

Lezione 1 MSC

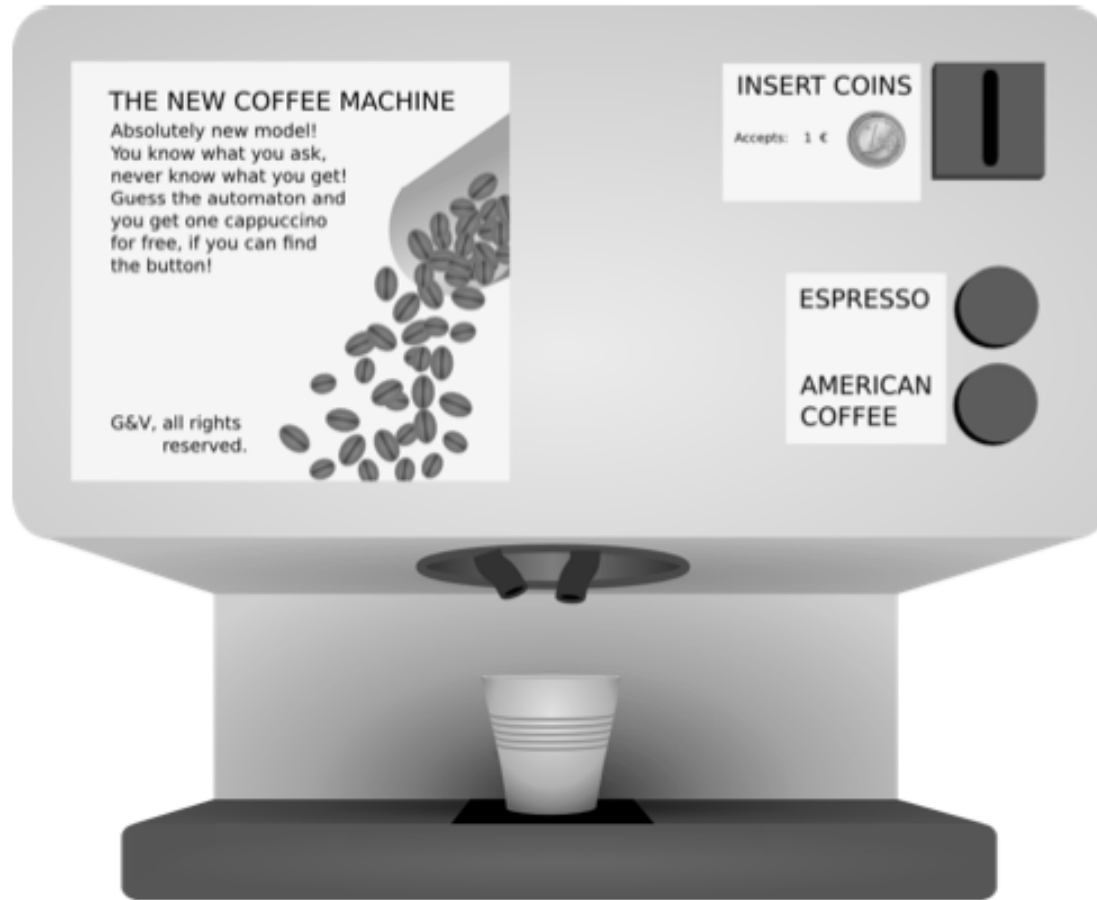
Overview del corso – Prima parte

Roberto Gorrieri

Modelli e Sistemi Concorrenti

- Come modellare un sistema concorrente
- Quando due sistemi sono equivalenti
- Quali facilitazioni linguistiche sono disponibili
- Proprietà di congruenza e assiomatizzazioni
- Logiche di specifica di proprietà
- Strumenti automatici di verifica
- Espressività
- Reti di Petri

Un semplice sistema reattivo: vending machine



Caratteristiche della vending machine

- Caratteristiche **visibili**: punto d'inserzione delle monete, bottoni per selezionare le bevande e apertura per raccogliere la bevanda (punti sui quali l'utente può interagire)
- Caratteristiche **invisibili**: fornitura di acqua e di caffè macinato (su cui l'utente non può interagire, ma che possono influenzare il funzionamento della macchina)

Come costruire un modello della vending machine

Individuare le astrazioni opportune: **azioni** eseguibili sulla (**input**) macchina (e.g., inserzione di una moneta) o dalla (**output**) macchina (e.g., restituzione di un caffè) e come gli **stati** interni in cui la macchina si può trovare (ad esempio, quando è pronta per partire) evolvono con l'esecuzione di quelle azioni (ad esempio, come cambia lo stato interno dopo l'inserimento della moneta).

