

## Przykłady problemów NP-zupełnych

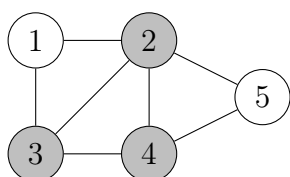
W czasie poprzedniego wykładu poznaliśmy dwa ważne problemy NP-zupełne SAT oraz 3SAT związane ze spełnialnością formuł logicznych. W czasie dzisiejszego wykładu wykorzystamy je do wykazania NP-zupełności kilku innych ciekawych problemów obliczeniowych.

Przypomnijmy, problem nazywamy NP-zupełnym jeśli należy do klasy NP oraz każdy inny problem z klasy NP redukuje się do niego w czasie wielomianowym. Dowodząc NP-zupełności nowego problemu wystarczy więc wykazać, że dowolny znany już problem NP-zupełny redukuje się do niego w czasie wielomianowym.

---

### Problem pokrycia wierzchołkowego

---



Niech  $G$  będzie grafem nieskierowanym. Podzbiór  $S$  jego wierzchołków nazywamy pokryciem wierzchołkowym  $G$ , jeśli każda krawędź  $G$  ma co najmniej jeden koniec w zbiorze  $S$ . Przykładowo, dla grafu umieszczonego na rysunku pokryciem wierzchołkowym może być zbiór  $S = \{2, 3, 4\}$ .

Problem pokrycia wierzchołkowego polega na rozstrzygnięciu czy graf posiada pokrycie wierzchołkowe ustalonego rozmiaru.

$$\text{VERTEX-COVER} = \left\{ (G, k) : G \text{ ma pokrycie wierzchołkowe rozmiaru } k \right\}.$$

**Twierdzenie 13.1**

Problem VERTEX-COVER jest NP-zupełny.

**Dowód****(1) VERTEX-COVER  $\in$  NP**

Dla dowolnego podzbioru  $S$  składającego się z  $k$  wierzchołków grafu  $G$  deterministyczna maszyna Turinga jest w stanie w czasie wielomianowym zweryfikować, czy jest on pokryciem wierzchołkowym  $G$ .

**(2) VERTEX-COVER jest NP-zupełny**

Pokażemy redukcję problemu 3SAT do problemu VERTEX-COVER działającą w czasie wielomianowym. W tym celu dla dowolnej formuły logicznej  $\phi$  w postaci 3CNF zbudujemy graf  $G$  oraz wyznaczymy liczbę  $k$  w taki sposób, że formuła  $\phi$  będzie spełnialna wtedy i tylko wtedy, gdy graf  $G$  będzie miał pokrycie wierzchołkowe rozmiaru  $k$ .

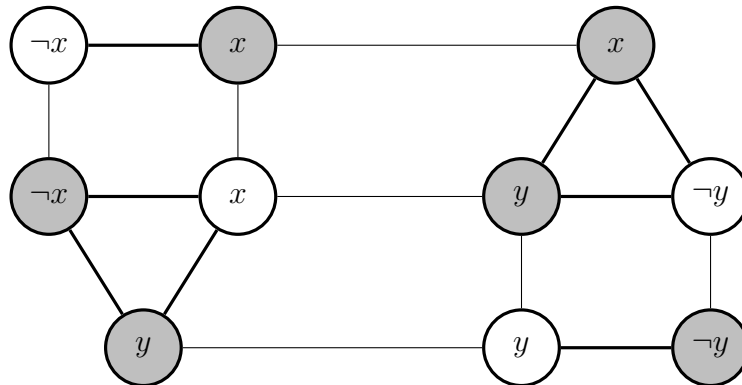
Niech  $\phi$  będzie formułą logiczną w postaci 3CNF składającą się z  $m$  zmiennych oraz  $l$  klauzul. Odpowiadający jej graf  $G$  konstruujemy według następującego schematu:

- Dla każdej zmiennej  $x$  występującej w  $\phi$  tworzymy *podgraf zmiennej* składający się z dwóch wierzchołków etykietowanych  $x$  oraz  $\neg x$  połączonych krawędzią.
- Dla każdej klauzuli  $(x \vee y \vee z)$  tworzymy *podgraf klauzuli* składający się z trzech wierzchołków z etykietami odpowiadającymi występującym w niej literalom połączonych krawędziami każdy z każdym.
- Każdy wierzchołek  $v$  występujący w podgrafie klauzuli łączymy z wierzchołkiem  $v'$  w podgrafie zmiennej o identycznej etykiecie.

