

Co umie policzyć komputer?

Zadania kwalifikacyjne

WWW20

Michał Horodecki

Wstęp

Forma rozwiązań dowolna, byle czytelna.

Nie trzeba robić wszystkiego, ale zachęcam żeby zrobić jak najwięcej ;)

W razie jakichkolwiek pytań i wątpliwości śmiało piszcie maila na michalhorodecki2002+www@gmail.com.

Same zadania zaś proszę wysyłać przez stronę warsztatów.

Definicje do zadań

Alfabetem nazywamy dowolny niepusty zbiór Σ

Słowem nad alfabetem Σ nazywamy dowolny skończony ciąg elementów z Σ . Słowo może mieć długość 0 i takie słowo oznaczamy jako ε .

Językiem nad alfabetem Σ nazywamy dowolny zbiór słów nad Σ

Język pełny (zawierający wszystkie możliwe słowa) nad Σ oznaczamy przez Σ^*

Nieformalnie **problemem decyzyjnym** nazywamy taki problem algorytmiczny na który odpowiedzią jest "TAK" lub "NIE". Przykładowym problemem decyzyjnym jest sprawdzanie parzystości.

Podobnie, **problem obliczeniowy** to taki problem na który odpowiedzią jest wartość (np. liczba, napis, tablica, itp.). Przykładowym problemem obliczeniowym jest znalezienie liczby parzystej.

Problem (czyli ogólny meta opis zadania) należy odróżnić od jego **instancji** czyli konkretnego wejścia dla którego należy znaleźć wyjście oraz **algorytmu**, który podaje odpowiedź.

Deterministyczny Automat Skończony (DFA) $A = (Q, \Sigma, \delta, q_0, F)$ to:

1. zbiór stanów – Q
2. alfabet nad którym operuje – Σ
3. funkcja przejścia między stanami na zadanej literze – $\delta : Q \times \Sigma \rightarrow Q$
4. wyróżniony stan początkowy $q_0 \in Q$
5. zbiór stanów akceptujących $F \subseteq Q$

W każdym kroku automat konsumuje kolejną literę a słowa wejściowego w nad Σ i ze stanu q przechodzi do stanu $\delta(q, a)$. Mówimy że automat rozpoznaje w jeśli po skonsumowaniu wszystkich liter automat znajduje się w stanie akceptującym $q \in F$.

Niedeterministyczny Automat Skończony (NFA) $A = (Q, \Sigma, \delta, q_0, F)$ to:

1. zbiór stanów – Q
2. alfabet nad którym operuje – Σ
3. funkcja przejścia między stanami na zadanej literze – $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$

4. wyróżniony stan początkowy $q_0 \in Q$
5. zbiór stanów akceptujących $F \subseteq Q$

Zasada działania NFA jest bardzo podobna co DFA z tym, że z jednego stanu można przejść do wielu stanów. O niedeterminizmie można więc myśleć dwojako:

1. albo automat znajduje się w wielu stanach równoległe – wtedy rozpoznanie słowa rozumiemy przez wystąpienie stanu akceptującego w tych stanach po skonsumowaniu słowa
2. albo automat losuje/wybiera do którego stanu przechodzi – wtedy rozpoznanie rozumiemy jako istnienie szansy/wyborów prowadzących do jakiegoś stanu akceptującego

Zadanie 1 [3pkt]

Podaj 3 przykładowe problemy decyzyjne i dowolne ich instancje.

(Fajnie jakby te problemy były ciekawsze niż "sprawdź czy liczba jest parzysta", zachęcam do rozwiązania w miarę trudnych/praktycznych problemów)

Zadanie 2 [3pkt]

Podaj 3 przykładowe problemy obliczeniowe i dowolne ich instancje.

(Uwaga jak w poprzednim zadaniu – zachęcam do zastanowienia)

Zadanie 3 [4pkt]

Dla każdej z podanych złożoności podaj:

- problem (decyzyjny lub obliczeniowy)
- dowolną jego instancję
- algorytm rozwiązujący ten problem w zadanej złożoności (dosłownie w paru słowach, jeśli algorytm jest ogólnie znany to wystarczy podać nazwę)

Oczywiście dla dowolnej złożoności istnieje algorytm który w niej działa – może po prostu czekać wymaganą liczbę kroków. Prosiłbym o podanie różnych problemów i rozsądnych algorytmów, bez próby obejścia systemu :P

1. $\Theta(\lg n)$
2. $\Theta(n)$
3. $\Theta(n^2)$
4. $\Theta(2^n)$

Zadanie 4 [2 pkt]

Skonstruuj:

1. DFA rozpoznający język $L = \{w : w \text{ jest parzystej długości}\} \subseteq \{a\}^*$
2. NFA rozpoznający język $L = \{w : w \text{ nie zawiera spójnego pod słowa } ab\} \subseteq \{a, b\}^*$

Automat najprościej jest przedstawić albo jako graf w którym wierzchołki to stany a krawędzie to przejścia albo jako tabelkę przejść $Q \times \Sigma$.

Zadanie 5 [5pkt]

Udowodnij że DFA i NFA mają taką samą siłę ekspresji, czyli że dowolny język $L \subseteq \Sigma^*$ jest rozpoznany przez jakiś DFA wtedy i tylko wtedy gdy jest rozpoznawany przez jakiś NFA.

W jakiej złożoności (od liczby stanów) umiemy skonstruować równoważny NFA/DFA ?

Zadanie 6 [4pkt]

Rozważmy język $L = \{w : n\text{-ta litera od końca to } a\} \subseteq \{a, b\}^*$.

1. Skonstruuj DFA który rozpoznaje L
2. Skonstruuj NFA o co najwyżej $2n$ stanach który rozpoznaje L