# COMPLESSITÀ

## RIDUZIONI DI TEMPO POLINOMIALE

# Riduzioni di tempo polinomiale mediante gadgets

- Quando costruiamo una riduzione di tempo polinomiale da $3SAT$ ad un linguaggio, cerchiamo strutture in quel linguaggio che possono simulare le variabili e le clausole nelle formule booleane.
- Tali strutture sono a volte chiamate *gadget* (tecnica di "riduzione mediante progettazione di componenti" o "gadgets").
- Occorre:
  - definire per ogni variabile una componente (*gadget*) (modella l'assegnazione di verità alla variabile)
  - definire per ogni clausola una componente (modella la soddisfacibilità della clausola)
  - Collegare i due insiemi di componenti per garantire che l'assegnazione alle variabili soddisfi tutte le clausole.
  - Non è l'unica tecnica.

# Riduzioni di tempo polinomiale mediante progettazione di componenti

- Esempio: $3SAT \leq_P CLIQUE$
- ogni variabile è simulata da un nodo
- ogni clausola è simulata da una tripla di nodi
- I nodi sono collegati da archi tali che un assegnamento alle variabili che soddisfa tutte le clausole è quello che assegna valore vero ai letterali corrispondenti ai nodi nella clique.

Sia $G = (V, E)$ un grafo non orientato.

- Sia $V' \subseteq V$, sia $(u, v) \in E$. Se $V'$ contiene almeno uno dei due vertici dell'arco $(u, v)$, diremo che $V'$ **copre** l'arco $(u, v)$.

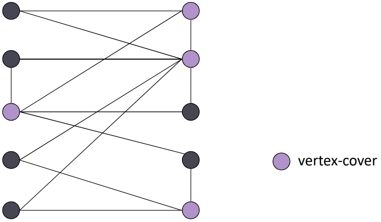
- Un **vertex cover** $V'$ di $G$ è un sottoinsieme $V'$ di $V$ tale che per ogni $(u, v) \in E$ risulta $\{u, v\} \cap V' \neq \emptyset$.
  ($V'$ copre ogni arco $(u, v)$ in $G$.)

# vertex-cover

Dato un grafo non orientato G = (V, E) , un vertex cover è un sottoinsieme V' di vertici, $V' \subseteq V$, tale che per ogni arco in E almeno uno dei suoi estremi è in V'

Ex. esiste un vertex-cover di cardinalità 4

Ex. Non esiste un vertex-cover di cardinalità $\leq 3$




(autore slide: Kevin Wayne)

# *VERTEX-COVER* è *NP*-completo

Il problema di stabilire se un grafo non orientato $G = (V, E)$ ha un vertex cover di cardinalità $k$ si può formulare come un problema di decisione, il cui linguaggio associato è *VERTEX-COVER*.

*VERTEX-COVER* $= \{\langle G, k\rangle \mid G$ è un grafo non orientato che ha un vertex cover di cardinalità $k\}$

**Teorema**

*VERTEX-COVER* $\in$ *NP*

**Dimostrazione.**

Un algoritmo $V$ che verifica *VERTEX-COVER* in tempo polinomiale:

$V =$ "Sull'input $\langle\langle G, k\rangle, c\rangle$:

1. Verifica se $c$ è un insieme $V'$ di $k$ nodi di $G$, altrimenti rifiuta.
2. Verifica se $V'$ copre ogni arco in $G$, accetta in caso affermativo; altrimenti rifiuta."

$\exists c : \langle\langle G, k\rangle, c\rangle \in L(V) \Leftrightarrow \langle G, k\rangle \in$ *VERTEX-COVER* $\square$

Prova alternativa: utilizzare le macchine di Turing non deterministiche.

# *VERTEX-COVER* è *NP*-completo

Teorema

*VERTEX-COVER è NP-completo.*

Dimostrazione.

Abbiamo provato che $VERTEX\text{-}COVER \in NP$.

Per concludere la prova, dimostriamo che

$3SAT \leq_P VERTEX\text{-}COVER$.

# *VERTEX-COVER* è *NP*-completo

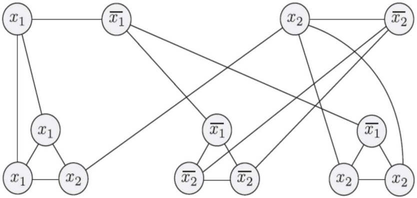



**FIGURA 7.45**
Il grafo che la riduzione produce a partire da
$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$

Figura tratta da M. Sipser, Introduzione alla teoria della computazione.

# $3SAT$ si riduce in tempo polinomiale a $VERTEX\text{-}COVER$

- Sia $\phi$ una formula $3CNF$ con $\ell$ clausole e $m$ variabili:

$$(a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_\ell \vee b_\ell \vee c_\ell)$$

- Definiamo un grafo non orientato $G$ e un intero $k$ tale che $\phi$ è soddisfacibile se e solo se $G$ ha un vertex cover di cardinalità $k$.
- Inoltre $\langle G, k \rangle$ può essere costruita in tempo polinomiale nella lunghezza di $\langle \phi \rangle$.

# $3SAT$ si riduce in tempo polinomiale a $VERTEX\text{-}COVER$

Costruzione:

- $G$ contiene due vertici per ogni variabile $x$ etichettati con $x$ e $\overline{x}$ (gadget per le variabili). Chiamiamo $V_1$ questo insieme di vertici.
- $G$ contiene tre vertici per ogni clausola, etichettati con i tre letterali della clausola (gadget per le clausole). Chiamiamo $V_2$ questo insieme di vertici.
- Connettiamo i due vertici associati a una variabile con un arco
- Connettiamo i tre vertici associati a una clausola tra loro in un triangolo
- Connettiamo con un arco ogni vertice nel triangolo (associato a una clausola) al vertice in $V_1$ (gadget per le variabili) che ha la stessa etichetta.
- Se $\phi$ ha $\ell$ clausole e $m$ variabili allora $G$ ha $2m + 3\ell$ vertici e $V = V_1 \cup V_2$.
- Prendiamo $k = m + 2\ell$.

