



Lezione 27 – complessità di problemi

Lezione del 14/05/2025



Complessità di problemi e codifica

- Siamo pronti ad affrontare il paragrafo 7.5
- Ci eravamo riproposti di estendere ai problemi quello che abbiamo studiato relativamente alla complessità di linguaggi
- a patto, come abbiamo chiarito, di utilizzare *codifiche ragionevoli* per codificare le istanze dei problemi
- Resta da capire come trasformare un problema in un linguaggio!
- E se questa trasformazione è indolore
 - o se ci costringe a considerare qualche nuova questioncina...

Da problema a linguaggio

- Sia $\Gamma = \langle \mathfrak{I}_\Gamma, S_\Gamma, \pi_\Gamma \rangle$ un problema decisionale
- Osserviamo che l'insieme $\mathfrak{I}_\Gamma$ delle istanze di Γ è partizionato in **due** sottoinsiemi:
 - l'insieme delle **istanze sì** – ossia le istanze che verificano π_Γ
 - l'insieme delle **istanze no** - ossia le istanze che non verificano π_Γ
- Sia $\chi : \mathfrak{I}_\Gamma \rightarrow \Sigma^*$ una codifica (ragionevole) per Γ .
- La codifica χ partiziona Σ^* in **tre** sottoinsiemi di parole:
 - l'insieme **Y_Γ** delle parole che codificano istanze sì di Γ ;
 - l'insieme **N_Γ** delle parole che codificano istanze no di Γ ;
 - l'insieme delle parole che non codificano istanze di Γ .
- Il linguaggio associato a Γ mediante la codifica χ è il sottoinsieme **$L_\Gamma(\chi)$** di Σ^* **contenente le parole appartenenti a Y_Γ** , ossia,

$$L_\Gamma(\chi) = \{ x \in \Sigma^* : \exists y \in \mathfrak{I}_\Gamma [x = \chi(y) \wedge \pi_\Gamma(y, S_\Gamma(y))] \}.$$

Da problema a linguaggio

- Sia $\Gamma = \langle \mathfrak{I}_\Gamma, S_\Gamma, \pi_\Gamma \rangle$ un problema decisionale
- Il linguaggio associato a Γ mediante la codifica χ è il sottoinsieme $L_\Gamma(\chi)$ di Σ^* contenente le parole che codificano l'insieme Y_Γ , ossia,

$$L_\Gamma(\chi) = \{ x \in \Sigma^* : \exists y \in \mathfrak{I}_\Gamma [x = \chi(y) \wedge \pi_\Gamma(y, S_\Gamma(y))] \}.$$

- Dunque, decidere se una istanza y di Γ è una istanza sì corrisponde a decidere se $x = \chi(y)$ è contenuto in $L_\Gamma(\chi)$
- e, d'altro canto, data $x \in \Sigma^*$, per decidere se $x \in L_\Gamma(\chi)$ occorre:
 - decidere se x è la codifica di un'istanza y di Γ
 - e poi, in caso affermativo, decidere se il predicato $\pi_\Gamma(y, S_\Gamma(y))$ è soddisfatto

Complessità di un problema

- A questo punto, possiamo definire la complessità computazionale di un problema decisionale.

- **Definizione 7.3:** Sia $\Gamma = \langle \mathfrak{I}_\Gamma, S_\Gamma, \pi_\Gamma \rangle$ un problema decisionale e sia C una classe di complessità

- data una funzione f totale e calcolabile

- $C \in \{ \text{DTIME}[f(n)] , \text{DSpace}[f(n)] , \text{NTIME}[f(n)] , \text{NSPACE}[f(n)] \}$

- Diciamo che

$\Gamma \in C$ se esiste una codifica ragionevole $\chi : \mathfrak{I}_\Gamma \rightarrow \Sigma^*$ per Γ tale che $L_\Gamma(\chi) \in C$.

