Linguaggi per il global computing
Esercizi B e D + Barbershop

Federico Perin - 2029215 - Ottobre 2021

Indice

1 Esercizi

1.1 Esercizio B

Dimostrare che ogni processo CCS finito termina in un numero finito di passi.

1.1.1 Sintassi

$$P, Q ::= \alpha.P \mid (P \mid Q) \mid \sum_{i \in I} P_i \mid P \setminus L \mid P[f] \mid \mathbf{0}$$

Note:

Nel CCS finito non sono previste le costanti $\mathcal{K}$.

Il processo $\mathbf{0}$ ha un solo stato e non ha interazioni con altri processi.

La dimostrazione viene divisa in due casi:

- Nel primo caso si hanno somme finite, cioè l'insieme I contenente le scelte della somma non deterministica, è finito;
- Nel secondo caso l'insieme I sarà infinito.

1.1.2 Dimostrazione per somme finite

La dimostrazione prevede i seguenti punti:

1. Ogni processo del CCS finito termina con un numero finito di passi;
2. Ogni processo del CCS finito ha un numero finito di stati.

1.1.2.1 Punto 1

Dato che non esistono costanti $\mathcal{K}$, non è possibile rigenerare passi eseguiti in precedenza, perciò dopo un certo numero finito di passi ogni processo P in CCS finito terminerà perché non avrà più passi da eseguire. Si deduce perciò che ogni processo ha un numero di passi limitato da un limite superiore è quindi tale numero è finito.

Per dimostrare quanto scritto si procede attraverso una dimostrazione induttiva sull'altezza di derivazione di un processo P, con ipotesi induttiva: $P \xrightarrow{\alpha} P' \Rightarrow Size_p(P') < Size_p(P)$, dove $Size_p(P')$ si intende il numero di passi del processo P dopo aver fatto l'azione α.

Definiamo $Size_p(P)$:

$$Size_p(P) \begin{cases} P = \alpha.R, & 1 + Size_p R \\ P = \sum_{i \in I} P_i, & max(Size_p(P_i)) \\ P = R \mid Q, & Size_p R + Size_p Q \\ P = R \setminus L, & Size_p R \\ P = R[f], & Size_p R \end{cases}$$

Si dimostrerà che $Size_p(P)$ indica il limite superiore del numero di passi eseguiti dal processo P per terminare.

1.1.2.1.1 Dimostrazione

Caso Base:

$\overline{\alpha.P \xrightarrow{\alpha} P}\ ^{ACT}$ $\qquad Size_p(\alpha.P) = 1 + Size_p(P) > Size_p(P)$

Si ha che l'azione α concatenata al processo P aggiunge un passo in più, perciò risulta essere corretto il limite superiore $Size_p(\alpha.P) = 1 + Size_p(P)$ passi.

Caso Induttivo:

* $\dfrac{P_j \xrightarrow{\alpha} P'}{\sum_{i \in I} P_i \xrightarrow{\alpha} P'}\ SUM\ j \in I$

  Si osserva che nell'esecuzione di un passo α, per ipotesi induttiva vale che: $Size_p(P_j) > Size_p(P')$.

  Perciò $Size_p(\sum_{i \in I} P_i) > Size_p(P_j) > Size_p(P')$ allora $max(Size_p(P_i)) > Size_p(P')$

  Il limite superiore $Size_p(\sum_{i \in I} P_i) = max(Size_p(P_i))$ passi, risulta essere corretto.

* $\dfrac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}\ PAR\text{-}L$

  Si osserva che nell'esecuzione di un passo α, per ipotesi induttiva vale che: $Size_p(P) > Size_p(P')$.

  E quindi dato che $Size_p(P|Q) = Size_p(P) + Size_p(Q)$ mentre $Size_p(P'|Q) = Size_p(P') + Size_p(Q)$, per ipotesi induttiva $Size_p(P|Q) > Size_p(P'|Q)$

* $\dfrac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'}\ PAR\text{-}R$

  Si osserva che nell'esecuzione di un passo α, per ipotesi induttiva vale che: $Size_p(Q) > Size_p(Q')$

  E quindi dato che $Size_p(P|Q) = Size_p(P) + Size_p(Q)$ mentre $Size_p(P|Q') = Size_p(P) + Size_p(Q')$, per ipotesi induttiva $Size_p(P|Q) > Size_p(P|Q')$

* $\dfrac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\overline{\alpha}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}\ PAR\text{-}\tau$

  Si osserva che nell'esecuzione di un passo di sincronizzazione, per ipotesi induttiva vale che: $Size_p(P) > Size_p(P')$ e $Size_p(Q) > Size_p(Q')$

  E quindi dato che $Size_p(P|Q) = Size_p(P) + Size_p(Q)$ mentre $Size_p(P'|Q') = Size_p(P') + Size_p(Q')$, per ipotesi induttiva $Size_p(P|Q) > Size_p(P'|Q')$

Perciò si è dimostrato che il limite superiore $Size_p(P|Q)$ =$Size_p(P) + Size_p(Q)$ passi, risulta essere corretto in tutti i tre casi PAR-L, PAR-R e PAR-τ.