# $3SAT$ si riduce in tempo polinomiale a $VERTEX\text{-}COVER$

Proviamo che $\phi$ è soddisfacibile se e solo se $G = (V, E)$ ha un vertex cover di cardinalità $k$.

- Sia $\phi$ soddisfacibile e sia $\tau$ un assegnamento che soddisfa $\phi$. Consideriamo il sottoinsieme $V'$ di $V$ che contiene:
  - tutti i vertici in $V_1$ (gadget per le variabili) che hanno come etichette i letterali veri in $\tau$
  - due vertici per ogni triangolo (gadget per una clausola), escludendone uno che ha etichetta uguale a un vertice selezionato al passo precedente (ne esiste almeno uno).
- Se $\phi$ ha $\ell$ clausole e $m$ variabili allora questo sottoinsieme $V'$ di $V$ ha taglia $k = m + 2\ell$.
- Inoltre $V'$ è un vertex cover:
  - tutti gli archi in un triangolo sono coperti (dai due vertici selezionati)
  - tutti gli archi tra due vertici di $V_1$ o tra un vertice di $V_1$ e un vertice di $V_2$ sono coperti (dalla scelta dei vertici in $V_1$ o $V_2$).

# $3SAT$ si riduce in tempo polinomiale a $VERTEX\text{-}COVER$

- Supponiamo che $G = (V, E)$ abbia un vertex cover $V'$ di cardinalità $k = m + 2\ell$ e proviamo che $\phi$ è soddisfacibile.

- $V'$ deve contenere almeno 2 vertici di ogni triangolo e, per ogni arco tra due vertici di $V_1$ (gadget per le variabili), almeno 1 dei 2 vertici.

- Siccome il numero dei triangoli è $\ell$ e il numero degli archi tra i vertici in $V_1$ è $m$, l'insieme $V'$ contiene **esattamente** due vertici di ogni triangolo e, per ogni arco tra due vertici di $V_1$, uno dei 2 vertici.

# $3SAT$ si riduce in tempo polinomiale a $VERTEX\text{-}COVER$

- Assegniamo valore vero ai letterali che sono etichette di vertici in $V' \cap V_1$.
- Proviamo che questo assegnamento $\tau$ soddisfa $\phi$. Cioè che questo assegnamento rende vera ogni clausola.
- Infatti, per ogni triangolo esiste un vertice $u$ che non è in $V'$.
- Ma $(u, v)$, con $v \in V_1$ deve essere coperto da $V'$. Quindi $v \in V'$.
- Ma $u$ e $v$ hanno la stessa etichetta (per costruzione di $G$) che corrisponde a un letterale a cui è assegnato valore 1 (per costruzione di $\tau$).
- Dunque, per ogni clausola $c$, c'è un letterale a cui $\tau$ assegna valore 1 e quindi $\phi$ è soddisfacibile. $\square$

# Esempi di linguaggi in *NP*: *SUBSET-SUM*

*SUBSET-SUM*: Dato un insieme finito $S$ di numeri interi e un numero intero $t$, esiste un sottoinsieme $S'$ di $S$ tale che la somma dei suoi numeri sia uguale a $t$?

$SUBSET\text{-}SUM = \{\langle S, t\rangle \mid S = \{x_1, \ldots, x_k\}$ ed esiste $S' \subseteq S$ tale che $\sum_{s \in S'} s = t\}$

Esempio: $\langle\{4, 11, 16, 21, 27\}, 25\rangle \in SUBSET\text{-}SUM$ perché $4 + 21 = 25$.

Nota: È possibile definire *SUBSET-SUM* in cui $S, S'$ sono multinsiemi (cioè insiemi in cui alcuni elementi si ripetono).

Teorema

*SUBSET-SUM* $\in$ *NP*

Dimostrazione.

Un algoritmo $V$ che verifica *SUBSET-SUM* in tempo polinomiale:

$V =$ "Sull'input $\langle\langle S, t\rangle, c\rangle$:

1. Verifica se $c$ è un insieme di numeri la cui somma è $t$, altrimenti rifiuta.
2. Verifica se $S$ contiene tutti i numeri in $c$, accetta in caso affermativo; altrimenti rifiuta."

$\exists c: \langle\langle S, t\rangle, c\rangle \in L(V) \Leftrightarrow \langle S, t\rangle \in$ *SUBSET-SUM* $\square$

Prova alternativa: utilizzare le macchine di Turing non deterministiche.

Teorema

*SUBSET-SUM è NP-completo.*

Dimostrazione

Abbiamo già provato che *SUBSET-SUM* è in *NP*.

Per concludere la prova, basta provare che $3SAT \leq_P SUBSET\text{-}SUM$.

- Sia $\phi$ una formula $3CNF$ con variabili $x_1, \ldots x_\ell$ e clausole $c_1, \ldots, c_k$.
- Associamo a $\phi$ un insieme $S$ di numeri e un numero $t$ tali che $\phi$ è soddisfacibile se e solo se $\langle S, t \rangle \in SUBSET\text{-}SUM$. I numeri in $S$ e il numero $t$ sono espressi nella notazione decimale ordinaria.
- Inoltre $\langle S, t \rangle$ può essere costruita in tempo polinomiale nella lunghezza di $\langle \phi \rangle$.

