Corso di Laurea Magistrale in Informatica
Modelli e Computazione - Modelli della Concorrenza

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 relazione di transizione debole è definita come segue:
$p \Rightarrow^{\alpha} p'$ se e solo se:
- $p \rightarrow^{\tau *} \rightarrow^{\alpha} \rightarrow^{\tau *} p'$, se $\alpha \neq \tau$
- $p \rightarrow^{\tau *} p'$, 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 relazione di transizione forte, nel caso della Bisimulazione debole il difensore usa sempre la relazione di transizione debole e se deve rispondere ad una azione τ dell'attaccante, il difensore può eseguire una qualsiasi sequenza di τ oppure può non fare nessuna mossa.

Bisimulazione (debole) come Gioco

Ci sono 2 giocatori, con due diversi ruoli: il giocatore I o Attaccante cerca di dimostrare che i due processi *non* sono Bisimili; il giocatore II, il Difensore, cerca di dimostrare invece che i processi sono Bisimili.

Una *partita* di un gioco $G(p_0, q_0)$ è costituita da una sequenza finita o infinita di configurazioni: $(p_0, q_0), (p_1, q_1), ..., (p_i, q_i), ...$, dove data la configurazione (p_i, q_i), la successiva (p_{i+1}, q_{i+1}) è determinata da una delle seguenti due mosse:

- l'Attaccante sceglie una transizione $p_i \rightarrow^{\alpha} p_{i+1}$ e allora il Difensore sceglie una (sequenza di) transizione con la stessa etichetta $q_i \Rightarrow^{\alpha} q_{i+1}$.
- l'Attaccante sceglie una transizione $q_i \rightarrow^{\alpha} q_{i+1}$ e allora il Difensore sceglie una (sequenza di) transizione con la stessa etichetta $p_i \Rightarrow^{\alpha} p_{i+1}$.

L'Attaccante sceglie sempre per primo e decide ogni volta su quale dei due processi agire; il Difensore, conoscendo la mossa fatta dall'Attaccante, deve scegliere una mossa corrispondente sull'altro processo. Mentre l'Attaccante usa sempre la relazione di transizione forte, il Difensore usa quella debole, cioè può eseguire tutte le τ-transizioni che vuole.

La partita continua fino a che uno dei due giocatori vince. Se in una configurazione (p_i, q_i) uno dei due processi può fare una transizione a e l'altro no, allora i due processi p_i e q_i sono distinguibili. Di conseguenza una configurazione è di vittoria per l'Attaccante se in quella configurazione i due processi sono distinguibili e quindi l'Attaccante è in grado di scegliere una transizione in uno dei due processi, mentre il Difensore non è in grado di rispondere con una scelta corrispondente sull'altro processo. Ogni partita che non raggiunge mai una tale configurazione viene vinta dal Difensore. Il Difensore vince quindi se la partita è *infinita* o se raggiunge una configurazione (nil, nil) in cui non è più possibile alcuna mossa (l'Attaccante non può fare alcuna mossa).

Diverse partite possono concludersi con vincitori diversi. Tuttavia, per ogni gioco, *uno solo dei due giocatori è in grado di vincere ogni partita indipendentemente dalle mosse del suo avversario.* (Due processi o sono bisimili o non lo sono.) Per questo è essenziale la nozione di *strategia*.

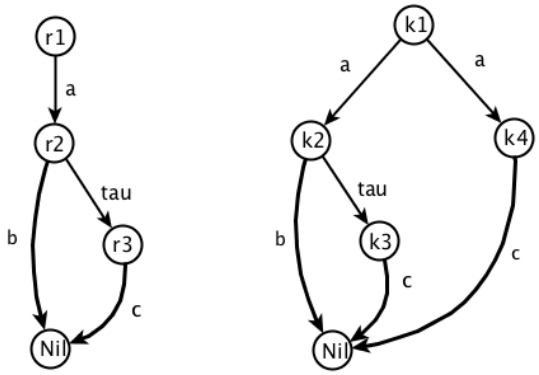
Una strategia per un giocatore è un insieme di regole che indicano di volta in volta che mossa fare. Tali regole non dipendono dalla storia precedente della partita ma solo dalla configurazione corrente. Per l'Attaccante la strategia dovrà suggerire in ogni configurazione su quale processo agire e quale mossa fare; per il Difensore invece la strategia deve suggerire solo quale mossa fare a seconda della mossa fatta dall'avversario. Un giocatore usa una strategia se tutte le sue mosse obbediscono alle regole di quella strategia. Una strategia è *vincente* se il giocatore vince ogni partita in cui usa quella strategia.