Tak skonstruowany graf  $G$  ma  $2m + 3l$  wierzchołków. Ustalamy rozmiar pokrycia wierzchołkowego na  $k = m + 2l$ .

**Przykład 13.1**

Niech  $\phi = (x \vee y \vee \neg x) \wedge (x \vee y \vee \neg y)$ . Rozważana formuła przyjmuje wartość *true* dla wartościowania  $(x = \text{true}, y = \text{false})$ , a zatem jest spełnialna. Odpowiadający jej graf nieskierowany z zaznaczonym pokryciem wierzchołkowym rozmiaru  $2 + 2 \cdot 2 = 6$  znajduje się na rysunku poniżej. Podgrafy zmiennych oraz klauzul zostały wyróżnione grubszymi liniami.



Musimy teraz udowodnić, że graf  $G$  skonstruowany dla formuły  $\phi$  według schematu opisanego powyżej ma pokrycie wierzchołkowe rozmiaru  $k = m + 2l$  wtedy i tylko wtedy, gdy formuła  $\phi$  jest spełnialna.

$\Rightarrow$ :

Założmy, że formuła  $\phi$  jest spełnialna. Pokrycie wierzchołkowe  $S$  rozmiaru  $k = m + 2l$  w grafie  $G$  konstruujemy następująco:

- Z każdego podgrafu zmiennej do  $S$  wybieramy wierzchołek odpowiadający literalowi mającemu wartość *true*.
- Z każdego podgrafu klauzuli wybieramy jeden literal mający wartość *true* i dodajemy do  $S$  pozostałe 2 wierzchołki.

Zauważmy, że skonstruowaliśmy podzbiór  $S$  rozmiaru  $k = m + 2l$  zbioru wszystkich wierzchołków  $G$  o następujących własnościach:

- Każda krawędź należąca do podgrafu zmiennej jest pokryta.
- Każda krawędź należąca do podgrafu klauzuli jest pokryta.
- Każda krawędź łącząca podgraf klauzuli z podgrafem zmiennej jest pokryta przez wierzchołek z wartością *true* z podgrafu zmiennej lub jeden z dwóch pozostałych wierzchołków z podgrafu klauzuli.

Zatem  $S$  jest pokryciem wierzchołkowym grafu  $G$  o rozmiarze  $k = m + 2l$ .

$\Leftarrow$ :

Założmy, że zbiór  $S$  jest pokryciem wierzchołkowym grafu  $G$  o rozmiarze  $k = m + 2l$ . Wówczas zbiór  $S$  musi zawierać po jednym wierzchołku z każdego podgrafu zmiennej oraz po dwa wierzchołki z każdego podgrafu klauzuli – razem  $m + 2l$  wierzchołków. Wartościowanie formuły  $\phi$  otrzymujemy przypisując wartość *true* wszystkim wierzchołkom wybranym z podgrafów zmiennej. Dzięki temu co najmniej jeden literal w każdej klauzuli jest prawdziwy, a zatem cała formuła również jest prawdziwa.

