



Università degli Studi dell'Insubria
Dipartimento di Scienze Teoriche e Applicate

Programmazione Concorrente e Distribuita Soluzioni del problema dei 5 filosofi

Luigi Lavazza

Dipartimento di Scienze Teoriche e Applicate

luigi.lavazza@uninsubria.it



Algoritmo del banchiere per il problema dei filosofi

- I bastoncini sono in mezzo al tavolo, accessibili da tutti
- Regole:
 - ▶ Se non è l'ultimo bastoncino, puoi prenderlo
 - ▶ Se è l'ultimo e ti basta per mangiare (perché ne hai già uno) prendilo pure
 - ▶ In tutti gli altri casi, aspetta



Esercizio 1

- Implementare l'algoritmo del banchiere nei seguenti due modi:
 - ▶ Acquisizione del bastoncino non sospensiva (implica polling). Tra un controllo e l'altro, il filosofo non può mangiare e fa altro.
 - ▶ Acquisizione del bastoncino sospensiva

Esercizio 2

- Implementare il sistema con le regole alternative dell'algoritmo del banchiere:
 - ▶ Se c'è una risorsa disponibile e non è l'unica disponibile, puoi prenderla
 - ▶ Se c'è un'ultima risorsa disponibile ed è l'ultima che ti manca, puoi prenderla
 - ▶ Se c'è una risorsa disponibile e c'è un altro processo che sta usando le risorse, puoi prenderla
 - ▶ In tutti gli altri casi, aspetta

Esercizio 3

- Scrivere un programma che risolve il problema dei filosofi usando un gestore delle risorse
- Quando un filosofo ha bisogno dei bastoncini li chiede al gestore
- Il gestore alloca i bastoncini ai filosofi facendo in modo che:
 - ▶ Non si verifichi alcun deadlock
 - ▶ Non si verifichi starvation
 - ▶ Possibilmente le richieste dei filosofi siano soddisfatte in modo da non favorire ne' sfavorire alcun filosofo



Estensione

- Generalizzare i programmi precedenti in modo che gestiscano un numero N di filosofi, che hanno bisogno di K risorse.



Estensione

- Attrezzare i programmi precedenti in modo che si possa valutare il grado di parallelismo e l'utilizzo delle risorse.

Algoritmo di Chandy & Misra

- I bastoncini possono essere sporchi o puliti. Inizialmente sono tutti sporchi.
- Inizializzazione: per ogni coppia di filosofi che si contende una risorsa, crea un bastoncino (sporco) e dallo al filosofo con l'ID più basso.
- Quando un filosofo vuole mangiare, deve ottenere le risorse che gli mancano dai vicini. A questo scopo manda una richiesta esplicita.
- Quando un filosofo che detiene un bastoncino riceve una richiesta, la ignora se il bastoncino è pulito; se invece è sporco, lo pulisce e lo cede.
- Quando un filosofo finisce di mangiare, i suoi bastoncini sono sporchi. Se un altro filosofo aveva richiesto un bastoncino, il filosofo che ha finito di mangiare lo pulisce e glielo cede.



Esercizio 4

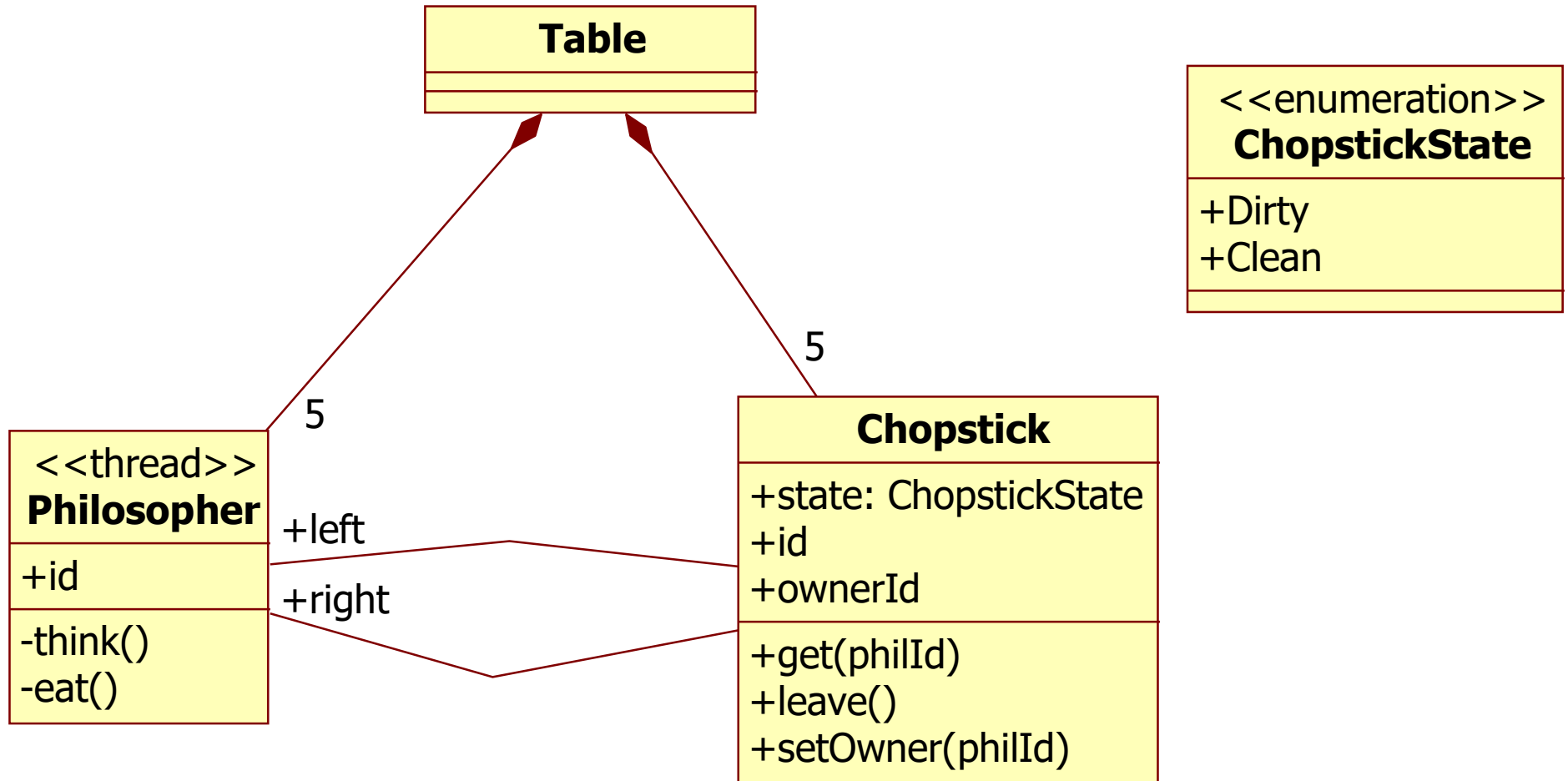
- Implementare il sistema con l'algoritmo di Chandy e Misra