* $\dfrac{P \overset{\alpha}{\to} P'}{P \backslash L \overset{\alpha}{\to} P' \backslash L}$ RES se $\alpha, \overline{\alpha} \notin$ L

  Si osserva che nell'esecuzione di un passo α, per ipotesi induttiva vale che: $Size_p(P) > Size_p(P')$

  Applicare una restrizione ad un processo non fa aumentare il numero massimo di passi di esecuzione, ma avere una possibile variazione delle possibili interazioni con altri processi in parallelo, vale che:
  $Size_p(P \backslash L) = Size_p(P)$ mentre $Size_p(P' \backslash L) = Size_p(P')$, per ipotesi induttiva $Size_p(P \backslash L) > Size_p(P' \backslash L)$

  Perciò si è dimostrato che il limite superiore $Size_p(P \backslash L) = Size_p(P)$ passi, risulta essere corretto.

* $\dfrac{P \overset{\alpha}{\to} P'}{P[\mathbf{f}] \overset{f(\alpha)}{\to} P'[\mathbf{f}]}$ REL

  Si osserva che nell'esecuzione di un passo α, per ipotesi induttiva vale che: $Size_p(P) > Size_p(P')$

  Applicare un relabelling ad un processo non fa aumentare il numero massimo di passi di esecuzione, ma avere una possibile variazione delle possibili interazioni con altri processi in parallelo, vale che:
  $Size_p(P[\mathbf{f}]) = Size_p(P)$ mentre $Size_p(P'[\mathbf{f}]) = Size_p(P')$, per ipotesi induttiva $Size_p(P[\mathbf{f}]) > Size_p(P'[\mathbf{f}])$

  Perciò si è dimostrato che il limite superiore $Size_p(P[\mathbf{f}]) = Size_p(P)$ passi, risulta essere corretto.

1.1.2.2 Punto 2

Si vuole dimostrare che ogni processo CCS ha un numero finito di stati, cioè esiste un limite superiore del numero di stati di esecuzione per un processo P.

Definiamo $Size_s(P)$:

$$Size_s(P) \begin{cases} P = \mathbf{0}, & 1 \\ P = \alpha.R, & 1 + Size_s R \\ P = \sum_{i \in I} P_i, & \sum_{i \in I} Size_s P_i \\ P = R \mid Q, & Size_s R * Size_s Q \\ P = R \setminus L, & Size_s R \\ P = R[f], & Size_s R \end{cases}$$

Si dimostrerà di seguito che $Size_s(()$P) calcola il limite superiore del numero di stati dell'esecuzione del processo P.

1.1.2.2.1 Dimostrazione

Tramite dimostrazione per induzione sull'esecuzione di un processo P si dimostra che i limiti di $Size_s(P)$ sono corretti.

Caso Base:

0
Il processo **0** ha un solo stato, quello di partenza, per definizione quindi $Size_s(\mathbf{0}) = 1$ stato.

Caso Induttivo:

In un passo si raggiunge uno sotto processo dal quale poi in n passi finiti si raggiungerà un stato terminante. Si utilizza la seguente ipotesi induttiva:
Un processo composto da sotto processi con un limite superiore ai stati di esecuzione ha anch'esso un limite superiore di stati di esecuzione.
Tale ipotesi scritta verrà utilizzata per dimostrare che il sotto processo raggiunto, da lui allo stato terminante, ci saranno un numero finiti di stati perché limitati da un limiti superiore. Si ricorda inoltre che il numero di stati raggiunti è finito se gli n passi sono finiti. Grazie alla dimostrazione del punto 1 si sa che n è finito.

* **ACT**
  Con l'azione α concatenata al processo P viene aggiunto a P uno stato in più, infatti $\alpha.P \xrightarrow{\alpha} P$. P per ipotesi induttiva ha al più $Size_s(P)$ stati, quindi $\alpha.P$ avrà al più $Size_s(\alpha.P) = 1 + Size_s(P)$ stati.

