Testing vs Bisimulazione

Luca Ferragina Lorenzo Mori

universita di Pisa

8 maggio 2014

Sommario

Definizione

Un SISTEMA DI TRANSIZIONE ETICHETTATO (LTS) è una tripla $\langle P, Act, \rightarrow \rangle$, dove

1. P è un insieme i cui elementi sono detti stati.
2. Act è un insieme i cui elementi sono detti azioni, che contiene una distinta azione interna τ.
3. $\rightarrow \subseteq P \times Act \times P$ è la relazione di transizione.

Definizione

Un SISTEMA DI TRANSIZIONE ETICHETTATO (LTS) è una tripla $\langle P, Act, \rightarrow \rangle$, dove

1. P è un insieme i cui elementi sono detti stati.
2. Act è un insieme i cui elementi sono detti azioni, che contiene una distinta azione interna τ.
3. $\rightarrow \subseteq P \times Act \times P$ è la relazione di transizione.

Scriveremo $p \xrightarrow{a} p'$ al posto di $\left\langle p, a, p' \right\rangle \in \rightarrow$. Considereremo inoltre solo LTS finitamente ramificati, cioè tali che per ogni $p \in P$, $|\{q | \exists a. p \rightarrow q\}| < \infty$

Definizione

Sia $\Pi \subseteq P \times P$ una relazione. R è una Π-bisimulazione se $R \subseteq \Pi$ e pRq implica i seguenti fatti:

1. $\mathrm{p} \xrightarrow{a} p' \Rightarrow \exists q'.q \xrightarrow{a} q' \wedge p'Rq'$
2. $\mathrm{q} \xrightarrow{a} q' \Rightarrow \exists p'.p \xrightarrow{a} p' \wedge p'Rq'$

Definizione

Sia $\Pi \subseteq P \times P$ una relazione. R è una Π-bisimulazione se $R \subseteq \Pi$ e pRq implica i seguenti fatti:

1. p$\xrightarrow{a} p' \Rightarrow \exists q'.q \xrightarrow{a} q' \wedge p'Rq'$
2. q$\xrightarrow{a} q' \Rightarrow \exists p'.p \xrightarrow{a} p' \wedge p'Rq'$

Definizione

$p \sim_{\Pi} q$ se e solo se esiste una Π-bisimulazione R tale che pRq.

1. Se Π è una relazione di equivalenza, anche $\sim_{\Pi}$ lo è.

1. Se Π è una relazione di equivalenza, anche $\sim_\Pi$ lo è.
2. Se $\Pi = P \times P$, allora ritroviamo la definizione classica di bisimulazione.

1. Se Π è una relazione di equivalenza, anche $\sim_{\Pi}$ lo è.
2. Se $\Pi = P \times P$, allora ritroviamo la definizione classica di bisimulazione.
3. $\sim_{\Pi}$ è la più grande Π-bisimulazione.

Definizione

Siano $\prod \subseteq P \times P$ e $\Psi \subseteq P \times Act$. R è una $\langle \Pi, \Psi \rangle$-prebisimulazione se R$\subseteq \Pi$ e pRq implica i seguenti fatti:

1. $\langle q, a \rangle \in \Psi \Rightarrow [\langle p, a \rangle \in \Psi \wedge (\ p \xrightarrow{a} p' \Rightarrow \exists q'.q \xrightarrow{a} q' \wedge p' R q')]$.
2. $\langle q, a \rangle \notin \Psi \Rightarrow [\langle p, a \rangle \notin \Psi \wedge (\ p \xrightarrow{a} p' \Rightarrow \exists q'.q \xrightarrow{a} q' \wedge p' R q')]$

Definizione

Siano $\prod \subseteq P \times P$ e $\Psi \subseteq P \times Act$. R è una $\langle \Pi, \Psi \rangle$-prebisimulazione se R$\subseteq \Pi$ e pRq implica i seguenti fatti:

1. $\langle q, a \rangle \in \Psi \Rightarrow [\langle p, a \rangle \in \Psi \wedge (\ p \xrightarrow{a} p' \Rightarrow \exists q'.q \xrightarrow{a} q' \wedge p'Rq')]$.
2. $\langle q, a \rangle\ \Psi \Rightarrow [\langle p, a \rangle\ \Psi \wedge (\ p \xrightarrow{a} p' \Rightarrow \exists q'.q \xrightarrow{a} q' \wedge p'Rq')]$

Definizione

$p \sqsubseteq_{\langle \Pi, \Psi \rangle} q$ se e solo se esiste una $\langle \Pi, \Psi \rangle$-prebisimulazione R tale che pRq.

1. Se Π è un preordine, anche $\sqsubseteq_{\langle \Pi, \Psi \rangle}$ lo è.

1. Se Π è un preordine, anche $\sqsubseteq_{\langle \Pi, \Psi \rangle}$ lo è.
2. Prendendo $\prod = P \times P$ e $\Psi = P \times Act$ ritroviamo la definizione classica di prebisimulazione e possiamo eliminare dalla definizione la condizione 2.

1. Se Π è un preordine, anche $\sqsubseteq_{\langle \Pi, \Psi \rangle}$ lo è.
2. Prendendo $\prod = P \times P$ e $\Psi = P \times Act$ ritroviamo la definizione classica di prebisimulazione e possiamo eliminare dalla definizione la condizione 2.
3. Analogamente, prendendo $\prod = \emptyset$ e $\Psi = P \times Act$ possiamo eliminare la condizione 1.

1. Se Π è un preordine, anche $\sqsubseteq_{\langle \Pi, \Psi \rangle}$ lo è.
2. Prendendo $\prod = P \times P$ e $\Psi = P \times Act$ ritroviamo la definizione classica di prebisimulazione e possiamo eliminare dalla definizione la condizione 2.
3. Analogamente, prendendo $\prod = \emptyset$ e $\Psi = P \times Act$ possiamo eliminare la condizione 1.
4. $\sqsubseteq_{\langle \Pi, \Psi \rangle}$ è la più grande $\langle \Pi, \Psi \rangle$-prebisimulazione.

Definizione

Siano $s, s' \in (\ Act\text{-}\{\tau\})^*$ sequenze di azioni visibili e $a \in Act$. $\stackrel{s}{\Rightarrow}$ è definita induttivamente sulla struttura di s come segue:

- $\stackrel{\epsilon}{\Rightarrow} = \stackrel{\tau}{\rightarrow}^*$, che rappresenta la chiusura riflessiva e transitiva di $\stackrel{\tau}{\rightarrow}$.
- $\stackrel{as'}{\Rightarrow} = \stackrel{\epsilon}{\Rightarrow} \circ \stackrel{a}{\rightarrow} \circ \stackrel{s'}{\Rightarrow}$

Definizione

Siano $s, s' \in (\ Act\text{-}\{\tau\})^*$ sequenze di azioni visibili e $a \in Act$. La relzione di convergenza $p \downarrow s$ è definita induttivamente come segue:

- $p \downarrow \epsilon$ se e solo se non esiste alcuna sequenza infinita $\langle p_i \rangle_{i \geq 0}$ tale che p$\xrightarrow{\tau} p_0$ e $p_i \xrightarrow{\tau} p_{i+1}$.
- $p \downarrow as'$ se e solo se $p \downarrow \epsilon$ e quando $p \stackrel{s}{\Rightarrow} p'$ si ha che $p' \downarrow s'$.