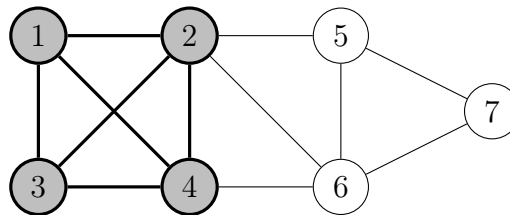
Pozostało nam jeszcze oszacować złożoność obliczeniową rozważanej redukcji. Dla formuły  $\phi$  składającej się z  $m$  zmiennych oraz  $l$  klauzul tworzymy graf składający się z  $2m+3l$  wierzchołków oraz  $6m+l$  krawędzi. Konstrukcja grafu może więc być wykonana w czasie  $O(n^2)$ , gdzie  $n$  jest długością formuły  $\phi$ . ■

## Problem kliki

Kliką w grafie nieskierowanym nazywamy podgraf, w którym każde dwa wierzchołki połączone są krawędzią. Klikę rozmiaru  $k$  nazywamy  $k$ -kliką. Problem kliki polega na rozstrzygnięciu, czy graf nieskierowany  $G$  zawiera klikę danego rozmiaru.

$$\text{CLIQUE} = \left\{ (G, k) : G \text{ zawiera klikę rozmiaru } k \right\}.$$

Na poniższym rysunku zaznaczona została 4-klika w grafie nieskierowanym.



### Twierdzenie 13.2

Problem CLIQUE jest NP-zupełny.

#### Dowód

##### (1) CLIQUE $\in$ NP

Dla podanego grafu  $G$  niedeterministyczna maszyna Turinga potrafi w czasie wielomianowym wybrać zbiór  $S$  składający się z  $k$  wierzchołków i rozstrzygnąć czy jest on kliką rozmiaru  $k$ . Podobnie, deterministyczna maszyna Turinga potrafi zweryfikować w czasie wielomianowym, czy wybrany zbiór  $S$  jest  $k$ -kliką w grafie  $G$ .

##### (2) CLIQUE jest NP-zupełny

Pokażemy redukcję w czasie wielomianowym problemu 3SAT do problemu CLIQUE. Dla formuły logicznej  $\phi$  skonstruujemy graf  $G$ , który będzie zawierał  $k$ -klikę wtedy i tylko wtedy, gdy formuła  $\phi$  jest spełnialna.

Niech  $\phi$  będzie formułą logiczną składającą się z  $k$  klauzul:

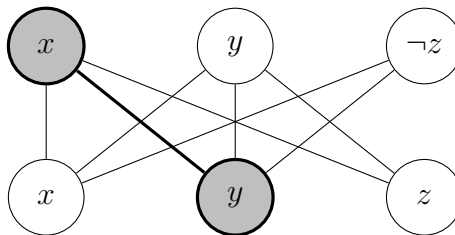
$$\phi = (x_1 \vee y_1 \vee z_1) \wedge \dots \wedge (x_k \vee y_k \vee z_k).$$

Odpowiadający jej graf  $G$  konstruujemy według następującego schematu:

- Tworzymy  $3k$  wierzchołków etykietowanych wystąpieniami wszystkich literałów w formule  $\phi$ .
- Grupujemy wierzchołki trójkami odpowiadającymi klauzulom  $\phi$ .
- Każdy wierzchołek grafu  $G$  o etykiecie  $x$  łączymy krawędzią ze wszystkimi pozostałymi poza wierzchołkami należącymi do tej samej trójki oraz wierzchołkami etykietowanymi  $\neg x$ .

### Przykład 13.2

Niech  $\phi = (x \vee y \vee \neg z) \wedge (x \vee y \vee z)$ . Rozważana formuła jest prawdziwa dla wartościowania  $x = \text{true}, y = \text{true}, z = \text{false}$ , a zatem jest spełnialna. Odpowiadający jej graf znajduje się na poniższym rysunku.



Pogrubiczonymi liniami zaznaczone zostały wierzchołki 2-kliki. Literały, których prawdziwość gwarantuje prawdziwość całej formuły  $\phi$ , zostały wyróżnione kolorem.

Musimy teraz udowodnić, że w grafie  $G$  skonstruowanym dla formuły  $\phi$  według schematu opisanego powyżej istnieje  $k$ -klika wtedy i tylko wtedy, gdy formuła  $\phi$  jest spełnialna.

