

# Modelli della concorrenza

## Introduzione

Corso di laurea magistrale in informatica  
Anno accademico 2022/2023 — I semestre

Insegnamento: Modelli e computazione

Modulo: Modelli della concorrenza (6 CFU)

## Argomenti

### Docenti

- Lucia Pomello
- Luca Bernardinello

### Tutor

- ?

- 1 Semantica e correttezza dei programmi sequenziali
- 2 Modelli di sistemi concorrenti: CCS e bisimulazione
- 3 Modelli di sistemi concorrenti: reti di Petri
- 4 Logiche temporali e *model-checking*

Sito di consultazione: [elearning.unimib.it](http://elearning.unimib.it) (calendario delle lezioni, materiale di studio, riassunto delle lezioni, avvisi, forum)  
Per accedere ai contenuti è necessario iscriversi

Orario:

- lun: 15:30-18:30, aula U4-02
- gio: 15:30-18:30, aula U24-C1
- Consultate il sito per eventuali variazioni!

Nel corso della prima settimana pubblicheremo il calendario completo (sempre suscettibile di variazioni)

Contatto: [luca.bernardinello@unimib.it](mailto:luca.bernardinello@unimib.it)

Orario di ricevimento: su appuntamento

Esame: prova scritta (esercizi su tutti gli argomenti) e colloquio obbligatorio

- KR Apt, FS de Boer, E-R Olderog [Verification of Sequential and Concurrent Programs](#), Texts in Computer Science, Springer, 2009
- M Ben-Ari [Mathematical Logic for Computer Science](#), third edition, Springer, 2012
- L Aceto, A Ingólfssdóttir, KG Larsen, J Srba [Reactive Systems: Modelling, Specification and Verification](#), Cambridge Univ. Press, 2007.

Consultate il sito del corso per indicazioni più complete, dispense, esercizi svolti.

# Requisiti (?)

## Insiemi e relazioni

Relazioni binarie, di equivalenza, d'ordine parziale  
Prodotto cartesiano, nozioni di combinatoria  
Principio di induzione, definizioni induttive

## Automi e linguaggi formali

Automi a stati finiti  
Linguaggi formali, grammatiche

## Logica matematica

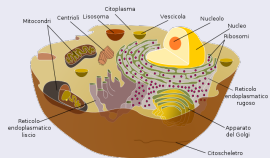
Logica proposizionale e dei predicati del primo ordine  
Tavole di verità dei connettivi  
Dimostrazione, regole di inferenza, assiomi

## Programmazione

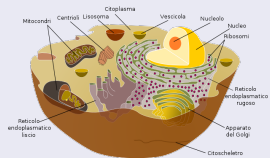
Fondamenti di programmazione concorrente, *thread*, mutua esclusione, *deadlock*



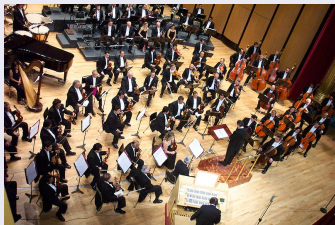
## La cellula vivente



## La cellula vivente



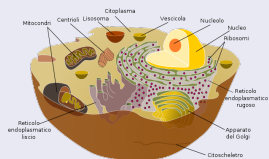
## L'orchestra



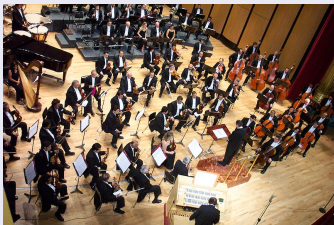


# Concorrenza

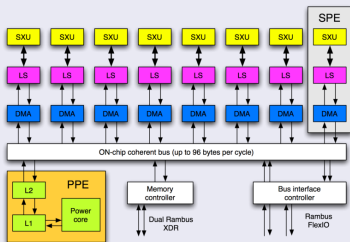
## La cellula vivente



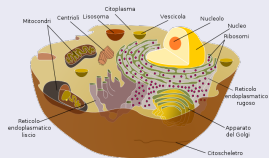
## L'orchestra



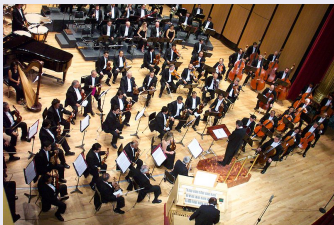
## Processore multicore



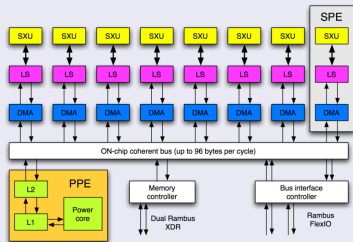
## La cellula vivente



## L'orchestra



## Processore multicore



## Squadra antincendio



## Caratteristiche generali

- Competizione per l'accesso a risorse condivise
- Cooperazione per un fine comune
- Coordinamento di attività diverse
- Sincronia/asincronia

Mars Pathfinder

Therac-25

NASA Rover Ames K9

Come progettare e realizzare sistemi concorrenti corretti?