- Vediamo ora con un esempio cosa occorre fare per decidere se $x \in L_\Gamma(\chi)$
 - e, quindi, da cosa è caratterizzata la complessità di un **problema**
 - e in cosa si differenzia lo studio della complessità di problemi dallo studio della complessità di linguaggi

Decidere un problema

- **Esempio 7.6:** Ricordiamo il problema 3SAT e la codifica χ_1
 - se $X = \{x_1, x_2, x_3\}$ e $f = c_1 \wedge c_2$ con $c_1 = x_1 \vee x_2 \vee x_3$ e $c_2 = x_1 \vee \neg x_2 \vee \neg x_3$ allora
 $\chi_1(X, f) = 444\ 0\ 100\ 2\ 0\ 010\ 2\ 0\ 001\ 3\ 0\ 100\ 2\ 1\ 010\ 2\ 1\ 001$
 - che abbiamo visto essere una codifica ragionevole
- Allora, una parola $x \in \{0, 1, 2, 3, 4\}^*$ è in $L_{3SAT}(\chi_1)$ se sono verificati i due fatti seguenti.
 - 1) x deve essere la codifica secondo χ_1 di qualche coppia $\langle X, f \rangle$ istanza di 3SAT:
 - ad esempio, è facile verificare che 4021011103240111 non è la codifica di alcuna istanza
 - Se x non è una codifica valida, possiamo subito concludere che $x \notin L_{3SAT}(\chi_1)$.
 - 2) Se x è la codifica secondo χ_1 di una istanza $\langle X, f \rangle$ di 3SAT, affinché $x \in L_{3SAT}(\chi_1)$ occorre che f sia soddisfacibile.
- ossia, come abbiamo visto, dati un problema Γ e una sua codifica ragionevole χ , per verificare che una parola sia in $L_\Gamma(\chi)$ **occorre innanzi tutto verificare che essa sia la codifica di una istanza.**

Il linguaggio delle istanze

- Dato un problema Γ ed una codifica ragionevole $\chi : \mathfrak{I}_\Gamma \rightarrow \Sigma^*$ per $\mathfrak{I}_\Gamma$,
- definiamo il **linguaggio delle istanze di Γ** , ossia, il linguaggio

$$\chi(\mathfrak{I}_\Gamma) = \{ x \in \Sigma^* : \exists y \in \mathfrak{I}_\Gamma [x = \chi(y)] \}.$$

- OSSERVAZIONE:

- χ è una codifica di $\mathfrak{I}_\Gamma$
- quindi, se $y, z \in \mathfrak{I}_\Gamma$ sono due istanze di Γ con $y \neq z$, allora $\chi(y) \neq \chi(z)$
- quindi χ è una funzione *invertibile*
- allora, possiamo definire il linguaggio $L_\Gamma(\chi)$ anche nella maniera seguente:

$$L_\Gamma(\chi) = \{ x \in \Sigma^* : x \in \chi(\mathfrak{I}_\Gamma) \wedge \pi_\Gamma(\chi^{-1}(x)), \mathfrak{S}_\Gamma(\chi^{-1}(x)) \}$$

- Dunque, se, per decidere se una parola x appartiene a $L_\Gamma(\chi)$ dobbiamo anche verificare se x è effettivamente la codifica di un'istanza di Γ ,
- allora per definire la complessità del problema decisionale Γ occorre considerare anche la complessità di decidere il linguaggio $\chi(\mathfrak{I}_\Gamma)$

Esempio 7.7

- Consideriamo un nuovo problema decisionale PHC (Percorso in Ciclo Hamiltoniano):
 - sia dato un particolare grafo non orientato $G=(V,E)$
 - G è un grafo particolare: contiene un ciclo che passa una ed una sola volta per ciascuno dei suoi nodi (che si chiama ciclo hamiltoniano)
 - siano dati, inoltre, due suoi nodi $u, v \in V$;
 - si chiede di decidere se esiste in G un percorso che collega u a v .
- Formalizziamo il problema precedente mediante la tripla $\langle \mathfrak{I}_{\text{PHC}}, S_{\text{PHC}}, \pi_{\text{PHC}} \rangle$:
 - $\mathfrak{I}_{\text{PHC}} = \{ \langle G = (V,E), u, v \rangle : G \text{ è un grafo non orientato } \wedge \exists \text{ un ciclo } c \text{ in } G \text{ che passa una e una sola volta attraverso ciascun nodo di } G \wedge u, v \in V \}$;
 - $S_{\text{PHC}}(G,u,v) = \{ p : p \text{ è un percorso in } G \}$;
 - $\pi_{\text{PHC}}(G,u,v, S_{\text{PHC}}(G,u,v)) = \exists p \in S_{\text{PHC}}(G,u,v) \text{ che connette } u \text{ a } v$.
- ATTENZIONE: Se sappiamo che un grafo contiene un ciclo che passa (una e una sola volta) attraverso tutti i nodi di G , allora, qualunque coppia di nodi u,v si consideri, una porzione di quel ciclo è un percorso da u a v
- Questo significa che ogni istanza del problema PHC è una istanza sì.