➔ macchine a stati/transizioni (simili agli automi), detti **labeled transition systems (LTS)**

Assunzioni: sistemi discreti untimed

Le **transizioni** sono etichettate da **azioni** che sono:

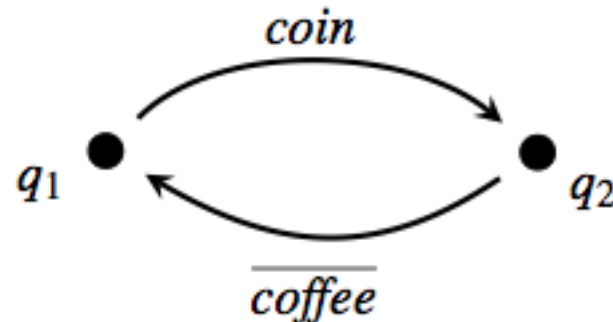
- **atomiche** (all-or-nothing),
- **istantanee** (cioè non impiegano tempo; non si modella il tempo fisico che passa ma solo l'ordine temporale (*prima/dopo*) delle azioni; si può immaginare che il tempo passi negli stati).

Gli **stati** costituiscono:

- Un insieme finito o infinito contabile (**sistema discreto**) - in antitesi coi sistemi continui usati, ad esempio, per modellare sistemi real-time

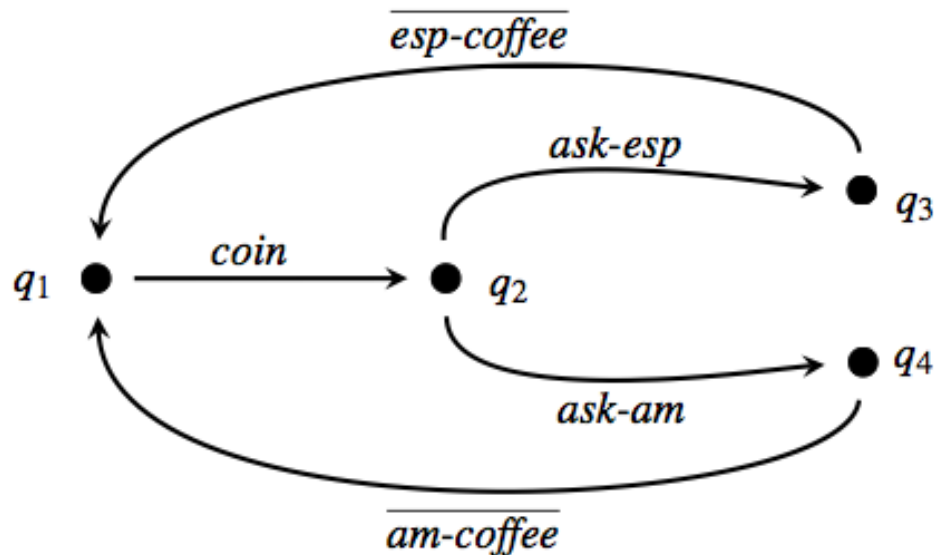
Primo modello, molto astratto

- Punto di vista input/output (nota che, per convenzione, le azioni di output sono barrate): astraggo i due tipi di bevanda in una sola generica.
- Comportamento banale: alternarsi delle due azioni (e.g., non è possibile inserire due monete di fila)
- Esercizio: modificare il modello per permettere l'inserzione di due monete di fila
- Comportamento irrealistico: non mostra le opzioni, servizio sempre disponibile, ...



