Modelli della Concorrenza - Domande Orale
Antonio Vivace

Definizione di transizione debole e di bisimulazione debole

Bisimulazione debole e verifica con la tecnica "attaccante-difensore"

Definizione di Bisimulazione debole

Una relazione binaria $\mathcal{R}$ tra processi CCS è una *bisimulazione debole* se, dati p e q tali che $p\mathcal{R}q$, allora $\forall \alpha \in \mathcal{A} = A \cup \bar{A} \cup \{\tau\}$ sono verificate le seguenti condizioni:

- se $p \rightarrow^{\alpha} p_1$, allora esiste $q \Rightarrow^{\alpha} q_1$ tale che: $p_1 \mathcal{R} q_1$ (e viceversa:)

- se $q \rightarrow^{\alpha} q_1$, allora esiste $p \Rightarrow^{\alpha} p_1$ tale che: $p_1 \mathcal{R} q_1$

Due processi p e q sono *debolmente bisimili* ($p \approx^{Bis} q$) se e solo se esiste una relazione di bisimulazione $\mathcal{R}$ tale che $p\mathcal{R}q$.

Dove la regola di transizione debole è definita come segue: $p \Rightarrow^{\alpha} p$se e solo se:

- $p \rightarrow^{\tau *} \rightarrow^{\alpha} \rightarrow^{\tau *} p$, se $\alpha = \tau$- $p \rightarrow^{\tau *} p$, se $\alpha = \tau$.

Si può dimostrare che la relazione $\approx^{Bis} = \cup \{\mathcal{R} : \mathcal{R}$ bisimulazione debole$\}$ è una relazione di equivalenza, è la più grande bisimulazione debole e che soddisfa la seguente proprietà:

$p \approx^{Bis} q$ se e solo se:

$\forall \alpha \in \mathcal{A}$,

- se $p \rightarrow^{\alpha} p_1$, allora esiste $q \Rightarrow^{\alpha} q_1$ tale che: $p_1 \approx^{Bis} q_1$ e viceversa:

- se $q \rightarrow^{\alpha} q_1$, allora esiste $p \Rightarrow^{\alpha} p_1$ tale che: $p_1 \approx^{Bis} q_1$

Per la verifica della bisimulazione con la tecnica tratta dalla teoria dei giochi dell'*attaccante e difensore* si veda la sezione 3.5 e 3.5.1 del testo "Reactive Systems" al link sul sito.

Si noti che nel testo viene presentata la tecnica per la Bisimulazione forte e solo nella sezione 3.5.1 quella per la Bisimulazione debole. La differenza sta nel fatto che, mentre l'attaccante usa sempre la regola di transizione forte, nel caso della Bisimulazione debole il difensore usa sempre la regola

Che cos'è una rete di Petri?

Una rete di petri è un'estensione di una RE.

Una **RE è una coppia (N=(B,E,F), C in B)**.

Una rete N=(B,E,F):

- **B** condizioni O
- **E** transizioni []
- **F** $\subseteq$ **(BxE)U(ExB)**. Relazione di flusso ->
- **dom(F) U tau(F) = B U E** (non ci sono elementi isolati).

Una rete può essere:

- **Aciclica**
- **B-Semplice**
- **E-Semplice**
- **Non isolata** per tutte le b in B, neighboorhood(b) != {}

A cosa servono le reti di Petri?

Permettono di rappresentare sistemi ed il loro comportamento.Rappresentati con un Grafo Orientato Bipartito con un numero finito di vertici. Un nodo rappresenta una possibile transizione o stato, gli archi gli input ed output associati ai nodi.

Regola di scatto (concorrenza)

Sia N una rete elementare e $c \subseteq B$ una configurazione. L'evento $e \in E$ è abilitato in c se e solo se $\bullet e \subseteq c \wedge e \bullet \cap c = \varnothing$ e si indica con c[e >. Se c[e > allora quando e occorre in c si genera una nuova configurazione $c' = (c - \bullet e) \cup e \bullet$.

Come si definisce l'ordine parziale di una rete di occorrenze?