Proposizione 1
Per ogni gioco $G(p_0, q_0)$, solo uno dei due giocatori ha una strategia vincente.

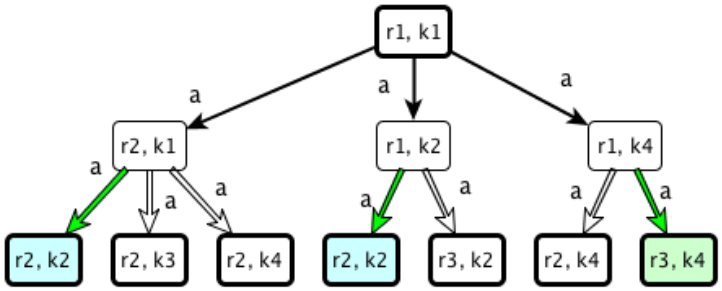
Proposizione 2
L'Attaccante ha una strategia vincente per $G(p_0, q_0)$ se e solo se $p_0 \not\approx^{Bis} q_0$.
Il Difensore ha una strategia vincente per $G(p_0, q_0)$ se e solo se $p_0 \approx^{Bis} q_0$

In generale quindi per dimostrare che i processi non sono Bisimili, bisogna mostrare che in ogni configurazione, l'Attaccante è in grado di scegliere su quale processo operare e con quale azione, in modo che per ogni successiva mossa del Difensore, l'Attaccante ha almeno una mossa che lo porterà a vincere. Per dimostrare che i processi sono Bisimili, bisogna mostrare che, per ogni mossa dell'Attaccante, il Difensore ha almeno una mossa che lo porterà a vincere.

Esempio 1 (processi bisimili)
Consideriamo i seguenti processi CCS
r_1 = `a.( b.nil + `τ`.c.nil )`
k_1 = `a.( b.nil + `τ`.c.nil ) + a.c.nil`
e i rispettivi sistemi di transizioni etichettati dati nella seguente figura



Una possibile rappresentazione del gioco è data nella seguente figura, dove ogni possibile partita è rappresentata da un cammino nell'albero dalla radice ad una foglia; in ogni cammino i nodi disegnati con tratto più grosso rappresentano le varie configurazioni della partita, quelle con tratto più leggero lo stato in cui in ogni mano ha appena 'mosso' l'Attaccante e deve ancora 'muovere' il Difensore.



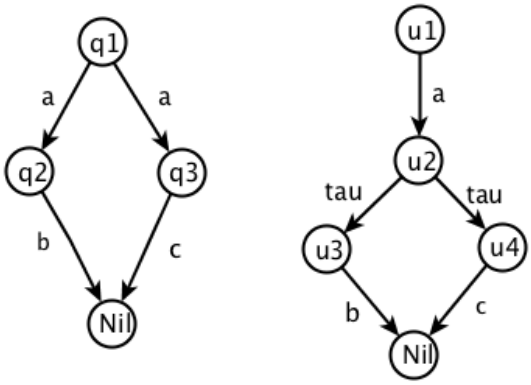
I due processi r_1 e k_1 sono debolmente bisimili ($r_1 \approx^{Bis} k_1$), infatti il Difensore ha la seguente strategia vincente:

- se l'Attaccante esegue sul processo r_1 l'azione a: $r_1 \rightarrow^a r_2$,
  il Difensore, che avrebbe tre possibilità, esegue: $k_1 \Rightarrow^a k_2$,
  e vince, infatti i sistemi di transizioni r_2 e k_2 sono isomorfi e quindi sono debolmente Bisimili.
- se l'Attaccante esegue sul processo k_1 l'azione a: $k_1 \rightarrow^a k_2$,
  il Difensore, che avrebbe due possibilità, esegue: $r_1 \Rightarrow^a r_2$,
  e vince anche in questo caso, infatti viene raggiunta la medesima configurazione (r_2, k_2) di prima.