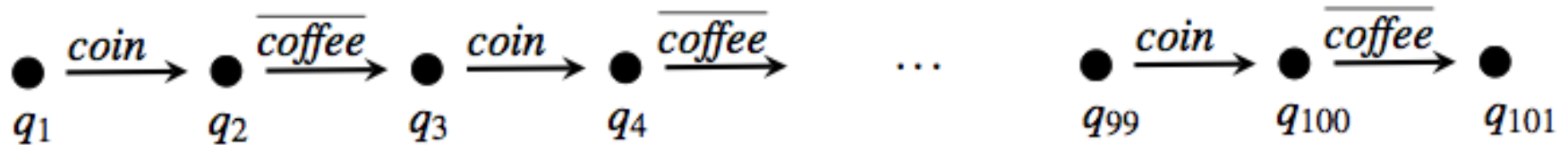
Secondo modello – black-box view

- Le azioni sono tutte e sole quelle relative ai bottoni/aperture (modello della **interaction interface**!)
- Caratteristiche: differenziazione della bevanda, non tiene credito, servizio sempre disponibile.



Terzo modello – service unavailable

- Caratteristiche: bevanda astratta, non tiene credito, servizio non disponibile dopo 50 caffè. (Stato q_{101} è di **deadlock**)



- Esercizio 1: aggiungi una azione *refill* (eseguibile solo da un operatore) che porta dallo stato 101 allo stato 1, che modella il rifornimento di acqua e caffè macinato.
- Esercizio 2: modifica il secondo modello, tenendo presente che il servizio non è più disponibile dopo 2 bevande erogate.

Quanti modelli? Quale modello?

- Modello **non è unico**: dipende dal livello di accuratezza che si vuole raggiungere.
- Spesso, più il modello è accurato (cioè fedele al comportamento reale), più il modello è complesso (in termine di numero di stati e transizioni).
- Spesso, più il modello è complesso, più è difficile da analizzare (a volte perfino impossibile, quando il numero degli stati è infinito).
- **Scelta del modello**: in base a quali caratteristiche del sistema si vogliono studiare e alla possibilità del modello di essere analizzato.

Modelli e Sistemi Concorrenti

- Come modellare un sistema concorrente
- Quando due sistemi sono equivalenti
- Quali facilitazioni linguistiche sono disponibili
- Proprietà di congruenza e assiomatizzazioni
- Logiche di specifica di proprietà
- Strumenti automatici di verifica
- Espressività
- Reti di Petri

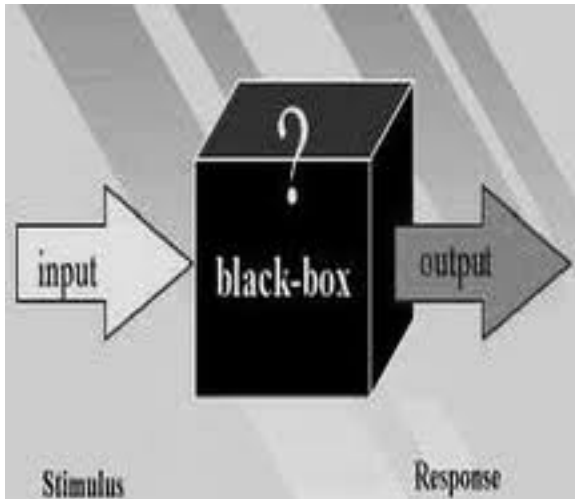
Equivalenza tra sistemi

Perché ci interessa vedere quando due (modelli di) sistemi sono tra loro equivalenti?

- **Intercambiabilità** perché hanno lo stesso **comportamento**. Ma cos'è il **comportamento** di un sistema? Abbiamo molte possibilità, alcune sensate, altre meno.... In particolare, è poco sensata l'equivalenza a tracce di esecuzione (automi)
- **Equivalence-checking**: Confronto tra modello astratto (**specifica**) e modello concreto (**implementazione**)

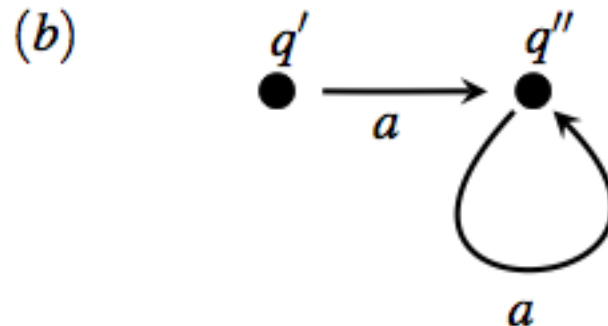
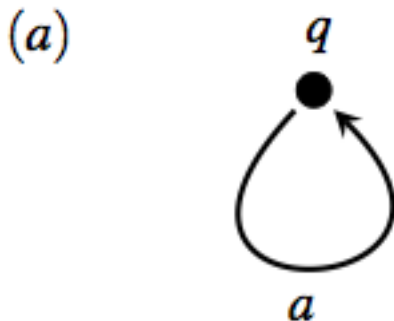
Osservatore: Black-box experiment

