Lezione 24 – costanti, problemi NP-completi e... oltre NP

Lezione del 04/06/2024

Dimostrazioni di NP-completezza

- Concludiamo il corso con una lunga esercitazione
- che ci permetterà di chiarire una serie di questioni che abbiamo toccato nel corso delle lezioni:
  - il ruolo delle costanti nella complessità dei problemi
  - NP e … non NP

Il ruolo delle costanti

- La scorsa lezione abbiamo incontrato un problema, 3-colorability, che è una restrizione del più generale colorability
- e la restrizione è ottenuta fissando ad un valore costante una parte dell'istanza
  - il numero di colori disponibili
- Abbiamo visto che
  - per taluni valori della costante il problema cade nella classe P (2-COL)
  - mentre per la maggior parte dei valori costanti il problema rimane NP-completo, come nel caso generale (k-COL, per ogni $k \geq 3$)
- La stessa cosa accade con il problema SAT, quando fissiamo ad un valore costante la dimensione delle clausole
  - 2SAT $\in$ P
  - kSAT è NP-completo, per ogni $k \geq 3$ (lo abbiamo dimostrato per $k = 3$, potete dimostrarlo per esercizio per $k \geq 4$)
- E, adesso, vediamo insieme altri esempi nei quali fissiamo ad una costante una parte delle istanze dei problemi

Il ruolo delle costanti

- Ad esempio, cosa accade al problema 3SAT quando stabiliamo che il numero di variabili booleane è una costante?
- Sia $k \in \mathbb{N}$ una costante, e sia $X = \{ x_1, x_2, \dots , x_k \}$ – X è un insieme costante!
  - Allora, il numero di assegnazioni di verità alle variabili in X, che è 2^k, è costante
- Chiamiamo 3SAT[k] la restrizione di 3SAT in cui l'insieme delle istanze contiene tutte le espressioni booleane f in 3CNF costruite su un insieme di variabili di dimensione costante k
- Verificare, una alla volta, tutte le assegnazioni di verità alla ricerca di una che soddisfi f ha costo in $O(2^k |f|)$
- E, ricordiamo (lezione 16), il numero di clausole di 3 letterali su un insieme di k variabili è $\leq (2k)^3$ – ossia, costante!
  - ossia, $|f|$ è costante
- Questo significa che decidiamo una istanza di 3SAT[k] in tempo costante
- ossia, 3SAT[k] $\in$ P

Il ruolo delle costanti

- Nell’esempio precedente, in effetti, abbiamo considerato un problema le cui istanze hanno dimensione costante
  - infatti, se h è una costante e |X|=h allora il numero di clausole in f è al più $\binom{2h}{3} \leq (2h)^3$
- e decidere istanze di dimensione costante non può che richiedere tempo costante!

- In altri casi, fissare la dimensione di *una parte* dell’istanza ad un valore costante non fa sì che l’intera istanza abbia dimensione costante
- tuttavia permette di ridurre sostanzialmente la dimensione dello spazio di ricerca
- E che cos’è lo spazio di ricerca?
  - Informalmente, è l’insieme all’interno del quale dobbiamo ''cercare la soluzione''

- Per essere più formali dobbiamo fare un passo indietro...

Il ruolo delle costanti

- Ricordiamo che

  se un linguaggio L è in NP allora esistono una macchina di Turing T_V e una costante c tali che

  una parola x appartiene al linguaggio se e soltanto se esiste una parola binaria y_x tale che $| y_x | \leq |x|^c$ e $T_V(x, y_x)$ accetta e **dtime**$(T_V, x, y_x) \in O(|x|^c)$

- e ricordiamo anche che per dimostrarlo abbiamo costruito una macchina NT che opera come segue: con input x
  - FASE 1: genera non deterministicamente una parola binaria y di lunghezza $\leq |x|^c$
    - **in tempo non deterministico O($|x|^c$)**
  - FASE 2: invoca $T_V(x, y)$ e termina nello stesso stato
    - **in tempo (deterministico) O($|x|^c$)**
- così che, per ogni parola x del linguaggio, **ntime**(NT, x) $\in O(|x|^c)$

Il ruolo delle costanti

- Naturalmente, possiamo trasformare NT in una m una macchina deterministica T che opera come segue: con input x
  - FASE 1: genera deterministicamente l'insieme $\mathcal{U}$ di tutte le parole binarie y di lunghezza $\leq |x|^c$
    - **in tempo (deterministico) $O(2^{|x|^c})$**
  - FASE 2: per ogni parola y ∈ $\mathcal{U}$, invoca $T_V(x, y)$ e termina in q_A se $T_V(x, y)$ termina in q_A

    se nessuna parola y ∈ $\mathcal{U}$ ha indotto T_V ad accettare, termina in q_R
    - **in tempo (deterministico) $O(|x|^c\ 2^{|x|^c})$**
- T è un algoritmo di ricerca esaustiva per il linguaggio L
- e, in generale, un algoritmo di ricerca esaustiva impiega tempo esponenziale
  - come descritto sopra, **dtime**(T,x) ∈ $O(|x|^c\ 2^{|x|^c})$
- Se, però, il numero di possibili parole y che *potrebbero* indurre T_V ad accettare fosse polinomiale in x,
  - se, cioè, fosse $|\mathcal{U}| \leq |x|^k$, per qualche ***costante*** k
- allora l'algoritmo di ricerca esaustiva impiegherebbe tempo polinomiale!

