Lezione 16 MSC
Congruenze comportamentali

Roberto Gorrieri

Cos’è una congruenza?

- Una relazione di equivalenza $\propto$ su un algebra di termini (come CCS, che usa operatori algebrici per costruire sistemi complessi a partire da sistemi più elementari) è una congruenza se è preservata dagli operatori (chiusa per contesti):
- Se $P \propto Q$, allora

$$\mu.P \propto \mu.Q$$
$$P + R \propto Q + R$$
$$P \mid R \propto Q \mid R$$
$$(\nu a)P \propto (\nu a)Q$$

Se questo vale, allora l’equivalenza $\propto$ è detta congruenza comportamentale.

Perché è utile avere congruenze?

- Equivalenza ➔ Intercambiabilità
- Congruenza ➔ Intercambiabilità in ogni contesto!
- Un sistema complesso è composto da molte sottocomponenti; se una di queste si guasta, possiamo sostituirla con un'altra che sia "solo" equivalente, ma non congruente?

  Se $P \propto Q$, ma non è vero che $P \mid R \propto Q \mid R$ (cioè $\propto$ non è composizionale rispetto al parallelo), allora quando sostituiamo P con Q, il comportamento del nuovo sistema $Q|R$ non è più lo stesso di $P|R$.

Congruenza → Equivalence-checking composizionale

- Supponiamo di dover confrontare

$$p_1 \mid p_2 \quad e \quad q_1 \mid q_2$$

- Se dimostriamo che p_1 è congruente a q_1 e che p_2 è congruente a q_2, allora, per congruenza rispetto al parallelo, siamo sicuri che $p_1 \mid p_2$ è congruente a $q_1 \mid q_2$, risparmiando molto in termini di complessità.
- In generale, la congruenza è la base necessaria per poter fare ragionamento composizionale.

Congruenza → minimizzazione composizionale

- Supponiamo di avere un processo
$$P = p_1 \mid p_2 \mid \ldots \mid p_n$$
- Spesso, lo spazio degli stati di P è enorme, a volte intrattabile. Può essere utile minimizzare rispetto alla congruenza scelta (ad esempio strong bis) ogni p_j ottenendo un processo p'_j, quindi comporre a due a due i processi sempre poi minimizzando il risultato.
- Lo spazio degli stati risultante Q è spesso molto più piccolo di quello per P. Ma P e Q sono congruenti! Quindi possiamo tranquillamente fare le nostre analisi su Q.
- La congruenza è la base necessaria per avere minimizzazione composizionale.

Isomorfismo non è una congruenza

- Consideriamo a.0 e a.(0+0): questi generano lts isomorfi.
- Consideriamo ora il contesto C[X] = X + a.0.
- Osserva che C[a.0] e C[a.(0+0)] non generano lts isomorfi: il primo ha solo 2 stati, mentre il secondo ha 3 stati!
- Questo vuol dire che l'equivalenza per isomorfismo non è una congruenza per l'operatore +.

Strong bisimilarity è una congruenza

- Teorema: Se $P \sim Q$, allora

$\mu.P \sim \mu.Q$	per ogni μ
$P + R \sim Q + R$	per ogni R
$P \mid R \sim Q \mid R$	per ogni R
$(\nu a)P \sim (\nu a)Q$	per ogni a

  Dimostrazione: sia S una bisimulazione contenente (P,Q).
  - Allora $S1 = S \cup \{(\mu.P, \mu.Q) \mid \mu \in Act\}$ è una bisimulazione.
  - Anche $S2 = S \cup \{(P+R,Q+R) \mid R$ un processo CCS$\} \cup Id$ è una bisimulazione.
  - Anche $S3 = \{(P'|R', Q'|R') \mid (P',Q')$ in S e R' un processo CCS$\}$ è una bisimulazione.
  - Infine $S4 = \{((\nu a)P',(\nu a)Q') \mid (P',Q')$ in S e $a \in L\}$ è pure una bisimulazione.
- N.B: i casi simmetrici $R + P \sim R + Q$ e $R|P \sim R|Q$ derivano per commutatività di quegli operatori.

Trace equivalence is a congruence

Exercise 4.28. Prove that trace equivalence (see Definition 2.9) is a congruence for the CCS operators: if $Tr(p) = Tr(q)$, then

1) $Tr(\mu.p) = Tr(\mu.q)$ for any $\mu \in Act$,
2) $Tr(p+r) = Tr(q+r)$ for any $r \in \mathscr{P}$,
3) $Tr(p|r) = Tr(q|r)$ for any $r \in \mathscr{P}$,
4) $Tr((\nu a)p) = Tr((\nu a)q)$ for any $a \in \mathscr{L}$.

