ANNO ACCADEMICO 2024/2025

Modelli Concorrenti e Algoritmi Distribuiti

Esercizi

Altair's Notes

UNIVERSITÀ DI TORINO

DIPARTIMENTO DI INFORMATICA

Premessa

Licenza

Formato utilizzato

Box di "Concetto sbagliato":

Concetto sbagliato 0.1: Testo del concetto sbagliato

Testo contente il concetto giusto.

Box di "Corollario":

Corollario 0.0.1 Nome del corollario

Testo del corollario. Per corollario si intende una definizione minore, legata a un'altra definizione.

Box di "Definizione":

Definizione 0.0.1: Nome delle definizione

Testo della definizione.

Box di "Domanda":

Domanda 0.1

Testo della domanda. Le domande sono spesso utilizzate per far riflettere sulle definizioni o sui concetti.

Box di "Esempio":

Esempio 0.0.1 (Nome dell'esempio)

Testo dell'esempio. Gli esempi sono tratti dalle slides del corso.

Box di "Note":

Note:-

Testo della nota. Le note sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive.

Box di "Osservazioni":

Osservazioni 0.0.1

Testo delle osservazioni. Le osservazioni sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive. A differenza delle note le osservazioni sono più specifiche.

1

Introduzione alla Programmazione Concorrente

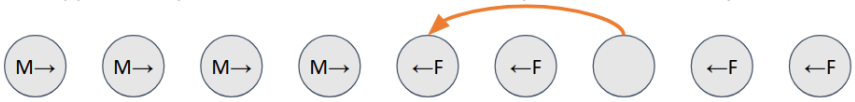
1.1 Salto dei ranocchi

Esercizio 1

Salto dei ranocchi

Ci sono ***2n+1*** pietre allineate. Sulle ***n*** pietre più a sinistra stanno ***N*** ranocchi maschi rivolti a destra, e sulle ***n*** pietre più a destra stanno ***N*** ranocchie femmine rivolte a sinistra. La pietra centrale è vuota.

I ranocchi si spostano nella direzione verso cui sono rivolti saltando nella pietra adiacente, se è vuota, oppure oltrepassando una rana, nella seconda pietra adiacente se questa è vuota.



Lo stato finale è dato dalla configurazione:

Si consideri il diagramma degli stati per n = 1 e per n = 2 e, per entrambi i casi, si risponda alle seguenti domande:

a. esiste una computazione che, partendo dallo stato iniziale porti allo stato finale?

b. tutte le computazioni raggiungono lo stato finale?

c. esistono computazioni che non terminano, cioè non raggiungono uno stato in cui non si può eseguire nessuna mossa?

Soluzione Esercizio 1

Diagramma degli stati del salto dei ranocchi:

- Il numero massimo di stati possibili è dato da:

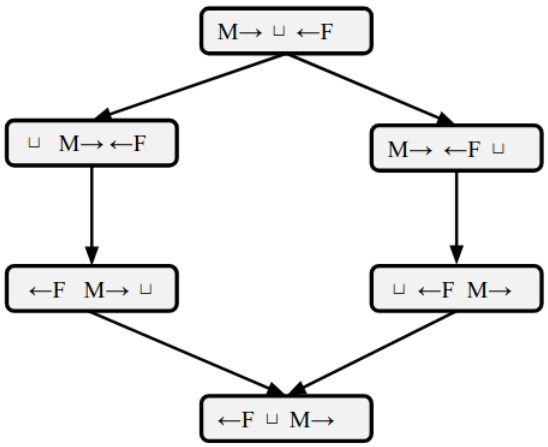
Tutte le possibili permutazioni di (2n + 1) elementi

$$\frac{(2n+1)!}{n! \times n!}$$

Siccome le rane M ed F sono indistinguibili, non importa l'ordine nelle permutazioni. Quindi dividiamo per n! permutazioni delle rane M, e per altri n! per le rane F.

- quindi per n=1, il numero di stati è 6, mentre per n=2 è 30.
- Nel seguito è mostrato, per n=1 l'intero diagramma degli stati, mentre per n=2 è mostrato solo la parte del diagramma con le mosse che si possono presentare per una computazione che parte con una mossa maschile, l'altra parte del diagramma è simmetrica.

Diagramma degli stati del salto dei ranocchi
n = 1



Esistono due sole computazioni e portano entrambe allo stato finale.

Non esistono computazioni che non terminano.