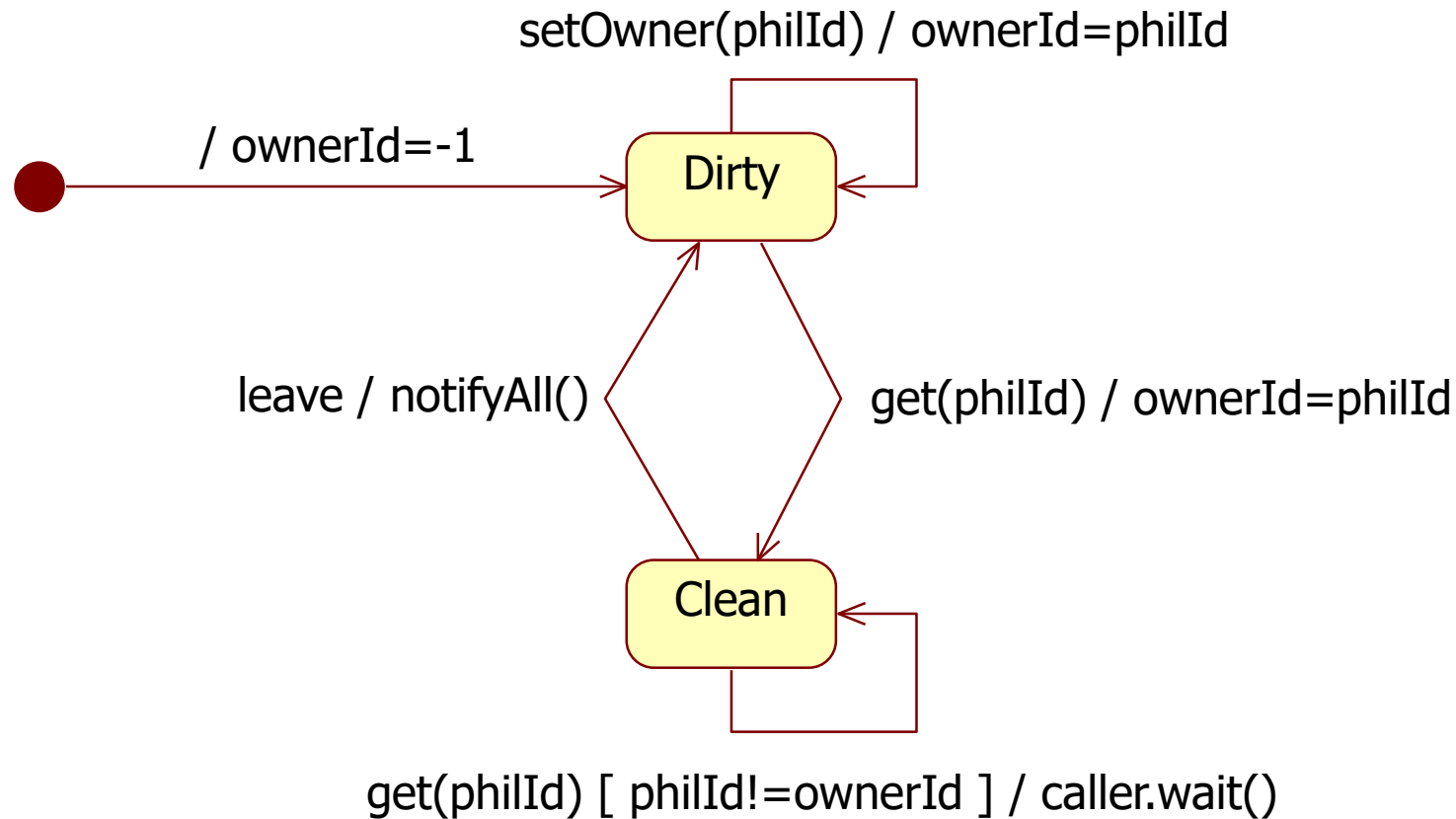
Implementazione dell'algoritmo di Chandy & Misra

- Per semplicità, anziché far comunicare esplicitamente i filosofi, mettiamo le informazioni rilevanti nelle classi che rappresentano i bastoncini.
- Quindi ogni bastoncino
 - ▶ È sporco o pulito
 - ▶ È in possesso di un filosofo

Chandy-Misra: class diagram



Chandy-Misra: stati del chopstick



Chandy-Misra: sequence diagram di ciascun filosofo