* **SUM**
  Si ha la seguente esecuzione: $\sum_{i \in I} P_i \xrightarrow{\alpha}$ P', dove qualunque P_i sia scelto per ipotesi induttiva ha un numero finito di stati d'esecuzione. Quindi tutti i processi che compongono la somma non deterministica hanno un numero finito di stati. Perciò $\sum_{i \in I} P_i$ ha al più $Size_s(\sum_{i \in I} P_i) = \sum_{i \in I} Size_s P_i$ stati inoltre, dato che il numero di processi P_i è finito, anche la somma dei stati lo sarà.

  È corretto il limite superiore presentato precedentemente perché potrebbero esserci stati non condivisi tra i vari processi P_i, quindi tutti i $Size_s(P_i)$.

* **PAR-L**
  Si ha la seguente esecuzione: $P|Q \xrightarrow{\alpha} P'|Q$, dove P|Q raggiunge lo stato P'|Q.

  Per ipotesi induttiva P'|Q ha un numero finito di stati d'esecuzione, cioè ha al più $Size_s(P'|Q) = Size_s(P') * Size_s(Q)$ stati, quindi P|Q ha al più $Size_s(P|Q) = (Size_s(P') + 1) * Size_s(Q)$ stati. Dato che è possibile che non ci sia alcun stato condiviso durante l'esecuzione dei due processi, il limite superiore ad essi è uguale al totale delle combinazioni possibili tra gli stati dei due singoli processi.

* **PAR-R**
  Si ha la seguente esecuzione: $P|Q \xrightarrow{\alpha} P|Q'$, dove P|Q raggiunge lo stato P|Q'.

Per ipotesi induttiva P|Q' ha un numero finito di stati d'esecuzione, cioè ha al più $Size_s(P|Q') = Size_s(P) * Size_s(Q')$ stati, quindi P|Q ha al più $Size_s(P|Q) = Size_s(P) * (Size_s(Q') + 1)$ stati. Dato che è possibile che non ci sia alcun stato condiviso durante l'esecuzione dei due processi, il limite superiore ad essi è uguale al totale delle combinazioni possibili tra gli stati dei due singoli processi.

* **PAR-τ**
  Si ha la seguente esecuzione: $P|Q \xrightarrow{\tau} P'|Q'$, dove P|Q raggiunge lo stato P'|Q'.

  Per ipotesi induttiva P'|Q' ha un numero finito di stati d'esecuzione, cioè ha al più $Size_s(P'|Q') = Size_s(P') * Size_s(Q')$ stati, quindi P|Q ha al più $Size_s(P|Q) = Size_s(P') * Size_s(Q') + 1$ stati. Dato che è possibile che non ci sia alcun stato condiviso durante l'esecuzione dei due processi, il limite superiore ad essi è uguale al totale delle combinazioni possibili tra gli stati dei due singoli processi.

  Dunque $Size_s(P|Q) = Size_s(P) * Size_s(Q)$ stati, risulta essere corretto.

* **RES**
  Si ha la seguente esecuzione: $P\backslash L \xrightarrow{\alpha} P'\backslash L$.

  Per ipotesi induttiva P' \L ha al più $Size_s(P'\backslash L) = Size_s(P')$ stati.
  Dato che $Size_s(P'\backslash L)$ è finito, allora P \L avrà al più:
  $Size_s(P\backslash L) = 1 + Size_s(P'\backslash L) = 1 + Size_s(P') = Size_s(P)$ stati, perché applicando una funzione di restrizione non si aumentano il numero massimo di stati d'esecuzione, vale quindi il limite superiore $Size_s(P\backslash L) = Size_s(P)$.

* **REL**
  Si ha la seguente esecuzione: $P[\mathbf{f}] \xrightarrow{f(\alpha)} P'[\mathbf{f}]$.

  Per ipotesi induttiva $P'[\mathbf{f}]$ ha al più $Size_s(P'[\mathbf{f}]) = Size_s(P')$ stati.
  Dato che $Size_s(P'[\mathbf{f}])$ è finito, allora $P[\mathbf{f}]$ avrà al più:
  $Size_s(P[\mathbf{f}]) = 1 + Size_s(P'[\mathbf{f}]) = 1 + Size_s(P') = Size_s(P)$ stati, perché applicando una funzione di relabelling non si aumentano il numero massimo di stati d'esecuzione ma si cambiano solo le possibili interazioni con altri processi in parallelo, vale quindi il limite superiore $Size_s(P[\mathbf{f}]) = Size_s(P)$.

1.1.2.2.2 Conclusione prima parte

Si è dimostrato nei punti precedenti che i processi definiti attraverso il CCS finito hanno sempre un limite superiore sia per il numero di passi e sia per il numero di stati d'esecuzione, di conseguenza il numero di passi e stati sono entrambi finiti.