Definizione

- $L(p) = \left\{ s \in (Act - \{\tau\})^* \mid \exists q . p \stackrel{s}{\Rightarrow} q \right\}$ sono le tracce di p.

Definizione

- $L(p) = \left\{ s \in (Act - \{\tau\})^* \mid \exists q . p \stackrel{s}{\Rightarrow} q \right\}$ sono le tracce di p.
- $S(p) = \left\{ a \in Act \mid \exists q . p \stackrel{a}{\rightarrow} q \right\}$.

Definizione

- $L(p) = \left\{ s \in (Act - \{\tau\})^* \,|\exists q.p \stackrel{s}{\Rightarrow} q \right\}$ sono le tracce di p.
- $S(p) = \left\{ a \in Act | \exists q.p \stackrel{a}{\rightarrow} q \right\}$.
- $D(p, a) = \left\{ q | p \stackrel{a}{\rightarrow} q \right\}$; $\mathrm{D}(p) = \bigcup_{a \in Act} \mathrm{D}(p, a)$.

Definizione

- $L(p) = \left\{ s \in (Act - \{\tau\})^* \,|\exists q.p \stackrel{s}{\Rightarrow} q \right\}$ sono le tracce di p.
- $S(p) = \left\{ a \in Act|\exists q.p \stackrel{a}{\rightarrow} q \right\}$.
- $D(p,a) = \left\{ q|p \stackrel{a}{\rightarrow} q \right\}$; $\mathrm{D}(p) = \bigcup_{a \in Act} \mathrm{D}(p,a)$.
- $Q^\tau = \left\{ p|\exists q \in Q.q \stackrel{\epsilon}{\Rightarrow} p \right\}$ è la τ-chiusura di un insieme di stati Q.

Definizione

- $L(p) = \left\{ s \in (Act - \{\tau\})^* \,|\exists q.p \stackrel{s}{\Rightarrow} q \right\}$ sono le tracce di p.
- $S(p) = \left\{ a \in Act|\exists q.p \stackrel{a}{\rightarrow} q \right\}$.
- $D(p, a) = \left\{ q|p \stackrel{a}{\rightarrow} q \right\}$; $\mathrm{D}(p) = \bigcup_{a \in Act} \mathrm{D}(p, a)$.
- $Q^\tau = \left\{ p|\exists q \in Q.q \stackrel{\epsilon}{\Rightarrow} p \right\}$ è la τ-chiusura di un insieme di stati Q.
- D(Q,a)= $\bigcup\limits_{q \in Q} D(q, a)$.

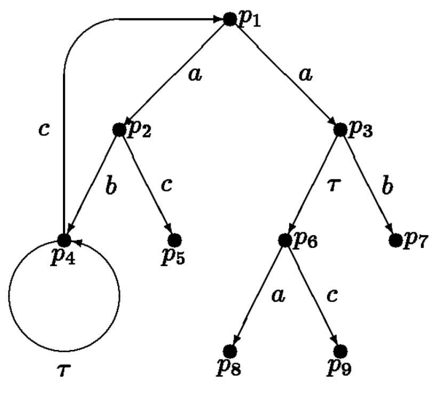
Definizione

- $L(p) = \left\{ s \in (Act - \{\tau\})^* \mid \exists q . p \stackrel{s}{\Rightarrow} q \right\}$ sono le tracce di p.
- $S(p) = \left\{ a \in Act \mid \exists q . p \stackrel{a}{\rightarrow} q \right\}$.
- $D(p, a) = \left\{ q \mid p \stackrel{a}{\rightarrow} q \right\}$; $\mathrm{D}(p) = \bigcup_{a \in Act} \mathrm{D}(p, a)$.
- $Q^\tau = \left\{ p \mid \exists q \in Q . q \stackrel{\epsilon}{\Rightarrow} p \right\}$ è la τ-chiusura di un insieme di stati Q.
- D(Q,a)$= \bigcup\limits_{q \in Q} D(q, a)$.
- Q$\downarrow s$ se e solo se $\bigwedge\limits_{q \in Q} q \downarrow s$.

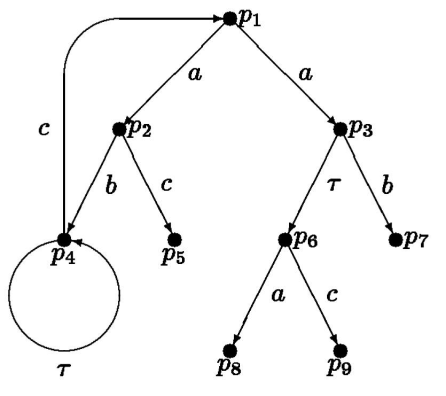
Definizione

- $L(p) = \left\{ s \in (Act - \{\tau\})^* \mid \exists q.p \stackrel{s}{\Rightarrow} q \right\}$ sono le tracce di p.
- $S(p) = \left\{ a \in Act \mid \exists q.p \stackrel{a}{\rightarrow} q \right\}$.
- $D(p, a) = \left\{ q \mid p \stackrel{a}{\rightarrow} q \right\}$; $\mathrm{D}(p) = \bigcup_{a \in Act} \mathrm{D}(p, a)$.
- $Q^\tau = \left\{ p \mid \exists q \in Q.q \stackrel{\epsilon}{\Rightarrow} p \right\}$ è la τ-chiusura di un insieme di stati Q.
- D(Q,a)= $\bigcup\limits_{q \in Q} D(q, a)$.
- Q$\downarrow s$ se e solo se $\bigwedge\limits_{q \in Q} q \downarrow s$.
- $A(p, s) = \left\{ S(p') \mid p \stackrel{s}{\Rightarrow} p' \wedge \neg(p' \stackrel{\tau}{\rightarrow}) \right\}$ è l'insieme di accettazione di p.

Esempio

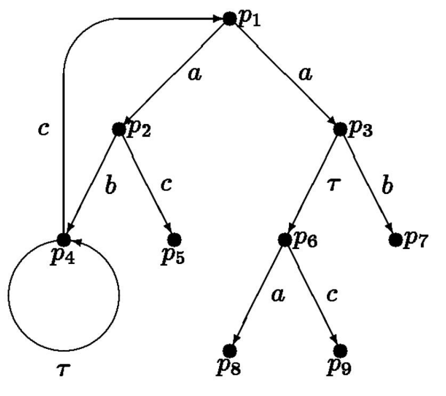


Esempio



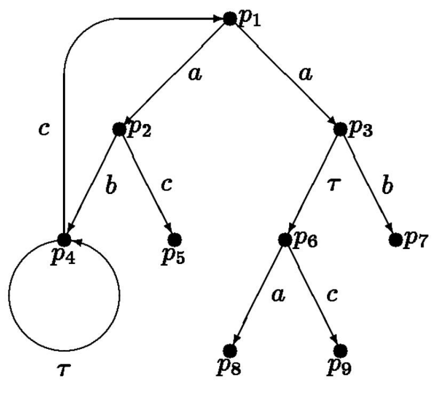
$S(p_1) = \{a\}$

Esempio



$D(p_1, a) = D(p_1) = \{p_2, p_3\}$

Esempio



$$A\left(p_1, < a >\right) = \left\{S\left(p_2\right), S\left(p_6\right)\right\} = \left\{\{a, b\}, \{a, c\}\right\}$$

Definizione

Siano A,B$\subseteq \mathcal{P}(Act)$.

