Strutture ad eventi

Mauro Piccolo

14 settembre 2007

Indice

Lo studio dei fenomeni e delle problematiche legate alla concorrenza sono di fatto un'attività basilare che interessa differenti ambiti nella realtà e in molti livelli, dal microchip fino ai sistemi multiprocessore. Il multitasking, ovvero la capacità di eseguire più di un compito parallelamente, da parte dei moderni sistemi reattivi è divenuto caratteristica essenziale.

Tuttavia un concetto che a prima vista può apparire semplice come eseguire due o più azioni (o insiemi di azioni) in modo contemporaneo, può essere fonte di problemi, spesso di difficile soluzione: problemi come il deadlock, la starvation (o livelock), l'inconsistenza dovuta all'accesso a risorse condivise, la mutua esclusione sono molto studiati soprattutto dai progettisti di sistemi concorrenti in quanto di difficile individuazione. Quindi, dire che un sistema concorrente è corretto (ovvero libero dai problemi sopra citati) è difficile da mostrare.

Ecco dunque che si rende necessario studiare il problema ad un livello più alto creando un modello astratto del sistema che si vuole costruire, sul quale verranno dimostrate le proprietà di correttezza desiderate, e (se il modello è abbastanza accurato) tali proprietà si riflettono sul sistema concreto.

In letteratura sono stati utilizzati vari formalismi per modellare sistemi reattivi concorrenti, come riportato in [WN95]. Lo scopo di questo capitolo è dare un'introduzione generale, per poi concentrarsi in uno specifico modello molto utilizzato per studiare processi concorrenti non deterministici, ovvero le strutture di eventi [NPW81, Win87, Win82].

1 Modelli ad interleaving vs modelli causali

I modelli per la concorrenza possono essere classificati in base a vari criteri. Una possibile classificazione prevede la distinzione fra *modelli ad interleaving* e *modelli causali*.

Nei modelli ad interleaving, la concorrenza è modellata attraverso la scelta non deterministica tra tutte le possibili sequenzializzazioni delle azioni concorrenti compiute dal sistema, mentre nei modelli causali viene definita una relazione di concorrenza e di mutua esclusione (conflitto) fra le azioni che compongono i vari task.

Esempio: Siano P_1 e P_2 due processi che devono eseguire in parallelo (indicato come $P_1 \| P_2$) e che compongono un sistema concorrente reattivo[1]. Il processo P_1 svolge prima l'azione a_1 e poi l'azione a_2, mentre il processo P_2 svolge l'azione b.

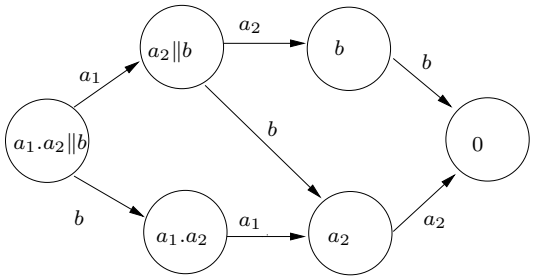
[1]Un sistema concorrente è difficile da caratterizzare: nel seguito utilizzeremo una sintassi CCS-like per descrivere un sistema concorrente con i relativi task.

Vediamo ora come $P_1 \| P_2$ possa essere modellato in ciascuna delle due istanze di modello

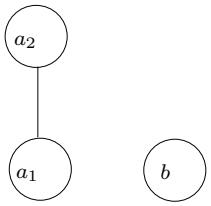
modelli ad interleaving: in questa tipologia di modello, $P_1 \| P_2$ viene modellato esprimendo tutte le possibili schedulazioni delle azioni compiute dai due processi in parallelo. Ciò può essere espresso dando tutte le possibili tracce del sistema, che nel nostro caso sono

$$\{a_1 a_2 b, a_1 b a_2, b a_1 a_2\}$$

oppure mediante un automa



modelli causali: in questa tipologia di modello, $P_1 \| P_2$ viene modellato esplicitando le relazioni di ordine e di conflitto che vi sono fra le azioni, per poter identificare quali azioni possono essere svolte in concorrenza. Nel nostro caso, abbiamo un insieme di azioni $A = \{a_1, a_2, b\}$ su cui è definita una relazione $<= \{(a_1, a_2)\}$ e da questo possiamo inferire che, l'azione b può essere svolta contemporaneamente (in concorrenza) con le azioni a_1 ed a_2



I modelli ad interleaving sono ideali per evidenziare aspetti osservazionali del sistema: ad esempio sono utili per focalizzare l'attenzione su tutte le possibili interazioni del sistema con l'ambiente esterno. In particolare, l'utilizzo di modelli ad interleaving è utile per delineare equivalenze osservazionali di due o più sistemi, modellate tramite relazioni di *bisimulazione*:

nella realtà, due sistemi sono equivalenti osservazionalmente se essi si comportano allo stesso modo [2]: nel modello significa che due oggetti sono bisimili se anch'essi, nel modello si evolvono allo stesso modo a partire da una data condizione[3].