1.1.3 Dimostrazione per somme infinite

Ora per quanto riguarda la somma non deterministica $\sum_{i \in I} P_i$, l'insieme I sarà infinito. Infatti i processi CCS che fanno parte della somma non deterministica, saranno infiniti e quindi hanno un numero infinito di stati ma hanno tutti una derivazione finita.

La dimostrazione sarà simile alla precedente per tutti i vari punti, tranne per il punto riguardante la somma non deterministica, infatti per dimostrare la finitezza dell'esecuzione si utilizzerà un'astrazione basata sulle generazioni della grammatica di CCS.

Si dimostra perciò per induzione sulla lunghezza di derivazione che ogni processo CCS termina in numero finito di passi.

Caso Base:

0

Il processo **0** termina in 0 passi per definizione.

Caso Induttivo:
La generazione della sequenza di interazioni avanza attraverso uno dei termini della grammatica e come ipotesi induttiva si ha che tale sequenza ha derivazione finita.

* **ACT, RES e REL**
  Con l'azione α concatenata al processo P si ha che, la derivazione produrrà un processo del tipo, $\alpha^{n+1}.P$ con $n \geq 0$, dove esiste un nuovo passo e uno nuovo stato nell'esecuzione di P. Perciò il numero di stati e di passi d'esecuzione rimangono finiti.

* **PAR-L, PAR-R, PAR-τ**
  Come dimostrato per il caso del CCS finito con somme finite, nell'esecuzione parallela la derivazione produrrà dei sotto processi che per ipotesi induttiva sono finiti e quindi il processo di cui fanno parte sarà anch'esso finito.

* **RES**
  Con **RES** si avrà che la derivazione produrrà un certo numero finito di restrizioni. Perciò si avranno sempre passi e stati in numero finito e quindi vale quanto dimostrato nel CCS finito con somme finite.

* **REL**
  Con **REL** si avrà che la derivazione produrrà un certo numero finito di relabelling. Perciò si avranno sempre passi e stati in numero finito e quindi vale quanto dimostrato nel CCS finito con somme finite.

* **SUM** In questo caso il processo derivato sarà del tipo $P_1 + P_2 + \ldots$. Per ipotesi induttiva ogni P_i ha una derivazione di lunghezza finita, perciò si può dimostrare che la derivazione del processo P è finita per induzione sul numero di sotto processi che vengono usati nella somma.

  Caso Base:

  Processo **0** per definizione è finito.

  Caso Induttivo:

  L'ipotesi induttiva afferma che la scelta non deterministica tra i processi ha una derivazione di lunghezza finita. Inoltre si deve tener conto che nella scelta non deterministica, viene scelto un solo processo da eseguire tra tutti quelli presenti.

  Quindi aggiungendo il processo P_{n+1}alla scelta, che per ipotesi induttiva anch'esso ha derivazione finita; la scelta diventerà tra l'esecuzione di uno dei processi già presenti e il processo appena aggiunto P_{n+1}. Solo uno dei processi verrà eseguito e quindi la derivazione avrà una lunghezza superiormente limitata dalla massima lunghezza di derivazione dei sotto processi. Si è perciò dimostrato che l'insieme dei processi della scelta non deterministica può essere illimitato ma finito senza perdere la finitezza di esecuzione. Questo perché la derivazione esegue solo uno dei processi della scelta come scritto in precedenza e in più si sa che l'altezza della derivazione di P è limitata superiormente dall'altezza della derivazione del processo con l'altezza maggiore + 1, ovvero $max_h(P_i) + 1$.

Purtroppo pero il numero di stati che si ha durante l'esecuzione non è più limitato superiormente, perché essendo il limite superiore dato dalla $\sum_{i \in I} P_i$ con I infinito, il numero di processi CCS risulta essere infinito, e quindi la somma dei stati d'esecuzione sarà anch'essa infinita. Perciò il numero di stati non é limitato superiormente.

1.1.4 Conclusione

È stato dimostrato che ogni processo CCS finito termina in un numero finito di passi, indipendente dal fatto si utilizzi una grammatica che permette scelte non deterministiche in un insieme o infinito o finito. Si sottolinea che scelta di un insieme infinito o finito determinerà, se il numero di stati d'esecuzione raggiungibili sarà finito o infinito.

1.2 Esercizio D

Dimostrare che la trace equivalence è una congruenza per il CCS.

Prima di illustrare la dimostrazione si definisce che cosa si intende con i concetti di trace equivalence e congruenza.