Il ruolo delle costanti

- Ad esempio, cosa accade al problema VC quando ci proponiamo di cercare un vertex cover di dimensione costante in un grafo?
- Sia $h \in \mathbb{N}$ una costante
- Chiamiamo h-VC la restrizione di VC nella quale una istanza è un grafo ⟨G=(V,E)⟩ e viene richiesto di decidere se G ha un vertex cover di al più h nodi
- In che modo potrebbe aiutarci (a progettare un algoritmo polinomiale) sapere che siamo alla ricerca di un vertex cover di dimensione costante?
  - Ricordiamo che Vertex Cover è un problema in NP
  - che un certificato per una istanza ⟨G=(V,E),k⟩ di Vertex Cover è un sottoinsieme di k nodi
  - e che verificare se un certificato è effettivamente un vertex cover per G richiede tempo polinomiale in |⟨G=(V,E),k⟩|
- Perciò, possiamo sfruttare il fatto che un certificato per una istanza ⟨G=(V,E)⟩ di h-VC ha dimensione costante per generare tutti i possibili certificati in tempo polinomiale in |⟨G=(V,E)⟩
- e ottenere un algoritmo di ricerca esaustiva che decide h-VC in tempo polinomiale

Il ruolo delle costanti

- Un algoritmo di ricerca esaustiva che decide h-VC in tempo polinomiale:
- **Input**: un grafo non orientato $G = (V,E)$, con $V = \{v_1, v_2, \ldots ,v_n\}$ e $E = \{e_1, e_2, \ldots ,e_m\}$
- $\mathcal{U} \leftarrow \{ V' \subseteq V: |V'| = h \}$; **perché è sufficiente considerare solo i sottoinsiemi di cardinalità *esattamente* h?**

trovato $\leftarrow$ falso;
while ($\mathcal{U} \neq \phi \wedge$ trovato = falso) **do begin**
    estrai un elemento V' da $\mathcal{U}$;
    trovato $\leftarrow$ vero;
    for ($(u,v) \in E$) **do**
        if ($u \notin V' \wedge v \notin V'$) **then** trovato $\leftarrow$ falso;
end
if (trovato =vero) **then** $q \leftarrow q_A$;
else $q \leftarrow q_R$;
Output: q.

Il ruolo delle costanti

- Un algoritmo di ricerca esaustiva che decide h-VC in tempo polinomiale: infatti,
- per costruire $\mathcal{U}$
  - dobbiamo elencare tutti i sottoinsiemi di V contenenti h nodi
  - e il numero di tali sottoinsiemi è $\binom{|V|}{h} \leq |V|^h$
  - ossia, $|\mathcal{U}| \leq |V|^h$
  - e si può mostrare che $\mathcal{U}$ si può costruire in tempo $O(|V|\,|\mathcal{U}|) \subseteq O(|V|^{h+1})$
  - ossia, in tempo polinomiale in $|\langle G=(V,E)\rangle|$
- Pertanto, l'algoritmo decide h-VC in tempo in $O(|V|^{h+2}|E|)$
- e, quindi, h-VC $\in$ P

Il ruolo delle costanti

- Lo stessa ragionamento può essere ripetuto per molti problemi: sia h ∈ ℕ una costante
  - **h-IS** – nel quale è richiesto di verificare se un grafo contiene un insieme indipendente di almeno h nodi
  - **h-CL** - nel quale è richiesto di verificare se un grafo contiene una clique di almeno h nodi
  - **h-LP** - nel quale è richiesto di verificare se un grafo contiene un percorso fra una data coppia di nodi di lunghezza almeno h
- In tutti questi problemi abbiamo considerato il caso in cui viene fissata ad un valore costante la dimensione dell'oggetto che si sta cercando
- In particolare, in tutti questi problemi l'oggetto che si cerca è un sottoinsieme di un insieme che è parte dell'istanza
  - e la cardinalità del sottoinsieme viene fissata a un valore costante
- Pensandoci bene, questo accade anche nei problemi di soddisfacibilità
- infatti, possiamo definire le varie versioni di SAT nel modo seguente: data una coppia ⟨X,f⟩, decidere se esiste un sottoinsieme X' di X tale che quando si assegna il valore **vero** alle variabili in X' e il valore **falso** alle variabili in X-X' l'espressione f assume il valore **vero**

