ZLOŻONOŚĆ OBLICZENIOWA - WYK. 6

1. Maszyny niedeterministyczne:
   - definicja jak zwykle, ale *relacja przejścia* zamiast funkcji częściowej:
     $$\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R, Z\}$$
   - poza tym nie ma stanów odrzucających (są niepotrzebne)
   - relacja przejścia na konfiguracjach: jak w deterministycznych maszynach
   - bieg maszyny: ciąg konfiguracji zgodny z relacją przejścia
   - maszyna akceptuje słowo w jeśli istnieje na tym słowie bieg akceptujący
   - maszyna działa w czasie $f(n)$ jeśli *każdy bieg* (niekoniecznie ten akceptujący) zatrzymuje się po $f(n)$ krokach. Analogicznie dla pamięci, przy czym wymagamy dodatkowo własności stopu.
   - klasy $\mathbf{NTIME}(f(n))$, $\mathbf{NSPACE}(f(n))$
   - $\mathbf{NL} = \mathbf{NSPACE}(log(n))$, $\mathbf{NP} = \bigcup_{k\in\mathbb{N}} \mathbf{NTIME}(n^k)$, $\mathbf{NPSPACE} = \bigcup_{k\in\mathbb{N}} \mathbf{NSPACE}(n^k)$.
2. Alternatywna definicja **NP**, czyli model ze świadkami
   - *Relacja* R to język słów postaci $v\$w$, dla $v, w \in \Sigma^*$ i \$ szczególnego symbolu
   - Relację nazywamy *wielomianową*, jeśli:
     - $R \in \mathbf{P}$, oraz
     - istnieje wielomian p taki że $v\$w \in R$ implikuje $|w| \leq p(|v|)$.
   - $\exists R = \{v \mid \exists w. v\$w \in R\}$.
   - *Twierdzenie:* $L \in \mathbf{NP}$ wtw. istnieje wielomianowa relacja R taka, że $L = \exists R$.
   - *Dowód:* z prawa w lewo: jeśli istnieje relacja R, to zróbmy maszynę, która za słowem wejściowym wypisze (niedeterministycznie) dowolne słowo długości $p(n)$ (wszystkie wielomiany są czasowo konstruowalne), a potem uruchomi (deterministyczną) maszynę sprawdzającą relację R.

     Z lewa w prawo: niech L będzie rozpoznawalny maszyną niedeterministyczną M w czasie $p(n)$. Bez utraty ogólności załóżmy, że relacja przejścia M ma stopień wyjściowy ≤ 2. (Każdą maszynę można przetłumaczyć na taką, spowalniając obliczenia liniowo.) To znaczy, że na każdym akceptowanym słowie v istnieje ciąg wyborów długości co najwyżej $p(|v|)$, który determinuje bieg akceptujący. Słowa wraz z kodowaniami akceptujących biegów bierzemy jako relację R. Jest to relacja wielomianowa, w szczególności jest rozpoznawana maszyną deterministyczną w czasie wielomianowym.

3. Ćwiczenie: wymyślić analogiczny model dla **NL**. Uwaga: świadki długości logarytmicznej to głupi pomysł!

4. Dla każdej klasy $\mathcal{C}$, klasa **co**$\mathcal{C}$ to dopełnienia języków z klasy $\mathcal{C}$. Klasy deterministyczne (np. $\mathbf{L}, \mathbf{P}, \mathbf{PSPACE}$) są w trywialny sposób równe swoim ko-klasom, ale dla klas niedetrministycznych sprawa nie jest taka prosta. Znany problem otwarty: czy $\mathbf{NP} = \mathbf{coNP}$?

5. $\mathbf{DTIME}(f(n)) \subseteq \mathbf{NTIME}(f(n))$, $\mathbf{DSPACE}(f(n)) \subseteq \mathbf{NSPACE}(f(n))$.

   Dowód: Trywialne, bo każdą maszynę deterministyczną można traktować jako niedeterministyczną.

6. $\mathbf{NTIME}(f(n)) \subseteq \mathbf{DSPACE}(f(n))$, o ile $f(n)$ pamięciowo konstruowalna.

   Dowód: Zajmij $f(n)$ pamięci roboczej i generuj tam po kolei wszystkie słowa. Każde kolejne słowo traktuj jako ciąg wyborów dla maszyny niedeterministycznej, dla każdego słowa symuluj ją w pamięci roboczej (użyjesz jej tylko $\mathcal{O}(f(n))$, bo symulowane obliczenie jest takie długie). Po każdym słowie możesz zresetować pamięć roboczą (poza ciągiem wyborów).