- Un osservatore è in grado di vedere solo parte di un sistema, ad esempio solo l'interfaccia (cioè quali input accetta e quali output fornisce di conseguenza), ma non può vedere come è fatto dentro, in particolare non sa quanti stati interni abbia il sistema.
- Due sistemi sono equivalenti (ovvero hanno lo stesso comportamento) se nessun osservatore riesce a vederne alcuna differenza.
- Diverse equivalenze a seconda del potere dell'osservatore: da semplice osservatore che sa solo mandare in esecuzione il sistema (e vede le azioni su un display), a vero utente che interagisce e che magari sa anche far andare più copie in parallelo dello stesso sistema.



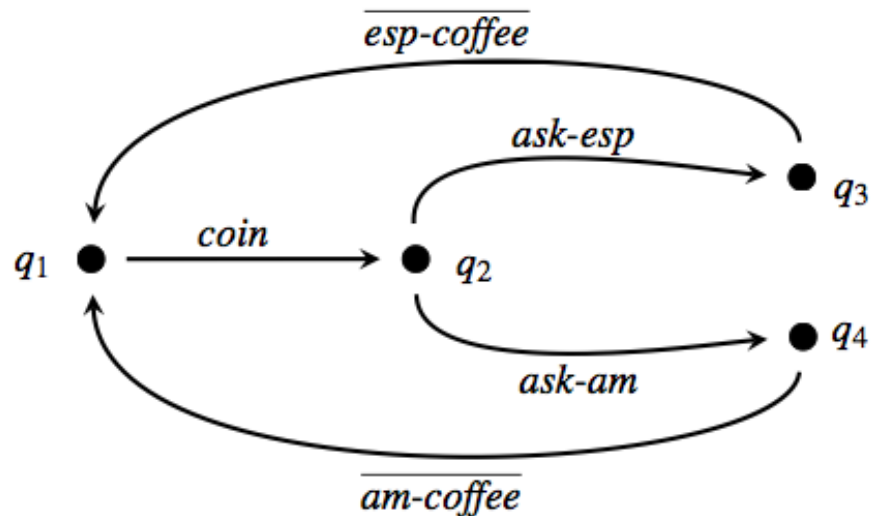
Equivalence-checking

- **Specifica** in (a) ed **implementazione** in (b) di un semplicissimo sistema che può eseguire infinite a
- Nessun “osservatore” sarebbe in grado di distinguerli. Consideriamo (b) una corretta implementazione di (a) perché i due sono equivalenti (rispetto ad una nozione di equivalenza non ancora specificata)