Su di una rete di occorrenze, che è una RE N=(B,E,F) che:

- $\forall b \in B, \ |\bullet b| \leq 1 \ \wedge |b \bullet| \leq 1$
- $\forall x,y \ \in (B \cup E) \mid (x,y) \in F^+ \rightarrow (y,x) \ \in F^+$

La relazione d'ordine è

$\{X, \ \leq\} = \{B \cup E, \ F^+\}$

con F+ chiusura transitiva e riflessiva della relazione di flusso.
Questa relazione soddisfa le seguenti proprietà:

- **x, y** $\in$ **X:** x,y elementi che occorrono nella storia di X
- **x** $\leq$ **y:** x causa y
- **x li y:** x $\leq$ y $\vee$ y $\leq$ x. Causalmente dipendenti.
- **x co y:** !x $\leq$ y $\vee$!y $\leq$ x. Causalmente **in**dipendenti.

li e co sono simmetriche e riflessive ma non transitive.

co-set: insieme C $\subseteq$ **X di elementi tali che per** $\forall x,y \in C : \ x \ co \ y$**. ovvero tutti casualmente indipendenti tra loro.**

li-set: insieme C $\subseteq$ **X di elementi tali che per** $\forall x,y \in C : \ x \ li \ y$**. ovvero tutti casualmente dipendenti tra loro.**

Definizione Linea

Una linea è un li-set massimale (aggiungendo un elemento rendo falsa la condizione)

Definitione Taglio

Un taglio è un co-set massimale (aggiungendo un elemento rendo falsa la condizione).

Significato di una tripla di Hoare

Una tripla di Hoare sempre scritta nella formula {P}C{Q} dove:

- P precondizioni
- C Programma
- Q Postcondizioni

La formula {P}C{Q} è vera sse eseguendo il programma P in uno stato della memoria in cui è vero p raggiungo uno stato finale in cui è vero q.
$\sigma \models Q$ significa che il σ è uno stato che soddisfa Q.

Struttura di una dimostrazione

Definizione di Invariante

È una proprietà utile alla derivazione dei programmi in cui compare un'iterazione (ciclo while). Deve essere soddisfatta prima del ciclo, durante le sue esecuzioni e al termine del ciclo stesso.

Forma generale di una regola di derivazione

È costituita da una serie di premesse(anche l'insieme vuoto è una premessa valida, ovvero non è necessaria alcuna premessa) che se provate, permettono di derivare la conclusione.

Eventi in Conflitto

Due eventi e1, e2, in una configurazione c in B di una rete elementare N=(B,E,F) sono in conflitto se c[e1> e c[e2> ma !c[{e1,e2}>. Ovvero, se entrambi sono abilitati (possono scattare) ma l'occorrenza di uno disabilità l'altro.

Eventi Concorrenti

Due eventi e1, e2, in una configurazione c in B di una rete elementare N=(B,E,F) sono concorrenti se e solo se sono abilitati entrambi nella stessa configurazione: c[{e1,e2}>

Definizione formale di processo non sequenziale

Sia $\sum = (S, T, F, c_{in})$ un sistema elementare senza contatti e finito ($S \cup T$ è finito).

Diciamo che $< N = (B, E, F); \phi >$ è un processo non sequenziale di $\sum$ se e solo se:

1) *(B, E, F)* è una rete di occorrenze (si ammettono posti isolati)
2) $\phi : B \cup E \rightarrow S \cup T$ è una mappa dove:
   a) $\phi(B) \subseteq S$ e $\phi(E) \subseteq T$
   b) $\forall x_1, x_2 \in (B \cup E) \mid \phi(x_1) = \phi(x_2) \rightarrow (x_1 \leq x_2) \vee (x_2 \leq x_1)$
   c) $\forall e \in E \mid \phi(\circ e) = \circ \phi(e) \wedge \phi(e \circ) = \phi(e) \circ$
   d) $\phi(Min(N)) = c_{in}$
      Dove $Min(N) = \{x \in (B \cup E) \mid !\exists y \; per \; cui \; (y, x) \in F\}$ (stati locali iniziali)

Se $< N = (B, E, F); \phi >$ è un processo non sequenziale di $\sum = (S, T, F, c_{in})$ sistema elementare finito e senza contatti, allora:

1) *N = (B, E, F)* è k-densa
2) $\forall k \subseteq B$, k b-taglio di *N* è tale che *k* è finito ed $\exists c \in C_{\sum} \mid \phi(k) = c$

Sostanzialmente i B-tagli corrispondono ai casi raggiungibili.

Significato di bisimulazione

Se due processi sono bisimili allora possiamo scambiare l'uno con l'altro mantenendo inalterato il risultato

Congruenza

Bisimulazione e reti di Petri

Che cos'è una strategia?