Figura tratta da M. Sipser, Introduzione alla teoria della computazione.

| | 1 | 2 | 3 | 4 | $\cdots$ | $l$ | $c_1$ | $c_2$ | $\cdots$ | $c_k$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $y_1$ | 1 | 0 | 0 | 0 | $\cdots$ | 0 | 1 | 0 | $\cdots$ | 0 |
| $z_1$ | 1 | 0 | 0 | 0 | $\cdots$ | 0 | 0 | 0 | $\cdots$ | 0 |
| $y_2$ | | 1 | 0 | 0 | $\cdots$ | 0 | 0 | 1 | $\cdots$ | 0 |
| $z_2$ | | 1 | 0 | 0 | $\cdots$ | 0 | 1 | 0 | $\cdots$ | 0 |
| $y_3$ | | | 1 | 0 | $\cdots$ | 0 | 1 | 1 | $\cdots$ | 0 |
| $z_3$ | | | 1 | 0 | $\cdots$ | 0 | 0 | 0 | $\cdots$ | 1 |
| $\vdots$ | | | | | $\ddots$ | $\vdots$ | $\vdots$ | | $\vdots$ | $\vdots$ |
| $y_l$ | | | | | | 1 | 0 | 0 | $\cdots$ | 0 |
| $z_l$ | | | | | | 1 | 0 | 0 | $\cdots$ | 0 |
| $g_1$ | | | | | | | 1 | 0 | $\cdots$ | 0 |
| $h_1$ | | | | | | | 1 | 0 | $\cdots$ | 0 |
| $g_2$ | | | | | | | | 1 | $\cdots$ | 0 |
| $h_2$ | | | | | | | | 1 | $\cdots$ | 0 |
| $\vdots$ | | | | | | | | | $\ddots$ | $\vdots$ |
| $g_k$ | | | | | | | | | | 1 |
| $h_k$ | | | | | | | | | | 1 |
| $t$ | 1 | 1 | 1 | 1 | $\cdots$ | 1 | 3 | 3 | $\cdots$ | 3 |

- Esempio: $(x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$

| Numero | 1 | 2 | 3 | 1 | 2 | 3 |
|---|---|---|---|---|---|---|
| $y_1$ | 1 | 0 | 0 | 1 | 0 | 0 |
| $z_1$ | 1 | 0 | 0 | 0 | 1 | 1 |
| $y_2$ | 0 | 1 | 0 | 1 | 0 | 1 |
| $z_2$ | 0 | 1 | 0 | 0 | 1 | 0 |
| $y_3$ | 0 | 0 | 1 | 1 | 1 | 0 |
| $z_3$ | 0 | 0 | 1 | 0 | 0 | 1 |
| $g_1$ | 0 | 0 | 0 | 1 | 0 | 0 |
| $h_1$ | 0 | 0 | 0 | 1 | 0 | 0 |
| $g_2$ | 0 | 0 | 0 | 0 | 1 | 0 |
| $h_2$ | 0 | 0 | 0 | 0 | 1 | 0 |
| $g_3$ | 0 | 0 | 0 | 0 | 0 | 1 |
| $h_3$ | 0 | 0 | 0 | 0 | 0 | 1 |
| $t$ | 1 | 1 | 1 | 3 | 3 | 3 |

**Esercizio 3**

Data la seguente formula booleana

$$\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

definire l'insieme $S$ e l'intero $t$ tali che $\langle S, t\rangle$ sia l'immagine di $\langle \phi \rangle$ nella riduzione polinomiale di 3-$SAT$ a $SUBSET$-$SUM$.

**Soluzione :**

| Numero | 1 | 2 | 3 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|---|
| $y_1$ | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| $z_1$ | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| $y_2$ | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| $z_2$ | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| $y_3$ | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| $z_3$ | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| $g_1$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| $h_1$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| $g_2$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| $h_2$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| $g_3$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| $h_3$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| $g_4$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $h_4$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $t$ | 1 | 1 | 1 | 3 | 3 | 3 | 3 |

$S = \{y_1, z_1, y_2, z_2, y_3, z_3, g_1, h_1, g_2, h_2, g_3, h_3, g_4, h_4\}$,
$t = 1113333$

## $3SAT \leq_P$ SUBSET-SUM

- Per ogni variabile $x_i$ con $1 \leq i \leq \ell$, definiamo due numeri $y_i$ e $z_i$ di $\ell + k$ cifre decimali:
- $y_i$ ha un 1 in posizione $i$ (da sinistra) e in ogni posizione $\ell + j$ (da sinistra) per cui $x_i$ appare nella clausola $c_j$, con $1 \leq j \leq k$,
- $z_i$ ha un 1 in posizione $i$ (da sinistra) e in ogni posizione $\ell + j$ (da sinistra) per cui $\overline{x_i}$ appare nella clausola $c_j$, con $1 \leq j \leq k$.
- Per ogni clausola $c_j$ con $1 \leq j \leq k$, definiamo due numeri uguali $g_j$ e $h_j$ di $\ell + k$ cifre decimali, tali che $g_j$ e $h_j$ hanno un 1 in posizione $\ell + j$ (da sinistra).
- Le cifre non specificate uguali a 1 sono uguali a zero.

$$S = \{y_i, z_i \mid 1 \leq i \leq \ell\} \cup \{g_j, h_j \mid 1 \leq j \leq k\}$$

$t$ è un numero di $\ell + k$ cifre decimali che ha un 1 in posizione $i$ (da sinistra), per $1 \leq i \leq \ell$, e ha un 3 in posizione $\ell + j$ (da sinistra), per $1 \leq j \leq k$.