Il ruolo delle costanti

- Così, sia $h \in \mathbb{N}$ una costante
- Chiamiamo h-min3SAT la versione di 3SAT nella quale una istanza è una coppia $\langle X,f \rangle$, in cui f è un'espressione booleana in 3CNF sulle variabili in X, e viene richiesto di decidere se esiste un'assegnazione di verità per X che soddisfi f e assegni il valore vero ad al più h variabili
- In questo caso, $\mathcal{U}$ contiene tutti i sottoinsiemi di X contenenti al più h variabili:

$$\mathcal{U} = \{ X' \subseteq X : |X'| \leq h \}$$

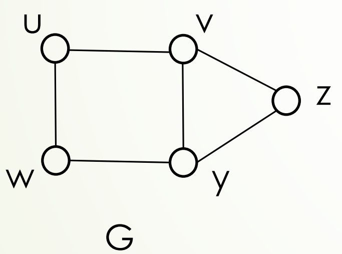
- e quindi $|\mathcal{U}| = \binom{|V|}{2} + \binom{|V|}{2} + \ldots + \binom{|V|}{h} \leq h\, |V|^h$
- pertanto, poiché h è costante, anche l'algoritmo di ricerca esaustiva che decide h-min3SAT ha complessità polinomiale
- e questo permette di concludere che h-min3SAT $\in$ P

Il ruolo delle costanti

- Ma cosa succede se, invece, nel problema min3SAT h ∈ ℕ non è una costante?
- anzi, invece di min3SAT consideriamo il problema min2SAT
  - e ricordiamo che 2SAT ∈ P
- Formalmente, min2SAT è il problema le cui istanze sono una triple ⟨X,f,h⟩, in cui f è un'espressione booleana in 2CNF sulle variabili in X e h ∈ ℕ, e viene richiesto di decidere se esiste un'assegnazione di verità per X che soddisfi f e assegni il valore vero ad al più h variabili
- Ovviamente, anche in questo caso, $|\mathcal{U}| = \binom{|V|}{2} + \binom{|V|}{2} + \ldots + \binom{|V|}{h} \leq h\ |V|^h$
- ma ora, poiché h non è costante (ma è parte dell'istanza), l'algoritmo di ricerca esaustiva che decide min3SAT ha complessità non polinomiale
- e questo non permette di concludere che min2SAT ∈ P
- In effetti, ora dimostreremo che min2SAT è NP-completo
  - e l'appartenenza a NP è lasciata per esercizio

Il ruolo delle costanti

- Dimostriamo ora che VC $\preccurlyeq$ min2SAT
- Sia $\langle G=(V,E),k\rangle$ un'istanza di VC
- l'istanza $\langle X,f,h\rangle$ corrispondente di min2SAT è così definita:
  - a ogni nodo $u \in V$ associamo una variabile booleana x_u - ossia, $X = \{ x_u : u \in V\}$
  - a ogni arco $(u,v) \in E$ associamo la clausola $c_{uv} = (x_u \vee x_v)$ – ossia $f = \{ c_{uv} : (u,v) \in E \}$
  - poniamo $h = k$



$$f = (x_u \vee x_v) \wedge (x_v \vee x_z) \wedge (x_z \vee x_y) \wedge (x_y \vee x_w) \wedge (x_w \vee x_u) \wedge (x_v \vee x_y)$$

Il ruolo delle costanti

- Dimostriamo ora che VC ≼ min2SAT
- Sia ⟨G=(V,E),k⟩ un'istanza di VC
- l'istanza ⟨X,f,h⟩ corrispondente di min2SAT è così definita:
  - a ogni nodo u ∈ V associamo una variabile booleana x_u - ossia, X = { x_u: u ∈ V}
  - a ogni arco (u,v) ∈ E associamo la clausola c_{uv} =(x_u ∨ x_v) – ossia f = { c_{uv} : (u,v) ∈ E }
  - poniamo h = k

- se esiste in G un Vertex Cover V' tale che |V'| ≤ k, allora
  - per ogni x_u ∈ X poniamo: a(x_u)= **vero** se u ∈ V', a(x_u)= **falso** se u ∉ V'
  - poiché |V'| ≤ k allora |X| ≤ h,
  - poiché per ogni (u,v) ∈ E si ha che u ∈ V' o v ∈ V', allora per ogni clausola c_{uv} in f si ha che a(x_u)= **vero** oppure a(x_v)= **vero** : ossia a soddisfa f