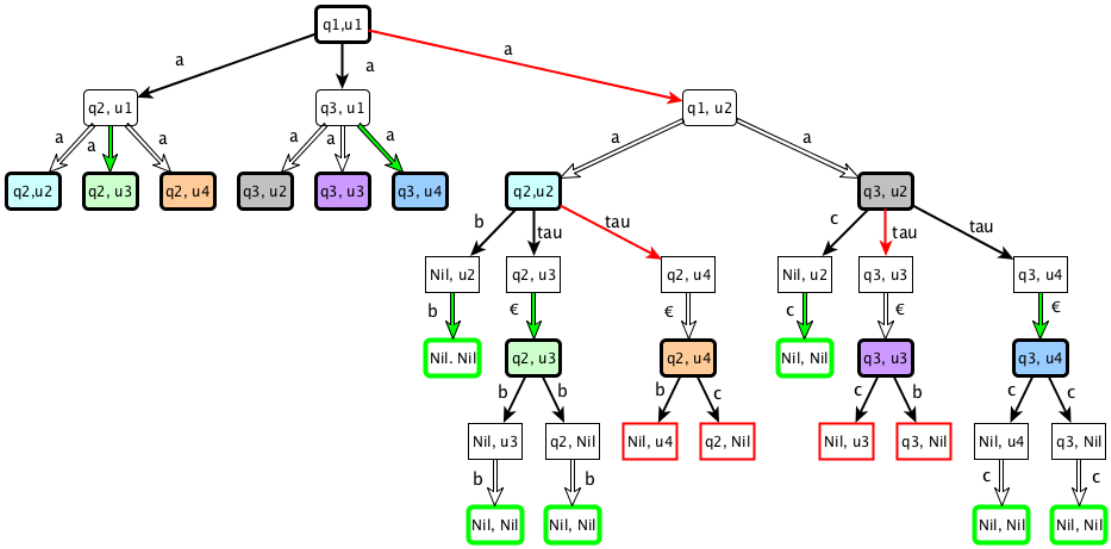
- se invece l'Attaccante esegue sul processo k_1 l'azione a: $k_1 \rightarrow^a k_4$,
  il Difensore, che avrebbe due possibilità, esegue: $r_1 \Rightarrow^a r_3$,
  e anche in questo caso vince perchè la configurazione raggiunta (r_3, k_4) è tale che i due sistemi di transizioni sono ancora isomorfi, da entrambi è possibile solo l'azione osservabile c.

Concludendo, r_1 e k_1 sono bisimili, $R = \{(r_1, k_1), (r_2, k_2), (r_3, k_3), (r_3, k_4), (Nil, Nil)\}$ è una relazione di bisimulazione.

Esempio 2 (processi non bisimili)
Consideriamo i seguenti processi CCS
q_1 `= a.b.nil + a.c.nil`
u_1 `= a.(` τ`.b.nil +` τ`.c.nil )`
e i rispettivi sistemi di transizioni etichettati dati nella seguente figura



Le possibili partite sono rappresentate dai cammini dalla radice alle foglie dell'albero dato in figura. (Si noti che l'albero non è completo, le prime sei foglie sono in realtà radici dei sottoalberi identificati dal colore dei nodi.)



I due processi q_1 e u_1 non sono debolmente bisimili ($q_1 \not\approx^{Bis} u_1$), infatti l'Attaccante ha la seguente strategia vincente:

- l'Attaccante esegue sul processo u_1 l'azione a: $u_1 \rightarrow^a u_2$,

a questo punto il Difensore ha due possibilità:

- $q_1 \Rightarrow^a q_2$

  in questo caso, nella configurazione (q_2, u_2), l'Attaccante esegue $u_2 \rightarrow^\tau u_4$ a cui il Difensore non può che rispondere con l'azione nulla ($q_2 \Rightarrow^\tau q_2$) restando in q_2 e perdendo perché $q_2 \not\approx^{Bis} u_4$, infatti da q_2 è possibile solo l'azione b, mentre da u_4 solo l'azione c.

- $q_1 \Rightarrow^a q_3$

  in questo caso, nella configurazione (q_3, u_2), l'Attaccante esegue $u_2 \rightarrow^\tau u_3$ a cui il Difensore non può che rispondere con l'azione nulla ($q_3 \Rightarrow^\tau q_3$) restando in q_3 e perdendo perché $q_3 \not\approx^{Bis} u_3$, infatti da q_3 è possibile solo l'azione c, mentre da u_3 solo l'azione b.