Innanzitutto per traces di un processo P che di seguito verrà indicata con Tr(P) si intende, le sequenze di interazioni $\alpha_1 .....\alpha_n \in Act$ con n>= 0 tale che esiste una sequenza di transizioni $P \overset{\alpha_1}{\to} P_1 \overset{\alpha_2}{\to} ... \overset{\alpha_n}{\to} P_n$, e quindi rappresentata tutte le possibili interazioni con un processo. Più formalmente Tr(P) = { $\alpha_1 .....\alpha_n | P \overset{\alpha_1}{\to} P_1 \overset{\alpha_2}{\to} ... \overset{\alpha_n}{\to} P_n$ }. Quindi due processi P e Q si dicono trace equivalence P$\sim_t$Q se Tr(P) = Tr(Q).

Per congruenza si intende, dati due processi P e Q in relazione tra loro (P R Q), allora per ogni contesto C[ ], C[P] R C[Q].

Perciò si dimostrerà che se P$\sim_t$Q $\Rightarrow$ $\forall$C[ ] C[P] $\sim_t$C[Q].

1.2.1 Dimostrazione

Siano P,Q e R processi CCS con P $\sim_t$ Q, allora

1. α.P $\sim_t$ α.Q
2. P + R $\sim_t$ Q + R
3. P|R $\sim_t$ Q|R
4. P\L $\sim_t$ Q\L
5. P [**f**] $\sim_t$ Q[**f**]

1.2.1.1 Prefisso C[ ] = α.

Si ha che Tr(C[P]) = Tr(α.P) = α.Tr(P), dato che con il contesto C[ ] si è aggiunto l'interazione α alle Tr(P). Per ipotesi induttiva Tr(P) = Tr(Q), inoltre aggiungendo l'interazione α sia a Tr(P) e sia a Tr(Q), si ha che α.Tr(P) = α.Tr(Q) = Tr(α.Q).

Perciò vale α.P $\sim_t$ α.Q.

1.2.1.2 Contesto non deterministico C[ ] = (+ R)

Nel caso del contesto non deterministico tra i processi P e R le Tr(P + R) = Tr(P) $\bigcup$ Tr(R). Se questo è vero, dato che per ipotesi induttiva Tr(P) = Tr(Q) e Tr(Q + R) = Tr(Q) $\bigcup$ Tr(R), allora Tr(P + R) = Tr(Q + R)

Perciò si deve dimostrare che il contesto non deterministico tra i processi P e R è uguale alla unione delle traccie dei due processi. Dimostrato questo ne consegue la veridicità di P + R $\sim_t$ Q + R.

($\subseteq$)
Sia t $\in$Tr(P + R) $\Rightarrow$ t $\in$(Tr(P) $\bigcup$ Tr(R)) Per induzione su |t|:

Caso Base |t| = 0
Allora t = ε $\in$(Tr(P) $\bigcup$ Tr(R))

Caso Induttivo |t| = n + 1

t ha una prima interazione seguita poi dalla traccia t', quindi P + R $\overset{\alpha_1}{\rightarrow}$ X' $\overset{t'}{\rightarrow}$ ovvero viene applicata una transizione secondo la regola della somma non deterministica arrivando in certo stato X'; ci sono perciò due possibilità:

- P+R ha effettuato una transizione usando la regola SUM-L:

  $$\frac{P \overset{\alpha_1}{\rightarrow} P'}{P + R \overset{\alpha_1}{\rightarrow} P' \overset{t'}{\rightarrow}} \; \textit{SUM-L}$$

  t = $\alpha_1.t'$ con |t'| = n, per ipotesi induttiva t' $\in Tr(P')$, quindi $\alpha_1.t' \in \alpha_1.Tr(P') \subseteq Tr(P)$ allora $t \in Tr(P) \bigcup Tr(R)$

- P+R ha effettuato una transizione usando la regola SUM-R:

  $$\frac{R \overset{\alpha_1}{\rightarrow} R'}{P + R \overset{\alpha_1}{\rightarrow} R' \overset{t'}{\rightarrow}} \; \textit{SUM-R}$$

  t = $\alpha_1.t'$ con |t'| = n, per ipotesi induttiva t' $\in Tr(R')$, quindi $\alpha_1.t' \in \alpha_1.Tr(R') \subseteq Tr(R)$ allora $t \in Tr(R) \bigcup Tr(P)$

($\supseteq$)

Sia t $\in$(Tr(P) $\bigcup$ Tr(R)) $\Rightarrow$ t $\in$Tr(P + R)
Perciò t può essere traccia sia di P e sia di R oppure solo uno dei due, quindi:

Se t $\in$ Tr(P), P + R può scegliere di fare una transizione attraverso la regola SUM-L, e allora vale che t $\in$ Tr(P + R).