Il ruolo delle costanti

- Dimostriamo ora che VC $\preccurlyeq$ min2SAT
- Sia ⟨G=(V,E),k⟩ un'istanza di VC
- l'istanza ⟨X,f,h⟩ corrispondente di min2SAT è così definita:
  - a ogni nodo u ∈ V associamo una variabile booleana x_u - ossia, X = { x_u: u ∈ V}
  - a ogni arco (u,v) ∈ E associamo la clausola c_{uv} =(x_u ∨ x_v) – ossia f = { c_{uv} : (u,v) ∈ E }
  - poniamo h = k
- se esiste una assegnazione di verità per X che soddisfa f e tale che |{ x_u ∈ X : a(x_u)= **vero**}| ≤ k, allora
  - per ogni u ∈ V poniamo: V' = {u ∈ V : a(x_u)= **vero** }
  - poiché |X|≤ h allora |V'|≤ k,
  - poiché a soddisfa f allora per ogni clausola c_{uv} in f si ha che a(x_u)= **vero** oppure a(x_v)= **vero** : quindi, per ogni (u,v) ∈ E si ha che u ∈ V' o v ∈ V' – ossia V' è un Vertex Cover per G
- infine, poiché calcolare ⟨X,f,h⟩ richiede tempo polinomiale in |⟨G,k⟩ |, questo completa la riduzione e la dimostrazione che min2SAT ∈ NPC

Ed ora, esercizi

- Terminiamo la lezione con una serie di esercizi simili fra loro e che ci permetteranno anche di sottolineare qualche ultima questioncina
- Consideriamo una serie di problemi che combinano, in tutti i modi possibili i problemi VC e 3-COL:
  - VC ∨ 3-COL:
  - VC ∧ 3-COL
  - VC ∧ ¬ 3-COL
  - VC ∨ ¬ 3-COL
  - ¬ VC ∧ 3-COL
  - ¬ VC ∨ 3-COL
  - ¬ VC ∨ ¬ 3-COL
  - ¬ VC ∧ ¬ 3-COL

VC v 3-COL

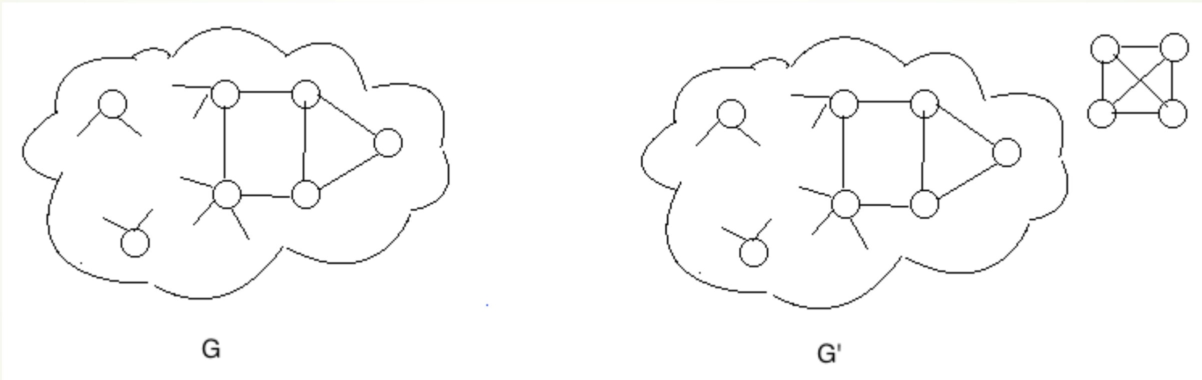
- Dati un grafo G = (V,E) ed un intero k, decidere se G ha un vertex cover di al più k nodi oppure è 3-colorabile

  - $\mathfrak{I}_{\mathbf{VCor3COL}} = \{ \langle G=(V,E), k \rangle : G$ è un grafo non orientato $\wedge\, k \in \mathbb{N} \}$
  - $\mathbf{S}_{\mathbf{VCor3COL}}(G, k) = \{ (V', \mathfrak{c}) : V' \subseteq V \wedge \mathfrak{c}: V \rightarrow \{1,2,3\} \}$
  - $\pi_{\mathbf{VCor3COL}}(G, k, \mathbf{S}_{\mathbf{VCor3COL}}(G, k)) = \exists\, (V', \mathfrak{c}) \in \mathbf{S}_{\mathbf{VCor3COL}}(G, k)$:
    $[\ |V'| \leq k \ \wedge \forall\, (u,v) \in E\ [\ u \in V' \ \vee\ v \in V'\ ]\ ] \ \vee\ [\forall\, (u,v) \in E\ [\ \mathfrak{c}(u) \neq \mathfrak{c}(v)\ ]\ ]$

