Corso di Programmazione Concorrente

Regioni Critiche

Valter Crescenzi
crescenz@dia.uniroma3.it
http://crescenzi.inf.uniroma3.it

Sommario

- Limiti dei Semafori
- Le Regioni Critiche
  - motivazioni
  - come strumento di competizione
  - sintassi & semantica
- Le Regioni Critiche Condizionali
  - come strumento di cooperazione
  - clausole condizionali
  - sintassi & semantica
  - es. 5 filosofi
- I limiti delle Regioni Critiche
- Oltre le Regioni Critiche

Caratteristiche, Vantaggi e Svantaggi dei Semafori

Dalle caratteristiche dei semafori:

- primitivi
- potenti
- flessibili

...seguono i principali...

- Vantaggi
  - espressivi: permettono di risolvere qualsiasi problema di sincronizzazione
- Svantaggi
  - non sono strutturati
  - difficili da testare
  - richiedono l'utilizzo di variabili comuni

…e nasce l'esigenza di strumenti a più alto livello di astrazione

Regioni Critiche (1)

- Nascono dall'esigenza di strumenti di sincronizzazione di più alto livello rispetto ai semafori

– Non hanno una controparte diretta tra i linguaggi di programmazione moderni più diffusi. Tuttavia:

✔ Rappresentano un passo intermedio prima dei *monitor* che sono il meccanismo alla base dei costrutti di sincronizzazione nei linguaggi moderni

✔ Consentono di motivare più facilmente le scelte progettuali alla base della semantica dei monitor

Regioni Critiche (2)

- Sintassi di base, come strumento per risolvere problemi di competizione
- Sia R una variabile condivisa appositamente dichiarata con questa sintassi

  var R**: shared** *T***;**

  ed associata ad una risorsa seriale R di tipo *T*, allora il costrutto di regione critica assume una forma del tipo

  region R **do**

  <lista istruzioni> // sezione critica con uso di R

  end region
- Ovviamente la semantica è quella di garantire l'indivisibilità della sezione critica
- N.B. Ci possono essere diverse occorrenze del costrutto sulla stessa risorsa condivisa sparse nel codice

