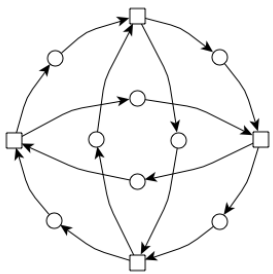
UNIVERSITA DEGLI STUDI DI MILANO–BICOCCA

CORSO DI LAUREA MAGISTRALE IN INFORMATICA

MODELLI DELLA CONCORRENZA

LOGICHE TEMPORALI E MODEL-CHECKING



```
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();
  }
}

public class Piatto
{
  private int valore;
  private boolean pieno;

  public synchronized int preleva () {
    while ( pieno == false ) {
      try {  wait();  }
      catch (InterruptedException e) { }
    }
    pieno = false;
    notifyAll ();
    return valore;
  }

  public synchronized void deposita (int v) {
    ...
  }
}
```

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

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

    public void run ()
    {
      for (int i = 1; i < 10; i++)
      {
        try {
          sleep((long)(Math.random()*1000));}
        catch (InterruptedException e) { }

        p.deposita (i);
        System.out.println(n + " deposita " + i);
      }
    }
}

public class Consumatore extends Thread
{
    ...
    public void run ()
    {
      ...
    }
}
```

Correttezza dei programmi concorrenti

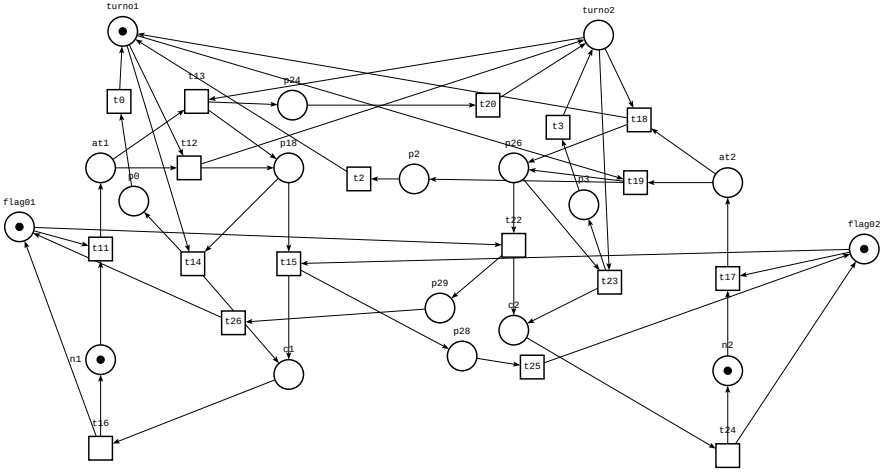
Ogni oggetto prodotto viene prima o poi consumato

Nessun oggetto viene consumato più di una volta

Il sistema non raggiunge mai uno stato di *deadlock*

...

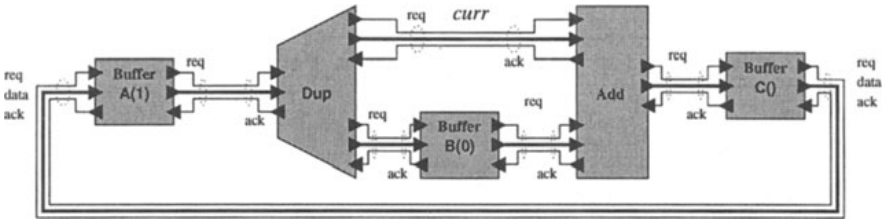
Reti di Petri



I due processi non si trovano mai contemporaneamente nella sezione critica (i posti c1 e c2 non sono mai marcati contemporaneamente)

Se un processo richiede l'accesso alla sezione critica, prima o poi avrà il permesso (se è marcato il posto at1, prima o poi sarà marcato il posto c1)

Circuiti



Micropipeline circuit for Fibonacci calculator

Tratta da H Barringer et al., *Abstract Modelling of Asynchronous Micropipeline Systems using Rainbow*.

Se un segnale è presente sulla linea `req` in uscita dal Buffer A(1), prima o poi un segnale sarà presente sulla linea `ack` in entrata al Buffer A(1)

...

Sistemi reattivi

Sistemi concorrenti, distribuiti, asincroni

Non obbediscono al paradigma *input-computazione-output*

Non si possono analizzare con gli strumenti della logica di Hoare

“Se un messaggio è stato spedito, prima o poi sarà consegnato al destinatario”

“La spia d’allarme resta accesa fino a quando il dispositivo viene spento”

“A partire da qualsiasi stato è possibile riportare il sistema allo stato iniziale”

“I due processi non si trovano mai contemporaneamente nella sezione critica”

Analisi di sistemi concorrenti

Problema stabilire se un sistema reattivo è “corretto”