1. A$\subset\subset B$ se $\forall S \in A. \exists R \in B. R \subseteq S$.

Definizione

Siano A,B$\subseteq \mathcal{P}(Act)$.

1. A$\subset\subset B$ se $\forall S \in A.\exists R \in B.R \subseteq S$.
2. minA=$\{S \in A | \neg\exists R \in A.R \subset S\}$.

Definizione

Siano A,B$\subseteq \mathcal{P}(Act)$.

1. A$\subset\subset B$ se $\forall S \in A.\exists R \in B.R \subseteq S$.
2. minA=$\{S \in A | \neg\exists R \in A.R \subset S\}$.

Lemma

Siano A,B$\subseteq \mathcal{P}(Act)$ tali che $glb_{\subseteq}(A) \in A$ e $glb_{\subseteq}(B) \in B$ allora:

1. *A$\subset\subset B$ se e solo se minA$\subset\subset$minB.*
2. *A$\subset\subset B$ e B$\subset\subset A$ se e solo se minA=minB*

Definizione

(Equivalenze may, must e testing)

Definizione

(Equivalenze may, must e testing)

1. $p << _{may}\ q$ se $L(p) \subseteq L(q)$.

Definizione

(Equivalenze may, must e testing)

1. p$<<_{may}$ q se L(p)$\subseteq$L(q).
2. p$<<_{must}$ q se per ogni
   $s \in (Act - \{\tau\})^* . p \downarrow s \Rightarrow (q \downarrow s \wedge A(q,s) \subset\subset A(p,s))$.

Definizione

(Equivalenze may, must e testing)

1. p$<<_{may}$ q se L(p)$\subseteq$L(q).
2. p$<<_{must}$ q se per ogni
   $s \in (Act - \{\tau\})^* . p \downarrow s \Rightarrow (q \downarrow s \wedge A(q, s) \subset\subset A(p, s))$.
3. p$<<_{t}$ q se p$<<_{may}$ q e p$<<_{must}$ q.

Definizione

(Equivalenze may, must e testing)

1. p$<<_{may}$ q se L(p)$\subseteq$L(q).
2. p$<<_{must}$ q se per ogni $s \in (Act - \{\tau\})^* . p \downarrow s \Rightarrow (q \downarrow s \wedge A(q,s) \subset\subset A(p,s))$.
3. p$<<_t$ q se p$<<_{may}$ q e p$<<_{must}$ q.
4. p$=_{may}$q se p$<<_{may}$ q e q$<<_{may}$ p.

Definizione

(Equivalenze may, must e testing)

1. p$<<_{may}$ q se L(p)$\subseteq$L(q).
2. p$<<_{must}$ q se per ogni $s \in (Act - \{\tau\})^{*}.p \downarrow s \Rightarrow (q \downarrow s \wedge A(q,s) \subset\subset A(p,s))$.
3. p$<<_{t}$ q se p$<<_{may}$ q e p$<<_{must}$ q.
4. p$=_{may}$q se p$<<_{may}$ q e q$<<_{may}$ p.
5. p$=_{must}$q se p$<<_{must}$ q e q$<<_{must}$ p.

Definizione

(Equivalenze may, must e testing)

1. p$<<_{may}$ q se L(p)$\subseteq$L(q).
2. p$<<_{must}$ q se per ogni $s \in (Act - \{\tau\})^* .p \downarrow s \Rightarrow (q \downarrow s \wedge A(q,s) \subset\subset A(p,s))$.
3. p$<<_{t}$ q se p$<<_{may}$ q e p$<<_{must}$ q.
4. p$=_{may}$q se p$<<_{may}$ q e q$<<_{may}$ p.
5. p$=_{must}$q se p$<<_{must}$ q e q$<<_{must}$ p.
6. p$=_{t}$q se p$<<_{t}$ q e q$<<_{t}$ p.

Definizione

Sia $\langle P, Act, \rightarrow \rangle$ un LTS, $\rightarrow$ è deterministica se $\xrightarrow{\tau} = \emptyset$ e $|D(p, a)| \leqslant 1$ per ogni p$\in$ P e per ogni a$\in$ Act.

Definizione

Un LTS $\langle T, Act, \rightarrow \rangle$ si dice T-grafo se:

Definizione

Un LTS $\langle T, Act, \rightarrow \rangle$ si dice T-grafo se:

1. $\rightarrow$ è deterministica.

Definizione

Un LTS $\langle T, Act, \rightarrow \rangle$ si dice T-grafo se:

1. $\rightarrow$ è deterministica.
2. Ogni t$\in T$ è etichettato da due pezzi di informazione, t.acc e t.closed, con t.acc$\subseteq \mathcal{P}(Act)$ e t.closed$\in \{true, false\}$.

Definizione

Un LTS $\langle T, Act, \rightarrow \rangle$ si dice T-grafo se:

1. $\rightarrow$ è deterministica.
2. Ogni t$\in$ T è etichettato da due pezzi di informazione, t.acc e t.closed, con t.acc$\subseteq \mathcal{P}\left(Act\right)$ e t.closed$\in \{true, false\}$.
3. $|t.acc| < \infty$, $\forall S \in t.acc.\, |S| < \infty$, t.acc=min t.acc.

Definizione

Un LTS $\langle T, Act, \rightarrow \rangle$ si dice T-grafo se:

1. $\rightarrow$ è deterministica.
2. Ogni t$\in$ T è etichettato da due
   pezzi di informazione, t.acc e t.closed, con t.acc$\subseteq \mathcal{P}\left(Act\right)$ e t.closed$\in \{true, false\}$.
3. $|t.acc| < \infty$, $\forall S \in t.acc. \, |S| < \infty$, t.acc=min t.acc.
4. t.closed=true se e solo se t.acc$\neq \emptyset$

Definizione

Un LTS $\langle T, Act, \rightarrow \rangle$ si dice T-grafo se:

1. $\rightarrow$ è deterministica.
2. Ogni t$\in$ T è etichettato da due pezzi di informazione, t.acc e t.closed, con t.acc$\subseteq \mathcal{P}\left(Act\right)$ e t.closed$\in \{true, false\}$.
3. $|t.acc| < \infty$, $\forall S \in t.acc. |S| < \infty$, t.acc=min t.acc.
4. t.closed=true se e solo se t.acc$\neq \emptyset$.
5. t.closed=false $\Rightarrow \forall t' \in D\left(t\right) t'.closed$=false.

