Progetto e analisi di algoritmi

Roberto Cordone
DTI - Università degli Studi di Milano
Polo Didattico e di Ricerca di Crema

Tel. 0373 / 898**089**
E-mail: **`cordone@dti.unimi.it`**
Ricevimento: **su appuntamento**
Web page: **`http://www.dti.unimi.it/~cordone`**
Lezioni: **Martedì dalle 11.00 alle 13.00**
Giovedì dalle 11.00 alle 13.00

Progetto e analisi di algoritmi

- Progetto è passare dalla descrizione di un problema alla descrizione di un algoritmo che lo risolve
- Analisi è passare dalla descrizione di un problema e di un algoritmo
  1. alla prova che l’algoritmo risolve il problema o no (correttezza)
  2. alla prova di quanto costa risolvere il problema con l’algoritmo (complessità)
- Algoritmo è uno strumento formale (cioè meccanico) per risolvere un problema

Un programma di lavoro

Per impostare rigorosamente il discorso dovremo tradurre fini e mezzi in concetti privi di ambiguità

1. Che cos'è un problema?
2. Che cos'è una soluzione?
3. Che significa risolvere un problema?
4. Con quali strumenti si risolve un problema?
   (a) Che cos'è una macchina?
   (b) Che cos'è un algoritmo?
5. Che cos'è il costo di un processo risolutivo?
   Come si misura? (complessità computazionale)

Una definizione informale

Problema è una domanda sulle proprietà di un sistema completamente descritto in termini quantitativi

- grandezze numeriche, fisiche, ordinali, logiche, . . .
- relazioni fra grandezze
  (ordine, appartenenza, funzioni, . . .)
- operazioni (regole di trasformazione del sistema)

Ogni problema che riguarda un sistema fisico
va prima tradotto in un modello matematico

Esempi di problema (1)

1) Il numero $2^{17} - 1$ è un numero primo?

- La domanda è definita dal numero $2^{17} - 1$
- La risposta è Sì oppure No

2) In quanto tempo si arriva in Duomo?

- La domanda è definita dalla rete stradale di Crema (con i relativi sensi unici, lunghezze e velocità)
- La risposta è un numero (ad es., in minuti)

Esempi di problema (2)

3) Quali sono i nuclei familiari di Crema?

- La domanda è definita dall'insieme dei cittadini e delle loro relazioni di parentela e convivenza
- La risposta è una collezione di sottoinsiemi di cittadini

4) In quale ordine si trovano nel vocabolario le parole *zebra, alce, renna, gnu, antilope*?

- La domanda è definita da una sequenza di parole
- La risposta è una permutazione delle stesse parole

Istanze e soluzioni

Capita spesso di porre la stessa domanda su sistemi diversi

- 2 è un numero primo?
- 3 è un numero primo?
- 10^{300} è un numero primo?
- $2^{17} - 1$ è un numero primo?

Definiremo

- Istanza (o *esemplare*) = descrizione quantitativa dello specifico sistema su cui si pone la domanda
- Soluzione = descrizione quantitativa della specifica risposta

Problemi mal posti

Esempio: Chi sono?

Se la domanda è esistenziale, è un problema mal posto (in senso matematico. . .)

- la domanda riguarda un sistema che non ha descrizione completa e quantitativa
- la risposta che ci si attende non è quantitativa

Chi sono? Una definizione

- Tizio, Caio e Sempronio abitano in un palazzo di tre piani (piano terra, primo piano e secondo piano)
- Ognuno di loro ha un diverso animale (un cane, un gatto e un pesce rosso)
- Ognuno abita a un diverso piano (con il proprio animale)
- Caio è allergico al pelo di gatto, mentre Tizio ha un gatto
- Sempronio non ha pesci rossi
- Il padrone del cane abita al piano terra
- Il padrone del pesce rosso non abita al secondo piano
- Io abito al primo piano

Chi sono? Un principio di modello

Il sistema è definito completamente e quantitativamente da

- un insieme di persone $\{T, C, S\}$
- un insieme di animali $\{C, G, P\}$
- un insieme di abitazioni $\{0, 1, 2\}$
- un insieme di relazioni persona-animale $\{C \nleftrightarrow G, T \leftrightarrow G, S \nleftrightarrow P\}$
- un insieme di relazioni animale-abitazione $\{C \leftrightarrow 0, P \nleftrightarrow 2\}$
- un insieme di relazioni persona-abitazione $\{? \leftrightarrow 1\}$

La risposta è definita completamente e quantitativamente da
un elemento di $\{T, C, S\}$

