Lezione 19 MSC
Bisimulation equivalence come punto fisso

Roberto Gorrieri

Bisimulazione come punto fisso

- Riformuliamo ~ come massimo punto fisso di una opportuna funzione F che trasforma relazioni binarie R su stati
- Useremo l'algoritmo per calcolare massimi punti fissi come algoritmo per determinare la relazione ~ .
- Osservazione: 2^{QxQ}, ovvero l'insieme di tutte le relazioni binarie su Q, è un reticolo completo (finito se Q è finito), con Top = QxQ.

Funzionale F: Trasformatore di relazioni binarie

Definition 2.25. Given an lts $TS = (Q, A, \rightarrow)$, functional $F : \wp(Q \times Q) \rightarrow \wp(Q \times Q)$ (i.e., a transformer of binary relations over Q) is defined as follows. If $R \subseteq Q \times Q$, then $(q_1, q_2) \in F(R)$ if and only if for all $\mu \in A$

- $\forall q_1'$ such that $q_1 \xrightarrow{\mu} q_1'$, $\exists q_2'$ such that $q_2 \xrightarrow{\mu} q_2'$ and $(q_1', q_2') \in R$
- $\forall q_2'$ such that $q_2 \xrightarrow{\mu} q_2'$, $\exists q_1'$ such that $q_1 \xrightarrow{\mu} q_1'$ and $(q_1', q_2') \in R$.

Proposition 2.10. *For any lts $TS = (Q, A, \rightarrow)$, we have that:*

1. *Functional F is monotone, i.e., if $R_1 \subseteq R_2$ then $F(R_1) \subseteq F(R_2)$.*
2. *Relation $R \subseteq Q \times Q$ is a bisimulation if and only if $R \subseteq F(R)$.*

Proof. The proof of (1) derives immediately form the definition of F: if $(q_1, q_2) \in F(R_1)$ then for all $\mu \in A$

- $\forall q_1'$ such that $q_1 \xrightarrow{\mu} q_1'$, $\exists q_2'$ such that $q_2 \xrightarrow{\mu} q_2'$ and $(q_1', q_2') \in R_1$
- $\forall q_2'$ such that $q_2 \xrightarrow{\mu} q_2'$, $\exists q_1'$ such that $q_1 \xrightarrow{\mu} q_1'$ and $(q_1', q_2') \in R_1$.

Since $R_1 \subseteq R_2$, the above implies that for all $\mu \in A$

- $\forall q_1'$ such that $q_1 \xrightarrow{\mu} q_1'$, $\exists q_2'$ such that $q_2 \xrightarrow{\mu} q_2'$ and $(q_1', q_2') \in R_2$
- $\forall q_2'$ such that $q_2 \xrightarrow{\mu} q_2'$, $\exists q_1'$ such that $q_1 \xrightarrow{\mu} q_1'$ and $(q_1', q_2') \in R_2$

which means that $(q_1, q_2) \in F(R_2)$.

The proof of (2) is also immediate: if R is a bisimulation, then if $(q_1, q_2) \in R$ then for all $\mu \in A$

- $\forall q_1'$ such that $q_1 \xrightarrow{\mu} q_1'$, $\exists q_2'$ such that $q_2 \xrightarrow{\mu} q_2'$ and $(q_1', q_2') \in R$
- $\forall q_2'$ such that $q_2 \xrightarrow{\mu} q_2'$, $\exists q_1'$ such that $q_1 \xrightarrow{\mu} q_1'$ and $(q_1', q_2') \in R$.

and, by using the reverse implication, this means that $(q_1, q_2) \in F(R)$, i.e., $R \subseteq F(R)$. Similarly, if $R \subseteq F(R)$, then the condition holding for $F(R)$ holds also for all the elements of R, hence R is a bisimulation. $\square$

~ è il massimo punto fisso di F

- Poiché F è monotona, il teorema del punto fisso di Knaster-Tarski assicura che il massimo punto fisso per F sia

$$Zmax = \bigcup \{R \mid R \subseteq F(R)\}$$

- Ma cos'è Zmax?