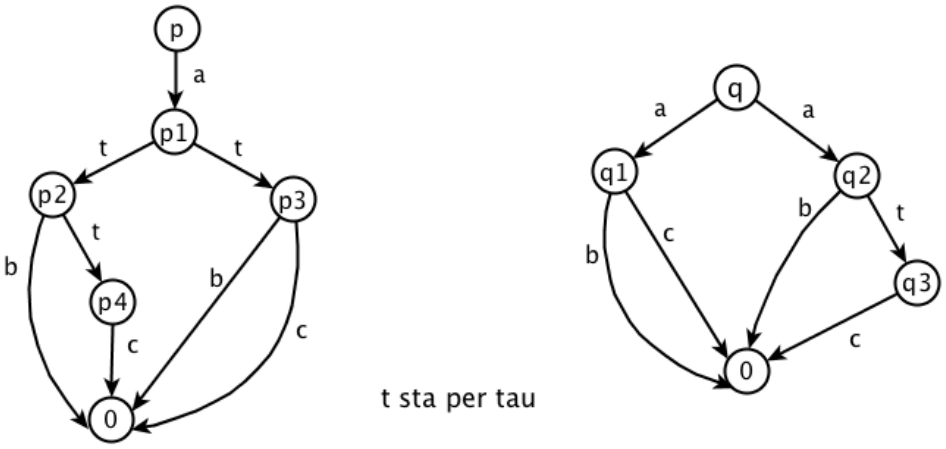
Si noti che se L'Attaccante avesse fatto inizialmente un'azione a sul processo $q1$, $q_1 \rightarrow^a q_2$ o $q_1 \rightarrow^a q_3$, allora avrebbe perso, infatti il Difensore avrebbe potuto in entrambi i casi eseguire sul processo u_1 un'azione a seguita da τ, passando nella configurazione (q_2, u_3) o nella configurazione (q_3, u_4) e tali configurazioni sono ovviamente entrambe di vittoria per il Difensore.

Esempio 3 (processi non bisimili)
Consideriamo i seguenti processi CCS

```
p = a.( τ.(b.nil + τ.c.nil) + τ.(b.nil + c.nil))

q = a.(b.nil + c.nil) + a.(b.nil + τ.c.nil)
```

e i rispettivi sistemi di transizioni etichettati dati in figura



I due processi p e q non sono debolmente bisimili ($p \not\approx^{Bis} q$), infatti l'Attaccante ha la seguente strategia vincente:

- l'Attaccante esegue sul processo p l'azione a: $p \rightarrow^a p_1$,

a questo punto il Difensore ha tre possibilità:

- $q \Rightarrow^a q_1$

  in questo caso l'Attaccante esegue $p_1 \rightarrow^\tau p_2$ a cui il Difensore non può che rispondere con l'azione nulla ($q_1 \Rightarrow^\tau q_1$) restando in q_1 e perdendo perché, come mostrato dopo, $p_2 \not\approx^{Bis} q_1$

- $q \Rightarrow^a q_2$

  in questo caso l'Attaccante esegue $p_1 \rightarrow^\tau p_3$. Il Difensore ha due possibilità: 1) il Difensore risponde con l'azione nulla ($q_2 \Rightarrow^\tau q_2$) restando in q_2 che abilita sia b che c, ma perde perché, come mostrato dopo, $p_3 \not\approx^{Bis} q_2$. 2) Se il Difensore avesse risposto invece con l'azione $q_2 \Rightarrow^\tau q_3$ avrebbe a maggior ragione perso perché ovviamente $p_3 \not\approx^{Bis} q_3$ in quanto in p_3 è ancora possibile l'azione b, che non è invece eseguibile da q_3.

- $q \Rightarrow^a q_3$

  questa mossa è chiaramente sbagliata, in quanto ora sul processo q_3 non è più possibile eseguire b, e l'Attaccante vince facilmente.

Mostriamo ora che $p_2 \not\approx^{Bis} q_1$ ($p_3 \not\approx^{Bis} q_2$).
L'Attaccante ha la seguente strategia vincente:
l'Attaccante esegue sul processo p_2 l'azione τ: $p_2 \rightarrow^\tau p_4$,
il Difensore deve rispondere con l'azione nulla ($q_1 \Rightarrow^\tau q_1$) restando in q_1;
a questo punto l'Attaccante vince perché cambia processo ed esegue b in q_1: $q_1 \rightarrow^b 0$; e infatti il Difensore non può eseguire b in p_4.