Linguaggio delle istanze e complessità

- Ogni istanza del problema PHC è una istanza sì.
- Quindi, indipendentemente dalla codifica utilizzata, decidere se una qualunque istanza del problema soddisfa il predicato del problema richiede costo costante.
- D'altra parte, data una qualunque codifica ragionevole (diciamo, binaria) χ per PHC, per decidere se una parola $x \in \{0,1\}^*$ è contenuta in $L_{\text{PHC}}(\chi)$, dobbiamo verificare
 - sia se x è la codifica di una istanza di PHC, ossia, di un grafo che contiene un ciclo che attraversa tutti i nodi una e una sola volta e di una coppia di suoi nodi,
 - sia se detto grafo contiene un percorso che connette i due nodi.
- Come vedremo, la prima di queste due verifiche (ossia decidere $\chi(\mathfrak{I}_{\text{PHC}})$) è un noto linguaggio NP-completo.
- E, quindi, concludiamo che $L_{\text{PHC}}(\chi)$ è NP-completo.
- Allora, anche se,
 - *una volta associato che una parola $x \in \{0,1\}^n$ è istanza di PHC,*
- decidere se x soddisfa $\pi_{\text{PHC}}(G,u,v, S_{\text{PHC}}(G,u,v))$ ha costo costante
- non possiamo affermare che decidere PHC è un problema in P

Da un problema al suo complemento

- Sia Σ un qualunque alfabeto (neanche a dirlo, finito)
- una (qualunque) codifica χ delle istanze di un problema decisionale Γ in parole di Σ^* induce una tri-partizione di Σ^* - ossia, una partizione di Σ^* in tre sottoinsiemi:
 - l'insieme Y_Γ delle parole di Σ^* che codificano istanze sì di Γ - il linguaggio $L_\Gamma(\chi)$
 - l'insieme N_Γ delle parole di Σ^* che codificano istanze no di Γ
 - parole di Σ^* che non codificano istanze di Γ
- Ora, ricordiamo, dato un qualunque linguaggio $L \subseteq \Sigma^*$, il linguaggio complemento di L è : $L^c = \Sigma^* - L$
 - è così che lo avevamo definito!
- Perciò, secondo definizione, il linguaggio complemento di $L_\Gamma(\chi)$ è $(L_\Gamma(\chi))^c = \Sigma^* - L_\Gamma(\chi)$
 - ossia, tutte le parole di Σ^* che codificano istanze no di Γ e tutte le parole di Σ^* che non codificano istanze di Γ
- Uhm...

Da un problema al suo complemento

- Perciò, secondo definizione, il linguaggio complemento di $L_\Gamma(x)$ è $(L_\Gamma(x))^c = \Sigma^* - L$
 - ossia, tutte le parole di Σ^* che codificano istanze no di Γ e tutte le parole di Σ^* che non codificano istanze di Γ in
- Uhm... Ma siamo sicuri che questo è proprio ciò che corrisponde al complemento di un problema decisionale?
- In effetti, se pensiamo al complemento di un problema di decisione, quello che ci viene in mente sono **le istanze del problema che non soddisfano il predicato**
 - ad esempio, il problema $3SAT^c$ è l'insieme delle istanze $\langle X, f \rangle$ di 3SAT tali che non è soddisfacibile
 - formalmente, $3SAT^c = \langle \mathfrak{I}_{3SAT}, \mathcal{S}_{3SAT}, \neg \pi_{3SAT} \rangle$
- Perciò, il linguaggio che vogliamo associare al problema complemento di Γ non è $(L_\Gamma(x))^c = \Sigma^* - L_\Gamma(x)$, bensì l'insieme N_Γ ,

$$L_{\Gamma^c}(x) = \{ x \in \Sigma^* : x \in \chi(\mathfrak{I}_\Gamma) \wedge \neg \pi_\Gamma(\chi^{-1}(x), \mathcal{S}_\Gamma(\chi^{-1}(x))) \}$$