- Linguaggi
- Modelli
- Logica, *model-checking*

# Sistemi concorrenti – linguaggi

```
public class Produttore extends Thread
{
    private Piatto p; // Buffer
    private int    n; // Identificatore

    public Produttore (Piatto b, int i)
    {
        p = b; n = i;
    }

    public void run ()
    {
        ...
    }
}
...
public class SaggioProdCons
{
    public static void main (String [] args)
    {
        Piatto p = new Piatto();
        Produttore a = new Produttore (p, 1);
        Consumatore b = new Consumatore (p, 2);

        a.start();
        b.start();
    }
}
```

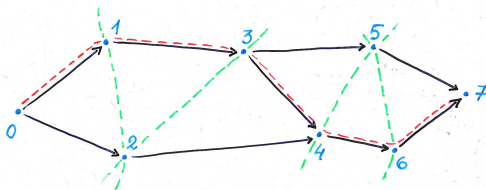
## Partitura

The image displays a musical score for five instruments: Horn in F, Violin 1, Violin 2, Viola, and Bass. The score is written in 2/4 time and consists of five staves. The Horn part is in F major and features a melodic line with eighth and sixteenth notes. The Violin 1 and Violin 2 parts are in B-flat major and play a similar melodic line. The Viola and Bass parts are in B-flat major and play a bass line with eighth and sixteenth notes. The score is presented in a system with five staves, each with its own clef and key signature.

W.A. Mozart, Ein musikalischer Spaß.

## Task graph — grafo delle attività

Grafi di attività (task graphs)



nodo = attività, evento  
arco = relazione di precedenza

Tagli :  $\{1,2\}, \{2,3\}, \{4,5\}, \{5,6\}, \{0\}, \{7\}$

Linee :  $\{0,1,3,4,6,7\}, \{0,1,3,5,7\} \dots$

« Cause » :  $4 \downarrow = \{0,1,2,3\}$      $5 \downarrow = \{0,1,3\} \dots$



## Sistema di equazioni - algebre di processi

$$Snd = in(x).\overline{sm}\langle x \rangle.S(x) \quad (1)$$

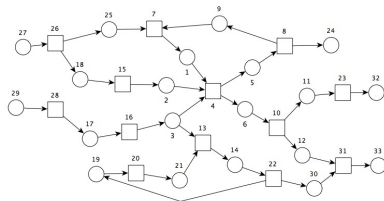
$$S(x) = ms.\overline{sm}\langle x \rangle.S(x) + ok.Snd \quad (2)$$

$$Med = sm(y).(mr\langle y \rangle.Med + \tau.ms.Med) \quad (3)$$

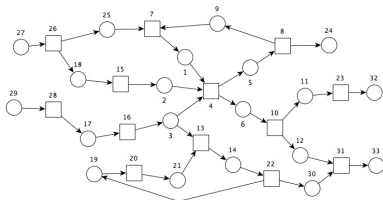
$$\dots \quad (4)$$

$$P = (Snd|Med|Rec) \quad (5)$$

# Modellare – le reti di Petri



# Modellare – le reti di Petri



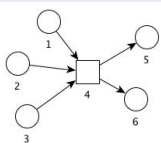
## Stato locale



## Evento

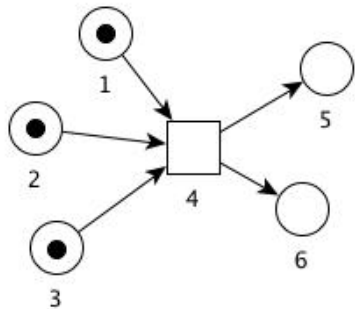


## Relazione di flusso

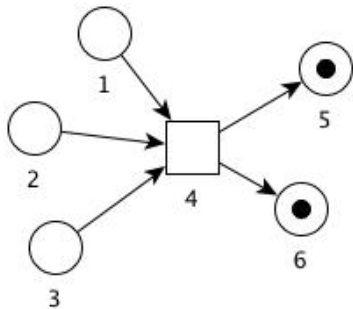
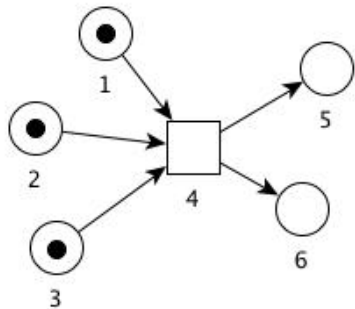


1, 2, 3: *precondizioni* di 4  
5, 6: *postcondizioni* di 4

## Reti di Petri – regola di scatto



## Reti di Petri – regola di scatto



## Produttore-buffer-consumatore

