PIZZO - zadanie domowe nr 2

Maria Wyrzykowska

Listopad 2021

1 Oznaczenia

Wprowadźmy następujące oznaczenia:
Formułę F z 3CNF (zakładamy, że jest w takiej postaci) możemy reprezentować jako zbiór {Σ, L, C},
gdzie Σ - zbiór zmiennych zdaniowych,
L - zbiór literałów,
C - zbiór (ponumerowanych) klauzul, reprezentowanych jako zbiory literałów, które je tworzą.
Dla przykładu F = $(p \vee q \vee \neg r) \wedge (\neg p \vee q \vee r)$ możemy reprezentować jako $\{\Sigma = \{p, q, r\}, L = \{p, \neg p, q, r, \neg r\}, C = \{c_1 = \{p, q, \neg r\}, c_2 = \{\neg p, q, r\}\}$.

2 3SAT $\leqslant_p$ Zainstaluj

Zdefiniujmy funkcję δ : {Σ, L, C} → {P, W, kolizje, wymagania}
(gdzie P, W zdefiniowane jak w treści zadania, kolizje, wymagania - zbiory kolizji i wymagań) następująco:

- P = $\{p_{a_0}$ dla a $\in \Sigma\} \cup \{p_{a_1}$ dla a $\in \Sigma\} \cup \{p_c$ dla i $= 1, ...|C|\}$
(Intuicyjnie: tworzymy pakiety odpowadające wszystkim możliwym literałom, które można utworzyć ze zmiennych z Σ, z rozróżnieniem czy literał jest pozytywny czy nie. Dodatkowo tworzymy po jednym pakiecie na klauzulę.)
- W = $\{p_c$ dla i $= 1, ...|C|\}$
(Intuicyjnie: wymagamy, żeby pakiety odpowiadające wszystkim klauzulom były zinstalowane)
- kolizje = $\{(p_{a_0}, p_{a_1})$ dla a $\in \Sigma$ }
(Intuicyjnie: w kolizji są pakiety odpowiadające pozytywnemu i negatywnemu literałowi danej zmiennej.)
- wymagania = $\{(p_{c_i}, (p_{x_y}$ t. że literał odpowiadający x_y wystąpił w klauzuli c_i)), dla i= $1, ..., |C|\}$
(Intuicyjnie: pakiety odpowiadające klauzulom wymagają któregoś z pakietów, które odpowiadają literałom tworzącym te klauzule)

Dla przykładu F opisanego w sekcji oznaczenia, $\delta(F)$:

- $P = \{p_{p_0}, p_{q_0}, p_{r_0}, p_{p_1}, p_{q_1}, p_{r_1}, p_{c_1}, p_{c_2}\}$
- $W = \{p_{c_1}, p_{c_2}\}$
- kolizje $= \{(p_{p_0}, p_{p_1}), (p_{q_0}, p_{q_1}), (p_{r_0}, p_{r_1})\}$
- wymagania $= \{(p_{c_1}, (p_{p_1}, p_{q_1}, p_{r_0})), (p_{c_2}, (p_{p_0}, p_{q_1}, p_{r_1})\}$

Pokażemy, że jest to odpowiednia redukcja wielomianowa.

1. Jest ona obliczalna w czasie wielomianowym: oczywiste. Na przykład wystarczy raz przejść po formule i notować występujące literały, klauzule i tworzyć wymagania. Następnie można przejść po wszystkich literałach i utworzyć kolizje.
2. F (formuła w 3CNF) jest w 3SAT wtedy i tylko wtedy, gdy $\delta(F)$ jest w Zainstaluj.
$\Rightarrow$: Weźmy dowolną formułę F z 3SAT. Istnieje wartościowanie ψ, które ją spełnia. Przekształćmy F funkcją δ. Pokażemy, że powstała instancja należy do Zainstaluj. Zbiór pakietów I potrzebny do spełnienia definicji może być wyznaczony na podstawie wartościowania ψ: wybieramy wszystkie pakiety odpowiadające klauzulom i te pakiety odpowiadające literałom, które przy wartościowaniu ψ są prawdziwe. Spełnia on definicję, ponieważ:

- Wszystkie pakiety z W są w nim zawarte (bo instalujemy wszystkie pakiety odpowiadające klauzulom).
- Żadna kolizja nie ma miejsca, ponieważ wartościowanie jest prawidłowe (zatem nie może istnieć taka zmienna p, że jednocześnie p i $\neg p$ są prawdziwe).
- Skoro przy wartościowaniu ψ F jest spełnione, to znaczy, że dla każdej klauzuli istnieje conajmniej jeden literał w niej występujący, który jest prawdziwy. Ze sposobu, w jaki działa δ, wynika że w takim razie wszystkie wymagania są spełnione jeśli zainstalujemy I.

$\Leftarrow$: Przez kontrapozycję. Weźmy dowolną F (formuła w 3CNF), taką, że $\delta(F)$ jest w Zainstaluj. Jest zatem zbiór pakietów I, który można wybrać by rozwiązać Zainstaluj dla $\delta(F)$. Pokażemy, że w takim razie F należy do 3SAT. Wartościowanie ψ, dla którego F jest spełnialna, można wyznaczyć następująco: wszystkie literały, których odpowiadające pakiety znajdują się w I, powinny być prawdziwe w ψ. Pozostałe literały muszą mieć zdefiniowaną jakąś wartość, niech więc wszystkie zmienne, które jeszcze nie mają przypisanej jakiejś wartości będą fałszywe (nie popsuje to żadnych wymagań). Zauważmy, że takie wartościowanie jest poprawne, ponieważ w I nie ma kolizji (nie istnieje zmienna z dwoma wartościami). Takie wartościowanie spełnia F, ponieważ wszystkie wymagania są spełnione, zatem w każdej klauzuli istnieje przynajmniej jeden literał obliczający się do prawdy.

3 Zainstaluj jest w NP

Pokażemy, że Zainstaluj jest rozwiązywalne przez niedeterministyczną maszynę Turinga w czasie wielomianowym. Niech maszyna działa następująco:
Faza "zgadywania":

1. Weź wszystkie pakiety z W do I.
2. Przejdź po wszystkich wymaganiach. Dla każdego wymagania, wylosuj, czy ma być spełnione czy nie. Jeśli ma być spełnione i aktualnie wybrane pakiety jeszcze go nie spełniają, wylosuj jeden z pakietów, które je spełnią i dodaj go oraz pakiet dla którego spełniamy to wymaganie do I.

Na pewno nie jest to efektywne, ale jeśli istnieje zbiór I spełniający daną instancję Zainstaluj, to da się go tak zgadnąć. Całość zajmie około $O(w * n)$, gdzie w - liczba wymagań, n - liczba pakietów w P.
Faza "sprawdzania":

1. Sprawdź, czy nie zachodzi żadna kolizja. Jeśli zachodzi: odrzucaj.
2. Sprawdź, czy wymagania dla wszystkich pakietów z I są spełnione. Jeśli nie: odrzucaj.
3. Akceptuj.

Ta faza zajmuje około $O(k + w * n)$, zatem cała maszyna działa w czasie wielomianowym, więc Zainstaluj jest w NP.