- Dimostriamo che VC v 3-COL è NP-completo
- 1) VC v 3-COL ∈ NP:
  - infatti, un certificato è una coppia (V', $\mathfrak{c}$) e verificare se soddisfa il predicato richiede tempo polinomiale in | ⟨ G=(V,E), k ⟩ |
- 2) per dimostrare la completezza di VC v 3-COL rispetto a NP, riduciamo VC a VC v 3-COL

VC v 3-COL

- 2) VC ≼ VC v 3-COL
  - data una istanza ⟨ G=(V,E), k ⟩ di VC, la corrispondente istanza ⟨ G'=(V',E'), k' ⟩ di VC v 3-COL è ottenuta aggiungendo a G una clique di 4 nodi (che non è 3-colorabile) e ponendo k' = k+3



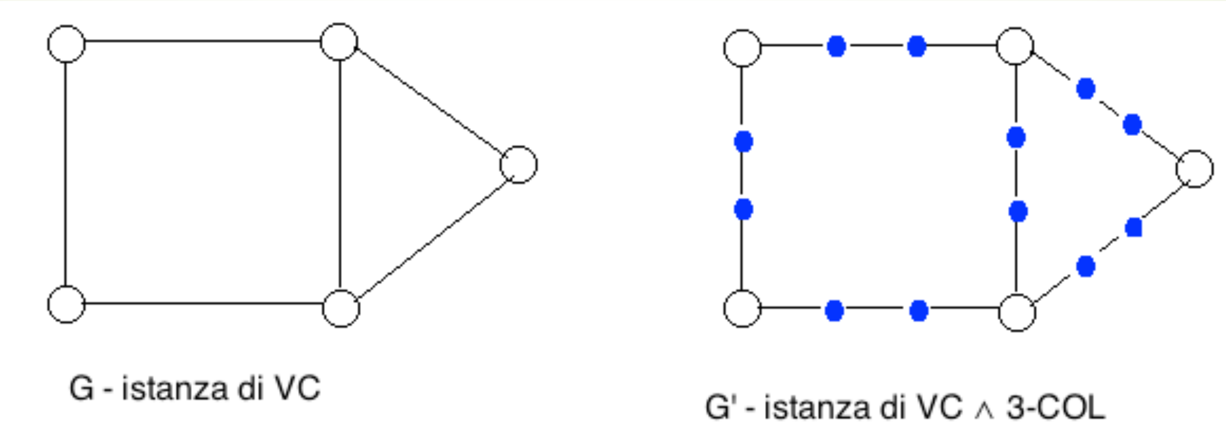
  - se G ha un vertex cover di k nodi, allora G' ha un vertex cover di k+3 nodi (3 nodi servono a coprire gli archi nella clique di 4 nodi) – ossia, ⟨ G'=(V',E'), k ⟩ è istanza sì di VC v 3-COL
  - se G non ha un vertex cover di k nodi, allora G' non ha un vertex cover di k+3 nodi (perché 3 nodi servono a coprire gli archi nella clique di 4 nodi) – allora, poiché G' non è 3-colorabile, ⟨ G'=(V',E'), k ⟩ è istanza no di VC v 3-COL

VC ∧ 3-COL

- Dati un grafo G = (V,E) ed un intero k, decidere se G ha un vertex cover di al più k nodi e, inoltre, è 3-colorabile
  - $\mathfrak{I}_{\mathbf{VCand3COL}}$ = { ⟨ G=(V,E), k ⟩ : G è un grafo non orientato ∧ k ∈ ℕ }
  - $\mathbf{S}_{\mathbf{VCand3COL}}$(G, k) = { (V', $\mathfrak{c}$): V' ⊆ V ∧ $\mathfrak{c}$: V → {1,2,3 } }
  - $\boldsymbol{\pi}_{\mathbf{VCand3COL}}$(G, k, $\mathbf{S}_{\mathbf{VCand3COL}}$(G, k))= ∃ (V', $\mathfrak{c}$) ∈ $\mathbf{S}_{\mathbf{VCand3COL}}$(G , k) :
    [| V' | ≤ k ∧ ∀ (u,v) ∈ E [u ∈ V' ∨ v ∈ V']] ∧ [∀ (u,v) ∈ E [$\mathfrak{c}$(u) ≠ $\mathfrak{c}$(v)]]
- Dimostriamo che VC ∧ 3-COL è NP-completo
- 1) VC ∧ 3-COL ∈ NP:
  - infatti, un certificato è una coppia (V', $\mathfrak{c}$) e verificare se soddisfa il predicato richiede tempo polinomiale in | ⟨ G=(V,E), k ⟩ |
- 2) per dimostrare la completezza di VC ∧ 3-COL rispetto a NP, riduciamo VC a VC ∧ 3-COL