  Poiché R è una bisimulazione sse $R \subseteq F(R)$, alllora:

$$Zmax = \bigcup \{R \mid R \text{ è una bisimulazione}\}$$

Ovvero Zmax = ~

Algoritmo per calcolare ~

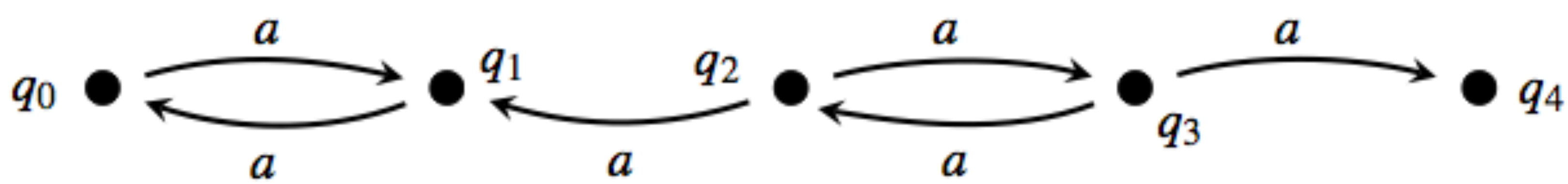
- Se Q è finito, allora 2^{QxQ} è un reticolo completo finito e possiamo applicare l'algoritmo per calcolare il massimo punto fisso di F:

  X := QxQ; Y := F(X);

  While X ≠ Y do { X:=Y; Y:=F(Y) }

  Return X
- Sappiamo che l'algoritmo termina sempre se Q è finito.
- Analogie con l'algoritmo con tabella a scala, visto a LP, per minimizzazione di automi deterministici.

Esempio



Q = {q0, q1, q2, q3, q4}

F^0(QxQ) = QxQ

F^1 (QxQ) = {(q0,q1), (q0,q2), (q0,q3), (q1,q2), (q1,q3), (q2,q3)} + simmetriche + riflessive

F^2(QxQ) = {(q0,q1), (q0,q2), (q1,q2)} + sim + rifl

F^3(QxQ) = {(q0,q1)} + sim + rifl = F^4(QxQ)

Minimizzazione

Dato TS = (Q, A $\rightarrow$) e calcolata ~ con l'algoritmo iterativo, possiamo costruire un lts minimo TS' = (Q', A, $\rightarrow$') dove:

$Q' = \{[q]_{\sim} \mid q \in Q\}$ e $[q]_{\sim} = \{q' \in Q \mid q' \sim q\}$

$\rightarrow' = \{([q]_{\sim}, a, [q']_{\sim}) \mid (q,a,q') \in \rightarrow\}$

LTS minimo rispetto a ~ va a fondere gli stati equivalenti; nell'esempio precedente fonde gli stati q0 e q1

Observe that in the definition of the minimum lts $TS_\sim$, any state $[q]_\sim$ is an equivalence class of states of TS: for all $q, q' \in Q$, $q \sim q'$ if and only if $[q]_\sim = [q']_\sim$. Moreover, if $([q]_\sim, \mu, [q']_\sim)$ is a transition in $TS_\sim$, then for all $q_1 \in Q$ such that $q \sim q_1$, there exists a $q_2 \in Q$ such that $q_1 \xrightarrow{\mu} q_2$ and $q' \sim q_2$, and so $([q_1]_\sim, \mu, [q_2]_\sim) = ([q]_\sim, \mu, [q']_\sim)$. In other words, the definition of $TS_\sim$ is independent of the choice of the representative state q for its equivalence class $[q]_\sim$.