Codifica di un problema

La definizione informale è inadatta a una macchina
Occorre

- codificare l'istanza perché una macchina possa manipolarla
- codificare la soluzione perché una macchina possa produrla
- definire il meccanismo che trasforma una codifica nell'altra

Codifica di istanza e soluzione

1. Si sostituiscono elementi, grandezze e relazioni con simboli
2. Si raccolgono i simboli in una struttura semplice (stringa)

Definizione formale di problema

- Istanza è una stringa di simboli I
- Soluzione è una stringa di simboli S
- Problema è un insieme P di coppie istanza-soluzione (I, S)

Esempio: il problema della primalità ("n è un numero primo?")

$$I = 3 \quad S_I = \textit{Sì} \qquad I' = 4 \quad S_{I'} = \textit{No}$$

$$P = \{(1, \textit{Sì}), (2, \textit{Sì}), (3, \textit{Sì}), (4, \textit{No}), \ldots\}$$

Se si vede la virgola come simbolo separatore fra I e S,

problema è un insieme di stringhe

Esempio

Istanza

- un insieme di persone $\{T, C, S\}$
- un insieme di animali $\{C, G, P\}$
- un insieme di abitazioni $\{0, 1, 2\}$
- un insieme di relazioni $\{C \not\leftrightarrow G, T \leftrightarrow G, \ldots\}$

Codifica dell'istanza:

$I = T\,C\,S \cdot C\,G\,P {\cdot} 0\,1\,2 \cdot C \not\leftrightarrow G\,T \leftrightarrow G\,S \not\leftrightarrow P \cdot C \leftrightarrow 0\,P \not\leftrightarrow 2 \cdot ? \leftrightarrow 1$

Soluzione: un elemento di $\{T, C, S\}$
Codifica della soluzione: $S = C$

Quale codifica?

Perché una codifica monodimensionale?

- Non è obbligatoria (l'uomo di solito non la usa)
- Una macchina potrebbe usare anche altre codifiche
- Però è la codifica più semplice per una macchina

(In pratica: legge i dati da un file. . .)

Quindi una macchina riceve stringhe di simboli e
deve saper riconoscere le istanze dalle stringhe non valide

(In pratica: deve saper distinguere
se il file d'ingresso ha un formato valido)

Problemi e linguaggi formali (1)

Istanze e soluzioni sono stringhe di simboli

Un problema è una collezione di stringhe di simboli

Ricordate nulla di analogo?

- Simbolo: entità atomica, non ulteriormente scomponibile
- Alfabeto $\mathcal{A}$: insieme finito e non vuoto di simboli
- Stringa s (o *parola*): insieme vuoto oppure sequenza ordinata finita di simboli di $\mathcal{A}$

In una stringa i simboli possono ripetersi

Problemi e linguaggi formali (2)

- $\mathcal{A}^*$ = collezione di tutte le stringhe definibili sull'alfabeto $\mathcal{A}$
- linguaggio $\mathcal{L}$ (sull'alfabeto $\mathcal{A}$): un sottoinsieme di $\mathcal{A}^*$ cioè una collezione di parole "valide"

Ma allora, definito un opportuno alfabeto $\mathcal{A}$

- Istanze e soluzioni sono parole
  (e così pure ogni coppia istanza-soluzione)
- Un problema è un linguaggio

Quale alfabeto?

Gli alfabeti possibili sono infiniti. Spiccano per la loro semplicità

- *alfabeto unario*: $\mathcal{A} = \{0\} \Rightarrow \mathcal{A}^* = \{\cdot, 0, 00, 000, 0000, \ldots\}$

L'informazione associata a ogni stringa in alfabeto unario (il "significato") è codificata nella sua lunghezza

- *alfabeto binario*: $\mathcal{A} = \{0, 1\}$
  $\Rightarrow \mathcal{A}^* = \{\cdot, 0, 1, 00, 01, 10, \ldots\}$

I simboli dell'alfabeto binario si definiscono bits (*binary digits*)

Traduzioni (1)

La scelta dell'alfabeto $\mathcal{A}$ è puramente convenzionale

Dati due alfabeti $\mathcal{A}$ e $\mathcal{A}'$ con pari numero di simboli ($|\mathcal{A}| = |\mathcal{A}'|$), si può tradurre meccanicamente ogni stringa di un linguaggio su $\mathcal{A}$ in una stringa di un linguaggio equivalente su $\mathcal{A}'$

1. stabilendo per convenzione una corrispondenza biunivoca fra i simboli dei due alfabeti
2. sostituendo ogni simbolo di $\mathcal{A}$ col corrispondente di $\mathcal{A}'$

$$\mathcal{A} \leftrightarrow \mathcal{A}' \Rightarrow \mathcal{L}_{\mathcal{A}} \leftrightarrow \mathcal{L}'_{\mathcal{A}'}$$

Traduzioni (2)

Se i due alfabeti hanno diversa dimensione ($|\mathcal{A}| > |\mathcal{A}'|$) ogni simbolo di $\mathcal{A}$ corrisponde a una sottostringa di simboli di $\mathcal{A}'$

Le parole di $\mathcal{A}'$ sono più lunghe delle corrispondenti parole di $\mathcal{A}$

È un fenomeno critico?