- Sia $\phi$ soddisfacibile e sia $\tau$ un assegnamento che soddisfa $\phi$. Consideriamo il sottoinsieme $S'$ di $S$ che contiene $y_i$ se $\tau$ assegna a $x_i$ valore 1, $z_i$ altrimenti.
- Se sommiamo ciò che abbiamo scelto fino ad ora, otteniamo un 1 in ciascuna delle prime $\ell$ cifre perché abbiamo selezionato $y_i$ o $z_i$ per ciascun $i$.
- Inoltre, ciascuna delle ultime $k$ cifre è un numero da 1 a 3 perché ciascuna clausola è soddisfatta e quindi contiene da 1 a 3 letterali veri.
- Quindi scegliamo un numero sufficiente di $g_j, h_j$ da aggiungere a $S'$ per portar ciascuna delle ultime $k$ cifre fino a 3 e ottenere $\sum_{s \in S'} s = t$.

- Viceversa supponiamo che esista un sottoinsieme $S'$ di $S$ tale che $\sum_{s \in S'} s = t$.

  Due osservazioni:

- Tutte le cifre negli elementi di $S$ sono 0 o 1.
- Ciascuna colonna nella tabella che descrive $S$ contiene al più cinque 1.
- Quindi, sommando elementi di un sottoinsieme di $S$ non si verifica mai un "riporto" nella colonna successiva.

- Sia $S'$ un sottoinsieme di $S$ tale che $\sum_{s \in S'} s = t$.
- Per ogni $i$, $1 \leq i \leq \ell$, $S'$ deve contenere $y_i$ o $z_i$ ma non entrambi.
- Sia $\tau$ l'assegnamento definito come segue: per ogni $i$, $1 \leq i \leq \ell$, assegniamo a $x_i$ valore 1 se $S'$ contiene $y_i$, valore 0 se $S'$ contiene $z_i$.
- Questo assegnamento $\tau$ soddisfa $\phi$.
- Infatti, poiché le ultime $k$ cifre di $t$ sono uguali a 3, in ciascuna delle $k$ colonne finali, la somma è sempre 3.
- Per ogni $j$, con $1 \leq j \leq k$, almeno un 1 nella colonna $c_j$ deve venire da qualche $y_i$ o $z_i$ nel sottoinsieme $S'$ perché da $g_j$ ed $h_j$ può venire al più 2.

- Per ogni $j$ nella colonna $c_j$ vi deve essere una cifra uguale a 1 corrispondente a un $y_i$ o $z_i$ in $S'$.
- Se è $y_i$, allora $x_i$ è presente in $c_j$ e gli viene assegnato 1, quindi $c_j$ è soddisfatta.
- Se è $z_i$, allora $\overline{x_i}$ è presente in $c_j$ e a $x_i$ viene assegnato 0, quindi $c_j$ è soddisfatta.
- Pertanto $\phi$ è soddisfatta.
- Infine, la riduzione può essere effettuata in tempo polinomiale. $\square$

# HAMPATH

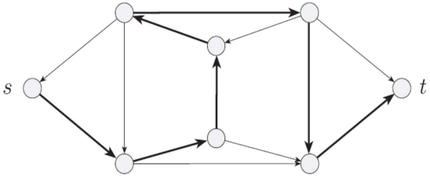



**FIGURA 7.17**
Un cammino Hamiltoniano attraversa ogni nodo esattamente una volta

Figura tratta da M. Sipser, Introduzione alla teoria della computazione.

Un cammino Hamiltoniano in un grafo orientato è un cammino (orientato) che passa per ogni vertice del grafo una e una sola volta.

Consideriamo il problema di stabilire se un grafo orientato contiene un cammino Hamiltoniano che collega due nodi specificati.

Questo si può formulare come un problema di decisione, a cui corrisponde un linguaggio associato, il linguaggio *HAMPATH*.

$$HAMPATH = \{\langle G, s, t\rangle \mid G \text{ è un grafo orientato e ha un cammino Hamiltoniano da } s \text{ a } t\}$$

**Teorema**

*HAMPATH* $\in$ *NP*

**Dimostrazione.**

Un algoritmo $N$ che verifica *HAMPATH* in tempo polinomiale:
$N =$ "Sull'input $\langle\langle G, s, t\rangle, c\rangle$, dove $G = (V, E)$ è un grafo orientato:

1. Verifica se $c = (u_1, \ldots, u_{|V|})$ è una sequenza di $|V|$ vertici di $G$, altrimenti rifiuta.
2. Verifica se i nodi della sequenza sono distinti, $u_1 = s$, $u_{|V|} = t$ e, per ogni $i$ con $2 \leq i \leq n$, se $(u_{i-1}, u_i) \in E$, accetta in caso affermativo; altrimenti rifiuta."

$\exists c : \langle\langle G, s, t\rangle, c\rangle \in L(N)$ se e solo se $\langle G, s, t\rangle \in$ *HAMPATH*. $\square$

Teorema

*HAMPATH è NP-completo.*

Dimostrazione.

Abbiamo già provato che $HAMPATH$ è in $NP$.

Per concludere la prova, basta provare che $3SAT \leq_P HAMPATH$.

## *HAMPATH* è *NP*-completo

Mostriamo che 3*SAT* è riducibile in tempo polinomiale a *HAMPATH*.

La riduzione converte formule booleane in 3CNF in grafi, in cui i cammini Hamiltoniani corrispondono ad assegnamenti soddisfacenti la formula.

I grafi contengono gadget che simulano variabili e clausole.

Il gadget di una variabile è una struttura romboidale che può essere attraversata in due modi, corrispondenti ai due assegnamenti di verità.

Il gadget di una clausola è un nodo.