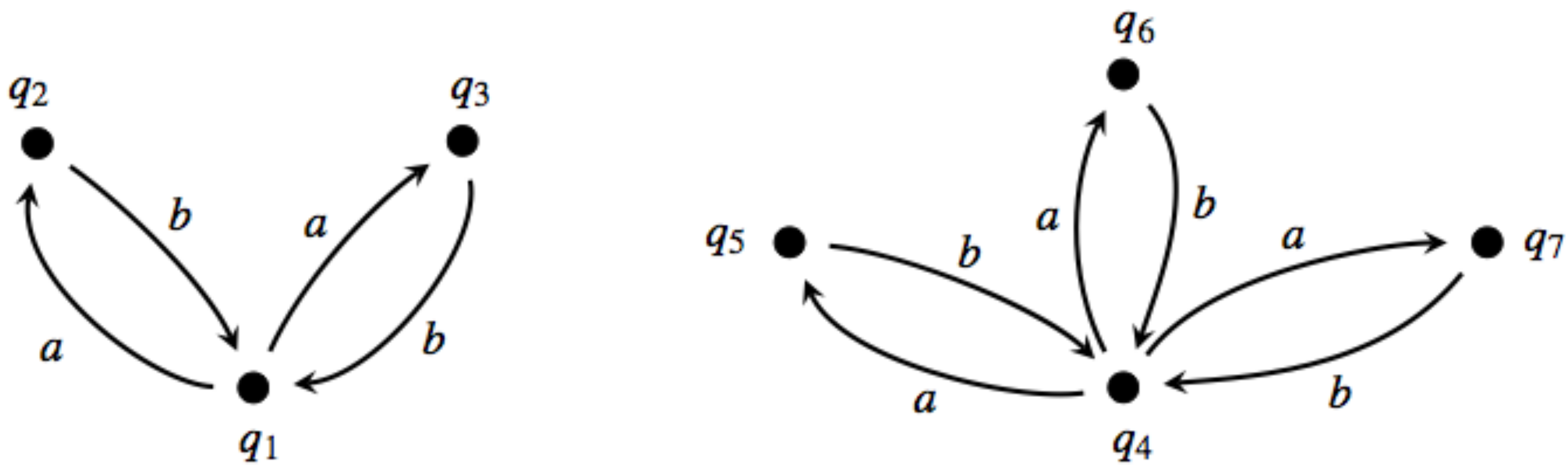
Proposition 2.12. *Given an lts $TS = (Q, A, \rightarrow)$ and its associated minimum lts $TS_\sim = (Q_\sim, A, \rightarrow_\sim)$, as defined in Definition 2.27, the following hold:*

- *$q \sim [q]_\sim$ for all $q \in Q$ and $[q]_\sim \in Q_\sim$, i.e., $TS_\sim$ is a correct realization of TS;*
- *for all $[q]_\sim, [q']_\sim \in Q_\sim$ we have that if $[q]_\sim \sim [q']_\sim$ then $[q]_\sim = [q']_\sim$, i.e., $TS_\sim$ is the minimum (up to isomorphism).*

Proof. For the proof of the first item, consider relation $R \subseteq Q \times Q_\sim$ defined as follows: $R = \{(q, [q]_\sim) \mid q \in Q\}$. It is easy to see that R is a bisimulation.

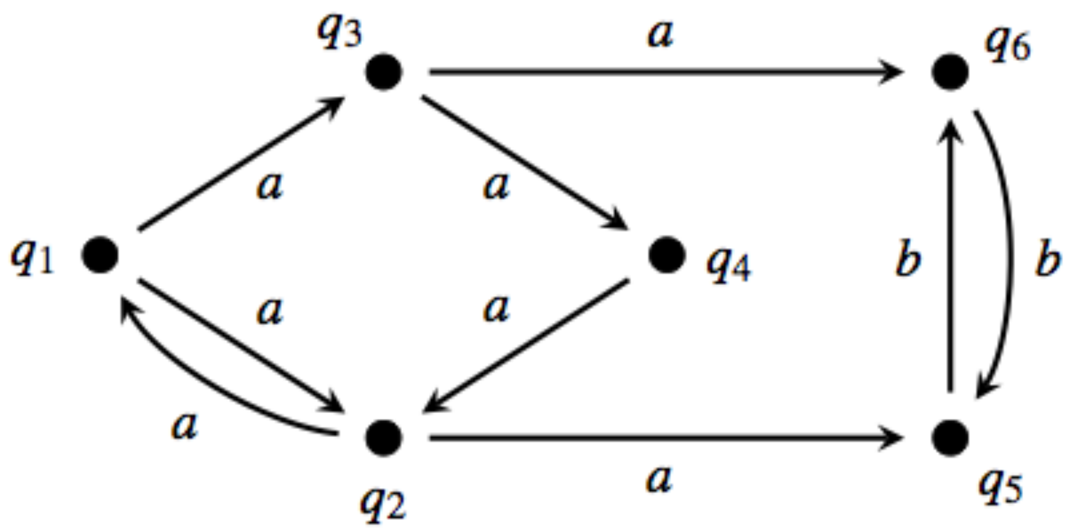
For the proof of second item, we have that $q \sim [q]_\sim$ as well as $q' \sim [q']_\sim$ by the previous item. Therefore, if $[q]_\sim \sim [q']_\sim$, then by transitivity we also have that $q \sim q'$ and so, by construction of $TS_\sim$, we have that $[q]_\sim = [q']_\sim$. $\square$

Esercizio (1)



Provate a calcolare ~ con l'algoritmo. Troverete che esistono solo due classi di equivalenza {q1,q4} e {q2,q3,q5,q6,q7} e quindi lts minimo ha solo due stati.