Semantica a tracce

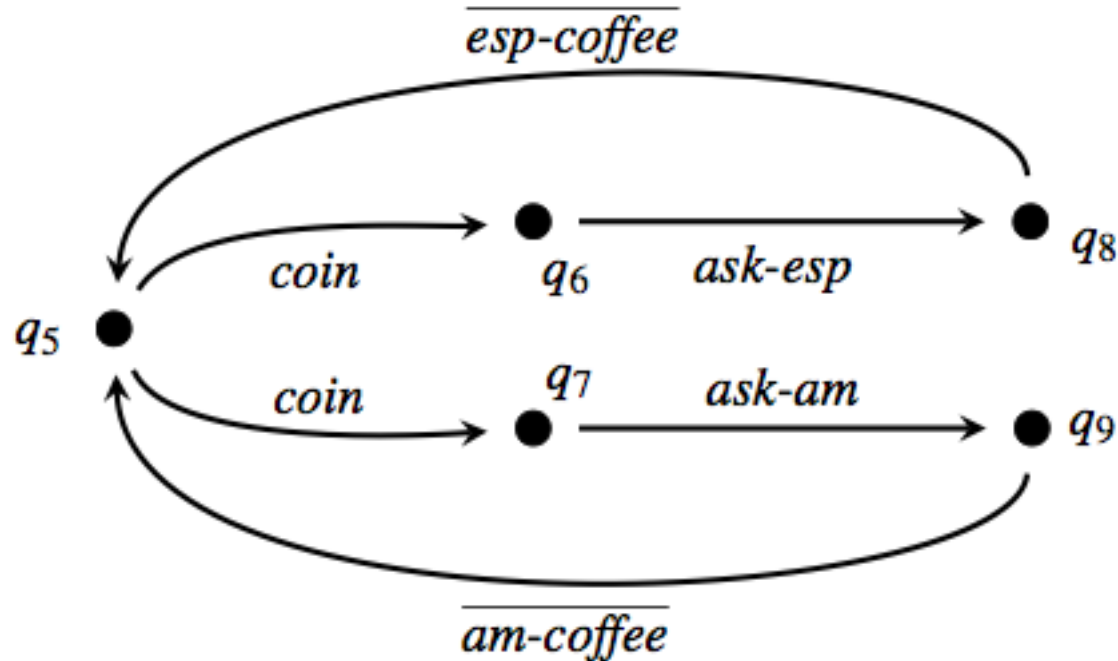
- Simile alla semantica degli automi (come se tutti gli stati fossero di accettazione)



$\text{Tracce}(q_1) = \{\varepsilon, \text{coin}, \text{coin ask-esp}, \text{coin ask-am}, \text{coin ask-esp esp-coffee}, \dots\}$

Scelta fra le bevande effettuata **dopo** l'inserimento della moneta: l'utente sceglie la bevanda.

Nondeterminismo: impolite vending machine



In questo caso è la macchina, e non l'utente, che sceglie la bevanda! (scelta interna)

Equivalenza a tracce

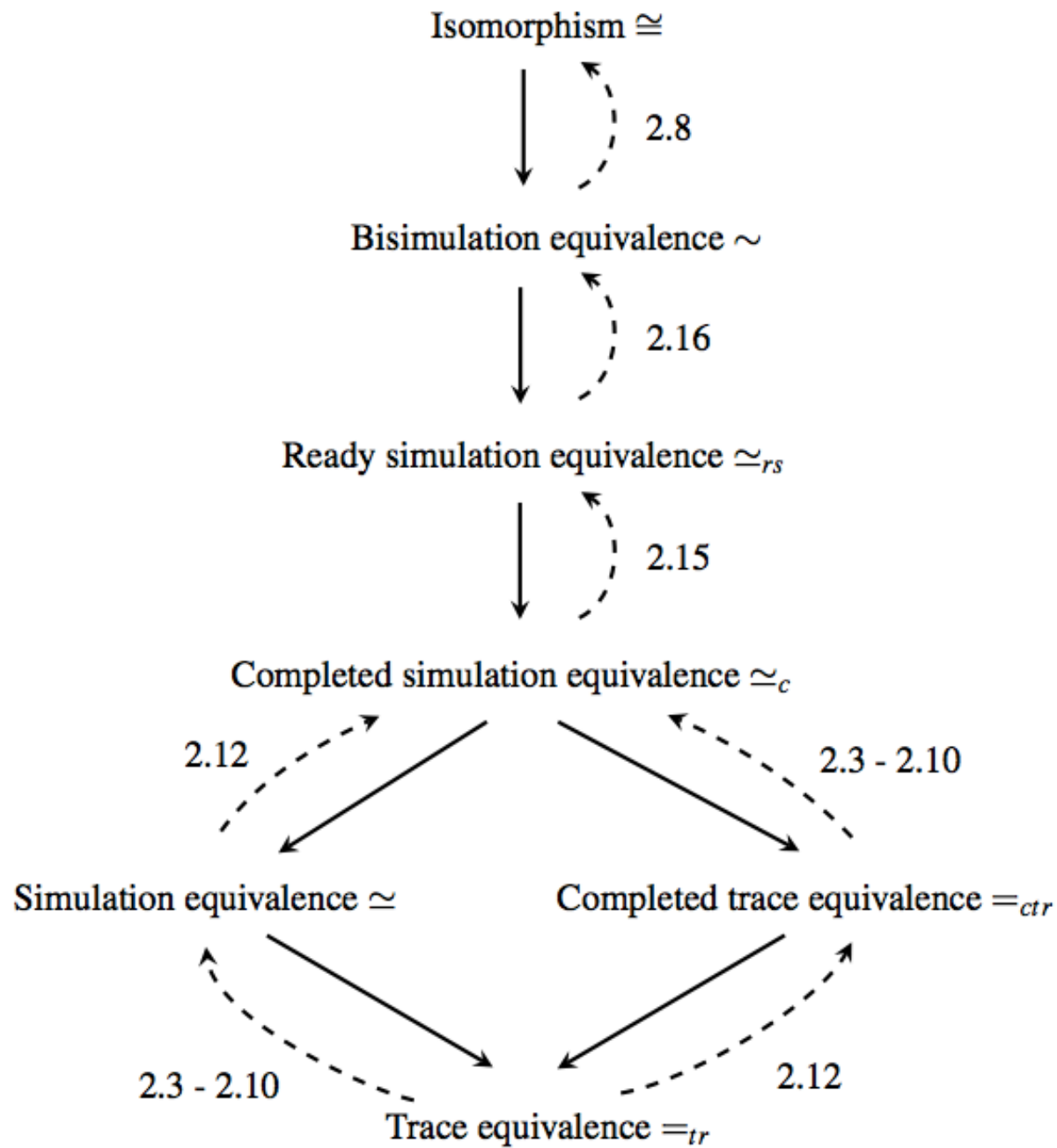
- Due stati q_1 e q_2 sono equivalenti a tracce quando $\text{Tracce}(q_1) = \text{Tracce}(q_2)$.
- Osserva che le due vending machine sono equivalenti a tracce
 - $\text{Tracce}(q_5) = \{\varepsilon, \text{coin}, \text{coin ask-esp}, \text{coin ask-am}, \text{coin ask-esp esp-coffee}, \dots\} = \text{Tracce}(q_1)$
- Allora, semantica a tracce troppo astratta, perché “the timing of a choice can be a crucial aspect of the behavior and cannot be neglected”