Se t $\in$ Tr(R), P + R può scegliere di fare una transizione attraverso la regola SUM-R, e allora vale che t $\in$ Tr(P + R).

Se t appartiene sia a P che R, qualsiasi regola venga applicata per fare la transizione vale sempre t $\in$ Tr(P + R).

Perciò si è dimostrato che Tr(P + R) = Tr(P) $\bigcup$ Tr(R) e quindi con Tr(P) = Tr(Q),
P + R $\sim_t$ Q + R come si voleva dimostrare.

1.2.1.3 Contesto parallello C[ ] = (| R)

Intuitivamente le traccie Tr(P|R) sono tutte le possibili combinazione tra Tr(P) e Tr(R), cioè quindi tutte le loro interazioni e sincronizzazioni. Se tale intuizione è vera allora dato che Tr(P) = Tr(Q), si potrebbe sostituire P con Q nelle Tr(P|R) ed ottenere le stesse combinazioni della versione precedente, quindi varrebbe che Tr(P|R) = Tr(Q|R) e di conseguenza P|R $\sim_t$ Q|R.

Per dimostrare ciò si deve prima dimostrare il seguente lemma che sarà utilizzato nella dimostrazione:
Siano A e B due processi CSS, se Tr(A) = Tr(B) $\Rightarrow \forall\alpha$ Tr(A')$\subseteq$Tr($\sum\{B'|B \overset{\alpha}{\rightarrow} B'\}$)

con A $\overset{\alpha}{\rightarrow}$ A'
Cioè se i processi A e B hanno le stesse traccie allora le traccie del sotto processo di A, A' sono incluse nell'insieme delle traccie relative ai sotto processi B', raggiunti con una

transizione α dal processo B.

Si dimostra di seguito tale lemma:

Tr($\sum\{B'|B \overset{\alpha}{\to} B'\}$) = {t'|t = α.t'∈Tr(B)}, dato che Tr(A) = Tr(B)

allora posso sostituire le Tr(B) con Tr(A) quindi, {t'|t = α.t'∈Tr(A)}.
Perciò Tr(A')⊆{t'|t = α.t'∈Tr(A)} = Tr($\sum\{B'|B \overset{\alpha}{\to} B'\}$) in accordo con quanto scritto precedentemente.

Si procede con la dimostrazione Tr(P|R) = Tr(Q|R):

(⊆)
Sia t ∈Tr(P|R) ⇒ t ∈(Q|R) Per induzione su |t|:

Caso Base |t| = 0
Allora t = ε ∈Tr(P|R) ⇒ ε ∈Tr(Q|R)

Caso Induttivo |t| = n + 1

t ha una prima interazione seguita poi dalla traccia t', quindi P|R $\overset{\alpha_1}{\to}$ X' $\overset{t'}{\to}$ ovvero viene applicata una transizione secondo la regola del parallelo arrivando in un certo processo X'; ci sono perciò tre possibilità:

- P|R ha effettuato una transizione usando la regola PAR-L:

  $$\frac{P \overset{\alpha_1}{\to} P'}{P|R \overset{\alpha_1}{\to} P' \overset{t'}{\to}} \; \textit{PAR-L}$$

  t = $\alpha_1.t'$ con |t'| = n, per ipotesi induttiva t' $\in Tr(P'|R) \Rightarrow t' \in Tr(Q'|R)$,
  si applica il lemma:
  Tr(P') ⊆ Tr($\sum\{Q'|Q \overset{\alpha_1}{\to} Q'\}$) e quindi t' ∈ Tr(P'|R) ⇒ t' ∈ Tr($\sum\limits_{Q\overset{\alpha_1}{\to}Q'} (Q'|R)$)

  Di conseguenza t ∈ α_1.Tr($\sum\limits_{Q\overset{\alpha_1}{\to}Q'} (Q'|R)$) = $\bigcup_{Q\overset{\alpha_1}{\to}Q'} Tr(\alpha_1.(Q'|R))$ e quindi, dato che il processo Q|R attraverso una transizione α_1 può arrivare al processo Q'|R, si dimostra che se t∈(P|R) ⇒ t ∈(Q|R).

- P|R ha effettuato una transizione usando la regola PAR-R:

  $$\frac{R \overset{\alpha_1}{\to} R'}{P|R \overset{\alpha_1}{\to} R' \overset{t'}{\to}} \; \textit{PAR-R}$$

  t = $\alpha_1.t'$ con |t'| = n, per ipotesi induttiva t' $\in Tr(P|R') \Rightarrow t' \in Tr(Q|R')$, e quindi t ∈Tr($\alpha_1$.(Q|R')).
  Dato che il processo Q|R attraverso una transizione α_1 può arrivare al processo Q|R', si dimostra che se t∈(P|R) ⇒ t ∈(Q|R).

