Azzolini Riccardo 2019-03-05

Strutture dati elementari

1 Vettori

Un **vettore** di *lunghezza* $n \in \mathbb{N}$ e *tipo base* $\mathcal{U}$ è un elemento di $\mathcal{U}^n$.

Le operazioni definite sui vettori sono:

- **proiezione** $\pi : \mathcal{U}^n \times \mathbb{N} \to \mathcal{U} \cup \{\bot\}$

$$\pi(A, i) = \begin{cases} a_i & \text{se } A = (a_1, \ldots, a_n),\ 1 \leq i \leq n \\ \bot & \text{altrimenti} \end{cases}$$

- **sostituzione** $\sigma : \mathcal{U}^n \times \mathbb{N} \times \mathcal{U} \to \mathcal{U}^n \cup \{\bot\}$

$$\sigma(A, i, a) = \begin{cases} (a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_n) & \text{se } A = (a_1, \ldots, a_n),\ 1 \leq i \leq n \\ \bot & \text{altrimenti} \end{cases}$$

In molti linguaggi, queste operazioni si effettuano mediante l'operatore *mix-fisso* (non è prefisso, postfisso o infisso, ma un misto) di accesso: `x = A[i]` (proiezione), `A[i] = z` (sostituzione).

1.1 Implementazione

Se un singolo oggetto di tipo $\mathcal{U}$ richiede k registri RAM (o byte, su una macchina reale) allora $A \in \mathcal{U}^n$ viene memorizzato in kn registri (byte) *consecutivi*.

In questo modo, noti l'indirizzo di base α del vettore e l'indice i, è possibile calcolare (in fase di esecuzione) l'indirizzo di $A[i]$ in tempo $O(1)$ (in base al CCU): il primo registro corrispondente al dato $A[i]$ è

- $R_{\alpha+k(i-1)}$ se gli indici iniziano da 1;
- $R_{\alpha+ki}$ se gli indici partono da 0 (in questo modo si evita una sottrazione).

La formula $\alpha+k(i-1)$ (o $\alpha+ki$) è chiamata **mappa di memorizzazione** del vettore.

2 Matrici

Una **matrice** di *ordine* $m \times n$, con $m, n \in \mathbb{N}$, e tipo base $\mathcal{U}$ è un elemento di $\mathcal{U}^{[m \times n]}$. In pratica, una matrice è un vettore bidimensionale.

Sulle matrici sono definite le stesse operazioni esistenti per i vettori:

- **proiezione** $\pi : \mathcal{U}^{[m \times n]} \times \mathbb{N} \times \mathbb{N} \to \mathcal{U} \cup \{\perp\}$

$$\pi(A, i, j) = \begin{cases} a_{ij} & \text{se } 1 \leq i \leq m,\ 1 \leq j \leq n \\ \perp & \text{altrimenti} \end{cases}$$

- **sostituzione** $\sigma : \mathcal{U}^{[m \times n]} \times \mathbb{N} \times \mathbb{N} \times \mathcal{U} \to \mathcal{U}^{[m \times n]} \cup \{\perp\}$

$$\sigma(A, i, j, a) = \begin{cases} B & \text{se } 1 \leq i \leq m,\ 1 \leq j \leq n, \\ & \text{dove } b_{pq} = a_{pq},\ p \neq i \vee q \neq j,\ b_{ij} = a \\ \perp & \text{altrimenti} \end{cases}$$

In molti linguaggi di programmazione, queste operazioni si effettuano con la sintassi:

- `x = A[i, j]` o `x = A[i][j]` (proiezione);
- `A[i, j] = z` o `A[i][j] = z` (sostituzione).

2.1 Implementazione

Se un singolo oggetto di tipo $\mathcal{U}$ richiede k registri RAM (o byte) allora $A \in \mathcal{U}^{[m \times n]}$ viene memorizzato in kmn registri (byte) *consecutivi*. A tale scopo, è necessario *linearizzare* la matrice, cioè scegliere l'ordine in cui memorizzare gli elementi:

- *memorizzazione per righe* (la più comune): si memorizzano tutti gli elementi della prima riga, poi tutti quelli della seconda, ecc.;
- *memorizzazione per colonne*: si memorizza la prima colonna, poi la seconda, ecc.

Noti l'indirizzo di base α e gli indici i, j, l'indirizzo α_{ij} del primo registro corrispondente a $A[i, j]$ si calcola in tempo $O(1)$, mediante una delle possibili mappe di memorizzazione:

Primo indice	Per righe	Per colonne
0	$\alpha_{ij} = \alpha + ink + jk$	$\alpha_{ij} = \alpha + jmk + ik$
1	$\alpha_{ij} = \alpha + (i-1)nk + (j-1)k$	$\alpha_{ij} = \alpha + (j-1)mk + (i-1)k$

Questa stessa implementazione può essere estesa a più di due dimensioni: una volta fissato il numero di dimensioni, il tempo di accesso rimane $O(1)$.

3 Record

Un **record** è costituito da campi eterogenei. Esso è caratterizzato da

- il numero di campi: n;
- i tipi dei campi: $\mathcal{U}_1, \ldots, \mathcal{U}_n$;
- il numero k_i di registri (o byte) occupati da un elemento del tipo $\mathcal{U}_i$, e quindi necessari per l'i-esimo campo;
- l'etichetta e_i dell'i-esimo campo;
- l'indirizzo di base α.

Un record R è quindi un elemento di $U_1 \times \cdots \times U_n$.

L'indirizzo del (primo registro/byte corrispondente al) campo $R.e_i$ è

$$\alpha + \sum_{j=1}^{i-1} k_j$$

che viene calcolato in fase di compilazione, quindi il tempo di accesso in esecuzione è $O(1)$. In compenso, proprio perché i calcoli sono effettuati in compilazione, non è possibile utilizzare una variabile come indice per accedere a un record.