$\Leftarrow$ :

Założmy, że formuła  $\phi$  jest spełnialna. Istnieje zatem wartościowanie, dla którego co najmniej jeden literał z każdej klauzuli ma wartość *true*. Z każdej trójki wierzchołków  $G$  odpowiadającej klauzuli formuły  $\phi$  wybieramy dowolny wierzchołek etykietowany literałem mającym wartość *true*. Otrzymaliśmy zbiór  $S$  składający się z  $k$  wierzchołków. Zauważmy, że żadna para wierzchołków ze zbioru  $S$  nie należy do tej samej trójki. Ponadto, ponieważ wszystkie wybrane literały mają wartość *true*, żadna para wierzchołków ze

zbioru  $S$  nie ma etykiet przeciwnych. Zatem, z konstrukcji grafu  $G$ , każda para wierzchołków ze zbioru  $S$  jest połączona krawędzią. Zbiór  $S$  jest więc  $k$ -kliką w grafie  $G$ .

$\implies$ :

Założmy, że w grafie  $G$  istnieje zbiór wierzchołków  $S$  tworzący  $k$ -klikę. Z konstrukcji  $G$  wynika, że żadna para wierzchołków ze zbioru  $S$  nie należy do tej samej trójki wierzchołków, a tym samym do odpowiadającej jej klauzuli formuły  $\phi$ . Ponadto, żadna para wierzchołków ze zbioru  $S$  nie może mieć etykiet przeciwnych. Przypisując wartość *true* literalom będącym etykietami wierzchołków wybranych do  $S$  zapewniamy, że w każdej klauzuli formuły  $\phi$  dokładnie jeden literal jest prawdziwy. Otrzymujemy więc wartościowanie spełniające formułę  $\phi$ .

Pozostało nam oszacowanie złożoności obliczeniowej opisanej redukcji. Dla formuły  $\phi$  zawierającej  $n$  literalów tworzymy graf składający się z  $n$  wierzchołków oraz  $O(n^2)$  krawędzi. Zatem rozważana redukcja ma złożoność czasową rzędu  $O(n^2)$ . ■

## Problem sumy podzbioru

Problem sumy podzbioru polega na rozstrzygnięciu, czy dla danego multizbioru liczb naturalnych  $S = \{x_1, x_2, \dots, x_n\}$  oraz ustalonej liczby naturalnej  $k$  istnieje podzbiór  $Y \subseteq S$ , którego elementy sumują się do  $k$ .

$$\text{SUBSET-SUM} = \{(S, k) : S = \{x_1, \dots, x_n\} \text{ t.ż. } \exists Y = \{y_1, \dots, y_m\} \sum y_i = k\}.$$

### Przykład 13.3

Rozważmy parę  $(X, k) = (\{11, 12, 14, 22, 34, 45\}, 25)$ . Wybierając podzbiór  $Y = \{11, 14\}$  otrzymujemy  $11 + 14 = 25$ , a zatem  $(X, k) \in \text{SUBSET-SUM}$ .

### Twierdzenie 13.3

Problem SUBSET-SUM jest NP-zupełny.

### Dowód

(1) SUBSET-SUM  $\in$  NP

Dla podanego multizbioru  $S$  niedeterministyczna maszyna Turinga potrafi w czasie wielomianowym wybrać zbiór  $Y$ , a następnie wyznaczyć sumę jego elementów. Z drugiej strony, maszyna deterministyczna potrafi w czasie

wielomianowym zweryfikować, czy suma elementów wybranego podzbioru  $Y$  jest równa  $k$ .

**(2) SUBSET-SUM jest NP-zupełny**

Pokażemy redukcję w czasie wielomianowym problemu 3SAT do problemu SUBSET-SUM.