Diagramma degli stati del salto dei ranocchi n=2

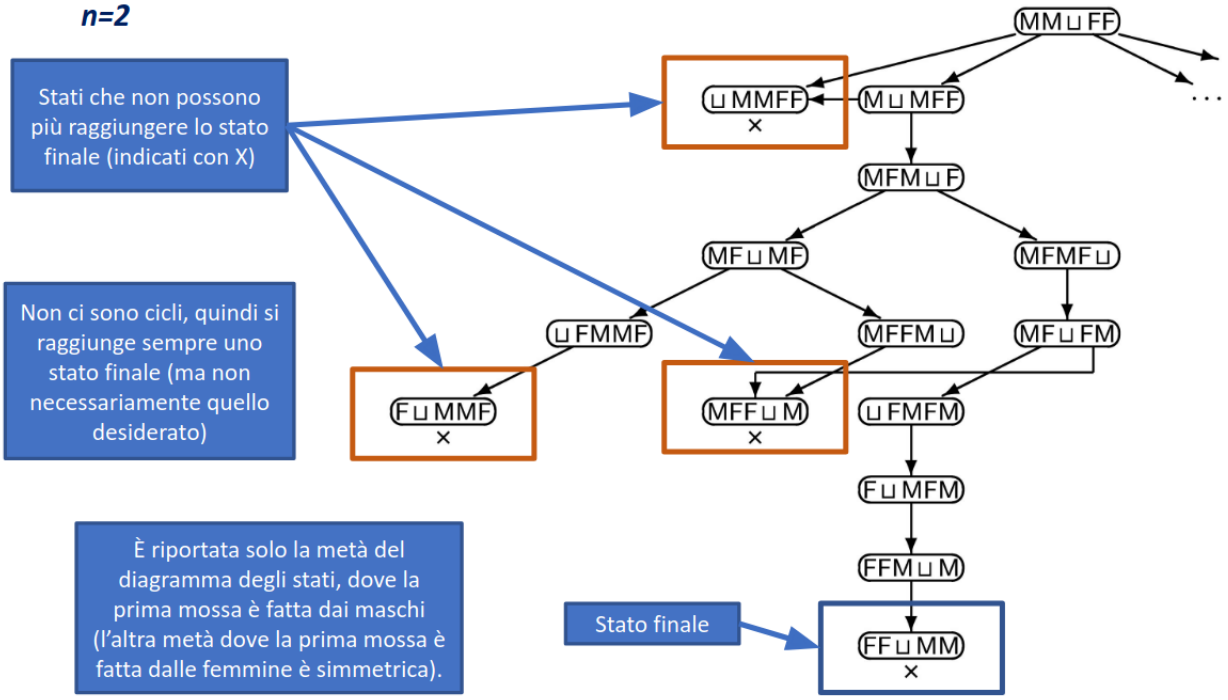
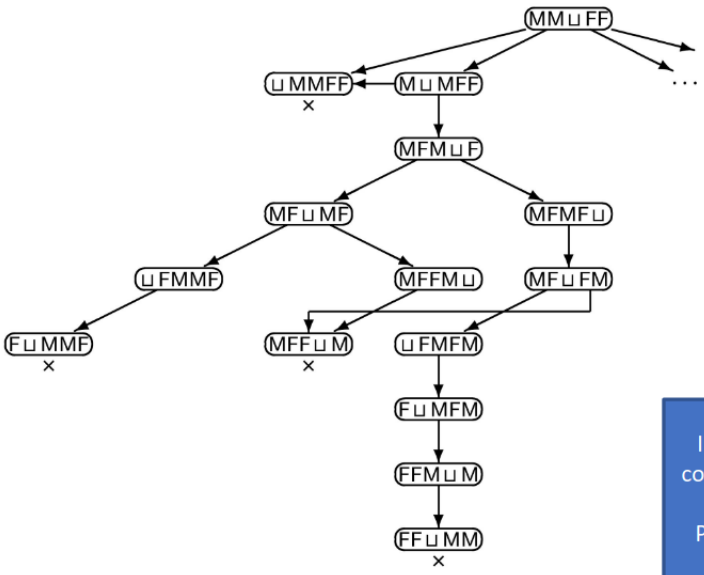


Diagramma degli stati del salto dei ranocchi n=2



a) Esistono computazioni che arrivano allo stato finale.
b) Non tutte le computazioni terminano nello stato finale.
c) Non esistono computazioni che non terminano.

Il diagramma degli stati riporta quindi tutte le combinazioni possibili e permette di esplorare in modo esaustivo un algoritmo concorrente. Purtroppo può essere molto grande e dunque difficile da generare.

1.2 Conteggio Concorrente

Esercizio 2

ESERCIZIO 2.

Si consideri il programma seguente:

Algoritmo Concurrent Counting	
integer n ← 1	
p	**q**
integer temp p1: do 10 times: p2: temp ← n p3: n ← temp + 1	integer temp q1: do 10 times: q2: temp ← n q3: n ← temp + 1

a) Si costruisca una computazione che dia 10 come valore finale di n
b) Si costruisca una computazione che dia 2 come valore finale di n

2
Test2