VC ∧ 3-COL

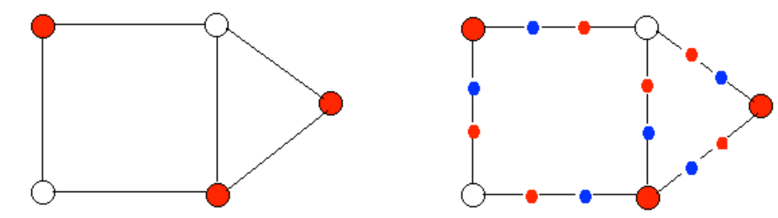
- 2) VC ≼ VC ∧ 3-COL
  - data una istanza ⟨ G=(V,E), k ⟩ di VC, la corrispondente istanza ⟨ G'=(V',E'), k' ⟩ di VC ∨ 3-COL è ottenuta sostituendo ogni arco in G con una catena di 4 nodi e ponendo k' = k+|E|



  - <u>Osserviamo</u> che G' è 3-colorabile: perciò, dobbiamo dimostrare che **G' è istanza sì di VC ∧ 3-COL se e solo se ha un vertex cover di k+|E| nodi**
  - <u>Osserviamo</u> che, in un qualunque vertex cover per G', |E| nodi devono essere nodi blu - servono a coprire gli archi nelle catene che abbiamo costruito

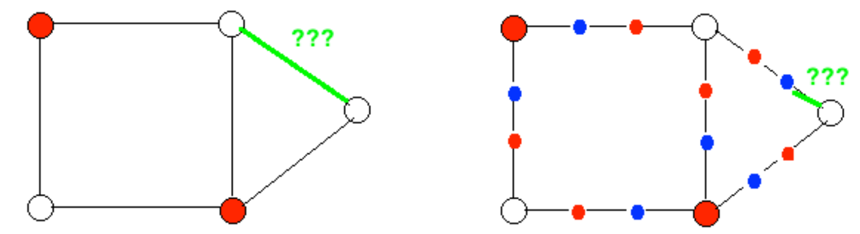
VC ∧ 3-COL

- 2) VC ≼ VC ∧ 3-COL
  - se G ha un vertex cover di k nodi, allora G' ha un vertex cover di k+|E| nodi (dei quali, |E| nodi blu) – ossia, ⟨ G'=(V',E'), k' ⟩ è istanza sì di VC ∧ 3-COL



<G, 3> - istanza sì di VC <G', 9> - istanza sì di VC ∧ 3-COL

  - se G non ha un vertex cover di k nodi, allora G' non ha un vertex cover di k+|E| nodi (poiché in G' |E| nodi servono a coprire gli archi adiacenti ai nodi blu nelle catene) – allora, ⟨ G'=(V',E'), k ⟩ è istanza no di VC ∧ 3-COL



<G,2> - istanza no di VC <G',8> - istanza no di VC ∧ 3-COL

VC ∧ ¬ 3-COL

- Dati un grafo G = (V,E) ed un intero k, decidere se G ha un vertex cover di al più k nodi e non è 3-colorabile
  - $\mathfrak{I}_{\mathbf{VCandnot3COL}} = \{ \langle G=(V,E), k \rangle : G$ è un grafo non orientato $\wedge\, k \in \mathbb{N} \}$
  - $\mathbf{S}_{\mathbf{VCandnot3COL}}(G, k) = \{ (V', \mathbf{c}) : V' \subseteq V \wedge \mathbf{c}: \mathbf{V} \rightarrow \{1,2,3\} \}$
  - $\pi_{\mathbf{VCandnot3COL}}(G, k, \mathbf{S}_{\mathbf{VCandnot3COL}}(G, k)) =$
    $\exists (V', \mathbf{c}) \in \mathbf{S}_{\mathbf{notVCand3COL}}(G, k) \, [\, (|V'| \leq k) \vee (\forall (u,v) \in E : u \in V' \vee v \in V') \,] \wedge$
    $\forall (V', \mathbf{c}) \in \mathbf{S}_{\mathbf{notVCand3COL}}(G, k) : [\exists (u,v) \in E \, [\, \mathbf{c}(u) = \mathbf{c}(v) \,] \,]$
- Osservate la complessità di questo predicato…
- Però, in effetti, la stessa riduzione utilizzata per mostrare che VC ≼ VC ∨ 3-COL funziona anche come riduzione da VC a VC ∧ ¬ 3-COL
  - infatti, poiché G' non è 3-colorabile, allora ⟨ G', k+3 ⟩ è una istanza sì di VC ∧ ¬ 3-COL se e soltanto se ⟨ G, k ⟩ è una istanza sì di VC
- allora, questo prova che VC ∧ ¬ 3-COL è NP-completo
- Siamo sicuri?!