(*Hint:* First define auxiliary operators on sets of traces: $\mu.L = \{\mu\sigma \mid \sigma \in L\}$; $L_1 \otimes L_2$ as the set of all the possible interleavings among each trace from L_1 and each trace from L_2; and $L \setminus a$ as the set composed of the traces in L with no occurrence of a or $\overline{a}$. Then, show that $Tr(\mu.p) = \{\varepsilon\} \cup \mu.Tr(p)$, $Tr(p+r) = Tr(p) \cup Tr(r)$, $Tr(p|r) = Tr(p) \otimes Tr(r)$, and, finally, that $Tr((\nu a)p) = Tr(p) \setminus a$. For simplicity's sake, you may restrict yourself to finite CCS only.) $\square$

Completed trace equivalence is not a congruence

- Dimostra che completed trace equivalence non è una congruenza per la restrizione, e nemmeno completed simulation equivalence.

  (Suggerimento: considera a.(b+c) e a.b+a.c)

Weak Bis è una congruenza, ma non per l'operatore +

- Teorema: Se $P \approx Q$, allora
  - $\mu.P \approx \mu.Q$ per ogni μ
  - $P \mid R \approx Q \mid R$ per ogni R
  - $(\nu a)P \approx (\nu a)Q$ per ogni a

Dimostrazione: sia S una weak bis contenente (P,Q).

- Allora $S1 = S \cup \{(\mu.P, \mu.Q) \mid \mu \in Act\}$ è una weak bis.
- Anche $S2 = \{(P'|R', Q'|R') \mid (P',Q')$ in S e R' un processo CCS$\}$ è una weak bis.
- Infine $S3 = \{((\nu a)P',(\nu a)Q') \mid (P',Q')$ in S e $a \in L\}$ è pure una weak bis

Non è una congruenza per l'operatore di scelta: $\tau.a \approx a$, ma $\tau.a + b$ non è equivalente al processo $a + b$

Rooted weak bis è una congruenza

- Teorema: Se P $\approx^c$ Q , allora
  μ.P $\approx^c$ μ.Q per ogni μ
  P + R $\approx^c$ Q + R per ogni R
  P | R $\approx^c$ Q | R per ogni R
  (va)P $\approx^c$ (va)Q per ogni a
- Dim.: Se P $\approx^c$ Q, allora P $\approx$ Q, allora μ.P $\approx^c$ μ.Q (eserc.)
- Se P+R-μ->S, allora o P-μ->S oppure R-μ->S. Nel primo caso, poiché P $\approx^c$ Q, deve essere Q=μ=>Q' con S $\approx$ Q'. Allora anche Q+R=μ=>Q' con S $\approx$ Q'. Nel secondo caso, Q+R-μ->S con S $\approx$ S. Simmetricamente, partendo da Q+R.
- Altri due casi per esercizio (si sfrutta che $\approx$ è una congruenza per parallelo e restrizione)

$\approx^c$ è la coarsest congruence contenuta in $\approx$

Theorem 4.5. *Assume that $fn(p) \cup fn(q) \neq \mathscr{L}$. Then $p \approx^c q$ if and only if, for all $r \in \mathscr{P}$, $p + r \approx q + r$.*

Proof. The implication from left to right follows by Theorem 4.4(2) and Exercise 2.74. For the implication from right to left, suppose that $p + r \approx q + r$ for all $r \in \mathscr{P}$. Take any action $a \in Act$ such that $a \notin fn(p) \cup fn(q)$[2] and assume $p \xrightarrow{\mu} p'$. Then also $p + a \xrightarrow{\mu} p'$ (by rule (Sum_1)). As $p + a \approx q + a$, then also $q + a$ must respond to this transition. We have to examine two different cases: either $\mu = \tau$ and $q + a \stackrel{\varepsilon}{\Longrightarrow} q + a$, or the transition truly originates from q, i.e., $q + a \stackrel{\mu}{\Longrightarrow} q'$ (with μ that can be τ). The former case is impossible: p' cannot be weakly bisimilar to $q + a$, as p' cannot perform a. Hence, the second case must be true; but this is indeed what is requested by rooted weak bisimulation: if $p \xrightarrow{\mu} p'$, then $q \stackrel{\mu}{\Longrightarrow} q'$ with $p' \approx q'$. The symmetric case when q moves first is omitted. $\square$

---

[2] The assumption that $\mathscr{L}$ is not covered by the free names of p and q is not strictly necessary [vGl05], but makes the proof easier. Such an assumption is satisfied when p and q are finitary CCS processes (see Proposition 4.5).