- ossia, formalmente (per gli interessati), $L_{\Gamma^c}(x) = (L_\Gamma(x))^c - \chi^c(\mathfrak{I}_\Gamma)$

Da un problema al suo complemento

- Dunque, il linguaggio che associamo al complemento di un problema decisionale Γ (codificato in Σ^* secondo una codifica χ) non è $(L_\Gamma(\chi))^c = \Sigma^* - L_\Gamma(\chi)$ ma $L_{\Gamma^c}(\chi)$
- Ora, dato un linguaggio L ed una classe di complessità $\mathcal{C}$, noi sappiamo (per definizione) che **se $L \in \mathcal{C}$ allora $L^c \in \text{co}\mathcal{C}$**
- Perciò, dato un problema decisionale Γ (codificato in Σ^* secondo una codifica χ), noi sappiamo che se $L_\Gamma(\chi) \in \mathcal{C}$ allora **$(L_\Gamma(\chi))^c \in \text{co}\mathcal{C}$**
- Bene.
- Ma, se sappiamo che se $L_\Gamma(\chi) \in \mathcal{C}$, cosa possiamo dire del linguaggio $L_{\Gamma^c}(\chi)$?
- Ossia: se sappiamo classificare (nell'ambito della complessità computazionale) un problema di decisione, sappiamo anche classificare il problema complemento???
- Prima di rispondere, vediamo un esempio

Da un problema al suo complemento

- **Esempio 7.8:** riprendiamo dall'esempio 7.7 il problema decisionale PHC: dato un grafo non orientato $G=(V,E)$ che contiene un ciclo che passa una ed una sola volta per ciascuno dei suoi nodi, e dati due suoi nodi $u, v \in V$, si chiede di decidere se esiste in G un percorso che collega u a v .
- PHC è formalizzato mediante la tripla $\langle \mathfrak{I}_{\text{PHC}}, S_{\text{PHC}}, \pi_{\text{PHC}} \rangle$:
 - $\mathfrak{I}_{\text{PHC}} = \{ \langle G = (V,E), u, v \rangle : G \text{ è un grafo non orientato } \wedge \exists \text{ un ciclo } c \text{ in } G \text{ che passa una e una sola volta attraverso ciascun nodo di } G \wedge u, v \in V \}$;
 - $S_{\text{PHC}}(G,u,v) = \{ p : p \text{ è un percorso in } G \}$;
 - $\pi_{\text{PHC}}(G,u,v, S_{\text{PHC}}(G,u,v)) = \exists p \in S_{\text{PHC}}(G,u,v) \text{ che connette } u \text{ a } v$.
- PHC^c è, allora:
 - dato un grafo non orientato $G=(V,E)$ che contiene un ciclo che passa una ed una sola volta per ciascuno dei suoi nodi, e dati due suoi nodi $u, v \in V$;
 - si chiede di decidere se non esiste in G alcun percorso che collega u a v .
- ed è formalizzato mediante la tripla $\langle \mathfrak{I}_{\text{PHC}}, S_{\text{PHC}}, \neg \pi_{\text{PHC}} \rangle$, con

$$\neg \pi_{\text{PHC}}(G,u,v, S_{\text{PHC}}(G,u,v)) = \nexists p \in S_{\text{PHC}}(G,u,v) \text{ che connette } u \text{ a } v.$$

