Azzolini Riccardo 2019-04-15

Algebre eterogenee e tabelle hash

1 Algebra eterogenea

Un'**algebra eterogenea** $A = \langle [A_1, \ldots, a_n], [f_1, \ldots, f_k] \rangle$ è costituita da:

- gli insiemi A_i;
- le funzioni f_j, che rappresentano operazioni sugli elementi degli insiemi A_i:

$$f_j : A_{c_{j1}} \times \cdots \times A_{c_{jr_j}} \to A_{i_j}, \quad c_{pq}, i_j \in \{1, \ldots, n\}$$

  dove r_j è il numero di argomenti di f_j.

Le algebre eterogenee formalizzano la nozione

$$\text{struttura dati} = \text{insiemi} + \text{operazioni}$$

2 Parti di A e parti di A ordinato

Dato un insieme A, si definisce l'algebra eterogenea **parti di A**,

$$\text{PA} = \langle [A, 2^A, \text{Boolean}], [\text{Member}, \text{Insert}, \text{Delete}] \rangle$$

dove:

- 2^A è l'*insieme delle parti* di A, cioè l'insieme di tutti i sottoinsiemi di A;
- Member : $A \times 2^A \to$ Boolean determina se un elemento di A (il primo argomento) è contenuto in un particolare sottoinsieme di A (il secondo argomento);
- Insert : $A \times 2^A \to 2^A$ aggiunge un elemento a un sottoinsieme, restituendo il sottoinsieme risultante.
- Delete : $A \times 2^A \to 2^A$ elimina un elemento dal sottoinsieme specificato, e restituisce il nuovo sottoinsieme.

Se l'insieme A è totalmente ordinato, si può definire anche l'algebra eterogenea **parti di A ordinato**:

$$\text{PAO} = \langle [A, 2^A, \text{Boolean}],\ [\text{Member}, \text{Insert}, \text{Delete}, \text{Min}] \rangle$$

- $\text{Min} : 2^A \to A$ restituisce l'elemento minimo del sottoinsieme specificato;
- le altre operazioni sono definite come nell'algebra PA.

3 Implementazione di PA

L'algebra eterogenea *parti di A* può essere implementata mediante:

- *liste concatenate*: sono semplici, ma il costo medio di ciascuna operazione è $O(n)$;
- **tabelle hash**: il costo medio di ogni operazione è $O(1)$, ma degenera a $O(n)$ se si sbaglia a impostare alcuni parametri (quindi questa soluzione non è adatta se servono prestazioni garantite), e l'implementazione è più complessa.

4 Tabella

Una **tabella** è una struttura dati elementare, composta da:

- un vettore di dimensione n;
- un intero, che indica quanti dati sono contenuti nella tabella (al massimo n).

5 Tipi di tabelle hash

Le tabelle hash si distinguono in base alla dimensione della tabella

- fissa: **hash statico**;
- variabile dinamicamente nel tempo (mediante la creazione di nuovi vettori): **hash dinamico**;

e a dove sono memorizzati i dati

- in delle liste concatenate, alle quali si accede mediante riferimenti contenuti nella tabella: **concatenazioni separate** (**separate chaining**);
- direttamente nella tabella: **indirizzamento aperto** (**open addressing**).

Ci sono quindi 4 casi possibili.

6 Principio delle tabelle hash

- Ogni dato è contraddistinto da una **chiave** $c \in U$ (che è una porzione del dato: ad esempio, per la scheda anagrafica di uno studente, la chiave potrebbe essere il numero di matricola).
- Ogni posizione in una tabella di dimensione M è identificata da un indirizzo $i \in \{0, \ldots, M-1\}$.
- Il numero di possibili chiavi è solitamente molto elevato, $k = \Theta(2^{|c|})$, dove $|c|$ è la lunghezza, cioè il numero di bit, della chiave. Di conseguenza, $M \ll k$.
- La posizione di un dato nella tabella viene determinata mediante una **funzione di hash** H, che trasforma le chiavi in indirizzi:

$$H : U \to \{0, \ldots, M-1\}$$

7 Funzioni di hash

Per essere accettabile, una funzione di hash deve avere le seguenti caratteristiche:

1. deve essere facile da calcolare, cioè avere costo $O(|c|)$;
2. deve distribuire uniformemente le chiavi su M indirizzi, ovvero

$$\forall c \in U,\ i \in \{0, \ldots, M-1\} \quad P(H(c) = i) = \frac{1}{M}$$

3. l'indirizzo calcolato deve dipendere da tutti i bit della chiave c.

7.1 Esempi di funzioni non accettabili

- La funzione identità $H(c) = c$ soddisfa 1, 2 e 3, ma sarebbe utilizzabile solo se $M = k$ (mentre in realtà si ha solitamente $M \ll k$).
- La funzione costante $H(c) = r$ soddisfa 1, ma non 2 e 3.