Assicurare che il cammino attraversa ciascun gadget di clausola corrisponde ad assicurare che ciascuna clausola è soddisfatta nell'assegnamento che soddisfa la formula.

## $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

- Sia $\phi$ una formula $3CNF$ con $k$ clausole:

$$(a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_k \vee b_k \vee c_k)$$

  dove ciascun $a$, $b$, e $c$ è un letterale $x_i$ or $\overline{x_i}$.
- Siano $x_1, \ldots x_\ell$ le $\ell$ variabili di $\phi$.
- Definiamo un grafo orientato $G$, con due vertici $s$ e $t$, tale che $\phi$ è soddisfacibile se e solo se $G$ ha un cammino Hamiltoniano da $s$ a $t$ .
- Inoltre $\langle G, s, t\rangle$ può essere costruita in tempo polinomiale nella lunghezza di $\langle \phi \rangle$.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

Rappresentiamo ciascuna variabile $x_i$ con una struttura di forma romboidale che contiene una riga orizzontale di nodi, come mostrato nella figura seguente.

Specificheremo dopo il numero di nodi che compaiono nella riga orizzontale.

Il numero dei vertici sulla linea orizzontale è legato al numero di clausole nella formula.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

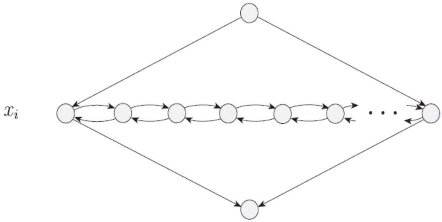



**FIGURA 7.47**
Rappresentazione della variabile $x_i$ con una struttura romboidale

Figura tratta da M. Sipser, Introduzione alla teoria della computazione.

## $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

Rappresentiamo ciascuna clausola di $\phi$ con un singolo nodo, come segue.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$


**FIGURA 7.48**
Rappresentazioni della clausola $c_j$ con un nodo

Figura tratta da M. Sipser, Introduzione alla teoria della computazione.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

La figura seguente riporta la struttura globale di $G$.

Mostra quasi tutti gli elementi di $G$ e le loro relazioni.

Mancano gli archi che rappresentano la relazione delle variabili con le clausole che le contengono.

# 3*SAT* si riduce in tempo polinomiale a *HAMPATH*

## La struttura di G

- (Mancano gli archi che rappresentano la relazione tra clausole e variabili)

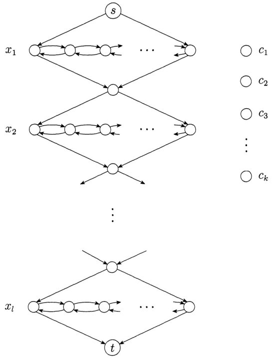



Figura tratta da M. Sipser, Introduzione alla teoria della computazione.

- Esempio

$$\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

Il grafo $G$ avrà tre rombi, corrispondenti alle variabili $x_1, x_2, x_3$ e quattro nodi $c_1, c_2, c_3, c_4$ associati alle quattro clausole.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

Ciascuna struttura romboidale contiene una riga orizzontale di nodi collegati tramite archi orientati in entrambe le direzioni.

La riga orizzontale contiene $3k + 1$ nodi in aggiunta ai due nodi alle estremità del rombo.

Questi nodi sono raggruppati in coppie adiacenti, una per ciascuna clausola, con nodi separatori aggiuntivi tra le coppie, come mostrato nella figura seguente.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

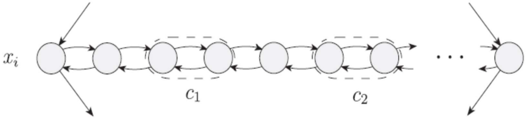



**FIGURA 7.50**
I nodi orizzontali in una struttura romboidale

- Sulla linea orizzontale del grafo associato a una variabile ci sono 2 vertici per ogni clausola e ogni coppia è separata dalla successiva da un vertice "separatore" (3k + 1 + 2 vertici)

Figura tratta da M. Sipser, Introduzione alla teoria della computazione.

## $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

Se la variabile $x_i$ è presente nella clausola $c_j$, aggiungiamo i due archi seguenti dalla coppia $j$-esima nell'$i$-esimo rombo al $j$-esimo nodo della clausola.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

- Se $x_i$ è un letterale di $c_j$:

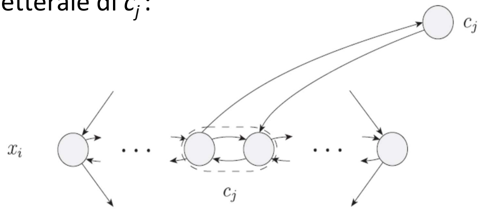



**FIGURA 7.51**
Gli archi aggiuntivi quando la clausola $c_j$ contiene $x_i$

Figura tratta da M. Sipser, Introduzione alla teoria della computazione.

- Esempio

$$\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

- Il grafo $G$ avrà tre rombi, corrispondenti alle variabili $x_1, x_2, x_3$ e quattro nodi $c_1, c_2, c_3, c_4$ associati alle quattro clausole.
- Quali sono i rombi collegati al nodo $c_1$ come nella precedente figura? E quelli collegati al nodo $c_2$ come in precedenza? E quelli collegati al nodo $c_3$ come in precedenza?
- Il rombo corrispondente a $x_1$ sarà collegato ai nodi $c_2$ e $c_3$ come nella precedente figura.
- Il rombo corrispondente a $x_2$ sarà collegato ai nodi $c_1$ e $c_2$ come nella precedente figura.
- Il rombo corrispondente a $x_3$ sarà collegato ai nodi $c_2$ e $c_3$ come nella precedente figura.