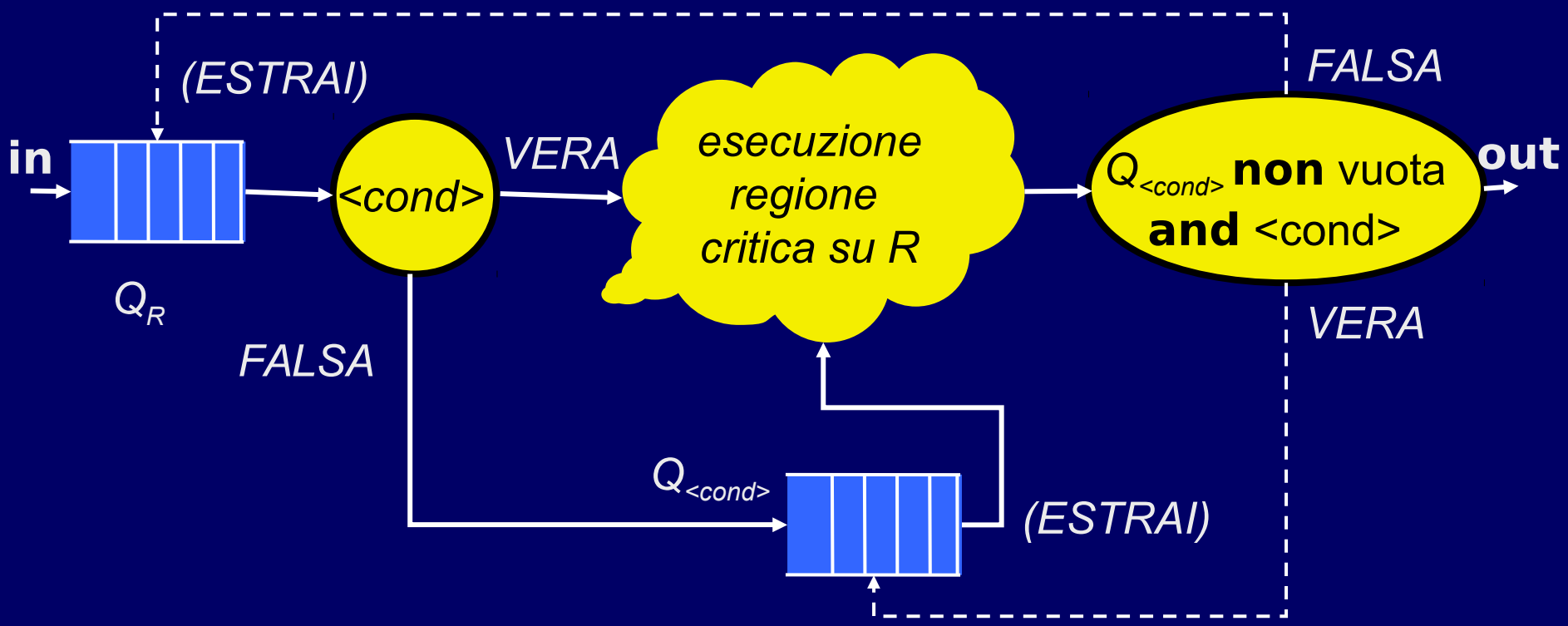
Implementazione delle Regioni Critiche

var R**: shared** *T***;**
region R **do**
<*lista istruzioni*> // sezione critica su R
end region

- L'implementazione prevede
  - un semaforo binario S_R associato alla variabile condivisa R
  - **P**(S_R) prima di ogni sezione critica su R
  - **V**(S_R) dopo ogni sezione critica su R

Regioni Critiche Condizionali

var R**: shared** *T***;**
region R **do**
when <*cond*> **do** <*lista istruzioni*>
end region



Regioni Critiche Condizionali: Semantica (1)

- Il f.d.e. appena entrato in sezione critica valuta una condizione (che coinvolge la variabile condivisa)
  - Se *vera*, si eseguono le istruzioni dopo il **do**
  - Se *falsa*, si esce dalla sezione critica
    - il f.d.e. rimane in attesa passiva in una coda $Q_{<cond>}$ associata alla condizione specificata
    - uscendo dalla sez. critica si dà modo ad altri f.d.e. di agire in sezione critica sulla variabile condivisa, cambiare lo stato della risorsa ed eventualmente rendere vera la condizione
- La condizione è *rivalutata* dopo l'esecuzione di ogni regione critica sulla stessa variabile e se soddisfatta, viene risvegliato il primo dei f.d.e. in attesa dentro la coda associata $Q_{<cond>}$

Regioni Critiche Condizionali: Semantica (2)

- Possono esistere diverse *condizioni* per ciascuna regione critica
- Ciascuna è *rivalutata* dopo l'esecuzione di ogni regione critica sulla stessa variabile e se soddisfatta, viene risvegliato il primo dei f.d.e. in attesa nella coda associata $Q_{<cond>}$
- Bisogna pertanto scegliere quale f.d.e. far entrare in sezione critica tra:
  - tutti quelli che si trovano in cima alle code $Q_{<cond>}$ per le quali la condizione <cond> risulta *vera*
  - i nuovi flussi in entrata in Q_R
- Semantica adottata: scelta *non-deterministica*

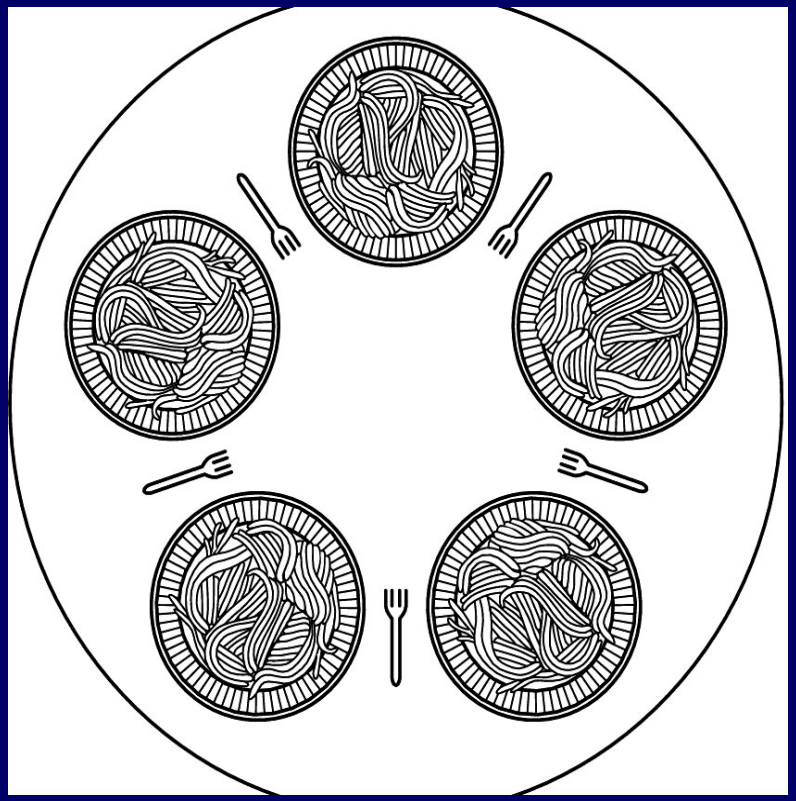
  Tuttavia altre scelte sono possibili e plausibili

  Cfr. monitor >>
- Risoluzione di un problema classico...