Da un problema al suo complemento

- Formalizzato il problema precedente mediante la tripla $\langle \mathfrak{S}_{\text{PHC}}, S_{\text{PHC}}, \neg \pi_{\text{PHC}} \rangle$:
 - $\mathfrak{S}_{\text{PHC}} = \{ \langle G = (V, E), u, v \rangle : G \text{ è un grafo non orientato } \wedge \exists \text{ un ciclo } c \text{ in } G \text{ che passa una e una sola volta attraverso ciascun nodo di } G \wedge u, v \in V \}$;
 - $S_{\text{PHC}}(G, u, v) = \{ p : p \text{ è un percorso in } G \}$;
 - $\neg \pi_{\text{PHC}}(G, u, v, S_{\text{PHC}}(G, u, v)) = \nexists p \in S_{\text{PHC}}(G, u, v) \text{ che connette } u \text{ a } v$.
- Data una qualunque codifica ragionevole χ per PHC^c , per decidere se una parola x è contenuta in $L_{\text{PHC}^c}(\chi)$, dobbiamo verificare
 - se x è la codifica di una istanza di PHC^c , ossia, di un grafo che contiene un ciclo che attraversa tutti i nodi una e una sola volta, e di una coppia di suoi nodi,
 - e se detto grafo non contiene percorsi che connettono i due nodi.
- Come abbiamo visto,
 - la verifica che x sia effettivamente la codifica di un'istanza di PHC è un problema NP-completo
 - verificare se una qualunque istanza del problema soddisfa il predicato del problema richiede costo costante – perché nessuna istanza soddisfa il predicato!
- E, quindi, concludiamo (ad occhio) che PHC^c è NP-completo.

Da un problema al suo complemento

- Riassumiamo:
- il problema PHC è NP-completo
- e il suo complemento PHC^c è anch'esso NP-completo.

!

- Quindi parrebbe che non possiamo trasportare ai problemi decisionali la teoria della complessità che abbiamo sviluppato per i linguaggi.
- E questo perché **la complessità di un problema decisionale dipende anche dalla complessità di decidere il linguaggio delle istanze**
- Ma se la decisione del linguaggio delle istanze richiede "poche risorse"
- Possiamo trasferire tutto ciò che abbiamo studiato relativamente alla complessità dei linguaggi alla complessità dei problemi decisionali
- Come mostra il prossimo teorema

Il ruolo del linguaggio delle istanze

- **Teorema 7.1:** Sia $\Gamma = \langle \mathfrak{S}_\Gamma, S_\Gamma, \pi_\Gamma \rangle$ un problema decisionale e sia $\chi : \mathfrak{S}_\Gamma \rightarrow \Sigma^*$ una sua codifica ragionevole. Se $\chi(\mathfrak{S}_\Gamma) \in \mathbf{P}$, allora valgono le seguenti implicazioni:
 - 1) se $L_\Gamma(\chi) \in \mathbf{NP}$ allora $L_{\Gamma^c}(\chi) \in \mathbf{coNP}$
 - 2) se $L_\Gamma(\chi) \in \mathbf{NEXPTIME}$ allora $L_{\Gamma^c}(\chi) \in \mathbf{coNEXPTIME}$
- Dimostriamo il Teorema 7.1 nel caso 1)

Teorema 7.1 – caso 1)

- Se $\chi(\mathfrak{F}_\Gamma) \in \mathbf{P}$, allora esistono una macchina deterministica T ed un intero h tali che, per ogni $x \in \Sigma^*$, T decide se $x \in \chi(\mathfrak{F}_\Gamma)$ e $\text{dtime}(T, x) \in O(|x|^h)$.
- Se $L_\Gamma(\chi) \in \mathbf{NP}$, allora esistono una macchina non deterministica NT ed un intero k tali che, per ogni $x \in L_\Gamma(\chi)$, NT accetta x e $\text{ntime}(NT, x) \in O(|x|^k)$.
- Combinando T e NT , costruiamo una nuova macchina non deterministica NT_0 che **accetta il linguaggio complemento di $L_{\Gamma^c}(\chi)$** , ossia, che **accetta $(L_{\Gamma^c}(\chi))^c$** .
- Due domande sorgono spontanee:
 - PRIMA DOMANDA: ma che ce ne importa di accettare $(L_{\Gamma^c}(\chi))^c$?
 - Beh, se riusciamo a mostrare che possiamo accettare $(L_{\Gamma^c}(\chi))^c$ in tempo non deterministico polinomiale, allora $(L_{\Gamma^c}(\chi))^c$ è in $\mathbf{NP}$ e, dunque, $L_{\Gamma^c}(\chi) \in \mathbf{coNP}$.
 - SECONDA DOMANDA: quali parole troviamo in $(L_{\Gamma^c}(\chi))^c$?
 - Poiché in $L_{\Gamma^c}(\chi)$ troviamo parole che codificano istanze no di Γ , allora in $(L_{\Gamma^c}(\chi))^c$ troviamo
 - a) parole che non codificano istanze di Γ
 - b) parole che codificano istanze sì di Γ , ossia, parole che appartengono a $L_\Gamma(\chi)$