Definizione

Un LTS $\langle T, Act, \rightarrow \rangle$ si dice ST-grafo se:

1. $\rightarrow$ è deterministica.
2. Ogni t$\in T$ è etichettato da due pezzi di informazione, t.acc e t.closed, con t.acc$\subseteq \mathcal{P}(Act)$ e t.closed$\in \{true, false\}$.
3. $|t.acc| < \infty$, $\forall S \in t.acc. |S| < \infty$, t.acc=min t.acc.
4. t.closed=true se e solo se t.acc$\neq \emptyset$.
5. $\forall t \in T$ t.closed=false $\Rightarrow$ D(t)=$\emptyset$.

Definizione

Un LTS $\langle T, Act, \rightarrow \rangle$ si dice D-grafo se:

1. $\rightarrow$ è deterministica.
2. Ogni t$\in$ T è etichettato da due pezzi di informazione, t.acc e t.closed, con t.acc$\subseteq \mathcal{P}(Act)$ e t.closed$\in \{true, false\}$.
3. $|t.acc| < \infty$, $\forall S \in t.acc. |S| < \infty$, t.acc=min t.acc.
4. $\forall t \in T$ t.closed=false.

Definizione

Sia $L = \langle P, Act, \rightarrow \rangle$ un LTS. Il T-grafo T(L)=$\langle T, Act, \rightarrow \rangle$ è definito come segue:

Definizione

Sia $L = \langle P, Act, \rightarrow \rangle$ un LTS. Il T-grafo T(L)=$\langle T, Act, \rightarrow \rangle$ è definito come segue:

1. T=$\{\langle Q, b \rangle \in \mathcal{P}\left(P\right) \times bool \ | Q = Q^{\tau} \wedge (Q \uparrow \epsilon \Rightarrow b = false)$.

Definizione

Sia $L = \langle P, Act, \rightarrow \rangle$ un LTS. Il T-grafo T(L)=$\langle T, Act, \rightarrow \rangle$ è definito come segue:

1. T=$\{\langle Q, b\rangle \in \mathcal{P}(P) \times bool \ | Q = Q^{\tau} \wedge (Q \uparrow \epsilon \Rightarrow b = false)$.
2. Per t=$\langle Q, b\rangle \in T$, definiamo
   - t.closed=b
   - t.acc=$\emptyset$ se b.closed=true
   - t.acc=min$\left\{ S(q) | q \in Q \wedge \neg \left(q \xrightarrow{\tau} \right) \right\}$ altrimenti.

Definizione

Sia $L = \langle P, Act, \rightarrow \rangle$ un LTS. Il T-grafo T(L)=$\langle T, Act, \rightarrow \rangle$ è definito come segue:

1. T=$\{\langle Q, b\rangle \in \mathcal{P}(P) \times bool \; | Q = Q^{\tau} \wedge (Q \uparrow \epsilon \Rightarrow b = false)$.
2. Per t=$\langle Q, b\rangle \in T$, definiamo
   - t.closed=b
   - t.acc=$\emptyset$ se b.closed=true
   - t.acc=min$\left\{ S(q) \,| q \in Q \wedge \neg \left(q \xrightarrow{\tau} \right) \right\}$ altrimenti.
3. Per $t_1 = \langle Q_1, b_1 \rangle$ e $t_2 = \langle Q_2, b_2 \rangle$, $t_1 \xrightarrow{a} t_2$ quando accade che:
   - $a \neq \tau$.
   - $(D(Q_1, a))^{\tau} = Q_2$
   - $[b_1 = false \Rightarrow b_2 = false] \wedge [(b_1 = true \wedge b_2 = false) \Rightarrow Q_2 \uparrow \epsilon]$

Definizione

Sia $L = \langle P, Act, \rightarrow \rangle$ un LTS. L' ST-grafo ST(L)=$\langle T, Act, \rightarrow \rangle$ è definito come segue:

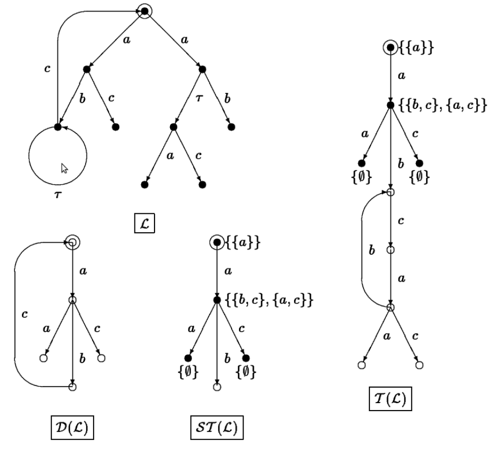
1. T=$\{\langle Q, b \rangle \in \mathcal{P}(P) \times bool \ | Q = Q^{\tau} \wedge (Q \uparrow \epsilon \Rightarrow b = false)$.
2. Per t=$\langle Q, b \rangle \in T$, definiamo
   - t.closed=b
   - t.acc=$\emptyset$ se b.closed=true
   - t.acc=min$\left\{ S(q) | q \in Q \wedge \neg \left(q \xrightarrow{\tau} \right) \right\}$ altrimenti.
3. Per $t_1 = \langle Q_1, b_1 \rangle$ e $t_2 = \langle Q_2, b_2 \rangle$, $t_1 \xrightarrow{a} t_2$ quando accade che:
   - $a \neq \tau$.
   - $(D(Q_1, a))^{\tau} = Q_2$
   - $b_1 = true \wedge (b_2 = false \Rightarrow Q_2 \uparrow \epsilon)$

Definizione

Sia $L = \langle P, Act, \rightarrow \rangle$ un LTS. Il D-grafo D(L)=$\langle T, Act, \rightarrow \rangle$ è definito come segue:

1. T=$\{\langle Q, false \rangle \,|\, Q = Q^{\tau}\}$.
2. Per t=$\langle Q, b \rangle \in T$, definiamo
   - t.closed=b
   - t.acc=$\emptyset$ se b.closed=true
   - t.acc=min$\left\{ S\left(q\right) | q \in Q \wedge \neg \left(q \xrightarrow{\tau} \right) \right\}$ altrimenti.
3. Per $t_1 = \langle Q_1, b_1 \rangle$ e $t_2 = \langle Q_2, b_2 \rangle$, $t_1 \xrightarrow{a} t_2$ quando accade che:
   - $a \neq \tau$.
   - $\left(D\left(Q_1, a\right)\right)^{\tau} = Q_2$.
   - $[b_1 = false \Rightarrow b_2 = false] \wedge [(b_1 = true \wedge b_2 = false) \Rightarrow Q_2 \uparrow \epsilon]$.

Esempio



$\mathcal{L}$
$\mathcal{D}(\mathcal{L})$
$\mathcal{ST}(\mathcal{L})$
$\mathcal{T}(\mathcal{L})$
$\{\{a\}\}$
$\{\{b,c\},\{a,c\}\}$
$\{\emptyset\}$

Definizione

Sia $p \in P$.