- Se $|\mathcal{A}'| \geq 2$, ogni stringa $s \in \mathcal{L}_\mathcal{A}$ si traduce in una stringa $s' \in \mathcal{L}'_{\mathcal{A}'}$ con $|s'| = \lceil \lg_{|\mathcal{A}'|} |\mathcal{A}| \rceil \, |s|$
- Se $|\mathcal{A}'| = 1$, ogni stringa $s \in \mathcal{L}_\mathcal{A}$ si traduce in una stringa

  $$s' \in \mathcal{L}'_{\mathcal{A}'} \text{ con } \sum_{i=1}^{|s|} |\mathcal{A}|^i < |s'| \leq \sum_{i=1}^{|s|} |\mathcal{A}|^i \Rightarrow |s'| \approx |\mathcal{A}|^{|s|}$$

Traduzioni (3)

- Traducendo in un alfabeto almeno binario, la dimensione di una stringa si moltiplica per un fattore costante (logaritmico rispetto all'alfabeto)
- Traducendo nell'alfabeto unario, la dimensione di una stringa cresce esponenzialmente

L'alfabeto binario combina semplicità e compattezza, per cui viene preferito agli altri

Avremo però altro da dire sull'alfabeto unario…

Tipi di problemi

Si classificano i problemi secondo la natura della stringa soluzione

- *problemi di decisione* (*esistenza*): S è *Sì* o *No* (*Vero* o *Falso*)
- *problemi di ricerca*: S è la descrizione quantitativa completa di un sottosistema che soddisfa certe condizioni
- *problemi di conteggio*: S è il numero dei sottosistemi che soddisfano certe condizioni
- *problemi di ottimizzazione*: S è il valore minimo o massimo di una funzione obiettivo definita sui sottosistemi che soddisfano certe condizioni
- *problemi di enumerazione*: S è l'insieme dei sottosistemi che soddisfano certe condizioni

Linguaggi e problemi di decisione (1)

Nei problemi di decisione, la soluzione è un singolo bit

Problemi di decisione e linguaggi formali sono in corrispondenza biunivoca

- Dato un problema di decisione P sull'alfabeto $\mathcal{A}$ il linguaggio associato $\mathcal{L}(P)$ è

$$\mathcal{L}(P) = \{X \in \mathcal{A}^* : (X, \textit{Sì}) \in P\}$$

- Dato un linguaggio $\mathcal{L}$ sull'alfabeto $\mathcal{A}$ il problema di decisione associato è

$$P(\mathcal{L}) = \{(X, \textit{Sì}) : X \in \mathcal{L}\} \cup \{(X, \textit{No}) : X \in \mathcal{A}^* \setminus \mathcal{L}\}$$

Linguaggi e problemi di decisione (2)

Dato un problema P, il linguaggio associato $\mathcal{L}(P)$ è l'insieme delle sue istanze positive (quelle con soluzione Sì)

Viceversa, dato un linguaggio $\mathcal{L}$, il problema associato $P(\mathcal{L})$ consiste nel chiedersi "La stringa X è una parola di $\mathcal{L}$?"

Di conseguenza

Risolvendo un'istanza di P, si riconosce se la stringa corrispondente appartiene a $\mathcal{L}(P)$

Riconoscendo se una stringa appartiene a $\mathcal{L}$, si risolve l'istanza corrispondente di $P(\mathcal{L})$

Che cos'è un problema?

Un problema è un insieme di stringhe : *qualsiasi insieme di stringhe?*

Alcuni linguaggi sono problemi più interessanti di altri,
ma è arbitrario indicare quali istanze definiscano un problema

Esempio: lo *Shortest Path Problem* si può definire su

- l'insieme dei grafi orientati
- l'insieme dei grafi orientati privi di circuiti negativi
- l'insieme dei grafi orientati con costi non negativi
- l'insieme dei grafi orientati privi di circuiti