VC ∧ ¬ 3-COL

- La stessa riduzione utilizzata per mostrare che VC ≼ VC ∨ 3-COL funziona anche come riduzione da VC a VC ∧ ¬ 3-COL
- allora, questo prova che VC ∧ ¬ 3-COL è NP-completo
- Siamo sicuri?!
- Non ci stiamo dimenticando qualcosa?...
- Per esempio, che **affinché un problema sia NP-completo, innanzi, tutto, quel problema deve appartenere a NP**?...
- Allora, vediamo se VC ∧ ¬ 3-COL appartiene a NP:
  - un certificato per l'istanza ⟨ G=(V,E), k ⟩ è l'insieme di tutte le coppie (V', c)
  - e verificare il certificato significa verificare che in tutte le coppie (V', c) V' non è un vertex cover di dimensione al più k e che, fra tutte queste coppie, ce n'è almeno una tale che è una 3-colorazione per G
  - solo in questo modo potremmo essere certi che il genio (burlone) ci ha detto il vero affermando che una certa istanza è una istanza sì!
- E questa verifica sembra richiedere tempo **più che polinomiale** in | ⟨ G=(V,E), k ⟩ |
- E, dunque, *non riusciamo a dimostrare* che VC ∧ ¬ 3-COL appartiene a NP!

Forse non in NP!

- Non riusciamo a dimostrare che VC ∧ ¬ 3-COL appartiene a NP!
- E nella stessa situazione si trovano i problemi
  - VC ∨ ¬ 3-COL
  - ¬ VC ∧ 3-COL
  - ¬ VC ∨ 3-COL
- In effetti, questi problemi appartengono a classi nelle quali NP è contenuta
  - classi che costituiscono la cosiddetta **gerarchia polinomiale**
- e si suppone che NP sia contenuta propriamente nella più piccola di queste classi
  - è un'altra delle congetture della complessità polinomiale
  - ma non ci interessiamo di questa questione in questo corso

¬ VC ∨ ¬ 3-COL

- Dati un grafo G = (V,E) ed un intero k, decidere se G non ha alcun vertex cover di al più k nodi oppure non è 3-colorabile
  - $\mathfrak{I}_{\mathbf{notVCornot3COL}}$ = { ⟨ G=(V,E), k ⟩ : G è un grafo non orientato ∧ k ∈ ℕ }
  - $\mathbf{S}_{\mathbf{notVCornot3COL}}$(G, k) = { (V', $\mathfrak{c}$): V' ⊆ V ∧ $\mathfrak{c}$: V → {1,2,3 } }
  - $\boldsymbol{\pi}_{\mathbf{notVCornot3COL}}$(G, k, $\mathbf{S}_{\mathbf{notVCornot3COL}}$(G, k))= ∀ (V', $\mathfrak{c}$) ∈ $\mathbf{S}_{\mathbf{notVCornot3COL}}$(G , k)
    [(| V' | > k ∨ ∃ (u,v) ∈ E : u ∉ V' ∧ v ∉ V') ∨ (∃ (u,v) ∈ E : $\mathfrak{c}$(u) = $\mathfrak{c}$(v))]
- anche in questo caso un certificato per l'istanza ⟨ G=(V,E), k ⟩ è l'insieme di tutte le coppie (V', $\mathfrak{c}$)
- e, di nuovo, verificare un certificato sembra richiedere tempo **più che polinomiale** in | ⟨ G=(V,E), k ⟩ |
- Tuttavia: se osserviamo bene il predicato di ¬ VC ∨ ¬ 3-COL ci accorgiamo che $\boldsymbol{\pi}_{\mathbf{notVCornot3COL}} = \neg\, \boldsymbol{\pi}_{\mathbf{VCand3COL}}$
- ossia, **¬ VC ∨ ¬ 3-COL = (VC ∧ 3-COL)^{c}**
- e poiché VC ∧ 3-COL ∈ NP, allora ¬ VC ∨ ¬ 3-COL ∈ coNP
- e poiché VC ∧ 3-COL è NP-completo, allora ¬ VC ∨ ¬ 3-COL è coNP-completo
  - per il teorema 6.25

¬ VC ∧ ¬ 3-COL

- Dati un grafo G = (V,E) ed un intero k, decidere se G non ha alcun vertex cover di al più k nodi e, inoltre, non è 3-colorabile

- valgono per questo problema identiche considerazioni a quelle che abbiamo fatto per il problema ¬ VC ∨ ¬ 3-COL

- Quindi, per esercizio:
  - formalizzate questo problema
  - dimostrate che è coNP-completo