Teorema 7.1 – caso 1)

- Se $\chi(\mathfrak{F}_T) \in \mathbf{P}$, allora esistono una macchina deterministica T ed un intero h tali che, per ogni $x \in \Sigma^*$, T decide se $x \in \chi(\mathfrak{F}_T)$ e $\text{dtime}(T, x) \in O(|x|^h)$.
- Se $L_T(\chi) \in \mathbf{NP}$, allora esistono una macchina non deterministica NT ed un intero k tali che, per ogni $x \in L_T(\chi)$, NT accetta x e $\text{ntime}(NT, x) \in O(|x|^k)$.
- Combinando T e NT , costruiamo una nuova macchina non deterministica NT_0 che **accetta il linguaggio complemento di $L_T^c(\chi)$** , ossia, che **accetta $(L_T^c(\chi))^c$**
- NT_0 opera in due fasi: con input $x \in \Sigma^*$,
 - Fase 1. Simula la computazione $T(x)$: se $T(x)$ termina nello stato di rigetto, allora NT_0 termina nello stato di accettazione, altrimenti ha inizio la Fase 2.
 - Fase 2. Simula la computazione $NT(x)$: se $NT(x)$ accetta allora NT_0 accetta
- $NT_0(x)$ accetta quando $x \notin \chi(\mathfrak{F}_T)$ oppure $x \in L_T(\chi)$, cioè
- $NT_0(x)$ accetta se e soltanto se x appartiene a $(L_T^c(\chi))^c$**
- Inoltre, è semplice verificare che $\text{ntime}(NT_0, x) \in O(|x|^{\max\{h, k\}})$.
- Quindi: **$(L_T^c(\chi))^c$** è in NP, e dunque **$L_T^c(\chi) \in \mathbf{coNP}$** .

Le assunzioni di lavoro

- Non è ragionevole che sia più complesso decidere se una parola è istanza di un problema, che decidere se una istanza di quel problema è una istanza sì
 - come avviene negli esempi 7.7 e 7.8
- perché la difficoltà nel risolvere un problema non dovrebbe essere nel riconoscere che i dati che ci vengono forniti siano effettivamente dati del nostro problema, ma nel trovare una soluzione (o nel verificare che una soluzione esiste) ad una data istanza del problema.
- Per questa ragione, da ora in avanti assumeremo sempre che
 - per ogni problema di decisione Γ e per ogni sua codifica ragionevole χ
- il linguaggio delle istanze sia in P, ossia $\chi(\mathfrak{I}_\Gamma) \in P$
- Questo significa che, ad esempio, la formalizzazione del problema PHC sarà:
 - $\mathfrak{I}_{\text{PHC}} = \{ \langle G = (V,E), u, v \rangle : G \text{ è un grafo non orientato } \wedge u, v \in V \};$
 - $S_{\text{PHC}}(G,u,v) = \{ p : p \text{ è un percorso in } G \};$
 - $\pi_{\text{PHC}}(G,u,v, S_{\text{PHC}}(G,u,v)) = \exists \text{ un ciclo } c \text{ in } G \text{ che passa una e una sola volta attraverso ciascun nodo di } G \wedge \exists p \in S_{\text{PHC}}(G,u,v) \text{ che connette } u \text{ a } v.$
- ossia, sposteremo nel predicato *tutte le proprietà che devono essere soddisfatte dai dati che costituiscono l'istanza*