Una strategia è un insieme di mosse (transizioni) che vengono fatte sui processi in esame durante il "gioco" dell'attaccante difensore

In quali casi vince il difensore?

Il difensore vince se l'attaccante non ha nessuna possibile strategia vincente, cioè se per ogni possibile mossa dell'attaccante il difensore può rispondere con una mossa valida.
Il difensore vince sse i due processi sono bisimili.

Differenze LTL, CTL

Con LTL posso esprimere proprietà che devono valere per forza di cose su tutti i cammini; con CTL, grazie all'operatore E, posso esprimere proprietà che valgono anche solo per un cammino.
Entrambe vengono utilizzate per definire proprietà per i cammini di un certo Kripke; sono entrambe logiche temporali utilizzate per specificare proprietà e caratteristiche da esaminare. In quanto logiche temporali appartengono alla famiglia delle logiche modali. CTL ed Ltl condividono tra loro gli operatori (F, X, G, U, R, W), CTL ha una capacità espressiva superiore ad LTL.

Espressioni CCS

Espressioni CCS:

Data una P espressione CCS definiamo altre espressioni CCS come:

- NIL
- $k \in K$
- $\alpha \cdot P$ con $\alpha \in Act$
- $\sum P_i = P_1 + \ldots + P_n$ (cioè un OR)
- $P_1 | P_2$: esecuzione in parallelo
- $P \backslash L$: restrizione ad $L \subseteq A$
- $P[f]$: relabeling con $f : Act \to Act$

A cosa servono reti di petri e ccs

Servono per descrivere e modellare un sistema concorrente

Cos'è il calcolo mu

Supponiamo una restrizione di CTL con solo X come operatore, ottenendo quindi solo EXa oppure AXa

Consideriamo la formula $f(a) = EFa = a \vee EXa \vee EXEXa \vee \ldots = a \vee EXf(a)$

Posso scrivere questa formula come: $\mu y(a \vee EXy)$

La formula AGa può essere scritta come: $\nu y(a \wedge AXy)$

Sia **f = AFb = b V AXf = μy.b V AXy**
allora S $\subseteq$ Q F(S) = [[b]] U { q $\in$ Q | se per ogni q $\rightarrow$ q', allora q' $\in$ S }

se invece consideriamo **f = EFb = μy.b V EXy** allora
allora S $\subseteq$ Q F(S) = [[b]] U {q $\in$ Q | $\exists$ q' $\in$ S | q $\rightarrow$ q' }

Sia invece **g = AGa = νy.a $\wedge$ AXy**
allora S $\subseteq$ Q G(S) = [[a]] $\cap$ { q $\in$ Q | se per ogni q $\rightarrow$ q', allora q' $\in$ S }

Sia invece **g = EGa = νy.a $\wedge$ EXy**
allora S $\subseteq$ Q G(S) = [[a]] $\cap$ { q $\in$ Q | $\exists$ q $\rightarrow$ q', allora q' $\in$ S }

Questo linguaggio si chiama μ-calculus, un linguaggio più semplice di CTL ma dalle stesse capacità espressive

Sintassi formule μ-calculus F_μ

- p $\in F_\mu$, $\forall$ p $\in$ P con P insieme di proposizioni
- y $\in F_\mu$ $\forall$ y variabile proposizionale
- !a $\in F_\mu$
- {a $\wedge$ b, a V b, a $\rightarrow$ b} $\in F_\mu$
- EXa, AXa $\in F_\mu$
- μyp(y) con p $\in F_\mu$(y)
- νyp(y) con p $\in F_\mu$(y)

Definizione Modello di Kripke

Un modello di Kripke è definito come $M = (S, T, v)$

- S = insieme di stati
- T = insieme di transizioni: $\forall s \in S, \exists (s, s^1) \in T$. Diciamo brutalmente che da ogni palletta deve uscire almeno un arco. Se non ne esce nessuno si mette il cappio. Così si ottiene sempre un cammino infinito.
- 2^P = insieme delle parti dell'insieme delle proposizioni atomiche (vere all'interno delle pallette)
- v = funzione $v : S \rightarrow 2^P$

La funzione v(s) indica le formule(proposizioni atomiche) vere nello stato s

Definizione Precondizioni e postcondizioni

$$pre(x) : {}^{\circ}x = \{y \in B \cup E | (y, x) \in F\}$$
$$post(x) : x^{\circ} = \{y \in B \cup E | (x, y) \in F\}$$