- $t(p) = \langle \{p\}^{\tau}, p \downarrow \epsilon \rangle$.
- $d(p) = \langle \{p\}^{\tau}, \mathit{false} \rangle$.

Definizione

Sia $p \in P$.

- $t\,(p) = \langle \{p\}^{\tau}\,, p \downarrow \epsilon\rangle$.
- $d\,(p) = \langle \{p\}^{\tau}\,, \mathit{false}\rangle$.

Definizione

Ridefiniamo il predicato $\downarrow s$ per i T-grafi come segue:

- $t \downarrow \epsilon$ se e solo se t.closed=true.
- $t \downarrow as'$ se e solo se t.closed=true $\wedge \left(t \xrightarrow{\tau} t' \Rightarrow t' \downarrow s'\right)$.

Lemma

Sia $L = \langle P, Act, \rightarrow \rangle$ *un LTS. Si ha che:*

1. *L(p)=L(t(p)), dove L(t(p)) è interpretato rispetto a T(L).*

Lemma

Sia $L = \langle P, Act, \rightarrow \rangle$ *un LTS. Si ha che:*

1. *L(p)=L(t(p)), dove L(t(p)) è interpretato rispetto a T(L).*
2. *L(p)=L(d(p)), dove L(d(p)) è interpretato rispetto a D(L).*

Lemma

Sia $L = \langle P, Act, \rightarrow \rangle$ un LTS. Si ha che:

1. *L(p)=L(t(p)), dove L(t(p)) è interpretato rispetto a T(L).*
2. *L(p)=L(d(p)), dove L(d(p)) è interpretato rispetto a D(L).*

In generale questa proprietà non vale fra p e il suo stato corrispondente in ST(L).

Tuttavia vale un fatto leggermente diverso.

Tuttavia vale un fatto leggermente diverso.

Lemma

Sia $s \in (Act - \{\tau\})^*$.

1. *Supponiamo che* $p\downarrow s$ *e consideriamo* $t(p)$ *in* $ST(L)$. *Allora* $s \in L(p)$ *se e solo se* $s \in L(t(p))$.

Tuttavia vale un fatto leggermente diverso.

Lemma

Sia $s \in (Act - \{\tau\})^*$.

1. *Supponiamo che* $p\downarrow s$ *e consideriamo t(p) in ST(L). Allora* $s \in L(p)$ *se e solo se* $s \in L(t(p))$.
2. $p\downarrow s$ *se e solo se* $t(p)\downarrow s$, *dove t(p) può essere interpretato sia in T(L) che in ST(L).*

Teorema

Sia $\Pi = \{\langle t, u\rangle \in T \times T | t.acc = \emptyset \vee u.acc \subset\subset t.acc\}$ *e siano p,q*$\in$*P.*
Valgono i seguenti fatti:

1. $p<<_{must}$ *q se e solo se* $t(p) \sqsubseteq_{\langle \Pi, \emptyset \rangle} t(q)$ *in ST(L).*
2. $p<<_{t}$ *q se e solo se* $t(p) \sqsubseteq_{\langle \Pi, T \times Act \rangle} t(q)$ *in T(L).*
3. $p<<_{may}$ *q se e solo se* $d(p) \sqsubseteq_{\langle D \times D, D \times Act \rangle} d(q)$ *in D(L).*

Teorema

Sia $\Pi = \{\langle t, u\rangle \in T \times T | t.acc = u.acc\}$ *e siano p,q*$\in$*P. Valgono i seguenti fatti:*

1. $p =_{may} q$ *se e solo se* $d(p) \sim_{D \times D} d(q)$ *in* $D(L)$.
2. $p =_{must} q$ *se e solo se* $t(p) \sim_{\Pi} t(q)$ *in* $ST(L)$.
3. $p =_{t} q$ *se e solo se* $t(p) \sim_{\Pi} t(q)$ *in* $T(L)$.

Definizione

Dato A un insieme di azioni e denotiamo con τ l'azione interna $\tau \in A$, definiamo la sintassi del Communicating Sequential Processes (CSP) come: $P \rightarrow STOP|\mu.P|P \sqcap P|P \square P|P\ |[L]|\ P|recx.P|P;P|P \backslash C|x$ in cui:

- $\mu \in A$ quando parleremo specificatemente di azioni visibili useremo la lettera a.
- $C, L \subseteq_f A$- $\{\tau\}$
- x è una variabile

Definizione

Un processo si dice chiuso se la variabile x appare solo nell'argomento della ricorsione.

Definizione

Un processo si dice chiuso se ogni occorrenza della variabile x si trova nell'argomento della ricorsione.

Dato A un insieme di azioni e P un processo costruito secondo le regole della sintassi, definiamo una semantica operazionale che ci permetta di associare a P un LTS.

$$P \rightarrow STOP | \mu.P | P \sqcap P | P \Box P | P \, |[L]| \, P | recx.P | P;P | P \backslash C | x$$

Definizione

Semantica operazionale per il CSP.

- operazione di prefixing $\dfrac{}{\mu.P \xrightarrow{\mu} P}$

$P \rightarrow STOP|\mu.P|P \sqcap P|P \square P|P\,|[L]|\,P|recx.P|P;P|P \backslash C|x$

Definizione

Semantica operazionale per il CSP.

- scelta non deterministica $\dfrac{}{P \sqcap Q \xrightarrow{\tau} P}$ $\dfrac{}{P \sqcap Q \xrightarrow{\tau} Q}$

$$P \rightarrow STOP|\mu.P|P \sqcap P|P\Box P|P\,|[L]|\,P|recx.P|P;P|P\backslash C|x$$

Definizione

Semantica operazionale per il CSP.

- scelta esterna $\dfrac{P \xrightarrow{\tau} P'}{P \Box Q \xrightarrow{\tau} P' \Box Q}$ $\qquad$ $\dfrac{Q \xrightarrow{\tau} Q'}{P \Box Q \xrightarrow{\tau} P \Box Q'}$

$$\dfrac{P \xrightarrow{a} P'}{P \Box Q \xrightarrow{a} P'} \qquad \dfrac{Q \xrightarrow{a} Q'}{P \Box Q \xrightarrow{a} Q'} \qquad a \neq \tau$$

$$P \rightarrow STOP|\mu.P|P \sqcap P|P\Box P|P\ |[L]|\ P|recx.P|P;P|P\backslash C|x$$

Definizione

Semantica operazionale per il CSP.

- parallelismo
  - $\dfrac{}{STOP|[L]|STOP \xrightarrow{\tau} STOP}$
  - $\dfrac{P \xrightarrow{\mu} P'}{P|[L]|Q \xrightarrow{\mu} P'|[L]|Q}$ $\dfrac{Q \xrightarrow{\mu} Q'}{P|[L]|Q \xrightarrow{\mu} P|[L]|Q'}$ se $a \notin L$
  - $\dfrac{Q \xrightarrow{a} Q' \ P \xrightarrow{a} P'}{P|[L]|Q \xrightarrow{a} P'|[L]|Q'}$ se $a \in L$

$$P \to STOP|\mu.P|P \sqcap P|P \Box P|P \mathbin{|[L]|} P|recx.P|P;P|P \backslash C|x$$

Definizione

Semantica operazionale per il CSP.