Niech  $\phi$  będzie formułą logiczną zawierającą zmienne  $x_1, x_2, \dots, x_m$  oraz klauzule  $c_1, c_2, \dots, c_k$ . Odpowiadający jej zbiór  $S$  tworzymy według następującego schematu. Dla każdej zmiennej  $x_i$  występującej w formule  $\phi$  do zbioru  $S$  dodajemy parę liczb dziesiętnych  $y_i$  oraz  $z_i$ :

$$y_i = 1 \underbrace{0 \dots 0}_{m-i} \underbrace{10 \dots 01}_k$$

$$z_i = 1 \underbrace{0 \dots 0}_{m-i} \underbrace{01 \dots 01}_k$$

Lewa część dodawanych liczb odpowiada numeracji zmiennych, natomiast prawa klauzulom. Liczba  $y_i$  zawiera 1 na pozycji  $j$ , jeśli w klauzli  $c_j$  występuje  $x_i$ , natomiast  $z_i$  zawiera 1 na pozycji  $j$ , jeśli w klauzli  $c_j$  występuje  $\neg x_i$ . Ponadto, do zbioru  $S$  dodajemy pary liczb  $g_1, h_1, \dots, g_k, h_k$  postaci

$$g_i = h_i = 1 \underbrace{0 \dots 0}_{k-i}$$

oraz przyjmujemy

$$z = \underbrace{1 \dots 1}_m \underbrace{3 \dots 3}_k.$$

**Przykład 13.4**

Rozważmy formułę w postaci 3CNF

$$\phi = (x_1 \vee x_2 \vee \neg x_1) \wedge (x_1 \vee x_2 \vee \neg x_2) \wedge (x_1 \vee \neg x_1 \vee \neg x_2).$$

Jest ona prawdziwa dla wartościowania ( $x_1 = false, x_2 = true$ ), a zatem jest spełnialna. Struktura odpowiadającego jej zbioru  $S$  jest przedstawiona w poniższej tabeli

	$x_1$	$x_2$	$c_1$	$c_2$	$c_3$
$y_1$	1	0	1	1	1
$z_1$	1	0	1	0	1
$y_2$		1	1	1	0
$z_2$		1	0	1	1
$g_1$			1	0	0
$h_1$			1	0	0
$g_2$				1	0
$h_2$				1	0
$g_3$					1
$h_3$					1
$z$	1	1	3	3	3

Musimy udowodnić, że zbiór  $S$  zawiera podzbiór  $Y$ , którego elementy sumują się do  $z$ , wtedy i tylko wtedy, gdy formuła  $\phi$  jest spełnialna.

$\Leftarrow$ :

Założmy, że formuła  $\phi$  jest spełnialna. Dla ustalonego wartościowania zmiennych spełniającego  $\phi$  do zbioru  $Y$  wybieramy  $y_i$  jeśli  $x_i$  ma wartość *true* lub  $z_i$  jeśli  $x_i$  ma wartość *false*. Po zsumowaniu elementów zbioru  $Y$  otrzymujemy liczbę dziesiętną, która na pierwszych  $m$  pozycjach ma cyfrę 1 natomiast na kolejnych  $k$  pozycjach cyfry z zakresy 1-3 (ponieważ co najmniej jeden literał w każdej klauzuli musi mieć wartość *true*). Aby otrzymać sumę elementów równą  $z$ , do zbioru  $Y$  dodajemy niezbędne elementy  $g_i$  oraz  $h_i$ .

Formuła z Przykładu 4 jest spełniona np. dla wartościowania ( $x_1 = \text{false}$ ,  $x_2 = \text{true}$ ). Do zbioru  $Y$  wybieramy więc  $y_1$  oraz  $z_2$  otrzymując sumę jego elementów równą 11122. Aby uzyskać sumę elementów równą 11333 do zbioru  $Y$  dodajemy dodatkowo  $g_1$ ,  $h_1$ ,  $g_2$  oraz  $g_3$ .

$\Rightarrow$ :

Założmy, że istnieje podzbiór  $Y \subseteq S$ , którego elementy sumują się do  $z$ . Zauważmy, że:

- Wszystkie cyfry liczb tworzących sumę są równe 0 lub 1
- Każda kolumna tablicy opisującej zbiór  $S$  zawiera co najwyżej pięć jedynek, więc przy dodawaniu nie występuje przeniesienie
- Aby otrzymać 1 na  $m$  najbardziej znaczących pozycjach sumy musimy wybrać dokładnie jeden element z każdej pary  $(y_i, z_i)$ .