Metodo

1. esprimiamo il criterio di correttezza come formula di un opportuno linguaggio logico;
2. rappresentiamo (*modelliamo*) il sistema nella forma di sistema di transizioni;
3. valutiamo se la formula è vera nel sistema di transizioni.

Strumenti sistemi di transizioni (modelli di Kripke), logiche temporali, algoritmi.

Sistemi di transizioni

Elementi di un sistema di transizioni: stati, transizioni di stato

$$A = (Q, T)$$

Q: insieme degli stati

$T \subseteq Q \times Q$: insieme delle transizioni di stato

Nozioni utili: cammino, cammino massimale

Cammino: $\pi = q_0 q_1 q_2 \cdots$, $(q_i, q_{i+1}) \in T$ per ogni i

Modelli di Kripke

$AP = \{z_1, z_2, \cdots\}$: insieme di *proposizioni atomiche*

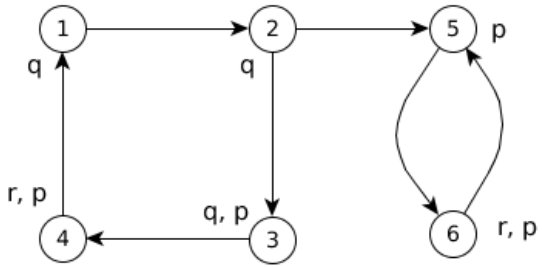
Dato un sistema di transizioni $A = (Q, T)$, associamo a ogni stato $q \in Q$ l'insieme delle proposizioni atomiche che sono *vere* in quello stato.

$$I : Q \longrightarrow 2^{AP}$$

Modello di Kripke

$$A = (Q, T, I)$$

Modelli di Kripke



$AP = \{p, q, r\} \quad Q = \{1, 2, 3, 4, 5, 6\}$

$T = \{(1, 2), (2, 3), (2, 5), (5, 6), (6, 5), \cdots\}$

$I(4) = \{p, r\} \quad I(2) = \{q\} \quad \cdots$

Logica temporale lineare

LINEAR TEMPORAL LOGIC (LTL)

Sintassi

Semantica

No apparato deduttivo (algoritmi)

Linear temporal logic (LTL) – sintassi

Proposizioni atomiche

$$AP = \{p_1, p_2, \cdots, p_i, \cdots\}$$

Esempi

- "la spia d'allarme è accesa"
- "il messaggio è stato spedito"
- "il processo P è nella sezione critica"
- "il buffer è pieno"

Linear temporal logic (LTL) – sintassi

Formule ben formate – $\mathrm{FBF}_{\mathrm{LTL}}$

- ogni proposizione atomica è una formula ben formata
- le costanti logiche TRUE e FALSE sono formule ben formate
- se α e β sono formule ben formate, allora $\neg\alpha$, $\alpha \vee \beta$, $\alpha \wedge \beta$,
- $\alpha \rightarrow \beta$ sono formule ben formate

Operatori temporali: se α e β sono formule ben formate, allora

- $\mathrm{X}\alpha$ "nel prossimo stato α"
- $\mathrm{F}\alpha$ "prima o poi α" (*eventually*)
- $\mathrm{G}\alpha$ "sempre α"
- $\alpha \mathrm{U} \beta$ "α fino a quando β"

sono formule ben formate

LTL – esempi di formule

FG α α è invariante da un certo istante in poi

LTL – esempi di formule

$\mathsf{FG}\,\alpha$	α è invariante da un certo istante in poi
$\mathsf{GF}\,\alpha$	α è vera in un numero infinito di stati

LTL – esempi di formule

$\mathsf{FG}\,\alpha$ α è invariante da un certo istante in poi

$\mathsf{GF}\,\alpha$ α è vera in un numero infinito di stati