- ricorsione $\dfrac{}{recx.P \xrightarrow{\tau} P[recx.P/x]}$

$$P \rightarrow STOP | \mu.P | P \sqcap P | P \Box P | P \; |[L]| \; P | recx.P | P;P | P \backslash C | x$$

Definizione

Semantica operazionale per il CSP.

- sequenzializzazione $\dfrac{P \xrightarrow{\mu} P'}{P;Q \xrightarrow{\mu} P';Q}$ $\qquad \dfrac{P \xrightarrow{\mu} STOP}{P;Q \xrightarrow{\mu} Q}$

$P \rightarrow STOP|\mu.P|P \sqcap P|P \Box P|P \,|[L]|\, P|recx.P|P; P|P \backslash C|x$

Definizione

Semantica operazionale per il CSP.

- hiding $\dfrac{P \xrightarrow{\mu} P'}{P \backslash C \xrightarrow{\mu} P' \backslash C} \quad \mu \notin C \qquad \dfrac{P \xrightarrow{a} P'}{P \backslash C \xrightarrow{\tau} P' \backslash C} \; a \in C$

- In questo modo per uno specifico processo abbiamo definito un LTS che ha come azioni quelle che occorrono nel processo, come regole di transizione definite dalla semantica operazionale e come stati i processi ottenuti nel corso delle transizioni.

- In questo modo per uno specifico processo abbiamo definito un LTS che ha come azioni quelle che occorrono nel processo, come regole di transizione definite dalla semantica operazionale e come stati i processi ottenuti nel corso delle transizioni.
- Gli LTS così costruiti sono finite branching.

Definizione

Siano P un processo e B$\subseteq A$.

- P si dice stabile se $\tau \notin S(P)$

Definizione

Siano P un processo e B$\subseteq A$.

- P si dice stabile se $\tau \notin L(P)$
- P ref B se e solo se (P è stabile e $B \cap S(P) = \emptyset$) e $B \subseteq A$

Definizione

Siano P un processo e B$\subseteq A$.

- P si dice stabile se $\tau \notin L(P)$
- P ref B se e solo se (P è stabile e $B \cap S(P) = \emptyset$) e $B \subseteq A$
- $\mathit{failures}(P) = \left\{ (s, X) \,|\, \exists Q : P \stackrel{s}{\Rightarrow} Q \wedge Q \mathit{ref} X \right\}$

Definizione

Siano P e Q processi.

- $P \sqsubseteq_F Q$ se e solo se $L(P) \supseteq L(Q) \wedge \mathit{failures}(P) \supseteq \mathit{failures}(Q)$

Definizione

Siano P e Q processi.

- $P \sqsubseteq_F Q$ se e solo se $L(P) \supseteq L(Q) \wedge \mathit{failures}(P) \supseteq \mathit{failures}(Q)$
- Se valgono $P \sqsubseteq_F Q$ e $Q \sqsubseteq_F P$ si scrive $P =_F Q$

Definizione

Siano P e Q processi.

- $P \sqsubseteq_F Q$ se e solo se $L(P) \supseteq L(Q) \wedge \mathit{failures}(P) \supseteq \mathit{failures}(Q)$
- Se valgono $P \sqsubseteq_F Q$ e $Q \sqsubseteq_F P$ si scrive $P =_F Q$

$=_F$ è in maniera evidente una relazione di equivalenza sugli stati.

Esempio

Siano a,b $\in$ A e P e Q processi definiti su A come segue:
$P = a.STOP \Box b.STOP \quad Q = a.STOP \sqcap b.STOP$

Esempio

Sia a,b $\in$ A e siano P e Q processi definiti su A come segue:

$P = a.STOP \Box b.STOP \quad Q = a.STOP \sqcap b.STOP$

vale che

- $L(P) = L(a.STOP \Box b.STOP) = L(a.STOP) \cup L(b.STOP) = L(a.STOP \sqcap b.STOP) = L(Q)$

Esempio

Siano a,b $\in$ A e P e Q processi definiti su A come segue:
$P = a.STOP \Box b.STOP \quad Q = a.STOP \sqcap b.STOP$
vale che

- $L(P) = L(a.STOP \Box b.STOP) = L(a.STOP) \cup L(b.STOP) = L(a.STOP \sqcap b.STOP) = L(Q)$
- $failures(P) = \{(a, X), (b, X), (\langle\rangle, \{\})\}$

Esempio

Siano a,b $\in$ A e P e Q processi definiti su A come segue:

$P = a.STOP \Box b.STOP \quad Q = a.STOP \sqcap b.STOP$

vale che

- $L(P) = L(a.STOP \Box b.STOP) = L(a.STOP) \cup L(b.STOP) = L(a.STOP \sqcap b.STOP) = L(Q)$
- $failures(P) = \{(a, X), (b, X), (\langle\rangle, \{\})\}$
  $failures(Q) = \{(a, X), (b, X), (\langle\rangle, X)\} \, \forall X \subseteq A$
  $failures(P) \subset failures(Q)$

Esempio

Siano a,b $\in$ A e P e Q processi definiti su A come segue:
$P = a.STOP \Box b.STOP \quad Q = a.STOP \sqcap b.STOP$
vale che

- $L(P) = L(a.STOP \Box b.STOP) = L(a.STOP) \cup L(b.STOP) = L(a.STOP \sqcap b.STOP) = L(Q)$
- $failures(P) = \{(a, X), (b, X), (\langle\rangle, \{\})\}$
  $failures(Q) = \{(a, X), (b, X), (\langle\rangle, X)\} \forall X \subseteq A$
  $failures(P) \subset failures(Q)$
  e quindi $Q \sqsubseteq_F P$ e $Q \neq_F P$

Definizione

Sia X un insieme. Denoto con $X_\perp = X \cup \{\perp\}$
se $f : X \longrightarrow Y_\perp$ scrivo che $f(x)\uparrow$ quando $f(x){=}\perp$ altrimenti $f(x)\downarrow$

Definizione

Sia X un insieme. Denoto con $X_\perp = X \cup \{\perp\}$
se $f : X \longrightarrow Y_\perp$ scrivo che $f(x)\uparrow$ quando $f(x){=}\perp$ altrimenti $f(x)\downarrow$

Definizione

Dati A insieme in input e K insieme in output un automa di Moore è una coppia data da $(S, (o_S, d_S))$ in cui S è un insieme di stati e (o_S, d_S) è una coppia di funzioni tali che:

1. $o_S : S \longrightarrow K_\perp$
2. $d_S : SxA \longrightarrow S_\perp$
3. $o_S(s)\uparrow \Rightarrow d_S(s,a)\uparrow \forall a \in A$