Se $\overline{x_i}$ è presente nella clausola $c_j$, aggiungiamo i due archi seguenti dalla coppia $j$-esima nell'$i$-esimo rombo al $j$-esimo nodo di clausola.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

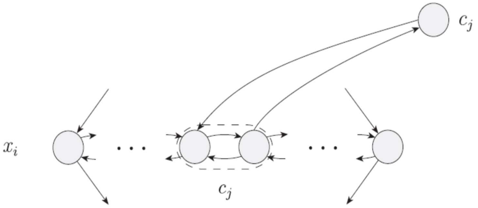



**FIGURA 7.52**
Gli archi aggiuntivi quando la clausola $c_j$ contiene $\overline{x_i}$

Figura tratta da M. Sipser, Introduzione alla teoria della computazione.

- Esempio

$$\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

- Il grafo $G$ avrà tre rombi, corrispondenti alle variabili $x_1, x_2, x_3$ e quattro nodi $c_1, c_2, c_3, c_4$ associati alle quattro clausole.
- Quali sono i rombi collegati al nodo $c_1$ come nella precedente figura? E quelli collegati al nodo $c_2$ come in precedenza? E quelli collegati al nodo $c_3$ come in precedenza?
- Il rombo corrispondente a $x_1$ sarà collegato ai nodi $c_1$ e $c_4$ come nella precedente figura.
- Il rombo corrispondente a $x_2$ sarà collegato ai nodi $c_3$ e $c_4$ come nella precedente figura.
- Il rombo corrispondente a $x_3$ sarà collegato ai nodi $c_1$ e $c_4$ come nella precedente figura.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

Dopo aver aggiunto tutti gli archi corrispondenti a ciascuna occorrenza di $x_i$ o di $\overline{x_i}$ in ciascuna clausola, la costruzione di $G$ è completa.

Per far vedere che questa costruzione funziona, dimostriamo che se $\phi$ è soddisfacibile, allora esiste un cammino Hamiltoniano da $s$ a $t$; e viceversa se esiste un tale cammino in $G$, allora $\phi$ è soddisfacibile.

# 3*SAT* si riduce in tempo polinomiale a *HAMPATH*

Supponiamo che $\phi$ sia soddisfacibile.

Per mostrare che esiste un cammino Hamiltoniano da $s$ a $t$, in un primo momento ignoriamo i nodi delle clausole. Il cammino inizia da $s$, attraversa ciascun rombo in successione, e termina in $t$.

Per raggiungere i nodi orizzontali in un rombo, il cammino può procedere a zig-zag da sinistra a destra oppure a zag-zig da destra a sinistra.

L'assegnamento che soddisfa $\phi$ determina in quale senso la riga è attraversata.

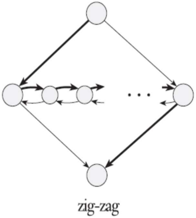
Se a $x_i$ viene assegnato 1, il cammino procede a zig-zag attraverso il rombo corrispondente.

Se a $x_i$ viene assegnato 0, il cammino procede a zag-zig.

Se a $x_i$ viene assegnato 1, il cammino attraversa la riga $i$ da sinistra a destra.
Se a $x_i$ viene assegnato 0, il cammino attraversa la riga $i$ da destra a sinistra.

Le due possibilità sono mostrate nella figura seguente.

# 3*SAT* si riduce in tempo polinomiale a *HAMPATH*



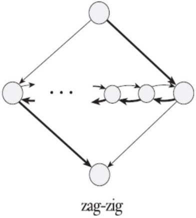




**FIGURA 7.53**
Zig-zag (da sinistra a destra) o zag-zig (da destra a sinistra), come stabilito dall'assegnamento soddisfacente

Figura tratta da M. Sipser, Introduzione alla teoria della computazione.

- Esempio

  $$\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

- Il grafo $G$ avrà tre rombi, corrispondenti alle variabili $x_1, x_2, x_3$ e quattro nodi $c_1, c_2, c_3, c_4$ associati alle quattro clausole.
- Un assegnamento che soddisfa $\phi$ è quello che assegna valore 1 a $x_1$ e $x_2$ e valore 0 a $x_3$. (Non è l'unico!)
- Siccome $x_1 = x_2 = 1$, $x_3 = 0$, il cammino attraversa ogni riga dei rombi di $x_1$ e $x_2$ da sinistra a destra e la riga del rombo di $x_3$ da destra verso sinistra.

Questo cammino copre tutti i nodi in $G$, eccetto i nodi delle clausole.

Per includerli aggiungiamo una deviazione per ognuno di tali nodi in questo modo.

Per ciascuna clausola $c_j$, scegliamo uno dei letterali in $c_j$ a cui è assegnato 1. Supponiamo che sia $x_i$ oppure $\overline{x_i}$.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

Se selezioniamo $x_i$ nella clausola $c_j$, attraversiamo la riga i corrispondente nella direzione “corretta” per raggiungere il vertice $c_j$ da uno dei vertici della $j$-esima coppia nella riga e tornare nell’altro vertice.

Cioè possiamo deviare alla $j$-esima coppia nell’$i$-esimo rombo. Questo è possibile perché $x_i$ deve essere 1, quindi il cammino procede a zig-zag da sinistra a destra attraverso il rombo corrispondente.

Quindi, gli archi verso il nodo $c_j$ sono nell’ordine corretto per permettere una deviazione ed un ritorno.