Vi sono tuttavia interessanti proprietà di un sistema concorrente, come ad esempio l'assenza di conflitto fra azioni, l'indipendenza delle scelte da eventuali politiche di scheduling e la sequenzialità che non sono percepite come proprietà osservazionali e che, utilizzando un modello ad interleaving, sono difficili da esprimere. Queste sono proprietà legate alla causalità delle azioni, che è una caratteristica catturata dai modelli causali, utilizzando i quali è possibile esprimere queste proprietà in modo semplice. Al contrario per i modelli ad interleaving, tali proprietà sono di difficile espressione, in quanto la causalità delle azioni non è una proprietà osservabile.

Quindi i modelli ad interleaving sono pensati per caratterizzare un sistema dal punto di vista osservazionale, mentre i modelli causali vengono utilizzati per esprimere proprietà legate alla causalità delle azioni.

2 Strutture di eventi

Nell'ambito dei modelli causali, le strutture di eventi sono state oggetto di numerosi studi, in quanto esse costituivano un framework generale, in cui altre strutture potevano trovare espressione [WN95]. Esse furono introdotte da Nielsen, Plotkin e Winskel [NPW81, Win80] come un modello per sistemi concorrenti non deterministici e sono apparse in forme differenti. Quella che verrà utilizzata nel seguito sarà la nozione di *struttura di eventi prima* [Win87].

2.1 Definizioni

Come affermato precedentemente, le strutture di eventi sono utilizzate per modellare sistemi concorrenti non deterministici. Informalmente, una struttura di eventi modella la computazione parallela attraverso

- eventi che sono occorrenze di azioni

[2] In un contesto macchina a stati, si intende dallo stesso stato oppure espresso in modo più concreto, dallo stesso contenuto di dati in memoria, i quali sono situati in una stessa posizione.

[3] più precisamente, se un oggetto è in grado di effettuare una determinata azione e ad arrivare ad un determinato stato (che nella realtà rappresenta una data configurazione del sistema), allora anche l'altro oggetto è in grado di fare altrettanto e di giungere ad uno stato che sia comparabile con lo stato a cui è arrivato l'oggetto precedente

- un ordinamento parziale che esprime la dipendenza causale delle azioni

mentre il non determinismo è espresso attraverso

- una relazione di conflitto che esprime come l'occorrenza di determinate azioni escluda l'occorenza di altre

Quindi due eventi sono detti concorrenti se non sono nè legati causalmente e nemmeno in conflitto, e due eventi sono in conflitto se vivono in differenti evoluzioni del sistema.

Definizione 2.1 (Struttura di eventi)**.** *Una struttura di eventi è una tripla* $\mathcal{E} = \langle E, \leq, \smile \rangle$ *dove*

- E *è un insieme finito o numerabile di* ***eventi***
- $\langle E, \leq \rangle$ *è un ordine parziale, detto* ***ordine causale***
- *per ogni* $e \in E$*, l'insieme* $[e) = \{e' | e' < e\}$*, detto* ***insieme abilitante di*** e*, è finito*
- $\smile$ *è una relazione simmetrica e non riflessiva, detta* ***relazione di conflitto****, soddisfacente la* ***proprietà di ereditarietà***

$$\forall e_1, e_2, e_3 \in E. e_1 \leq e_2 \wedge e_1 \smile e_3 \Rightarrow e_2 \smile e_3 \tag{1}$$

Con un abuso di notazione, nel seguito, data una strutura ad eventi $\mathcal{E}$ si scriverà che $e \in \mathcal{E}$ per indicare che e appartiene all'insieme degli eventi di $\mathcal{E}$. Inoltre dato $e \in \mathcal{E}$ si denoterà

- $parents(e)$ l'insieme degli eventi massimali secondo $\leq$ di $[e)$.
- $\lceil e \rceil = [e) \cup \{e\}$

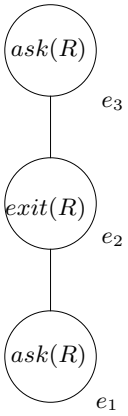
In base all'ereditarietà è possibile distinguere i conflitti tra eventi tra **conflitti ereditati** e non: quindi dati $e_1, e_2, e_3 \in \mathcal{E}$ diciamo che $e_2 \smile e_3$ è un conflitto ereditato da $e_1 \smile e_3$ se $e_1 < e_2$. Se $e_1 \smile e_3$ non è ereditato da nessun altro conflitto allora esso si dice immediato e si indica con $e_1 \smile_\mu e_3$. La **chiusura riflessiva dei conflitti** (risp. dei conflitti immediati) si indica con $\asymp$ (risp. $\asymp_\mu$).

Remark 2.1. *Si noti inoltre che il conflitto e l'ordinamento causale sono mutualmente esclusivi.*

Due eventi si dicono **concorrenti** se non sono nè in conflitto, nè legati causalmente.

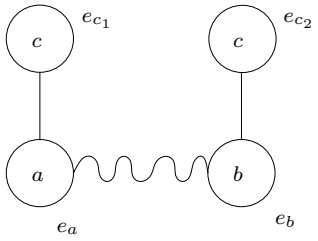
E' importante notare che gli eventi qui sono intesi come *occorrenze* di azioni. Si prenda ad esempio la seguente situazione.