Wartościowanie formuły  $\phi$  tworzymy przypisując zmiennej  $x_i$  wartość *true* jeśli  $y_i \in Y$  oraz *false* jeśli  $z_i \in Y$ . Ostatnie  $k$  pozycji liczby  $z$  odpowiada



poszczególnym klauzulom formuły  $z$ . Na każdej z nich występuje cyfra 3. Wybranie elementów  $g_i$  oraz  $h_i$  daje w sumie co najwyżej 2, zatem każda klauzula  $\phi$  zawiera co najmniej jeden literal ( $x_i$  lub  $\neg x_i$ ) mający wartość *true*. Formuła  $\phi$  jest więc spełnialna.

Pozostało oszacowanie złożoności czasowej opisanej redukcji. Dla formuły  $\phi$  składającej się z  $m$  zmiennych oraz  $k$  klauzul redukcja polega na stworzeniu tabeli zbioru rozmiaru  $O((k+m)^2)$ . Może więc być wykonana w czasie  $O(n^2)$ , gdzie  $n$  jest długością formuły  $\phi$ . ■

## Problem ścieżki Hamiltona

Ścieżką Hamiltona w grafie skierowanym  $G$  nazywamy ścieżkę skierowaną przechodzącą przez każdy jego wierzchołek dokładnie raz. Problem ścieżki Hamiltona polega na rozstrzygnięciu, czy w grafie skierowanym  $G$  istnieje ścieżka Hamiltona łącząca dwa ustalone wierzchołki  $v_1$  oraz  $v_2$ .

$$\text{HAMPATH} = \left\{ (G, v_1, v_2) : G \text{ zawiera ścieżkę Hamiltona } v_1 \rightsquigarrow v_2 \right\}.$$

### Twierdzenie 13.4

Problem HAMPATH jest NP-zupełny.

### Dowód

#### (1) HAMPATH $\in$ NP

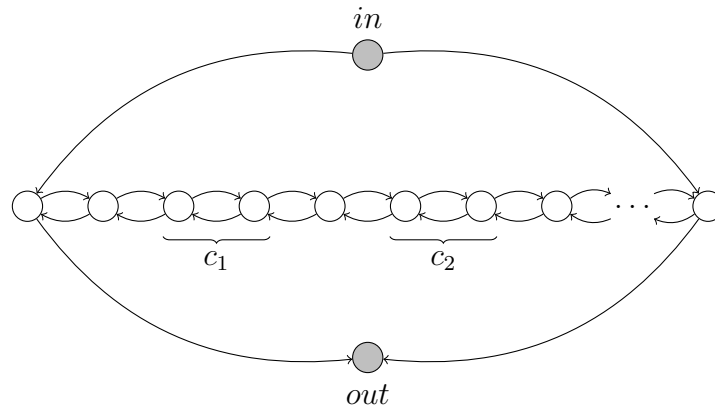
Dla grafu skierowanego  $G$  niedeterministyczna maszyna Turinga może w czasie wielomianowym wybrać ścieżkę łączącą wyróżnione wierzchołki  $v_1$  oraz  $v_2$ . Z drugiej strony, dla wybranej ścieżki maszyna deterministyczna może w czasie wielomianowym zweryfikować, czy jest ona ścieżką Hamiltona.

#### (2) HAMPATH jest NP-zupełny

Pokażemy redukcję problemu spełnialności 3SAT do problemu HAMPATH działającą w czasie wielomianowym. W tym celu dla dowolnej formuły logicznej  $\phi$  w postaci 3CNF zbudujemy graf  $G_\phi$  oraz wybierzemy parę jego wierzchołków  $v_1$  oraz  $v_2$  w taki sposób, że formuła  $\phi$  będzie spełnialna wtedy i tylko wtedy, gdy graf  $G$  będzie miał ścieżkę Hamiltona  $v_1 \rightsquigarrow v_2$ .