Problemi coNP-completi

- Gli ultimi due problemi sono coNP-completi
- abbiamo dimostrato questo fatto dimostrando che, ciascuno di essi, è il complemento di un problema NP-completo
- **e questo è l'unico modo per dimostrare che un problema è coNP-completo**
- Infatti, la classe **coNP** è definita come "**la classe dei linguaggi il cui complemento appartiene a NP**"
- e poi il teorema 6.25 completa l'opera, indicandoci quali dei problemi in coNP sono completi
- Quindi, di problemi coNP-completi ne conosciamo a bizzeffe: VC^C, IS^C, SAT^C, …
  - tutti i problemi che sono il complemento di un problema NP-completo
- Ma come riconosciamo che un problema è in coNP?
- Un indizio c'è …
- E l'indizio è nel predicato
- Vediamolo con un esempio

Problemi coNP-completi

- Dati un grafo G = (V,E) ed un intero k, decidere se G non ha alcun vertex cover di al più k nodi
  - $\mathfrak{I}_{\mathbf{notVC}}$ = { ⟨ G=(V,E), k ⟩ : G è un grafo non orientato ∧ k ∈ ℕ }
  - $\mathbf{S}_{\mathbf{notVC}}$(G, k) = { V': V' ⊆ V }
  - $\boldsymbol{\pi}_{\mathbf{notVC}}$(G, k, $\mathbf{S}_{\mathbf{notVC}}$(G, k)) = ∀ (V', c) ∈ $\mathbf{S}_{\mathbf{notV}}$(G , k) [| V' | > k ∨ ∃ (u,v) ∈ E : u ∉ V' ∧ v ∉ V']
- Il predicato inizia con "∀"
  - questo significa che un certificato per l'istanza ⟨ G=(V,E), k ⟩ è l'insieme di tutti i sottoinsiemi V' di V
  - e poiché il numero di sottoinsiemi di V è $2^{|V|}$, non riusciamo a verificare un ***siffatto*** certificato in tempo polinomiale
- D'altra parte, negando il predicato, otteniamo un nuovo predicato che inizia con "∃"
  - e ci dice che un certificato è un singolo sottoinsieme V' di V!
- **ATTENZIONE**: un predicato che inizia con "∃" è solo un **indizio** di appartenenza a NP
  - abbiamo visto esempi di problemi non in NP il cui certificato inizia con "∃" (VC ∧ ¬ 3-COL)
  - **perciò verificate sempre che il problema ammetta certificati verificabili in tempo polinomiale!**

Problemi coNP-completi

- Dati un grafo G = (V,E) ed un intero k, decidere se G non ha alcun vertex cover di al più k nodi
  - $\mathfrak{I}_{\mathbf{notVC}}$ = { ⟨ G=(V,E), k ⟩ : G è un grafo non orientato ∧ k ∈ ℕ }
  - $\mathbf{S}_{\mathbf{notVC}}$(G, k) = { V': V' ⊆ V }
  - $\boldsymbol{\pi}_{\mathbf{notVC}}$(G, k, $\mathbf{S}_{\mathbf{notVC}}$(G, k)) = ∀ (V', c) ∈ $\mathbf{S}_{\mathbf{notV}}$(G , k) [| V' | > k ∨ ∃ (u,v) ∈ E : u ∉ V' ∧ v ∉ V']
- Il predicato negazione del predicato $\boldsymbol{\pi}_{\mathbf{notVC}}$(G, k, $\mathbf{S}_{\mathbf{notVC}}$(G, k)) è:
  ¬ $\boldsymbol{\pi}_{\mathbf{notVC}}$(G, k, $\mathbf{S}_{\mathbf{notVC}}$(G, k)) = ∃(V', c) ∈ $\mathbf{S}_{\mathbf{notV}}$(G , k) : | V' | ≤ k ∧ ∀(u,v) ∈ E [u ∉ V' ∨ v ∉ V']
- ed esso coincide con il predicato di VC
- Poiché VC è NP-completo, questo ci permette di concludere che il problema in questione è coNP-completo
  - e mostrare che il complemento di un problema Γ è NP-completo è l'unico modo per dimostrare che il problema Γ è coNP-completo!

Esercizi

- Sul modello della definizione dei problemi che abbiamo visto in questa lezione, definite, formalizzate e analizzate (dal punto di vista della complessità computazionale) i seguenti problemi che combinano, in tutti i modi possibili, i problemi VC e 2-COL:
  - VC ∨ 2-COL:
  - VC ∧ 2-COL
  - VC ∧ ¬ 2-COL
  - VC ∨ ¬ 2-COL
  - ¬ VC ∧ 2-COL
  - ¬ VC ∨ 2-COL
  - ¬ VC ∨ ¬ 2-COL
  - ¬ VC ∧ ¬ 2-COL
- ricordando che 2-COL ∈ P
- Suggerimento: provate a farvi un'idea intuitiva di quale possa essere la complessità di ciascun problema e poi provate a dimostrarla o a confutarla