) \times \Sigma^k \to Q \times \Sigma^k \times \{L, R, Z\}^k$$

- ...

Fakt: Wszystkie rozpoznają tę samą klasę języków.

Uwaga: wystarczy pokazać że dla każdej maszyny typu X istnieje równoważna maszyna typu Y. W praktyce te konstrukcje są obliczalne, ale nie musimy tego wiedzieć. To będzie częsty motyw na tym przedmiocie: kiedy wystarczy że coś istnieje, a kiedy musimy to umieć (szybko) polliczyć?

4. Problemy decyzyjne vs. języki
   - przykład: osiągalność w grafie
   - różne reprezentacje grafów, wzajemnie tłumaczalne
   - zalety maszyn Turinga: model prosty, ale mocny (teza Turinga-Churcha, ale mocniejsza: problemy łatwe maszyny Turinga rozwiązują szybko)
5. Złożoność czasowa
   - Maszyna M działa w czasie $T(n)$ (dla funkcji $T : \mathbb{N} \to \mathbb{N}$) jeśli dla każdego słowa $w \in \Sigma^*$ zatrzymuje się po co najwyżej $T(|w|)$ krokach. (Czyli m.in. ma własność stopu).

     Język $L \in \Sigma^*$ jest *rozpoznawalny* w czasie $T(n)$ jeśli istnieje maszyna **wielotaśmowa**, która go akceptuje w czasie $T(n)$. Aby wariant maszyny nie grał roli, mówimy o czasie $\mathcal{O}(T(n))$.

     Ćwiczenie: Maszyna dwutaśmowa rozpoznaje język palindromów w czasie liniowym. Maszyna jednotaśmowa potrzebuje czasu kwadratowego.
6. Twierdzenie o liniowym przyspieszeniu: jeśli język L jest rozpoznawalny w czasie $T(n)$, to dla każdej stałej $c > 0$ jest rozpoznawalny w czasie $c \cdot T(n) + \mathcal{O}(n)$. *Dowód*: ćwiczenie.
7. Podstawowe klasy złożoności czasowej:
   - $\mathbf{P} = \bigcup_{k \in \mathbb{N}} \mathbf{DTIME}(n^k)$. Inaczej: te problemy, dla których istnieje wielomian p i maszyna M działająca w czasie $p(n)$.
   - $\mathbf{EXPTIME} = \bigcup_{k \in \mathbb{N}} \mathbf{DTIME}(2^{n^k})$
8. Mierzenie złożoności pamięciowej: maszyny *off-line*
   - taśma wejściowa, tylko do odczytu
   - taśma robocza, mierzymy jej rozmiar

- przy obliczaniu funkcji: taśma wyjściowa, głowica idzie tylko w prawo (czyli: tylko do zapisu)

NB. Nie ma czegoś takiego jak "pamięć stała", w ten sposób rozpoznamy tylko języki regularne. Stała liczba zmiennych to pamięć logarytmiczna.

Język jest w klasie $\mathbf{DSPACE}(S(n))$ jeśli istnieje maszyna akceptująca ten język i działająca w pamięci $\mathcal{O}(S(n))$. Dodatkowo wymagamy, by miała własność stopu (bo może się pętlić w skończonej pamięci).

9. Podstawowe klasy złożoności czasowej:
   - $\mathbf{L} = \mathbf{DSPACE}(\log n)$
   - $\mathbf{PSPACE} = \bigcup_{k\in\mathbb{N}} \mathbf{DSPACE}(n^k)$. Inaczej: te problemy, dla których istnieje wielomian p i maszyna M działająca w pamięci $p(n)$.
10. Twierdzenie: $\mathbf{L} \subseteq \mathbf{P}$ (a także $\mathbf{PSPACE} \subseteq \mathbf{EXPTIME}$).

    Dowód: Maszyna, która używa $c \cdot \log n$ pamięci roboczej, ma

    $$|\Gamma|^{c\cdot\log n} \cdot n \cdot |Q|$$

    konfiguracji, czyli wielomianowo wiele względem n ($|\Gamma|$, c i $|Q|$ są stałe). Jeżeli maszyna ma własność stopu, to żadna konfiguracja nie może się powtórzyć, więc maszyna zatrzyma się po wielomianowej liczbie kroków.