Teoria della NP-completezza

Damiano Macedonio
mace@unive.it

Complessità di problemi decisionali

- Consideriamo un problema Q come una relazione $Q \subseteq I \times S$
  - I è l'insieme delle istanze di ingresso
  - S è l'insieme delle soluzioni
- Possiamo immaginare Q come un predicato che, dato in ingresso una istanza di input $x \in I$ e una soluzione $s \in S$, restituisce:
  - 1 se $(x,s) \in Q$ (s è soluzione del problema Q sull'istanza x)
  - 0 altrimenti (s non è soluzione del problema Q sull'istanza x)

Decidere, ricercare, ottimizzare

- Problemi di decisione
  - Problemi che richiedono una risposta binaria ($S = \{0,1\}$)
  - Es: “Un dato grafo x è connesso?”
  - Es: “Un elemento x è contenuto in un dizionario?”
- Problemi di ricerca
  - Data una istanza x, restituire una soluzione s tale che $(x,s) \in Q$
  - Es: “Trovare un albero ricoprente per il grafo x”
- Problemi di ottimizzazione
  - Data una istanza x, restituire la “migliore” soluzione s
  - Es: “Trovare un <u>minimo</u> albero ricoprente per il grafo x”

Classi di complessità

- Data una funzione $f(n)$, chiamiamo TIME($f(n)$) e SPACE($f(n)$) l'insieme di tutti i problemi decisionali che possono essere risolti in tempo e spazio risp. O($f(n)$)
  - Ossia tutti i problemi che ammettono un algoritmo A tale che per ogni istanza di input x, l'algoritmo restituisce 1 sse (x,1) risolve il problema. A deve avere complessità O($f(n)$) in tempo e spazio

Classi di complessità

- La classe P è la classe dei problemi risolvibili in tempo polinomiale nella dimensione n dell'istanza di ingresso

$$P = \cup_{c=0}^{\infty} TIME\left(n^{c}\right)$$

- La classe PSPACE è la classe dei problemi risolvibili in spazio polinomiale nella dimensione n dell'istanza di ingresso

$$PSPACE = \cup_{c=0}^{\infty} SPACE\left(n^{c}\right)$$

Classi di complessità

- La classe EXPTIME è la classe dei problemi risolvibili in tempo esponenziale nella dimensione n dell'istanza di ingresso

$$EXPTIME = \cup_{c=0}^{\infty} TIME\left(2^{n^c}\right)$$

Classi di complessità

- Un algoritmo che richiede tempo polinomiale riuscirà al più ad accedere ad un numero polinomiale di diverse locazioni di memoria, quindi:

$$P \subseteq PSPACE$$

- Poiché n^c locazioni di memoria possono trovarsi al più in 2^{n^c} stati diversi, si ha anche

$$PSPACE \subseteq EXPTIME$$

- Non è noto se le inclusioni di cui sopra sono strette o no (non si sa ad es se P $\subset$ PSPACE)

Esempi

- Una espressione booleana si dice in forma normale congiuntiva se è espressa come congiunzione di clausole
- Esempio $(x \vee \bar{y} \vee z) \wedge (y \vee z) \wedge (\bar{x} \wedge y \wedge \bar{z})$
- *Data una espressione booleana in forma normale congiuntiva, Il problema della soddisfacibilità richiede di verificare se esiste una assegnazione di valori alle variabili che rende l'espressione vera*
- Il problema della soddisfacibilità è in EXPTIME ∩ PSPACE
  - Se ci sono n variabili, basta tentare tutte le 2^n assegnazioni

La classe NP

- Come visto in precedenza, nei problemi di decisione siamo interessati a sapere se una istanza x del problema verifica una certa proprietà
  - Es: L'espressione booleana in forma normale congiuntiva K è soddisfacibile?
- Spesso però siamo anche interessati a conoscere un qualche oggetto *y*, che dipende da *x* e dal problema da risolvere, che possa certificare il fatto che *x* gode di tale proprietà
  - Es: una assegnazione di variabili che rende l'espressione booleana K vera

Esempio

- Il problema delle formule booleane quantificate richiede di verificare se una certa formula booleana quantificata è vera
- Es: $\exists x \forall y \exists z \forall w : (x \vee \bar{y} \vee z) \wedge (\bar{x} \vee w) \wedge y$
  - l'espressione sopra è falsa. Perché?
- Per il problema delle formule booleane quantificate non basta un certificato di dimensione polinomiale!
  - Quando l'espressione è vera, se tutte le variabili sono quantificate universalmente occorrerà fornire un numero *esponenziale* di possibili assegnamenti per certificare che l'espressione è vera!

