

Analisi della starvation mediante ASM

Gennaro (Rino) Vessio

gennaro.vessio@uniba.it

Proprietà (1)

Dal punto di vista di un sistema di transizione, una **proprietà** è un insieme di esecuzioni.

Un'**esecuzione** è una sequenza (eventualmente infinita) di stati.

Informalmente, una proprietà definisce un **comportamento**.

Proprietà (2)

Tradizionalmente, le proprietà sono distinte in due grandi classi:

- **safety**: qualcosa di «indesiderato» non deve mai accadere;
- **liveness**: qualcosa di «desiderato» deve prima o poi accadere.

Analisi delle proprietà (1)

Negli automi a stati finiti le proprietà sono **decidibili**.

Esistono algoritmi che, in un tempo finito, verificano se una determinata proprietà è soddisfatta.

E per le macchine di Turing?

E per le ASM?

Analisi delle proprietà (2)

Allo stato dell'arte, le proprietà di una ASM sono analizzate con tecniche di **model checking**.

Ma...

... abbiamo perdita di potere espressivo!

... le proprietà sono espresse in logica temporale!



Analisi delle proprietà (3)

Perché non dotare le ASM di specifiche operazionali di proprietà?

Pro:

- no perdita di potere espressivo;
- no logiche temporali.

Contro:

- le proprietà diventano semi-decidibili.



Starvation (1)

Cos'è la **starvation**?

In letteratura, non esiste una definizione univoca.
In generale, rappresenta l'impossibilità, da parte di un processo di un sistema distribuito, di compiere progressi computazionali.



Starvation (2)

La negazione della starvation è una proprietà: la **starvation-freedom**.

La starvation-freedom è una proprietà di safety o liveness?

È domain-dependent o domain-independent?

Starvation (3)

Condizione necessaria e sufficiente al rischio di starvation:

1. Esiste un sistema distribuito Λ
2. La computazione di uno o più dei suoi processi non è indipendente dal mondo esterno Λ
3. Il processo attende un servizio richiesto al mondo esterno (es. l'accesso a una risorsa)

ASM-based starvation (1)

Condizione necessaria e sufficiente al rischio di starvation nelle ASM:

1. Esiste una DASM \wedge
2. La computazione di uno o più dei suoi agenti non è indipendente dal mondo esterno \wedge
3. L'agente attende un servizio richiesto al mondo esterno (es. l'accesso a una risorsa)

ASM-based starvation (2)

La computazione di uno o più dei suoi agenti non è indipendente dal mondo esterno...

L'esecuzione di almeno una regola dipende da funzioni che rappresentano l'interazione col mondo esterno.

ASM-based starvation (3)

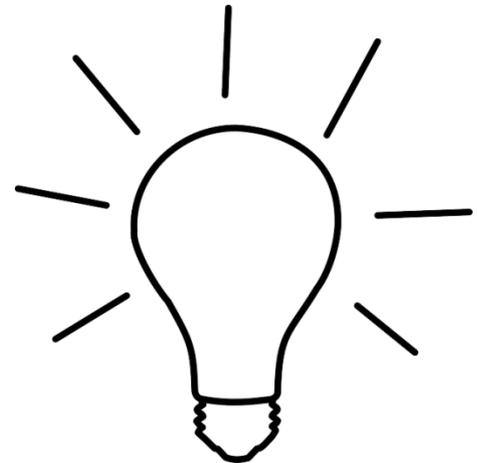
L'agente attende un servizio richiesto al mondo esterno (es. l'accesso a una risorsa)...

L'esecuzione o la non esecuzione della regola prima espressa non determina un cambiamento del valore dei predicati sugli stati.

Predicati sugli stati (1)

In una ASM la computazione può non evolvere anche se vengono eseguiti degli update.

... abbiamo bisogno di un'astrazione: i predicati sugli stati!