- P|R ha effettuato una transizione usando la regola PAR-τ:

$$\frac{P \xrightarrow{\alpha_1} P' \qquad R \xrightarrow{\overline{\alpha_1}} R'}{P|R \xrightarrow{\tau_1} P'|R' \xrightarrow{t'}}\ \textit{PAR-}\tau$$

t = $\tau_1.t'$ con |t'| = n, per ipotesi induttiva t' $\in Tr(P'|R') \Rightarrow t' \in Tr(Q'|R')$,
si applica il lemma:
Tr(P') $\subseteq$ Tr($\sum\{Q'|Q \xrightarrow{\alpha_1} Q'\}$) e quindi t' $\in$ Tr(P'|R') $\Rightarrow$ t' $\in$ Tr($\sum_{Q \xrightarrow{\alpha_1} Q'} (Q'|R')$)

Di conseguenza t $\in \tau_1$.Tr($\sum_{Q \xrightarrow{\alpha_1} Q'} (Q'|R')) = \bigcup_{Q \xrightarrow{\alpha_1} Q'} Tr(\tau_1.(Q'|R'))$ e quindi, dato che il processo Q|R attraverso una transizione τ_1 effettua la sincronizzazione tra Q e R per arrivare al processo Q'|R', si dimostra che se t$\in$(P|R) $\Rightarrow t \in$(Q|R).

($\supseteq$)
Sia t $\in$Tr(Q|R) $\Rightarrow$ t $\in$(P|R) Per induzione su |t|:

Caso Base |t| = 0
Allora t = ε $\in$Tr(Q|R) $\Rightarrow$ ε $\in$Tr(P|R)

Caso Induttivo |t| = n + 1

t ha una prima interazione seguita poi dalla traccia t', quindi Q|R $\xrightarrow{\alpha_1}$ X' $\xrightarrow{t'}$ ovvero viene applicata una transizione secondo la regola del parallelo arrivando in un processo stato X'; ci sono perciò tre possibilità:

- Q|R ha effettuato una transizione usando la regola PAR-L:

$$\frac{Q \xrightarrow{\alpha_1} Q'}{Q|R \xrightarrow{\alpha_1} Q' \xrightarrow{t'}}\ \textit{PAR-L}$$

t = $\alpha_1.t'$ con |t'| = n, per ipotesi induttiva t' $\in Tr(Q'|R) \Rightarrow t' \in Tr(P'|R)$,
si applica il lemma:
Tr(Q') $\subseteq$ Tr($\sum\{P'|P \xrightarrow{\alpha_1} P'\}$) e quindi t' $\in$ Tr(Q'|R) $\Rightarrow$ t' $\in$ Tr($\sum_{P \xrightarrow{\alpha_1} P'} (P'|R)$)

Di conseguenza t $\in \alpha_1$.Tr($\sum_{P \xrightarrow{\alpha_1} P'} (P'|R)) = \bigcup_{P \xrightarrow{\alpha_1} Q'} Tr(\alpha_1.(P'|R))$ e quindi, dato che il processo P|R attraverso una transizione α_1 può arrivare al processo P'|R, si dimostra che se t$\in$(Q|R) $\Rightarrow t \in$(P|R).

- P|R ha effettuato una transizione usando la regola PAR-R:

$$\frac{R \xrightarrow{\alpha_1} R'}{Q|R \xrightarrow{\alpha_1} R' \xrightarrow{t'}}\ \textit{PAR-R}$$

t = $\alpha_1.t'$ con |t'| = n, per ipotesi induttiva t' $\in Tr(Q|R') \Rightarrow t' \in Tr(P|R')$, e quindi t $\in$Tr(α_1.(P|R')).

Dato che il processo P|R attraverso una transizione α_1 può arrivare al processo P|R', si dimostra che se t∈(Q|R) ⇒ t ∈(P|R).

- P|R ha effettuato una transizione usando la regola PAR-τ:

$$\frac{Q \overset{\alpha_1}{\to} Q' \qquad R \overset{\overline{\alpha_1}}{\to} R'}{Q|R \overset{\tau_1}{\to} Q'|R' \overset{t'}{\to}} \; PAR\text{-}\tau$$

t = $\tau_1.t'$ con |t'| = n, per ipotesi induttiva t' ∈ $Tr(Q'|R') \Rightarrow t' \in Tr(P'|R')$,
si applica il lemma:
Tr(Q') ⊆ Tr($\sum\{P'|P \overset{\alpha_1}{\to} P'\}$) e quindi t' ∈ Tr(Q'|R') ⇒ t' ∈ Tr($\sum\limits_{P \overset{\alpha_1}{\to} P'} (P'|R')$)

Di conseguenza t ∈ τ_1.Tr($\sum\limits_{P \overset{\alpha_1}{\to} P'} (P'|R')) = \bigcup_{P \overset{\alpha_1}{\to} P'} Tr(\tau_1.(P'|R'))$ e quindi, dato che il processo P|R attraverso una transizione τ_1 effettua la sincronizzazione tra P e R per arrivare al processo P'|R', si dimostra che se t∈(Q|R) ⇒ t ∈(P|R).