Definizione

- Informalmente NP è la classe dei problemi decisionali che ammettono certificati verificabili in tempo polinomiale.

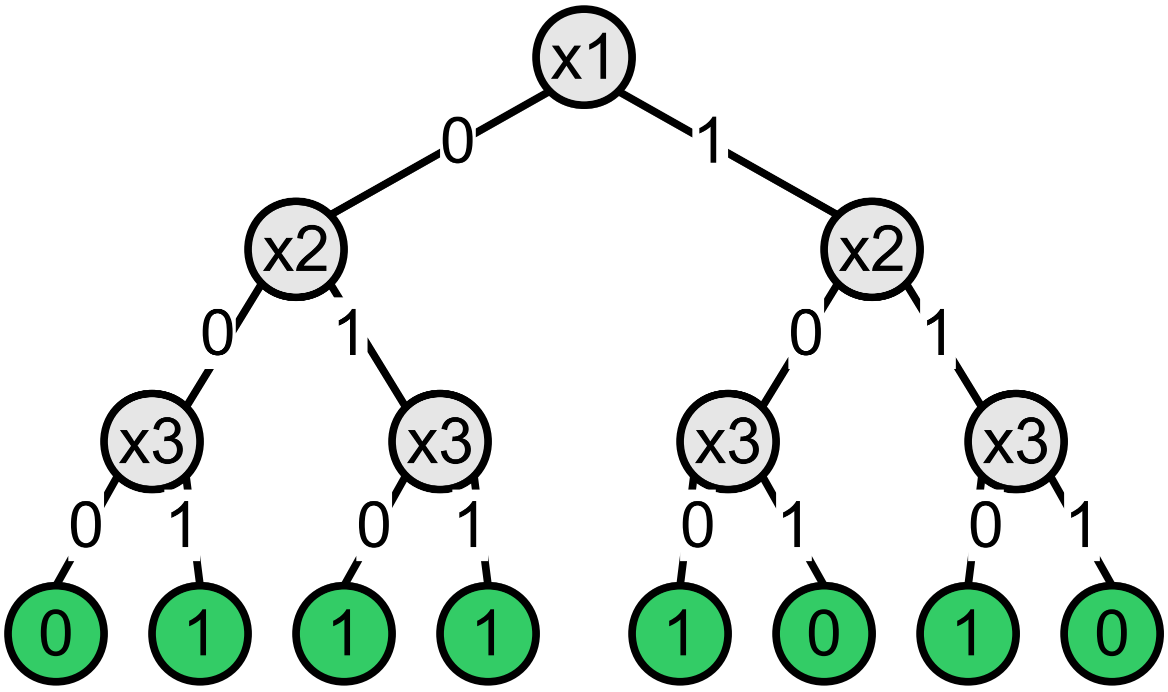
Non determinismo

- Negli algoritmi visti a lezione, ogni passo è sempre univocamente determinato dal passo precedente
- Un algoritmo non deterministico, invece, oltre alle normali istruzioni può eseguire istruzioni del tipo "indovina $z \in \{0,1\}$"
  - In altre parole, può indovinare un valore booleano "corretto" e far proseguire la computazione nella "giusta" direzione.
- Un algoritmo non deterministico può essere rappresentato da un albero di decisione

Esempio

- L'espressione seguente è soddisfacibile?

$$\left(x_1 \vee x_2 \vee x_3\right) \wedge \left(\bar{x}_1 \vee \bar{x}_3\right)$$