7. $\mathbf{NSPACE}(f(n)) \subseteq \bigcup_{c \in \mathbb{N}} \mathbf{DTIME}(c^{f(n)+\log n})$, jeśli $f(n)$ pamięciowo konstruowalna.

   Dowód: Weźmy niedeterministyczną maszynę M działającą w pamięci $f(n)$. Bez utraty ogólności można założyć, że taśm roboczych jest 1. Konfigurację takiej maszyny na ustalonym słowie wejściowym $|w| = n$ można reprezentować jako:

   - zawartość taśmy roboczej ($|\Gamma|^{f(n)}$ możliwości),
   - pozycje głowicy na taśmie roboczej ($f(n)$ możliwości),
   - pozycja głowicy na taśmie wejściowej (n możliwości),
   - stan (stała liczba możliwości).

   W sumie jest $d^{f(n)+\log n}$ konfiguracji, dla pewnej stałej d.

   Sprawdzenie, czy M rozpoznaje słowo w, to problem osiągalności w skierowanym grafie konfiguracji. Istnieje wielomianowy (kwadratowy) algorytm na osiągalność w grafie skierowanym.

   Uwaga: można wygenerować i zapisać cały graf na taśmie, ale nie jest to potrzebne: wystarczy tylko konstruować listę konfiguracji osiągalnych, a dla danej konfiguracji umieć rozstrzygać, które są z niej bezpośrednio osiągalne. W tym celu czasem trzeba zaglądać na odpowiednie miejsce taśmy wejściowej, bo taśma wejściowa nie jest zapisana w konfiguracji.

8. wnioski: $\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE}$. Myślimy że wszystkie inkluzje są ostre, ale wiemy tylko że któraś jest ostra (z tw. o hierarchii pamięciowej).

9. Rozważmy problem osiągalności, gdzie graf skierowany jest reprezentowany jako lista krawędzi (albo macierz incydencji, to nie ma znaczenia). Fakt: problem osiągalności jest w **NL**. *Dowód:* maszyna pamięta aktualny wierzchołek i zgaduje następny. Ewentualnie: świadek to ścieżka w grafie (*uwaga:* tu trzeba zrobić wcześniejsze ćwiczenie!)

10. Twierdzenie: osiągalność w grafie skierowanym jest w $\mathbf{DSPACE}(\log^2 n)$.

    Dowód: Rozważamy problem $PATH(x, y, i)$: czy wierzchołek y jest osiągalny z x ścieżką długości co najwyżej 2^i? Aby to rozwiązać, przeglądamy wszystkie wierzchołki z i dla każdego pytamy, czy $PATH(x, z, y-1)$ i $PATH(z, y, i-1)$. W tym celu musimy zbudować stos, na którym odkładamy trójki (x, y, i). Każda trójka ma rozmiar $\log n$ i jest ich $\log n$, więc w sumie użyjemy $\mathcal{O}(\log^2 n)$ pamięci.

11. Twierdzenie Savitcha (1970): Dla każdej funkcji pamięciowo konstruowalnej nie mniejszej niż $\log n$, mamy $\mathbf{NSPACE}(f(n)) \subseteq \mathbf{DSPACE}(f(n)^2)$.

    Dowód: Dla danej niedeterministycznej maszyny M działającej w pamięci $f(n)$, uruchamiamy algorytm z poprzedniego twierdzenia na grafie konfiguracji M na danym słowie wejściowym. Skoro $f(n) \geq \log n$, to ten graf ma $d^{f(n)}$ wierzchołków (i stopień wyjściowy ograniczony przez stałą). Z poprzedniego twierdzenia, osiągalność w tym grafie można obliczyć w pamięci $\mathcal{O}(\log^2(d^{f(n)})) = \mathcal{O}(f(n)^2)$

    Samego grafu nie wypisujemy, tylko sprawdzamy poszczególne krawędzie na dnie rekursji. Pamięciowa konstruowalność jest po to, by ograniczyć rozmiary sprawdzanych konfiguracji z i początkową liczbę i.

12. Wnioski: $\mathbf{NPSPACE} = \mathbf{PSPACE} = \mathbf{coNPSPACE}$.

13. Za dwa tygodnie będzie twierdzenie Immermana-Szelepcsenyiego (1987): $\mathbf{NL} = \mathbf{coNL}$. Wszystko to razem sugeruje, że niedeterminizm ma mniejszy wpływ na złożoność pamięciową niż czasową (bo nikt nie podejrzewa że $\mathbf{NP} = \mathbf{coNP}$), ale uwaga: nikt też nie podejrzewa że $\mathbf{L} = \mathbf{NL}$!