I Cinque Filosofi Mangiatori

Problema classico originariamente proposto da E.W. Dijkstra.
Cinque filosofi trascorrono la vita alternando, ciascuno indipendentemente dagli altri, periodi in cui pensano a periodi in cui mangiano degli spaghetti.

Per raccogliere gli spaghetti ogni filosofo necessita delle due forchette poste rispet-tivamente a destra ed a sinistra del proprio piatto.
Trovare una strategia che consenta ad ogni filosofo di ottenere sempre le due forchette richieste.



Pericolo di Stallo con i Filosofi

- Bisogna escludere una situazione di stallo che si può facilmente originare: tutti i filosofi possiedono esattamente la forchetta a destra (rispett. a sinistra) senza poter ottenere quella sinistra (destra)
- Per questo prevediamo che ogni filosofo segua un protocollo di questo tipo:

  loop *<<pensa>>;*

  <<impossessati delle due forchette>>;

  <<mangia>>;

  <<rilascia le due forchette>>;

  end
- le forchette sono modellate con una variabile condivisa

  var LIBERA**: shared array**[0..4] **of** *boolean***;**

CINQUE_FILOSOFI_MANGIATORI

```
concurrent program
   CINQUE_FILOSOFI_MANGIATORI;
type filosofo = concurrent procedure (I: 0..4);
   begin /* … */  end
var A, B, C, D, E: filosofo;
     J: 0..4;
     var LIBERA: shared array[0..4] of  boolean;
begin
   for J ← 0 to 4 do LIBERA[J] ← true;
   cobegin A(0) || B(1) || C(2) || D(3) || E(4) coend
end
```

type *filosofo*

```
concurrent program CINQUE_FILOSOFI_MANGIATORI;
type   filosofo = concurrent procedure (I: 0..4);
    begin loop
        <<pensa>>;
        region LIBERA
            when LIBERA[I] and LIBERA[(I+1) mod 5]      do
                LIBERA[I] ← false;
                LIBERA[(I+1) mod 5] ← false;
        end region;
        <<mangia>>;
        region LIBERA
                LIBERA[I] ← true;
                LIBERA[(I+1) mod 5] ← true;
        end region;
    end
end; /* ... */
```

Commenti a CINQUE_FILOSOFI_MANGIATORI

- Lo stallo viene evitato imponendo l'acquisizione contemporanea delle due forchette
- Quando un filosofo scopre che almeno una delle due forchette di cui necessita è occupata, abbandona la regione critica condizionale ed aspetta che uno dei vicini finisca di mangiare
- La soluzione proposta non gode della proprietà di fairness: non esiste nessuna garanzia esplicita contro la possibilità che un filosofo muoia di fame per colpa dei propri vicini di tavola che si alternano nel rubargli sistematicamente le forchette attigue

Limiti delle Regioni Critiche Condizionali

Le regioni critiche, sebbene riescano nello scopo di aumentare il livello di astrazione rispetto ai semafori, hanno due principali limitazioni

- efficienza: per la necessità di rivalutare le condizioni all'uscita di ogni regione critica
- coesione: le regioni critiche sulla stessa variabile condivisa possono essere sparse in tutto il programma;
  - ✔ per conoscere e comprendere tutti i modi in cui viene utilizzata può essere necessario esaminare grandi porzioni di codice

Esercizi

Esercizio: Risolvere il problema Produttori / Consumatori usando le regioni critiche condizionali

Esercizio: Risolvere il problema del barbiere dormiente usando le regioni critiche condizionali

Esercizio: Risolvere il problema dei cinque filosofi pensatori con le regioni critiche condizionali garantendo assenza di stallo, massimo parallelismo e fairness.

Esercizio: Tradurre la prima soluzione proposta al problema Produttori / Consumatori (basata su semafori di alto livello PIENE, VUOTE, USO_T, USO_D) riottenendo lo stesso livello di parallelismo ma utilizzando le regioni critiche condizionali.

Superare i Difetti delle Regioni Critiche

- Le diverse limitazioni delle regioni critiche condizionali sono di natura molto diversa tra loro
  - *limiti nell'efficienza*: sono legati alla rivalutazione delle condizioni all'uscita di ogni sezione critica, e questo spesso porta a valutare inutilmente condizioni ancora false
    - possibile soluzione: alcuni f.d.e. sanno quando una condizione attesa da altri f.d.e. diventa vera; si dovrebbe cercare metterli in comunicazione con i f.d.e. interessati
  - *limiti nella coesione*: sono legati più al paradigma di programmazione usato che non al costrutto di regione critica
    - possibile soluzione: utilizzare paradigmi (come quello OO) che aumentino la coesione tra le risorse e le operazioni per manipolarle