- Esempio

  $\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$

- Il grafo $G$ avrà tre rombi, corrispondenti alle variabili $x_1, x_2, x_3$ e quattro nodi $c_1, c_2, c_3, c_4$ associati alle quattro clausole.
- Un assegnamento che soddisfa $\phi$ è quello che assegna valore 1 a $x_1$ e $x_2$ e valore 0 a $x_3$.
- Siccome $x_1 = x_2 = 1$, $x_3 = 0$, il cammino attraversa ogni riga dei rombi di $x_1$ e $x_2$ da sinistra a destra e la riga del rombo di $x_3$ da destra verso sinistra.
- Selezioniamo $x_2$ in $c_1$, $x_1$ in $c_2$ e $c_3$. Quindi raggiungiamo il nodo $c_1$ dal rombo di $x_2$ e i nodi $c_2$ e $c_3$ dal rombo di $x_1$.

Allo stesso modo, se avessimo selezionato $\overline{x_i}$ nella clausola $c_j$, avremmo potuto deviare alla $j$-esima coppia nell'$i$-esimo rombo perché $x_i$ deve essere 0, quindi il cammino procede a zag-zig da destra a sinistra attraverso il rombo corrispondente.

Pertanto, gli archi verso il nodo $c_j$ sono nuovamente nell'ordine corretto per permettere una deviazione ed un ritorno.

(Si noti che ciascun letterale vero in una clausola fornisce una *opzione* di deviazione per raggiungere il nodo clausola. Come risultato, se diversi letterali in una clausola sono veri, viene presa soltanto una deviazione.)

Così abbiamo costruito il cammino Hamiltoniano.

- Esempio

  $\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$

- Il grafo $G$ avrà tre rombi, corrispondenti alle variabili $x_1, x_2, x_3$ e quattro nodi $c_1, c_2, c_3, c_4$ associati alle quattro clausole.
- Un assegnamento che soddisfa $\phi$ è quello che assegna valore 1 a $x_1$ e $x_2$ e valore 0 a $x_3$.
- Siccome $x_1 = x_2 = 1$, $x_3 = 0$, il cammino attraversa ogni riga dei rombi di $x_1$ e $x_2$ da sinistra a destra e la riga del rombo di $x_3$ da destra verso sinistra.
- Selezioniamo $\overline{x_3}$ in $c_4$. Quindi raggiungiamo il nodo $c_4$ dal rombo di $x_3$.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

Per la direzione inversa, mostriamo che se un grafo $G$ così costruito ha un cammino Hamiltoniano da $s$ a $t$ allora esiste un assegnamento che soddisfa $\phi$.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

Se il cammino Hamiltoniano è *normale*, cioè passa attraverso i rombi in ordine, da quello più in alto a quello più in basso, con deviazioni verso i nodi delle clausole come descritte prima, possiamo facilmente definire un assegnamento che soddisfa $\phi$:

- alla variabile $i$-esima $x_i$ assegniamo valore 1 se la riga orizzontale è attraversata da sinistra a destra, cioè se il cammino procede a zig-zag attraverso il rombo;
- alla variabile $i$-esima $x_i$ assegniamo valore 0 se la riga orizzontale è attraversata da destra a sinistra, cioè se il cammino procede a zag-zig attraverso il rombo.

L'assegnamento soddisfa la formula poiché ciascun nodo clausola è connesso a una coppia di una riga e quindi contiene un letterale vero.

Osservando come la deviazione verso il nodo clausola avviene, possiamo stabilire quale dei letterali nella clausola corrispondente è vero.

Tutto ciò che resta da mostrare è che un cammino Hamiltoniano deve essere normale.

La normalità può venir meno solo se il cammino entra in una clausola da un rombo e ritorna in un altro come nella figura seguente.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

Figura tratta da M. Sipser, Introduzione alla teoria della computazione.

a2 o a3 deve essere un vertice separatore.
Se a2 vertice separatore , l’unico modo per raggiungerlo nel cammino Hamiltoniano è da a3 (non potendo da a1).
Analogamente, se a3 vertice separatore, l’unico modo per raggiungere a2 è da a3 (non potendo da a1 e c).
Poi però il cammino non ha modo di uscirne e a2 non può essere l’ultimo vertice del cammino (che deve essere t).

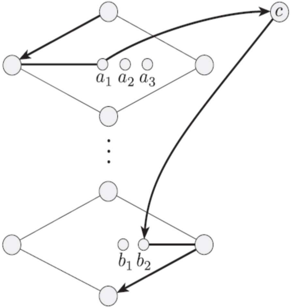



**FIGURA 7.54**
Questa situazione non può verificarsi

Il cammino va dal nodo $a_1$ a $c$; ma invece di ritornare nello stesso rombo, ritorna in $b_2$ in un rombo differente.

Se questo accade, $a_2$ oppure $a_3$ è un nodo separatore.

# $3SAT$ si riduce in tempo polinomiale a $HAMPATH$

- Se $a_2$ fosse un nodo separatore, gli unici archi entranti in $a_2$ sarebbero quelli da $a_1$ e $a_3$.
- Se $a_3$ fosse un nodo separatore, $a_1$ ed $a_2$ sarebbero nella stessa coppia associata alla clausola $c$, quindi gli unici archi entranti in $a_2$ sarebbero quelli da $a_1$, $a_3$ e $c$.

In ogni caso il cammino non potrebbe contenere il nodo $a_2$.

- Il cammino non può entrare in $a_2$ da $c$ o $a_1$ perché il cammino va altrove da questi nodi.
- Il cammino non può entrare in $a_2$ da $a_3$ perché $a_3$ è l'unico nodo disponibile a cui $a_2$ punta, quindi il cammino deve uscire da $a_2$ passando per $a_3$.

# 3*SAT* si riduce in tempo polinomiale a *HAMPATH*

Pertanto, il cammino Hamiltoniano deve essere normale.

Questa riduzione è ovviamente calcolabile in tempo polinomiale e la dimostrazione è completa.

