Dominik Wawszczak
numer indeksu: 440014
numer grupy: 1

Algorytmika
praca domowa 4

Zadanie 1

Skoro w każdej klauzuli każda zmienna występuje w co najwyżej jednym literale, to zbiór X jest zbiorem rozwiązującym wtedy i tylko wtedy, gdy każda klauzula zawiera co najwyżej dwie zmienne niebędące w X. Wynika z tego, że jeśli X nie jest zbiorem rozwiązującym, to istnieje klauzula, która zawiera przynajmniej 3 zmienne niebędące w X. Obserwacja ta prowadzi do następującego algorytmu:

Algorithm 1 SolvingSet

procedure SolvingSet($k, X = \emptyset$)
    if every clause has at most two variables not in X **then**
        return true
    end if
    if $k \leqslant 0$ **then**
        return false
    end if
    $(a, b, c) \leftarrow$ three variables not in X from one clause
    return SolvingSet($k-1, X \cup \{a\}$) **or** SolvingSet($k-1, X \cup \{b\}$) **or** SolvingSet($k-1, X \cup \{c\}$)
end procedure

Algorytm ten wykorzystuje fakt, że jeśli aktualny zbiór X jest podzbiorem pewnego zbioru rozwiązującego, to przynajmniej jedna ze zmiennych a, b lub c musi być w tym zbiorze.

W drzewie rekurencji tego algorytmu, na i-tym poziomie głębokości jest co najwyżej 3^i wierzchołków, dla $i = 0, \ldots, k$. Łącznie daje to maksymalnie $\frac{3^{k+1}-1}{2}$ wywołań rekurencyjnych. W każdym wywołaniu musimy jedynie znaleźć odpowiednią klauzulę i wybrać z niej trzy zmienne, bądź stwierdzić, że taka klauzula nie istnieje, co oczywiście da się zrobić w czasie $|\varphi|^{O(1)}$. Ostateczna złożoność czasowa naszego algorytmu wynosi zatem $3^k \cdot |\varphi|^{O(1)}$, co kończy rozwiązanie pierwszej części zadania.