Altre equivalenze

Oltre alla **equivalenza a tracce** (troppo astratta), abbiamo almeno:

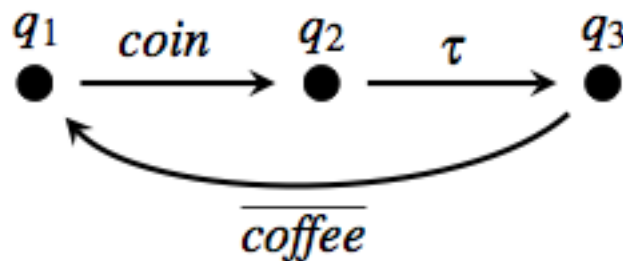
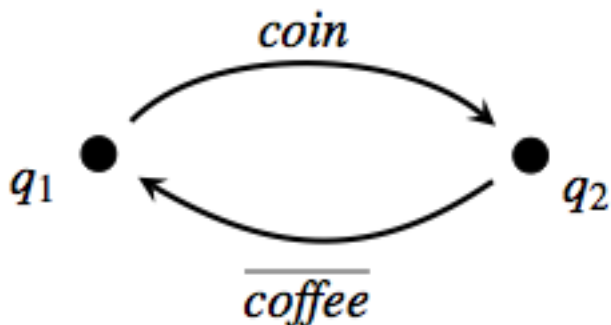
- **Isomorfismo** di LTS (troppo concreta)
- **Simulation preorder and equivalence** (interessante, ma piuttosto astratta)
- **Bisimulation equivalence** (buone proprietà ed intuitivamente interessante)

Non ce n'è una migliore di tutte le altre, ma molte che possono avere una loro funzione.



Azione non osservabile

- Azione interna su cui non è possibile interagire, di solito rappresentata con tau τ
- Tutte le nozioni di equivalenza viste possono essere adattate a tener conto del tau
- I due processi sotto devono essere considerati equivalenti



A volte il tau ha effetti osservabili

Il tau può escludere azioni possibili, cioè operare delle scelte! Quindi non è totalmente non osservabile

N.B. q_7 è un deadlock