$\mathsf{G}\,\neg(cs_1 \wedge cs_2)$ Mutua esclusione

LTL – esempi di formule

$\mathsf{FG}\,\alpha$ α è invariante da un certo istante in poi

$\mathsf{GF}\,\alpha$ α è vera in un numero infinito di stati

$\mathsf{G}\,\neg(cs_1 \wedge cs_2)$ Mutua esclusione

$\mathsf{G}\,(req \longrightarrow \mathsf{XF}\,ack)$

$\mathsf{G}\,(req \longrightarrow (req\ \mathsf{U}\ ack))$

LTL – esempi di formule

$\mathsf{FG}\,\alpha$ α è invariante da un certo istante in poi

$\mathsf{GF}\,\alpha$ α è vera in un numero infinito di stati

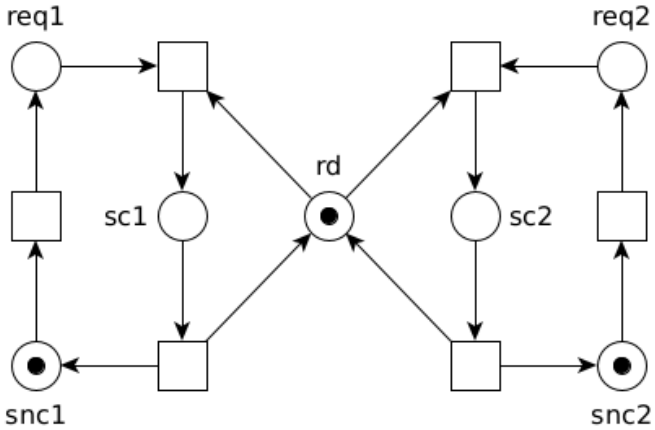
$\mathsf{G}\,\neg(cs_1 \wedge cs_2)$ Mutua esclusione

$\mathsf{G}\,(req \longrightarrow \mathsf{XF}\,ack)$

$\mathsf{G}\,(req \longrightarrow (req\ \mathsf{U}\ ack))$

$\mathsf{G}\,(req \longrightarrow ((req \wedge \neg ack)\ \mathsf{U}\ (ack \wedge \neg req)))$

Esempio



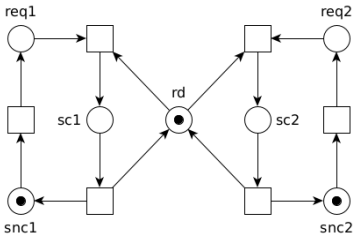
`rd`: risorsa disponibile

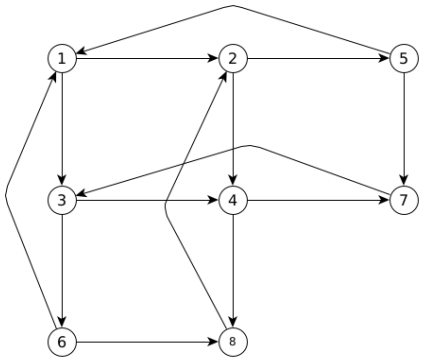
`snc`: sezione non critica

`sc`: sezione critica

`req`: richiesta pendente

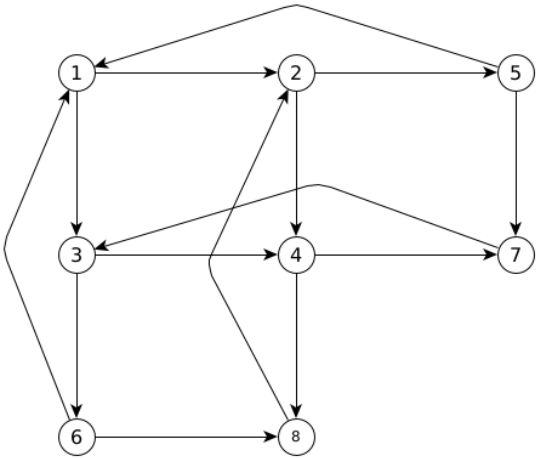
Esempio





1: rd, snc1, snc2
2: rd, req1, snc2
3: rd, snc1, req2
4: rd, req1, req2
5: sc1, snc2
6: snc1, sc2
7: sc1, req2
8: req1, sc2

Esempio



1: $\mathrm{rd}, \mathrm{snc1}, \mathrm{snc2}$
2: $\mathrm{rd}, \mathrm{req1}, \mathrm{snc2}$
3: $\mathrm{rd}, \mathrm{snc1}, \mathrm{req2}$
4: $\mathrm{rd}, \mathrm{req1}, \mathrm{req2}$
5: $\mathrm{sc1}, \mathrm{snc2}$
6: $\mathrm{snc1}, \mathrm{sc2}$
7: $\mathrm{sc1}, \mathrm{req2}$
8: $\mathrm{req1}, \mathrm{sc2}$

$\mathsf{G}\,\neg(\mathrm{sc1} \wedge \mathrm{sc2})$ $\quad$ $\mathsf{G}\,(\mathrm{req1} \longrightarrow \mathsf{F}\,\mathrm{sc1})$

Modelli di Kripke

$AP = \{p, p_1, p_2, \cdots\}$: insieme di *proposizioni atomiche*

Dato un sistema di transizioni $A = (Q, T)$, associamo a ogni stato $q \in Q$ l'insieme delle proposizioni atomiche che sono *vere* in quello stato.

$$I : Q \longrightarrow 2^{AP}$$

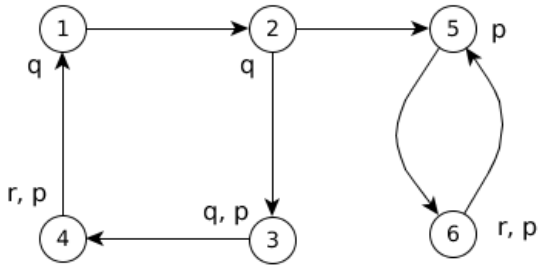
Modello di Kripke

$A = (Q, T, I)$

Cammino: $\pi = q_0 q_1 q_2 \cdots$, $(q_i, q_{i+1}) \in T$ per ogni i

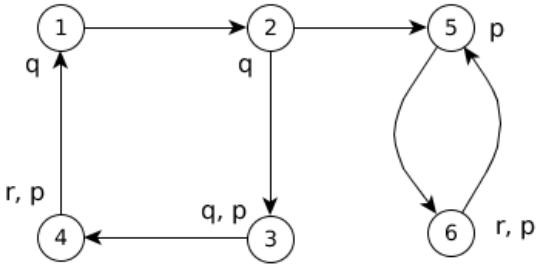
Suffisso di ordine i di π è il cammino $\pi^{(i)} = q_i q_{i+1} \cdots$

Modelli di Kripke



Famiglie di cammini massimali

Modelli di Kripke



Famiglie di cammini massimali

$(1234)^\omega \quad (1234)^*(12)(56)^\omega$

LTL - semantica

Interpretiamo le formule di LTL su un modello di Kripke

Procediamo in due fasi:

1) definiamo un criterio per stabilire se una formula α è vera in un cammino massimale π