Siano $(S, (o_S, d_S))$ $\quad (T, (o_T, d_T))$ due automi di Moore con A in input e K in output:

Siano $(S, (o_S, d_S)) \quad (T, (o_T, d_T))$ due automi di Moore con A in input e K in output:

Definizione

Un omomorfismo tra due automi di Moore è una funzione f tale che:

1. $f : S \to T$
2. $o_S(s) = o_T(f(s))$
3. $f_\perp(d_S(s, a)) = d_T(f(s), a)$

$\forall s \in S \forall a \in A$

Siano $(S, (o_S, d_S)) \quad (T, (o_T, d_T))$ due automi di Moore con A in input e K in output:

Definizione

Una bisimulazione tra automi di Moore è un insieme $R \subseteq SxT$ tale che per ogni $(s, t) \in R$:

1. $o_S(s) = o_T(t)$
2. $(d_S(s, a), d_T(t, a)) \in R_\perp \forall a \in A$

Siano $(S, (o_S, d_S)) \quad (T, (o_T, d_T))$ due automi di Moore con A in input e K in output:

Definizione

Una bisimulazione tra automi di Moore è un insieme $R \subseteq SxT$ tale che per ogni $(s, t) \in R$:

1. $o_S(s) = o_T(t)$
2. $(d_S(s, a), d_T(t, a)) \in R_{\perp} \forall a \in A$

Definition

Sia M un automa di Moore e $n, m \in M$. n e m sono bisimili $(n \sim m)$ se esiste una bisimulazione R tra e M e M stesso tale che nRm

Definizione

Siano A e K insiemi, σ si dice una serie di potenze formale parziale se si ha che:

1. $\sigma : A^* \to K_\perp$
2. $\forall w \in A^* \sigma(w) \uparrow \Rightarrow \sigma(wa) \uparrow \quad \forall a \in A$

Definizione

Siano A e K insiemi, σ si dice una serie di potenze formale parziale se si ha che:

1. $\sigma : A^* \to K_\perp$
2. $\forall w \in A^* \sigma(w) \uparrow \Rightarrow \sigma(wa) \uparrow \quad \forall a \in A$

Definizione

l'insieme delle serie di potenze formali parziali si denota con $K\langle A \rangle$

Definizione

Siano A e K insiemi, σ si dice una serie di potenze formale parziale se si ha che:

1. $\sigma : A^* \to K_\perp$
2. $\forall w \in A^* \sigma(w) \uparrow \Rightarrow \sigma(wa) \uparrow \quad \forall a \in A$

Definizione

l'insieme delle serie di potenze formali parziali si denota con $K\langle A \rangle$

Esempio

Se $A = \{x\}$ e $K = R$ si può vedere $A\langle R \rangle$ come l'insieme delle serie di potenze a coefficienti reali. Infatti ad ogni $\sigma \in A\langle R \rangle$ si può associare in modo naturale la serie: $\sum \sigma(x^n) x^n$.

Definizione

$\sigma_a(w) = \sigma(aw)$

Definizione

$\sigma_a(w) = \sigma(aw)$

Definizione

l'automa di Moore $(M_K\langle A\rangle, (o_M, d_M))$ con A in input e K in output e in cui

1. $o_M(\sigma) = \sigma(\varepsilon)$
2. $d_M(\sigma, a) = \sigma_a$

Definizione

$\sigma_a(w) = \sigma(aw)$

Definizione

l'automa di Moore $(M_K\langle A\rangle, (o_M, d_M))$ con A in input e K in output e in cui

1. $o_M(\sigma) = \sigma(\varepsilon)$
2. $d_M(\sigma, a) = \sigma_a$

è detto automa di Moore delle serie di potenze formali parziali.

Definizione

Un insieme M con una operazione $\oplus$ associativa e con elemento neutro è detto monoide

Definizione

Un insieme M con una operazione $\oplus$ associativa e con elemento neutro è detto monoide

Definizione

Un insieme A in cui siano definite due operazioni $\oplus$ e $\otimes$ per cui $(A, \oplus)$ e $(A, \otimes)$ siano monoidi commutativi e per le quali valga che :

1. $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
2. $a \otimes 0 = 0$

si dice semianello

Definizione

Sia V insieme e $A, B \subseteq_f \mathcal{P}_f(V)$:

Definizione

Sia V insieme e $A, B \subseteq_f \mathcal{P}_f(V)$:

1. A si dice saturato se $\forall X \in A$ vale che se $X \subseteq Y \subseteq \cup A$ allora Y$\in A$

Definizione

Sia V insieme e $A, B \subseteq_f \mathcal{P}_f(V)$:

1. A si dice saturato se $\forall X \in A$ vale che se $X \subseteq Y \subseteq \cup A$ allora Y$\in A$
2. S(A) è il piu piccolo insieme saturato che contiene A

Definizione

Sia V insieme e $A, B \subseteq_f \mathcal{P}_f(V)$:

1. A si dice saturato se $\forall X \in A$ vale che se $X \subseteq Y \subseteq \cup A$ allora Y$\in A$
2. S(A) è il piu piccolo insieme saturato che contiene A
3. $\mathfrak{F}(V)$ è la famiglia dei saturati di V

Definizione

Sia V insieme e $A, B \subseteq_f \mathcal{P}_f(V)$:

1. A si dice saturato se $\forall X \in A$ vale che se $X \subseteq Y \subseteq \cup A$ allora Y$\in A$
2. S(A) è il piu piccolo insieme saturato che contiene A
3. $\mathfrak{F}(V)$ è la famiglia dei saturati di V

Definizione

Sia A un insieme, la terna $(\mathfrak{F}(A), \oplus, \otimes)$in cui:

1. $F \oplus G = S(F \cup G)$
2. $F \otimes G = S(\{X \cup Y | X \in F, Y \in G\})$

è un semianello che chiamo K_T in cui gli elementi neutri sono:$\emptyset$ $e\{\emptyset\}$

Definizione

Sia A un insieme in input e $\sigma \in K_T \langle A \rangle$

Definizione

Sia A un insieme in input e $\sigma \in K_T \langle A \rangle$

1. 0 è la funzione che manda tutto in $\emptyset$

Definizione

Sia A un insieme in input e $\sigma \in K_T \langle A \rangle$

1. 0 è la funzione che manda tutto in $\emptyset$
2. Ω è la funzione che manda tutto in $\perp$

Definizione

Sia A un insieme in input e $\sigma \in K_T \langle A \rangle$

1. 0 è la funzione che manda tutto in $\emptyset$
2. Ω è la funzione che manda tutto in $\perp$
3. $supp(\sigma) = \{a | \sigma_a \neq 0\}$

Definizione

Sia A un insieme in input e $\sigma \in K_T \langle A \rangle$

1. 0 è la funzione che manda tutto in $\emptyset$
2. Ω è la funzione che manda tutto in $\perp$
3. $supp\left(\sigma\right) = \{a|\sigma_a \neq 0\}$
4. σ è consistente se: $\sigma \neq 0$ e se $\forall w \in A^*$ per cui $\sigma_w \neq 0$ si ha che $\sigma_w\left(\varepsilon\right) \neq \emptyset$ e $supp\left(\sigma_w\right) = \cup \sigma_w\left(\varepsilon\right)$

