Lezione 18 11/12/2023

Il model checking è l'analisi dei modelli. Per stabilire la correttezza di un modello stabiliamo una rappresentazione del modello (con diversi modi, ccs, reti di Petri, etc).

Model-checking

Partiamo da esempi concreti, questo è un programma concorrente in java:

```
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 ()
    {
      ...
    }
}
```

Ogni oggetto prodotto viene prima o poi consumato

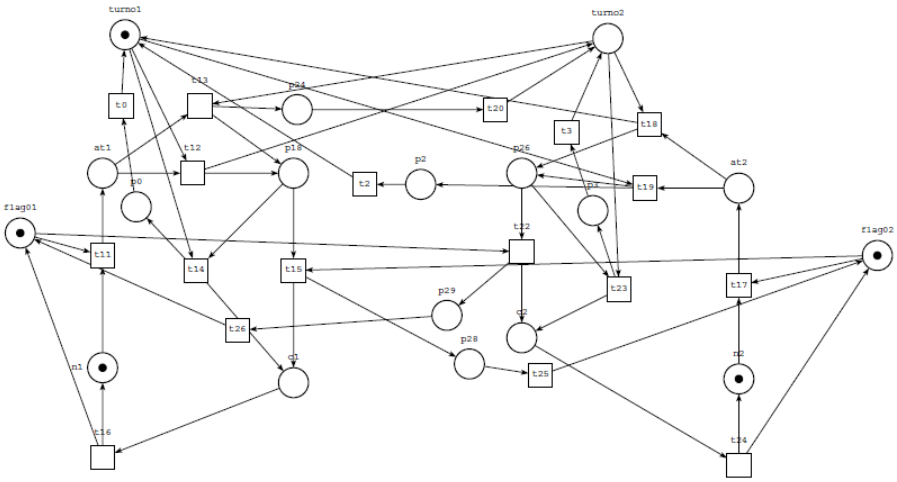
Nessun oggetto viene consumato più di una volta

Il sistema non raggiunge mai uno stato di *deadlock*

...

Queste sono proprietà per il corretto funzionamento di un programma concorrente(?).

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)

Nelle prossime lezioni vogliamo quindi sviluppare un metodo che ci permette di verificare che un modello di un sistema reattivo soddisfa le proprietà richieste. Dovremo quindi poter specificare le proprietà e poi verificare se valgono.

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"

Un sistema reattivo di solito non ha uno stato finale, quindi la nozione di tripla non è utile per analizzare il comportamento di questi sistemi.

Metodo per stabilire la correttezza di un sistema

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.

Esprimiamo in un linguaggio logico i nostri criteri di correttezza con una formula. Poi rappresentiamo il sistema con una formula matematica. Infine quindi valutiamo se il sistema è corretto.

Sistemi di transizioni (strumento 1)

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

Potremo anche rappresentarli tramite un grafo. Un cammino è una sequenza di stati se tutte le coppie di stati sono collegate da una transizione.

Un cammino è massimale se non può essere esteso, quindi il cammino massimale può anche essere infinito.

Logica temporale lineare (strumento 2)

Useremo questa logica inizialmente.

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"

Per la sintassi, partiamo da delle proposizione atomiche che consideriamo non scomponibili, per cui consideriamo di poter sapere se sono vere o false in un momento qualsiasi, senza guardare gli altri stati del sistema.

Formule ben formate – FBF_{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

Definiamo induttivamente l'insieme delle formule ben formate.

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

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

sono formule ben formate

Gli operatori temporali sono applicati alle formule ben formate, ottenendo altre fbf.

Semantica - 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

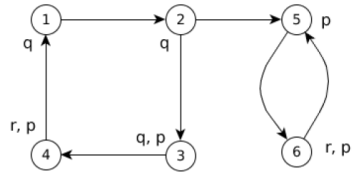
$$A = (Q, T, I)$$

Possiamo definire una funzione I, che associa ad ogni stato l'insieme delle proposizione atomiche che sono vere in quello stato. (I → funzione di interpretazione)

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 $\pi^{(i)} = q_i q_{i+1} \cdots$

Rappresentazione tramite grafo di un modello di Kripke:



I 6 nodi corrispondono ai 6 stati del modello.

Di fianco ad ogni stato ci sono le proposizioni atomiche che sono vere in quello stato.

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

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

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

Famiglie di cammini massimali

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

Un cammino massimale è per esempio 1234 ripetuto all'infinito, ma anche quello che poi rimane all'infinito dentro a 56.

In questo modello i cammini massimali sono tutti infiniti, perchè da ogni stato parte almeno un arco.

Il cammino che percorre all'infinito 1234 è segnato dentro alle parentesi, con omega come esponente che indica l'infinito. L'asterisco invece indica un numero qualunque ma finito.

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

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 $\pi \models \beta$

Il cammino pi può anche essere infinito come quelli che abbiamo appena visto, non necessariamente sono simboli tutti diversi.

Operatori temporali

Ipotesi: $\alpha \in FBF$

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

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

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

pigreco di 1 è l'attimo subito successivo.

F (prima o poi) basta che ci sia un pi(i) dove prima o poi la formula è vera.

G (sempre) per ogni pi(i) la formula deve essere vera.

Operatore until

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

$\pi \models \alpha \, U \, \beta$ se e solo se

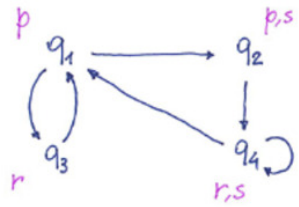
1. esiste $i \in N$ tale che $\pi^{(i)} \models \beta$ $(\pi \models F\beta)$
2. per ogni h, $0 \leqslant h < i$, $\pi^{(h)} \models \alpha$

Dire alfa util beta è uguale a U(alfa, beta), ogni tanto lo scriverà così.

Prima o poi Fbeta è vera, fino a quel momento beta deve essere vera.

Esercizio

Dato un sistema di transizioni, composto da 4 stati con le possibili transizioni



$I(q_1) = \{p\}$ $I(q_2) = \{p, s\} \dots$.

Formula $Fs \rightarrow$ prima o poi sarà vera s

α Fs
β Fr
γ G(p v r)
δ U(p,s)

Si analizza tale formula rispetto ai 4 cammini massimali

π_1 $q_1 q_3 q_1 q_3 q_1 q_3 \cdots$
π_2 $q_1 q_2 q_4 q_4 q_4 q_4 \cdots$
π_3 $q_1 q_2 q_4 q_4 q_1 q_3 q_1 q_3 q_1 q_3 \cdots$
π_4 $q_1 q_3 q_1 q_2 \cdots$

π_1

1. rimane sempre in q_1, q_3, dunque α non è vera nel cammino π_1.
2. In questo caso sarà vera in π_1
3. In questo caso sarà vera in π_1
4. Non è vera poichè s non diventa mai vero.

π_2

1. Si poichè al secondo stato è diventata vera s
2. Si poichè dopo due transizioni arriviamo a q_4 dove è vera r
3. Si in tutti e 3 o è vera p o s
4. Prima o poi s diventa vera e nei precedenti è vera p

π_3

1. Si poichè valida in q_2
2. Si
3. Si ed è vera in tutti gli stati
4. Si

π_4

1. ..
2. ...
3. ...
4. in q_3 non è vera.