È possibile definire una "versione non orientata" del problema del cammino Hamiltoniano.

- Un cammino Hamiltoniano in un grafo non orientato è un cammino che passa per ogni vertice del grafo una e una sola volta.

$$\mathit{UHAMPATH} = \{\langle G, s, t\rangle \mid G \text{ è un grafo non orientato e ha un cammino Hamiltoniano da } s \text{ a } t\}$$

Per mostrare che *UHAMPATH* è *NP*-completo, definiamo una riduzione di tempo polinomiale da *HAMPATH* a *UHAMPATH*.

**Teorema**
$UHAMPATH \in NP$

**Dimostrazione.**
Un algoritmo $N$ che verifica $UHAMPATH$ in tempo polinomiale:
$N =$ "Sull'input $\langle\langle G, s, t\rangle, c\rangle$, dove $G = (V, E)$ è un grafo non orientato:

1. Verifica se $c = (u_1, \ldots, u_{|V|})$ è una sequenza di $|V|$ vertici di $G$, altrimenti rifiuta.
2. Verifica se i nodi della sequenza sono distinti, $u_1 = s$, $u_{|V|} = t$ e, per ogni $i$ con $2 \leq i \leq n$, se $(u_{i-1}, u_i) \in E$, accetta in caso affermativo; altrimenti rifiuta."

$\exists c : \langle\langle G, s, t\rangle, c\rangle \in L(N)$ se e solo se $\langle G, s, t\rangle \in UHAMPATH$. $\square$

**Teorema**

*UHAMPATH è NP-completo.*

**Dimostrazione**

Abbiamo provato che *UHAMPATH* è in *NP*.

Per concludere la prova, dimostriamo che $HAMPATH \leq_P UHAMPATH$.

# *HAMPATH* si riduce in tempo polinomiale a *UHAMPATH*

- La riduzione di tempo polinomiale associa a un grafo orientato $G = (V, E)$ con vertici $s$ e $t$ un grafo non orientato $G' = (V', E')$ con vertici $s'$ e $t'$.
- Il grafo $G$ ha un cammino Hamiltoniano da $s$ a $t$ se e solo se $G'$ ha un cammino Hamiltoniano da $s'$ a $t'$.
- Inoltre $G'$ può essere costruito a partire da $G$ in tempo polinomiale.

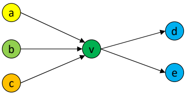
# *HAMPATH* si riduce in tempo polinomiale a *UHAMPATH*

## cammino Hamiltoniano in un grafo non orientato

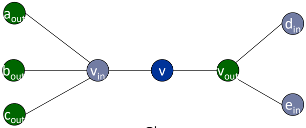
Dato grafo non orientato G' = (V', E') e due vertici s', t', esiste un cammino Hamiltoniano in G' da s' a t'?

Fatto. HAMPATH $\leq_P$ UHAMPATH.

Dim. Dato un grafo orientato G = (V, E) con n vertici, costruiamo un grafo non orientato G' con 3(n -2) + 2 vertici.




G




G'

(autore slide: Kevin Wayne)

# *HAMPATH* si riduce in tempo polinomiale a *UHAMPATH*

Costruzione di $G'$:

- Ogni vertice $u$ di $G$, diverso da $s$ e $t$ è rimpiazzato da tre vertici $u^{in}, u^{mid}$ e $u^{out}$ in $G'$.
- I vertici $s$ e $t$ sono sostituiti con i vertici $s^{out}$ e $t^{in}$ in $G'$.
- Per ogni $u \in V \setminus \{s, t\}$, $(u^{in}, u^{mid})$ e $(u^{mid}, u^{out})$ sono in $E'$.
- Se $(u, v) \in E$ allora $(u^{out}, v^{in}) \in E'$.

# *HAMPATH* si riduce in tempo polinomiale a *UHAMPATH*

- Dimostriamo che $G$ ha un cammino Hamiltoniano da $s$ a $t$ se e solo se $G'$ ha un cammino Hamiltoniano da $s^{out}$ a $t^{in}$.
- Se $G$ ha un cammino Hamiltoniano $P$ da $s$ a $t$:

$$P = s, u_1, u_2, \dots, u_k, t$$

allora $P'$:

$$P' = s^{out}, u_1^{in}, u_1^{mid}, u_1^{out}, u_2^{in}, u_2^{mid}, u_2^{out}, \dots, u_k^{in}, u_k^{mid}, u_k^{out}, t^{in}$$

è un cammino Hamiltoniano in $G'$ da $s^{out}$ a $t^{in}$.

# *HAMPATH* si riduce in tempo polinomiale a *UHAMPATH*

- Viceversa se $G'$ ha un cammino Hamiltoniano $P'$ da $s^{out}$ a $t^{in}$, è facile vedere che $P'$ deve essere della forma

$$P' = s^{out}, u_1^{in}, u_1^{mid}, u_1^{out}, u_2^{in}, u_2^{mid}, u_2^{out}, \dots, u_k^{in}, u_k^{mid}, u_k^{out}, t^{in}$$

- La prova è per induzione su $k$. Infatti $P'$ ha come primo vertice $s^{out}$ il quale è connesso solo a vertici della forma $u_i^{in}$. Quindi il secondo vertice è $u_i^{in}$ per qualche $i$. I vertici successivi devono essere $u_i^{mid}, u_i^{out}$ perché $u_i^{mid}$ è connesso solo a $u_i^{in}$ e $u_i^{out}$.
- Ma se $P'$ ha la forma suddetta allora

$$P = s, u_1, u_2, \dots, u_k, t$$

è un cammino Hamiltoniano da $s$ a $t$.

□