

Zadania domowe. Zestaw 4.1

Maciej Poleski

3 czerwca 2013

1

Zredukujemy problem subset-sum do problemu z treści zadania. Jeżeli S jest sumą wszystkich elementów, a B liczbą zadaną w problemie subset-sum to tworzymy dwa dodatkowe elementy $2S - B$ oraz $S + B$. Jeżeli w naszym zbiorze istnieje podzbiór o sumie równej B to po dodaniu do niego elementu $2S - B$ uzyskamy sumę $2S$ czyli tyle samo co pozostałe elementy $(S - B) +$ dodatkowy element $S + B$. Czyli zbiór wzbogacony o te dwa elementy da się podzielić na dwie części o równych sumach. Jeżeli się da to dwa dodatkowe elementy są w różnych podzbiorach (ponieważ ich suma to $3S$, tymczasem suma całego jednego podzbioru to $2S$). Wtedy w podzbiorze zawierającym element $2S - B$ suma pozostałych elementów jest równa B czyli tyle ile chcemy.

Mając rozwiązanie tego problemu możemy sprawdzić w czasie liniowym czy jest poprawne (sumując elementy z obu podzbiorów i sprawdzając czy obie sumy są równe). W takim razie jest to problem klasy NP.

Problem subset-sum jest NPC oraz może zostać zredukowany do problemu z treści zadania. W takim razie problem z treści zadania również jest problemem NPC.

2

Będziemy za pomocą wyroczni szukać wartościowania kolejnych zmiennych logicznych które będzie spełniać daną formułę.

Dla każdej zmiennej logicznej występującej w formule:

Podstaw wartość 1 pod wybraną zmienną

Zapytaj wyrocznię czy formuła jest spełnialna

Jeżeli nie to podstaw wartość 0

W każdej iteracji pętli wyszukujemy (w czasie wielomianowym) jakąkolwiek zmienną logiczną. Podstawiamy pod nią wartość 1 (redukując formułę) i pytamy wyrocznię czy taka formuła nadal jest spełnialna. Jeżeli tak - kolejny obieg pętli. Jeżeli formuła jest spełnialna, ale nie jest spełnialna gdy pod wybraną zmienną podstawimy 1 to znaczy że jest spełnialna gdy podstawimy pod nią 0. Więc podstawiamy i przechodzimy do następnego obiegu pętli. Wyszukiwanie w formule kolejnej zmiennej pod którą można coś podstawić jest wielomianowe względem długości formuły. Samo podstawienie również. Podstawienie zawsze skraca długość formuły. Pytanie do wyroczni zakładam że kosztuje wielomianowo. Ponieważ każda iteracja skraca długość formuły - iteracji będzie co najwyżej wielomianowo względem początkowej długości

formuły. Czyli pytań do wyroczni będzie co najwyżej wielomianowo wiele. Cały koszt również będzie co najwyżej wielomianowy. Redukcje:

$$a \vee 0 = a$$

$$a \wedge 1 = a$$

$$a \vee 1 = 1$$

$$a \wedge 0 = 0$$

$$\sim 0 = 1$$

$$\sim 1 = 0$$

Jeżeli formuła wejściowa jest spełnialna to istnieje wartościowanie (algorytm jakieś znajdzie), jeżeli nie to nie istnieje. Możemy na samym początku zapytać o to wyrocznię (jeżeli formuła wejściowa nie jest spełnialna - algorytm nie ma sensu).