Interleaving

Thursday, 16 November 2023 11:47

- Abbiamo 3 sequenza di simboli, e la 3 deve essere tanto lunga come la somma delle prime 2 sequenze
  X = <**C**, **I**, **A**, **O**>
  Y = <M, A, M, M, A>
  W = <**C**, **I**, M, A, A, M, M, **A**, **O**>
  Dobbiamo stabilire se la terza sequenza è un interveaving delle prime due
    Aka sia X che Y sono sottosequenza di W
    Ed in più sono sottosequenza disgiunte, cioè non usiamo gli stessi simboli
  Per comprenderlo meglio, esprimiamolo in forma insiemistica:
    $|X| = m$
    $|Y| = n$
    $|W| = |X| + |Y| = m + n$
    Stabiliamo: i indice per X, j indice per Y
      $\exists\{i_1, \dots, i_m\} \subseteq \{1, \dots, m+n\}, i_1 < i_2 < \cdots < i_m$
      $\exists\{j_1, \dots, j_n\} \subseteq \{1, \dots, m+n\}, j_1 < j_2 < \cdots < j_n$
    Qui abbiamo detto che, deve esistere i e j incluso strettamente un insieme lungo m+n e che i e j devono essere in ordine crescente di indice.
      $t.c.$
      $\{i_1, \dots, i_m\} \cap \{j_1, \dots, j_n\} = \epsilon$
      $\{i_1, \dots, i_m\} \cup \{j_1, \dots, j_n\} = \{1, \dots, m+n\}$
    Qui che l'intersezione tra i e j deve essere nulla e che l'unione deve essere M
    $\forall k \in \{1, \dots, m\}, x_k = w_{ik}$
    $\forall h \in \{1, \dots, n\}, y_h = w_{jh}$
    E che per ogni valore di x e j ci deve essere il corrispondente in m (chiamato w)
  Nota: CIAO e MAMMA in W devono essere ordinati
- Sottoproblemi
    Già notiamo che il problema richiede 2 indici: (i, j)
      Definiamo $S_{ij} = True$ se $Interleaving(W_{i+j}, X_i, Y_j)$
- Equazione di ricorrenza
  - Caso base: i=0 v j=0
    - $i = 0$ ^ $i = 0$

Qui sia X che Y sono vuoti, quindi W per definizione deve essere vuoto
Quindi $S_{ij} = true$

- $i = 0 \wedge j > 0$
  Qui X non è vuoto ma Y lo è, quindi dobbiamo controllare $Interleaving(W_j, \epsilon, Y_j)$
  - $w_j = y_j \rightarrow Int(W_j, \epsilon, Y_j) = Int(W_{j-1}, \epsilon, Y_{j-1})$
    Quindi $s_{ij} = S_{i,j-1}$
    Nota: i=0
  - $w_j \neq y_j$
    Allora $s_{ij} = false$
- $i > 0 \wedge j = 0$
  Stessa cosa di sopra ripetuta

- Passo ricorsivo: $i > 0, j > 0$
  - $w_{i+j} \neq x_i \wedge w_{i+j} \neq y_j$
    Qui semplicmeente
    $s_{ij} = False$
  - $w_{i+j} = x_i \wedge w_{i+j} \neq y_j$
    Quindi abbiamo trovato una x compatibile
    Quindi continuamo con x (decrementando i)
    $s_{ij} = s_{i-1,j}$
    $Int(W_{i+j}) = Int(W_{i+j-1})$
  - $w_{i+j} \neq x_i \wedge w_{i+j} = y_j$
    Stessa cosa detta sopra
  - $w_{i+j} = x_i \wedge w_{i+j} = y_j$
    Qui abbiamo 2 casi: prendiamo x, prendiamo y
    Quindi facciamo il massimo.
      Però siccome stiamo facendo il massimo tra valori booleani
      Dobbiamo usare il massimo dei booleani: OR
    $s_{i,j} = s_{i-1,j} \vee s_{i,j-1}$

Uniamo tutto:

$$s_{ij} = \begin{cases} True & i = 0 \wedge j = 0 \\ s_{i,j-1} & i = 0 \wedge j > 0 \wedge w_j = y_j \\ False & i = 0 \wedge j > 0 \wedge w_j \neq y_j \\ s_{i-1,j} & i > 0 \wedge j = 0 \wedge w_i = x_i \\ False & i > 0 \wedge j = 0 \wedge w_i \neq x_i \\ s_{i,j-1} & i = 0 \wedge j > 0 \wedge w_j = y_j \\ s_{i-1,j} & i > 0 \wedge j = 0 \wedge w_i = x_i \\ s_{i-1,j} & w_{i+j} = x_i \wedge w_{i+j} \neq y_j \end{cases}$$

$$
\begin{cases}
s_{i-1,j} & w_{i+j} = x_i \wedge w_{i+1} \neq y_j \\
s_{i,j-1} & w_{i+1} \neq x_i \wedge w_{i+j} = y_j \\
s_{i-1,j} \ v \ s_{1-j-1} & else
\end{cases}
$$

Si.... Quante belle casistiche (ops ne ho duplicata qualcuna)

- Pseudocodice
  - Ricorsivo

```
Def INTric(i, j):
        If i=0 ^ j=0:
                Return True
        If i=0 ^ j>0 ^ wj=yj:
                Return INTric(i, j-1)
        If i=0 ^ j>0 ^ wj!=yj:
                Return False
        If i>0 ^ j=0 ^ wi=xi:
                Return INTric(i-1, j)
        If i>0 ^ j=0 ^ wi!=xi:
                Return False
        If w[i+j] = xi ^ w[i+j] != yj:
                Return INTric(i-1, j)
        If w[i+j] != xi ^ w[i+j] = yj:
                Return INTric(i, j-1)
        Return INTric(i-1, j) v INTric(i, j-1)
```

  - Iterativo

```
Def INT(X, Y, W):
        S[0, 0] = True
        For i=1 to m:
                If wi = xi:
                        S[i, 0] = s[i-1, 0]
                Else:
                        S[i, 0] = False
        For j=1 to m:
                If wij= xj:
                        S[0, j] = s[0, j-1]
                Else:
                        S[0, j] = False
        For i=1 to m:
                For j=1 to n:
                        If w[i+j]! Xi ^ w[i+j] != yj:
                                S[i, j] = False
                        If W[i+j] =xi ^ w[i+j] != yj:
                                S[i, j] = S[i-1, j]
                        If W[i+k]!=xi ^ w[i+j] = yj:
```