Predicati sugli stati (2)

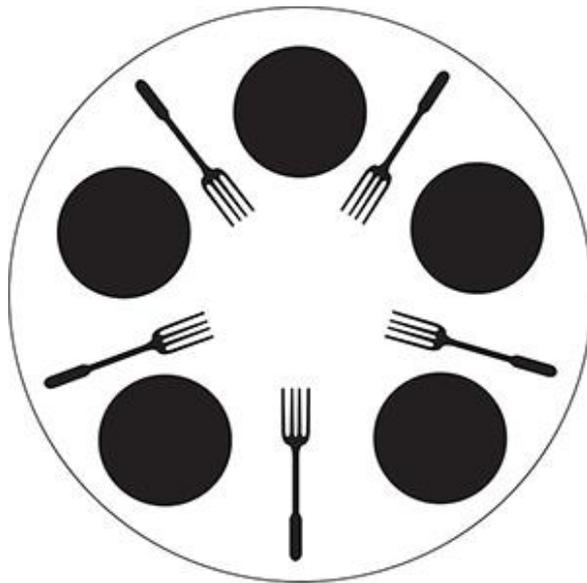
Un **predicato** p **sullo stato** s è una formula del prim'ordine sulle location di s , tale che p è soddisfatta in s .

È l'equivalente di una funzione di etichettatura: diversi stati possono soddisfare uno stesso predicato d'interesse, e uno stesso stato può soddisfare diversi predicati d'interesse.

Cinque filosofi a cena (1)

Cinque filosofi siedono a cena di fronte ad un piatto di spaghetti.

Ciascun filosofo nella sua vita pensa o mangia, rigorosamente con due forchette.



Cinque filosofi a cena (2)

Cosa succede se non neghiamo la resource holding?

Cinque filosofi a cena (3)

Neghiamo la resource holding.

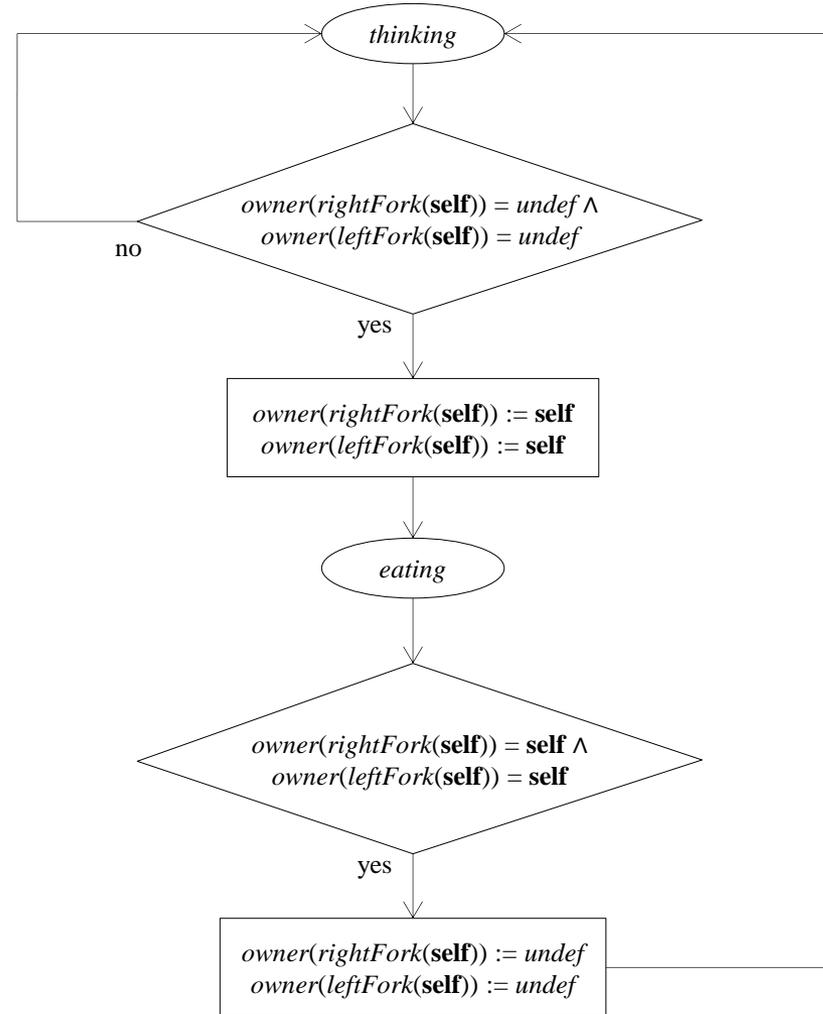
Obblighiamo ciascun filosofo a impugnare le forchette solo quando entrambe libere.