Definizione

Sia A un insieme in input e $\sigma \in K_T \langle A \rangle$

1. 0 è la funzione che manda tutto in $\emptyset$
2. Ω è la funzione che manda tutto in $\perp$
3. $supp\,(\sigma) = \{a | \sigma_a \neq 0\}$
4. σ è consistente se: $\sigma \neq 0$ e se $\forall w \in A^*$ per cui $\sigma_w \neq 0$ si ha che $\sigma_w\,(\varepsilon) \neq \emptyset$ e $supp\,(\sigma_w) = \cup \sigma_w\,(\varepsilon)$
5. $K_T^c \langle A \rangle = \{\sigma \in K_T \langle A \rangle\, | \sigma \grave{e}\, consistente\}$

Lemma

$K_T^c \langle A \rangle \cup \{0\}$ è chiuso per derivazione.

Lemma

$K_T^c \langle A \rangle \cup \{0\}$ è chiuso per derivazione.

Lemma

Sia σ consistente. $\forall a \in A$ vale una e una sola delle seguenti:

1. $\exists n : \sigma_{a^k} = 0 \forall k \geqslant n$
2. $\exists n : \sigma_{a^k} = \Omega \forall k \geqslant n$
3. $\sigma_{a^k} \neq 0, \Omega \forall k$

Definizione

Se A è un insieme di azioni definisco con $\mathcal{P}^e$ l'insieme dei termini chiusi derivabili dalla sintassi del CSP estesa con delle costanti $\bar{\sigma}$ $\forall \sigma \in K_T^c \langle A \rangle$

Definizione

Se A è un insieme di azioni definisco con $\mathcal{P}^e$ l'insieme dei termini chiusi derivabili dalla sintassi del CSP estesa con delle costanti $\bar{\sigma}$ $\forall \sigma \in K^c_T \langle A \rangle$

Se $D \subseteq_f \mathcal{P}^e$, $D = \{P1, P2, \ldots., Pn\}$ uso la seguente notazione:

1. $\underset{P \in D}{\sqcap} P$ al posto di $P1 \sqcap \ldots . \sqcap Pn$
2. $\underset{P \in D}{\Box} P$ al posto di $P1 \Box \ldots . \Box Pn$

Definizione

Se A è un insieme di azioni definisco con $\mathcal{P}^e$ l'insieme dei termini chiusi derivabili dalla sintassi del CSP estesa con delle costanti $\bar{\sigma}$ $\forall \sigma \in K^c_T \langle A \rangle$

Definizione

definisco $\bar{\sigma}$ come:

- $\displaystyle\bigsqcap_{L \in \sigma(\varepsilon)} \Box_{a \in L} a.\bar{\sigma}_a$ se $\sigma(\varepsilon) \downarrow$
- $rec_x \tau.x$ se $\sigma(\varepsilon) \uparrow$

Definizione

Se A è un insieme di azioni definisco con $\mathcal{P}^e$ l'insieme dei termini chiusi derivabili dalla sintassi del CSP estesa con delle costanti $\bar{\sigma}$ $\forall \sigma \in K_T^c \langle A \rangle$

Definizione

definisco $\bar{\sigma}$ come:

- $\underset{L \in \sigma(\varepsilon)}{\sqcap} \underset{a \in L}{\square} a.\bar{\sigma}_a$ se $\sigma(\varepsilon) \downarrow$
- $rec_x \tau.x$ se $\sigma(\varepsilon) \uparrow$

Definisco la semantica per la sintassi appena definita: constante:

$$\frac{\bar{\sigma} := P, P \xrightarrow{\mu} P'}{\bar{\sigma} \xrightarrow{\mu} P'}$$

Definizione

Sia Aut $= \left(\mathcal{P}^e \cup \bar{0}, (o, d)\right)$ in cui:

Definizione

Sia Aut $= \left(\mathcal{P}^e \cup \bar{0}, (o, d)\right)$ in cui:

1. - $o\left(\bar{0}\right) = \emptyset$
   - $o\left(P\right) = S\left(A\left(p, \varepsilon\right)\right)$ se $P \downarrow$
   - $o\left(P\right) = \perp$ se $P \uparrow$

Definizione

Sia Aut $= \left(\mathcal{P}^e \cup \bar{0}, (o, d)\right)$ in cui:

1.
   - $o\left(\bar{0}\right) = \emptyset$
   - $o\left(P\right) = S\left(A\left(p, \varepsilon\right)\right)$ se $P \downarrow$
   - $o\left(P\right) = \perp$ se $P \uparrow$
2.
   - $d\left(\bar{0}, a\right) = \bar{0} \quad \forall a \in A$
   - $d\left(P, a\right) = \oplus \left\{Q | P \stackrel{a}{\Rightarrow} Q\right\}$ se $P \downarrow a$
   - $d\left(P, a\right) = \perp$ se $P \uparrow a$

Definizione

Sia Aut $= \left(\mathcal{P}^e \cup \bar{0}, (o, d)\right)$ in cui:

1.
   - $o\left(\bar{0}\right) = \emptyset$
   - $o\left(P\right) = S\left(A\left(p, \varepsilon\right)\right)$ se $P \downarrow$
   - $o\left(P\right) = \perp$ se $P \uparrow$
2.
   - $d\left(\bar{0}, a\right) = \bar{0} \quad \forall a \in A$
   - $d\left(P, a\right) = \oplus \left\{Q | P \stackrel{a}{\Rightarrow} Q\right\}$ se $P \downarrow a$
   - $d\left(P, a\right) = \perp$ se $P \uparrow a$

Ho allora che Aut è un automa di Moore.

Definition

Se ho che $A, B \subseteq_f \mathcal{P}(V)$ dico che $A \subset\subset B$ se
$\forall X \in A \; \exists Y \in B$ tale che $Y \subseteq X$

Definition

Se ho che $A, B \subseteq_f \mathcal{P}(V)$ dico che $A \subset\subset B$ se $\forall X \in A \; \exists Y \in B$ tale che $Y \subseteq X$

Diamo ora una definizione di equivalenza testing specifica per il CSP.

Definition

Se ho che $A, B \subseteq_f \mathcal{P}(V)$ dico che $A \subset\subset B$ se $\forall X \in A\ \exists Y \in B$ tale che $Y \subseteq X$

Diamo ora una definizione di equivalenza testing specifica per il CSP.

Definition

Siano P e Q processi, Diremo che $P \simeq Q$ se $\forall w \in A^*$ si ha che:

1. $P \downarrow w \Leftrightarrow Q \downarrow w$
2. se $P \downarrow w$ allora $A(P, w) \asymp A(Q, w)$

Teorema

Siano $P, Q \in \mathcal{P}^e$. *Allora* $P \simeq Q \iff P \sim Q$ *in Aut*