Quindi con Tr(P) = Tr(Q), P|R $\sim_t$ Q|R come si voleva dimostrare.

1.2.1.4 Contesto restrizione C[ ] = \L

Il caso del contesto restrizione L sul processo P ha la seguente uguaglianza: Tr(P\L) = Tr(P)\{t=...α_x...|α_x ∈L} cioè le traccie che stanno in Tr(P) non ci sono nell'insieme di restrizione definito precedentemente. Se questo è vero, dato che per ipotesi induttiva Tr(P) = Tr(Q) e quindi Tr(Q\L) = Tr(Q)\{t=...α_x...|α_x ∈L}, allora Tr(P\L) = Tr(Q\L).

Perciò si deve dimostrare che il contesto restrizione L sul processo P è uguale a Tr(P)\{t=...α_x...|α_x ∈L}. Dimostrato questo ne consegue la veridicità di P\L $\sim_t$ Q\L.

(⊆)
Sia t ∈Tr(P\L) ⇒ t ∈Tr(P)\{t=...α_x...|α_x ∈L} Per induzione su |t|:

Caso Base |t| = 0
Allora t = ε ∈Tr(P)\{t=...α_x...|α_x ∈L}

Caso Induttivo |t| = n + 1

t ha una prima interazione seguita poi dalla traccia t', quindi P\L $\overset{\alpha_1}{\to}$ X' $\overset{t'}{\to}$ ovvero viene applicata una transizione secondo la regola della restrizione arrivando in un certo processo X', quindi:

$$\frac{P \overset{\alpha_1}{\to} P'}{P\backslash L \overset{\alpha_1}{\to} P'\backslash L \overset{t'}{\to}} \; RES \text{ se } \alpha_1, \overline{\alpha_1} \notin \text{L}$$

t = $\alpha_1.t'$ con |t'| = n, per ipotesi induttiva t' ∈ Tr(P'\L), quindi $\alpha_1.t' \in \alpha_1$.Tr(P'\L)⊆ Tr(P\L) con t ∉ {t=...α_x...|α_x ∈L} allora t ∈Tr(P)\{t=...α_x...|α_x ∈L}.

$(\supseteq)$

Sia t $\in$Tr(P)$\backslash$\{t=...α_x...|α_x $\in$L\} $\Rightarrow$ t $\in$Tr(P$\backslash$L) Per induzione su |t|:

Caso Base |t| = 0
Allora t = ε $\in$Tr(P$\backslash$L)

Caso Induttivo |t| = n + 1

t ha una prima interazione seguita poi dalla traccia t', quindi P$\backslash$L $\xrightarrow{\alpha_1}$ X' $\xrightarrow{t'}$ ovvero viene applicata una transizione secondo la regola della restrizione arrivando in un certo processo X'; quindi:

$$\frac{P \xrightarrow{\alpha_1} P'}{P\backslash L \xrightarrow{\alpha_1} P'\backslash L \xrightarrow{t'}}\ RES \text{ se } \alpha_1, \overline{\alpha_1} \notin \{\text{t=...}\alpha_x\text{...}|\alpha_x \in \text{L}\}$$

t = $\alpha_1.t'$ con |t'| = n, per ipotesi induttiva t' $\in$ Tr(P')$\backslash$ \{t=...α_x...|α_x $\in$L\}, quindi $\alpha_1.t' \in$ α_1.Tr(P')$\backslash$\{t=...α_x...|α_x $\in$L\} $\subseteq$ Tr(P)$\backslash$\{t=...α_x...|α_x $\in$L\} con t $\notin$ \{t=...α_x...|α_x $\in$L\} allora t $\in$Tr(P$\backslash$L).

Quindi dato che Tr(P$\backslash$L) = Tr(P)$\backslash$\{t=...α_x...|α_x $\in$L\} con Tr(P) = Tr(Q) allora si è dimostrato che P$\backslash$L $\sim_t$ Q$\backslash$L.

1.2.1.5 Contesto Relabelling C[ ] = [f]