Esercizio (2)



- Minimizzare questo lts, attraverso il calcolo di ~ con l'algoritmo iterativo.

Minimo lts rispetto a trace equiv.?

- Dato TS=(Q, A, →, q0), possiamo ottenere il minimo deterministico attraverso questi passi:
  - Trasforma TS in deterministico dTS, con la costruzione per sottoinsiemi
  - Calcola ~ sopra dTS (ricorda che ~ e trace equivalence coincidono su lts deterministici)
  - Minimizza dTS rispetto a ~ per ottenere il minimo lts deterministico per TS.
- Tuttavia è possibile che esistano lts nondeterministici equivalenti più piccoli.
- Esercizio: calcola il minimo lts deterministico equivalente a tracce a quello del lucido precedente, assumendo q1 come stato iniziale.

In generale … (anche se Q è infinito)

Definition 2.26. Given an lts $TS = (Q, A, \rightarrow)$, for each natural $i \in \mathbb{N}$, define the relations $\sim_i$ over Q as follows:

- $\sim_0 = Q \times Q$.
- $q_1 \sim_{i+1} q_2$ if and only if for all $\mu \in A$
  - $\forall q_1'$ such that $q_1 \xrightarrow{\mu} q_1'$, $\exists q_2'$ such that $q_2 \xrightarrow{\mu} q_2'$ and $q_1' \sim_i q_2'$
  - $\forall q_2'$ such that $q_2 \xrightarrow{\mu} q_2'$, $\exists q_1'$ such that $q_1 \xrightarrow{\mu} q_1'$ and $q_1' \sim_i q_2'$.

We denote with $\sim_\omega$ the relation $\bigcap_{i \in \mathbb{N}} \sim_i$. □

Proposition 2.11. *Prove that, for each $i \in \mathbb{N}$:*

1. *the relation $\sim_i$ is an equivalence relation,*
2. $\sim_{i+1} \subseteq \sim_i$,
3. $\sim_i = F^i(Q \times Q)$

Moreover, $\sim_\omega$ is an equivalence relation.

…. ovvero

catena, possibilmente infinita, con ~ω come limite

$$\sim_0 = F^0(Q \times Q) \supseteq \sim_1 = F^1(Q \times Q) \supseteq \ldots \supseteq \sim_i = F^i(Q \times Q) \supseteq \ldots \supseteq \sim_\omega$$

Ma che relazione c'è con ~?

Teorema:

Se TS = (Q, A, →) è image-finite, allora ~ω = ~

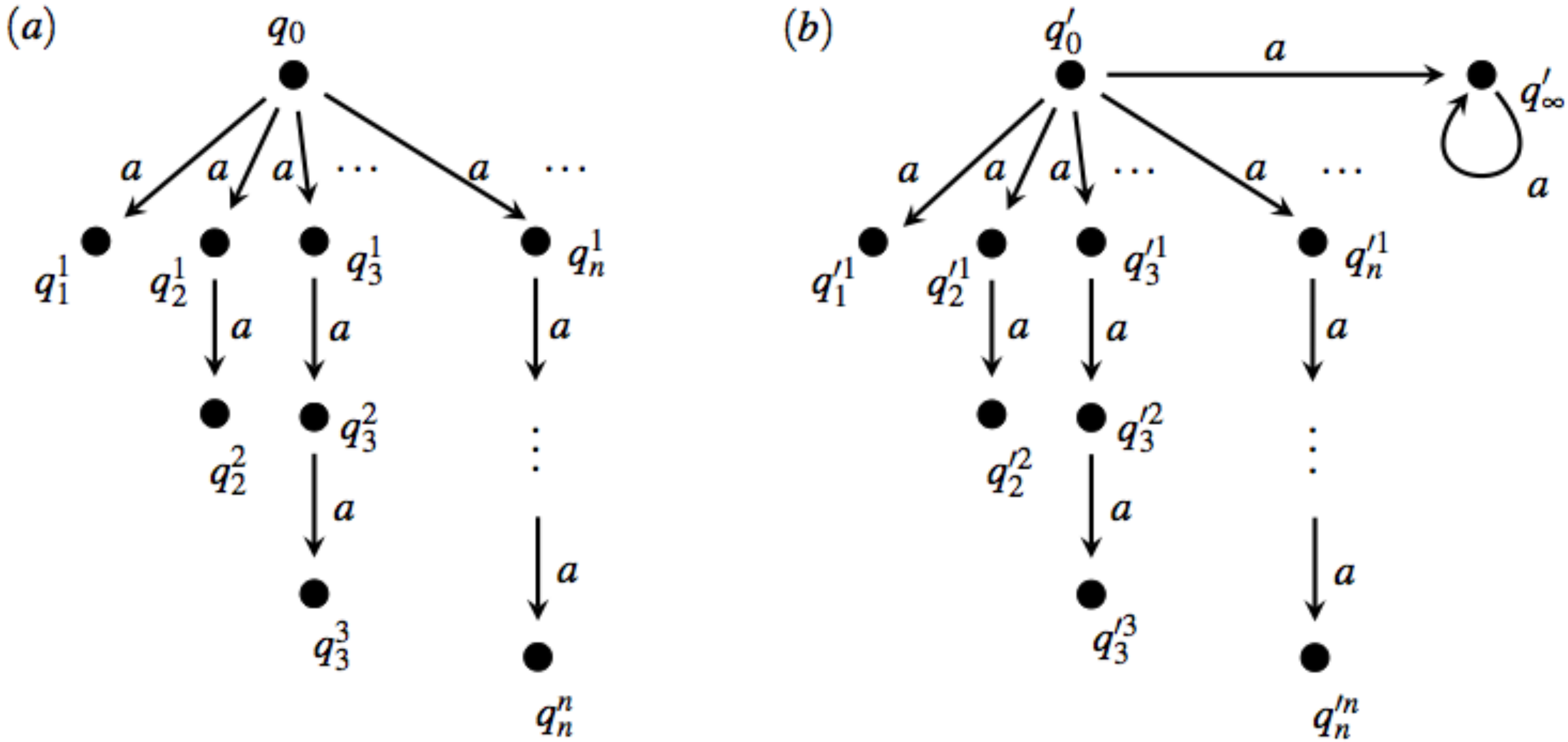
Dimostrazione

Theorem 2.2. *If the lts $TS = (Q, A, \rightarrow)$ is image-finite, then $\sim = \sim_\omega$.*

Proof. We prove first that $\sim \subseteq \sim_i$ for all i by induction on i. Indeed, $\sim \subseteq \sim_0$ (the universal relation); moreover, assuming $\sim \subseteq \sim_i$, by monotonicity of F and the fact that $\sim$ is a fixed-point for F, we get $\sim = F(\sim) \subseteq F(\sim_i) = \sim_{i+1}$. Hence, $\sim \subseteq \sim_\omega$.

Now we prove that $\sim_\omega \subseteq \sim$, by proving that relation $R = \{(q_1, q_2) \mid q_1 \sim_\omega q_2\}$ is a bisimulation. Assume $(q_1, q_2) \in R$, hence $q_1 \sim_i q_2$ for all $i \in \mathbb{N}$. If $q_1 \xrightarrow{\mu} q_1'$, then for all i, there exists q_{2_i} such that $q_2 \xrightarrow{\mu} q_{2_i}$ with $q_1' \sim_i q_{2_i}$. Since the lts is image-finite, the set $K = \{q_{2_k} \mid q_2 \xrightarrow{\mu} q_{2_k} \wedge q_1' \sim_k q_{2_k} \wedge k \in \mathbb{N}\}$ is finite; hence, there is at least one $q_{2_n} \in K$ such that $q_1' \sim_i q_{2_n}$ for infinitely many i. But since if $q \sim_i q'$ then $q \sim_j q'$ for any $j < i$, then we can conclude that $q_1' \sim_i q_{2_n}$ for all i, hence $q_1' \sim_\omega q_{2_n}$ and so $(q_1', q_{2_n}) \in R$. The symmetric case when q_2 moves first is analogous, hence omitted. So $R = \sim_\omega$ is a bisimulation, hence $\sim_\omega \subseteq \sim$. $\square$

Perché serve image-finite?



q0 ~ω q0' ma invece q0 non è bisimile a q0' perché da q0 non parte nessuna computazione infinita.