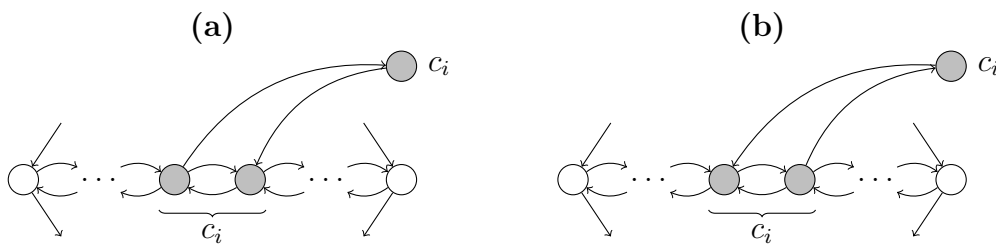
Niech  $\phi$  będzie formułą logiczną w postaci 3CNF zawierającą  $m$  zmiennych oraz  $k$  klauzul. Odpowiadający jej graf skierowany  $G_\phi$  konstruujemy według następującego schematu.

- Każdej klauzuli  $c_i$  formuły  $\phi$  odpowiada pojedynczy wierzchołek  $G_\phi$  z etykietą  $c_i$ .
- Każdej zmiennej  $x_i$  formuły  $\phi$  odpowiada podgraf postaci:



Wyróżnione kolorem wierzchołki *in* oraz *out* służą do łączenia ze sobą podgrafów zmiennych. Przekątna podgrafu odpowiada klauzulom formuły  $\phi$ . Każdej klauzuli odpowiada dokładnie jedna para wierzchołków. Pary odpowiadające poszczególnym klauzulom rozdzielone są dodatkowo pojedynczym wierzchołkiem.

- Wierzchołki odpowiadające klauzulom łączymy z podgrafami odpowiadającym zmiennym następująco:



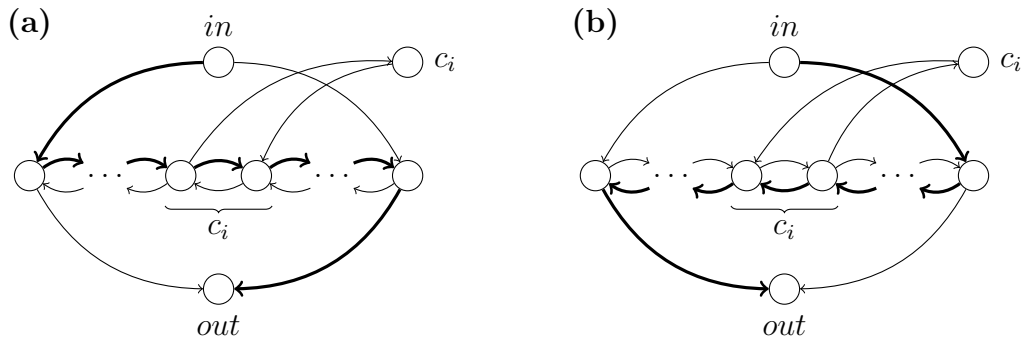
Jeśli klauzula  $c_i$  zawiera literal  $x_j$ , łączymy wierzchołek odpowiadający  $c_i$  z podgrafem odpowiadającym  $x_j$  (dokładnie z wyróżnioną parą wierzchołków na jego przekątnej) w sposób pokazany na rysunku (a). Jeśli natomiast klauzula  $c_i$  zawiera literal  $\neg x_j$ , łączymy wierzchołek odpowiadający  $c_i$  z podgrafem odpowiadającym  $x_j$  w sposób pokazany na rysunku (b).

- Podgrafy odpowiadające kolejnym zmiennym  $x_1, x_2, \dots, x_m$  łączymy w ten sposób, że wierzchołek *out* podgrafu zmiennej  $x_i$  staje się wierzchołkiem *in* podgrafu zmiennej  $x_{i+1}$ .
- Wierzchołkiem początkowym ( $v_1$ ) szukanej ścieżki Hamiltona jest wierzchołek *in* podgrafu odpowiadającego zmiennej  $x_1$ , natomiast wierzchołkiem końcowym ( $v_2$ ) jest wierzchołek *out* podgrafu odpowiadającego zmiennej  $x_m$ .