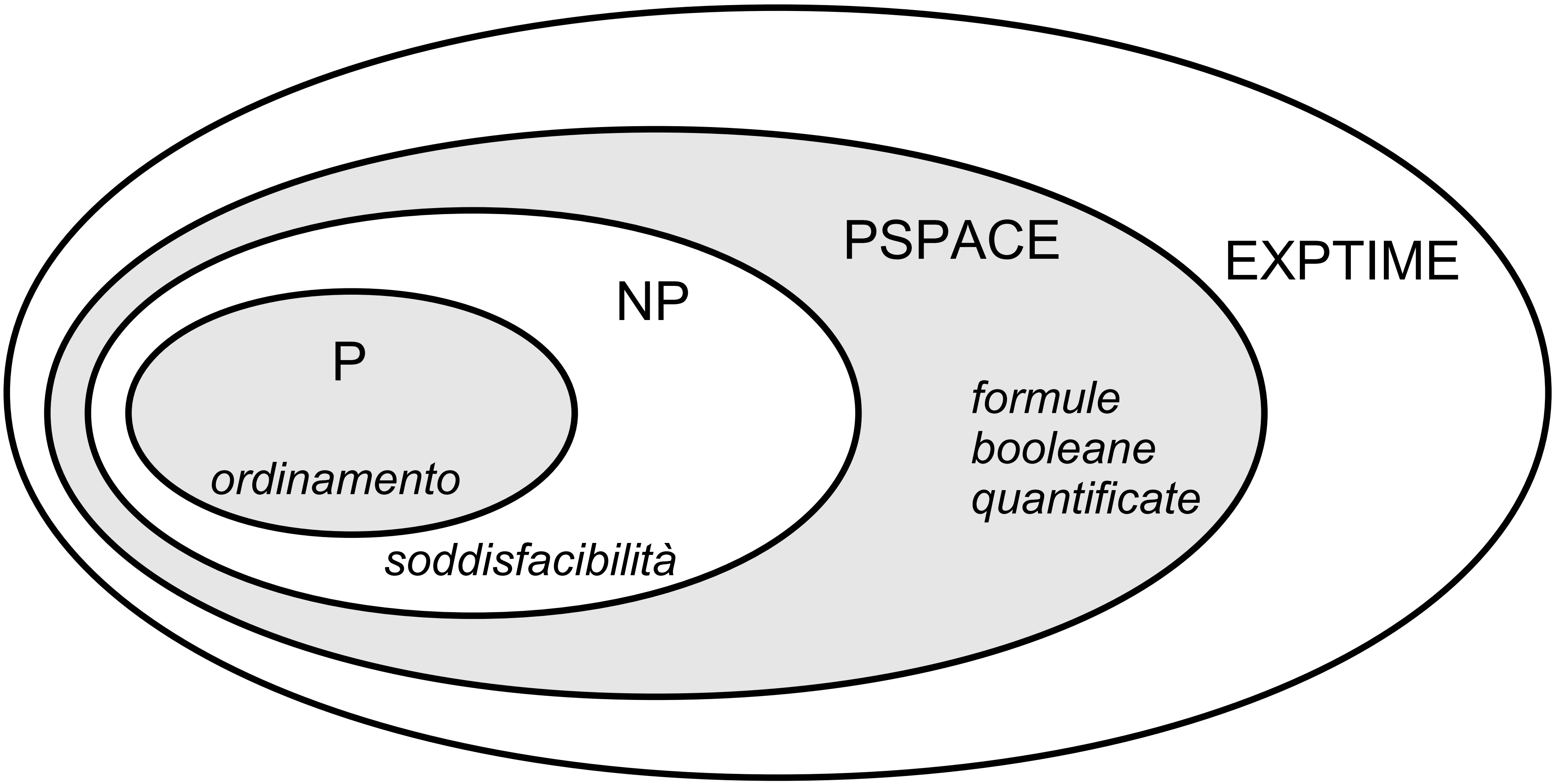


Definizione

- Data una funzione f(n), chiamiamo NTIME(f(n)) l'insieme dei problemi decisionali che possono essere risolti da un algoritmo non deterministico in tempo O(f(n)). La classe NP è la classe dei problemi risolvibili in tempo polinomiale non deterministico nella dimensione n dell'istanza di ingresso:

$$NP = \cup_{c=0}^{\infty} NTIME\left(n^{c}\right)$$

Uno sguardo alla gerarchia



- N: si **congettura** che le inclusioni siano proprie:

$P \subset NP \subset PSPACE \subset EXPTIME$

Riducibilità polinomiale

- Consideriamo due problemi decisionali $Q1:I1 \to \{0,1\}$ e $Q2:I2 \to \{0,1\}$
- Supponiamo di avere una funzione $f:I1 \to I2$ in grado di trasformare istanze di input per Q1 in istanze di input per Q2, tali che per ogni soluzione s

$$(x,s) \in Q1 \text{ se e solo se } (f(x), s) \in Q2$$

- Allora diremo che Q1 è riducibile polinomialmente a Q2

Riducibilità polinomiale

- Consideriamo un problema Q2 per il quale sia noto un algoritmo risolutivo polinomiale
  - Quindi $Q2 \in P$
- Supponiamo che Q1 sia riducibile polinomialmente a Q2
- Allora anche $Q1 \in P$

Definizioni

- Un problema decisionale Q si dice NP-arduo se ogni problema $W \in NP$ è riducibile polinomialmente a Q
- Un problema decisionale Q si dice NP-completo se appartiene alla classe NP ed è NP-arduo
- Nota: se un qualunque problema decisionale NP-completo appartenesse alla classe P, allora P = NP
- **Teorema di Cook**: Il problema della soddisfacibilità di espressioni booleane in forma normale congiuntiva è NP-completo

Riepilogo