2) diciamo che la formula è vera rispetto a uno stato q se è vera in tutti i cammini massimali che partono da q

LTL - semantica

Sia $\pi = q_0 q_1 q_2 \cdots$ un cammino e sia α una formula di LTL

$\pi \models \alpha$ significa che α è vera nel cammino π

Definiamo la relazione $\models$ per induzione sulla struttura delle formule.

LTL - semantica

Sia $\pi = q_0 q_1 q_2 \cdots$ un cammino e sia α una formula di LTL

$\pi \models \alpha$ significa che α è vera nel cammino π

Definiamo la relazione $\models$ per induzione sulla struttura delle formule.

Supponiamo che α e β siano due formule, p una proposizione atomica.

LTL - semantica

Sia $\pi = q_0 q_1 q_2 \cdots$ un cammino e sia α una formula di LTL

$\pi \models \alpha$ significa che α è vera nel cammino π

Definiamo la relazione $\models$ per induzione sulla struttura delle formule.

Supponiamo che α e β siano due formule, p una proposizione atomica.

$\pi \models p$ sse, $p \in I(q_0)$

$\pi \models \neg\alpha$ sse $\pi \not\models \alpha$

$\pi \models \alpha \vee \beta$ sse $\pi \models \alpha$ o $\pi \models \beta$

LTL – semantica

Operatori temporali

Ipotesi: $\alpha \in \mathrm{FBF}$

LTL – semantica

Operatori temporali

Ipotesi: $\alpha \in \mathrm{FBF}$

$\pi \models \mathrm{X}\alpha$ se e solo se $\pi^{(1)} \models \alpha$

LTL – semantica

Operatori temporali

Ipotesi: $\alpha \in \mathrm{FBF}$

$\pi \models X\alpha$ se e solo se $\pi^{(1)} \models \alpha$

$\pi \models F\alpha$ se e solo se $\exists i \in N : \pi^{(i)} \models \alpha$

LTL – semantica

Operatori temporali

Ipotesi: $\alpha \in \mathrm{FBF}$

$\pi \models \mathrm{X}\alpha$ se e solo se $\pi^{(1)} \models \alpha$

$\pi \models \mathrm{F}\alpha$ se e solo se $\exists i \in \mathrm{N} : \pi^{(i)} \models \alpha$

$\pi \models \mathrm{G}\alpha$ se e solo se $\forall i \in \mathrm{N} : \pi^{(i)} \models \alpha$

LTL – semantica

L'operatore $\mathcal{U}$

Ipotesi: $\alpha, \beta \in \mathrm{FBF}$

$\pi \models \alpha \,\mathcal{U}\, \beta$ se e solo se

LTL – semantica

L'operatore $\mathcal{U}$

Ipotesi: $\alpha, \beta \in FBF$

$\pi \models \alpha \mathcal{U} \beta$ se e solo se

1. esiste $i \in N$ tale che $\pi^{(i)} \models \beta$ $(\pi \models F\beta)$

LTL – semantica

L'operatore $\mathcal{U}$

Ipotesi: $\alpha, \beta \in \mathrm{FBF}$

$\pi \models \alpha \,\mathcal{U}\, \beta$ se e solo se

1. esiste $i \in \mathbb{N}$ tale che $\pi^{(i)} \models \beta$ $(\pi \models \mathrm{F}\beta)$
2. per ogni h, $0 \leqslant h < i$, $\pi^{(h)} \models \alpha$