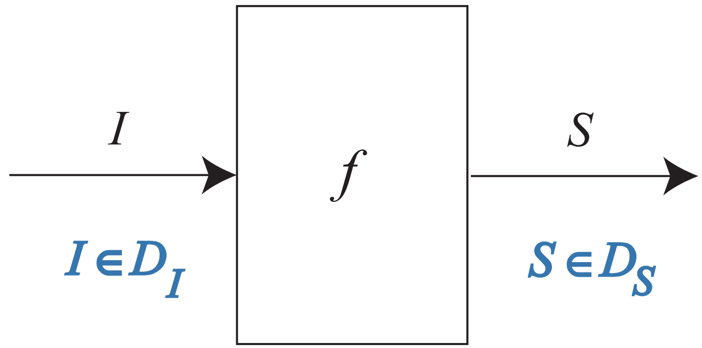
Sono insiemi di stringhe diversi e problemi diversi
(anche se sovrapposti e dotati di una "grammatica" comune)

In sintesi

La risoluzione meccanica di problemi ha forti legami con la linguistica formale

- problemi di decisione e linguaggi si corrispondono
- tutto l'armamentario teorico dei linguaggi formali si può trasferire alla risoluzione di problemi
- la codifica in stringhe permette di comunicare istanze a una macchina che riconosce linguaggi
- si può risolvere un'istanza con gli stessi metodi con cui si riconosce se una parola fa parte di un linguaggio

Problemi e funzioni (1)



Istanza: stringa I di simboli tratti da un alfabeto $\mathcal{A}$

- Codifica i dati e appartiene a un dominio D_I

Soluzione: stringa S di simboli tratti da un alfabeto $\mathcal{A}$

- Codifica i risultati e appartiene a un dominio D_S

Problema: corrispondenza f fra istanza I e soluzione S

$$f : D_I \rightarrow D_S$$

Quindi una funzione matematica!

Problemi e funzioni (2)

Dato un alfabeto composto di cifre,

istanze e soluzioni sono numeri naturali

$\Rightarrow$ corrispondenza biunivoca fra problemi e funzioni $\mathbb{N} \to \mathbb{N}$

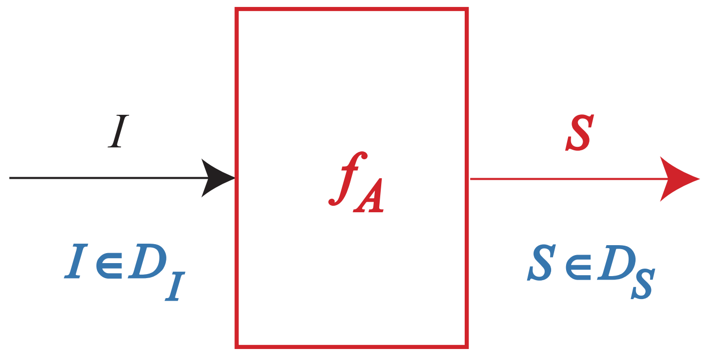
- dato l'argomento n, il valore $f(n)$ è soluzione del problema
- le coppie (I, S) di un problema definiscono una funzione $f : \mathbb{N} \to \mathbb{N}$

(in genere non sappiamo come calcolarla!)

Risolvere problemi e computare funzioni si equivalgono

Algoritmi

Algoritmo: una sequenza finita di operazioni elementari che trasforma ciascuna stringa I di un opportuno dominio in una stringa S di un altro dominio (è un manipolatore di stringhe)



Algoritmo risolvente (un problema dato): un algoritmo che trasforma ogni istanza di un problema nella soluzione corrispondente (cioè computa la funzione che le lega)

$$f_A(\cdot) \equiv f(\cdot) \Leftrightarrow f_A(I) = f(I) \, \forall I \in D_I$$

Algoritmi e problemi (1)

Ma allora sono la stessa cosa? No!

- Problema è una funzione
- Algoritmo è una funzione più il modo di computarla

La relazione fra problemi e algoritmi è intricata

- In genere un problema ha diversi algoritmi risolutivi

$A1: \quad 7 + 4n + 2n^2 + n^3$ (3 somme e 5 prodotti)

$A2: \quad 7 + n[4 + n(2 + n)]$ (3 somme e 2 prodotti)

$A1 \neq A2$ ma $f_{A1}(\cdot) \equiv f_{A2}(\cdot)$

Algoritmi e problemi (2)

- Un algoritmo risolve tutti i problemi contenuti nel suo problema associato (quindi risolve diversi problemi: disgiunti, intersecantisi, inclusi uno nell'altro)

Esempio: l'algoritmo del simplesso risolve mix produttivo, percorso minimo, flusso massimo...

- Esistono problemi per i quali non vi sono algoritmi!
- Esistono algoritmi non corretti, ma utili

Esempio: Algoritmi di stima per eccesso o per difetto (euristiche)
Algoritmi probabilistici (ad es., per la primalità)