Che algoritmo è?

Cosa succede se non neghiamo la no preemption?

Cinque filosofi a cena (4)

Modelliamo con
le ASM.



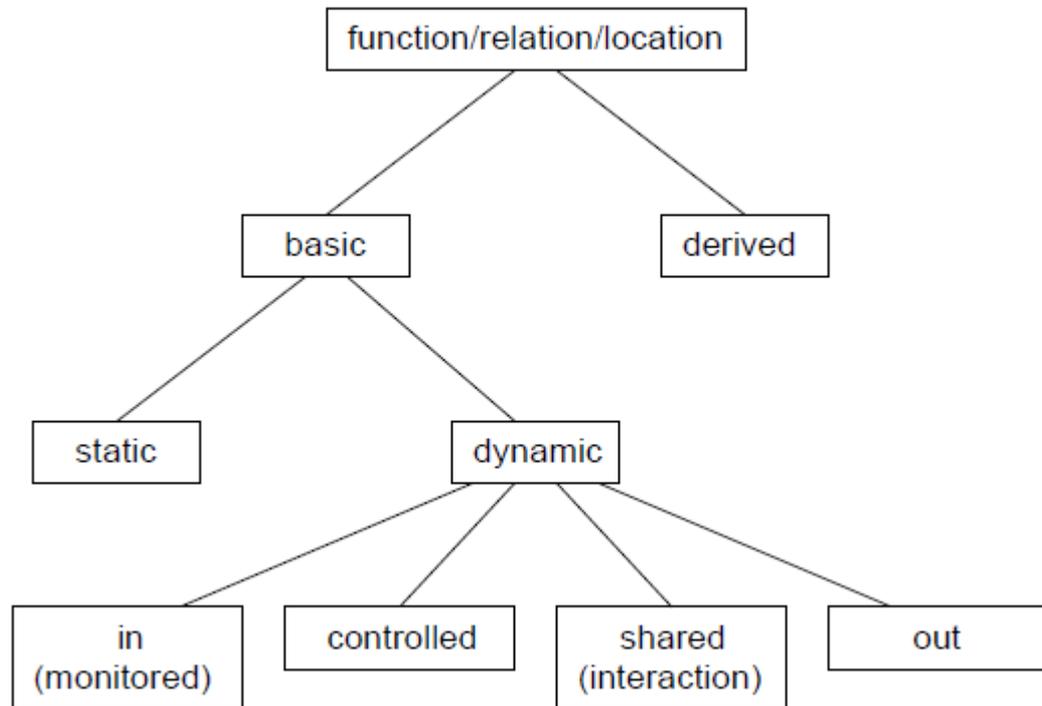
Vulnerable rule (1)

Il rischio di starvation compare in una regola nella forma **if** *cond* **then** *update_true* **else** *update_false* tale che:

1. Il valore di verità di *cond* dipende da funzioni di rischio;
2. Uno fra *update_true* e *update_false* genera una computazione che non cambia il valore dei predicati sugli stati;
3. La computazione evolve, attraverso l'altro update, in uno stato che soddisfa un altro predicato.

Vulnerable rule (2)

Qual è la natura delle funzioni «di rischio»?



Teorema

Una DASM starvation-free è una DASM senza vulnerable rule.

La dimostrazione si basa sul fatto che la definizione di vulnerable rule incapsula la condizione necessaria e sufficiente alla starvation.



Ultimo punto

Come si risolve anche la starvation nei cinque filosofi a cena?