Musimy teraz udowodnić, że formuła  $\phi$  jest spełnialna wtedy i tylko wtedy, gdy graf  $G_\phi$  zawiera ścieżkę Hamiltona  $v_1 \rightsquigarrow v_2$ .

$\implies$ :

Założmy, że formuła  $\phi$  jest spełnialna. Istnieje zatem wartościowanie zmiennych  $x_1, x_2, \dots, x_m$ , dla którego jest ona prawdziwa. Konstruujemy ścieżkę  $v_1 \rightsquigarrow v_2$ . Jeśli w rozważanym wartościowaniu formuły zmienna  $x_i$  ma wartość *true* odpowiadający jej podgraf przechodzimy w kierunku zaprezentowanym na rysunku (a). Jeśli natomiast zmienna  $x_i$  ma wartość *false* odpowiadający jej podgraf przechodzimy w kierunku zaprezentowanym na rysunku (b).



Zauważmy, że skonstruowana w ten sposób ścieżka zawiera wszystkie wierzchołki grafu  $G_\phi$  poza wierzchołkami odpowiadającymi klauzulom. Z każdej klauzuli  $c_i$  wybieramy dokładnie jeden literał  $x_j$  lub  $\neg x_j$ , który w rozważanym wartościowaniu ma wartość *true* i dołączamy go do fragmentu konstruowanej ścieżki przechodzącego przez podgraf odpowiadający  $x_j$ . Z konstrukcji  $G_\phi$  wynika, że krawędzie łączące wierzchołek odpowiadający  $c_i$  z wierzchołkami podgrafu odpowiadającego  $x_j$  mają odpowiedni zwrot. Skonstruowana ścieżka  $v_1 \rightsquigarrow v_2$  jest zatem ścieżką Hamiltona.

$\implies$ :

Założmy, że w grafie  $G_\phi$  istnieje ścieżka Hamiltona  $v_1 \rightsquigarrow v_2$  przechodząca kolejno przez podgrafy wszystkich zmiennych i odstupująca od tej reguły wyłącznie po to, żeby odwiedzić wierzchołki odpowiadające klauzulom  $c_j$ .

Wówczas łatwo jest znaleźć wartościowanie spełniające formułę  $\phi$ . Zależnie od kierunku przechodzenia podgrafu odpowiadającego zmiennej  $x_i$  nadajemy jej wartość *true* lub *false*.

Przypuśćmy teraz, że w grafie  $G_\phi$  istnieje ścieżka Hamiltona  $v_1 \rightsquigarrow v_2$  nie spełniająca warunków opisanych powyżej. Musi zatem zawierać fragment  $v_{x_a} \rightarrow v_{c_i} \rightarrow v_{x_b}$ , gdzie  $v_{x_a}$  jest wierzchołkiem należącym do wnętrza podgrafu odpowiadającego zmiennej  $x_a$ ,  $v_{x_b}$  jest wierzchołkiem należącym do wierzchołka podgrafu odpowiadającego zmiennej  $x_b$ , zaś  $v_{c_i}$  jest wierzchołkiem odpowiadającym klauzuli  $c_i$ . Z konstrukcji grafu  $G_\phi$  wynika jednak, że do rozważanej ścieżki nie będzie mógł należeć jeden z wierzchołków sąsiadujących z  $v_{x_a}$ . Zatem ścieżka  $v_1 \rightsquigarrow v_2$  nie będzie ścieżką Hamiltona.

Pozostało oszacowanie złożoności czasowej opisanej redukcji. Dla formuły  $\phi$  składającej się z  $m$  zmiennych oraz  $k$  klauzul redukcja polega na stworzeniu grafu  $G_\phi$  rozmiaru  $O(k \cdot m)$ . Może więc być wykonana w czasie  $O(n^2)$ , gdzie  $n$  jest długością formuły  $\phi$ . ■