Esempio 1: si intende modellare con una strutura di eventi la seguente semplice situazione: si supponga che un processo consumatore richieda due volte l'accesso ad una risorsa R: la prima volta la trova libera e vi accede e poi ne esce e poi la accede di nuovo. Dunque le azioni che compie sono $ask(R)$ e $exit(R)$ e poi nuovamente $ask(R)$. La corrispondente struttura di eventi è dunque:



Si noti che gli eventi e_1 ed e_3 sono occorrenze della stessa azione $ask(R)$.

Esempio 2: un esempio più strutturato un sistema concorrente può eseguire in modo mutualmente esclusivo le azioni a e b e può eseguire l'azione c solo dopo l'esecuzione di una delle due azioni precedenti. Vogliamo costruire la corrispondente struttura di eventi $\mathcal{E}$. Ovviamente avremo due eventi $e_a, e_b \in \mathcal{E}$ corrispondenti alle azioni a e b con $e_a \smile e_b$, le quali occorrono una sola volta. L'azione c invece occorre due volte: o dopo a o dopo b e quindi avremo due eventi $e_{c_1}, e_{c_2} \in \mathcal{E}$ con $e_a \leq e_{c_1}$ ed $e_b \leq e_{c_2}$: si noti che per ereditarietà abbiamo $e_{c_1} \smile e_{c_2}$. Nel seguito vi è una rappresentazione grafica della struttura di eventi costruita: i conflitti immediati sono indicati con linee ondulate.



Il precedente esempio induce ad introdurre una nozione di struttura di eventi, a cui ad ogni evento è associata l'azione (etichetta) corrispondente: abbiamo dunque la seguente

Definizione 2.2 (Struttura di eventi etichettata)**.** *Una struttura di eventi etichettata è una quadrupla* $\mathcal{E} = \langle E, \leq, \smile, \lambda \rangle$ *dove:*

- $\langle E, \leq, \smile \rangle$ *è una struttura di eventi.*
- $\lambda : E \to L$ *è una funzione di etichettaggio, in cui* L *è un insieme di etichette (azioni).*

Nel seguito, ove non specificato diversamente, quando si parlerà di struttura di eventi si intenderà una struttura di eventi etichettata, ovvero una struttura di eventi con una funzione di etichettatura.

Esempio: Prendendo come riferimento il precedente esempio, possiamo costruire la relativa event structure etichettata $\mathcal{E}_{lab}$ con $\lambda : E \to L$ dove $L = \{a, b, c\}$, con l'etichettatura $\lambda(e_a) = a, \lambda(e_b) = b, \lambda(e_{c_1}) = c, \lambda(e_{c_2}) = c$.

Il precedente esempio e la naturale interpretazione aiutano a formulare una nozione di stato (traccia) della computazione di una struttura di eventi $\mathcal{E}$. Una nozione ragionevole può essere data dall'insieme di azioni che sono occorse nella computazione (gli eventi) e in particolare per insiemi sì fatti ci si aspetta che

- se un evento appartiene a questo insieme, allora tutti gli eventi dai quali esso dipende causalmente devono appartenere a questo insieme
- non ci sono due eventi in conflitto fra loro che appartengono a questo insieme

Da qui quindi abbiamo la seguente:

Definizione 2.3 (configurazione)**.** *Sia* $\mathcal{E}$ *una struttura di eventi e sia* $C \subseteq \mathcal{E}$ *un insieme di eventi.* C *è una configurazione se valgono le seguenti proprietà*

assenza di conflitto

$$\forall e, e' \in C . e \not\smile e'$$

chiusura verso il basso

$$\forall e, e' \in \mathcal{E} \, . \, (e' \leq e \wedge e \in C) \Rightarrow e' \in C$$

Data una struttura di eventi $\mathcal{E}$, si denota con $\mathcal{D}(\mathcal{E})$ l'insieme delle configurazioni di $\mathcal{E}$. Si noti che dato $e \in \mathcal{E}$, $[e), \lceil e \rceil$ sono configurazioni. Si denota invece con $\mathcal{D}^0(\mathcal{E})$ l'insieme delle configurazioni finite.

Proposizione 2.1. *Sia $\mathcal{E} = \langle E, \leq, \smile \rangle$ una struttura di eventi. Allora*

- $e \leq e' \iff \forall C \in D^0(\mathcal{E}).e' \in C \Rightarrow e \in C$
- $e \smile e' \iff \forall C \in D^0(\mathcal{E}).e \in C \Rightarrow e' \notin C$

La dimostrazione è diretta ed è lasciata al lettore come esercizio.

Un evento e si dice *abilitato* da una configurazione C se $C \cup \{e\}$ è una configurazione. Due configurazioni C_1, C_2 si dicono *compatibili* se $C = C_1 \cup C_2 \in \mathcal{D}(\mathcal{E})$.

Lemma 2.1 (compatibilità)**.** *Sia $\mathcal{E}$ una struttura di eventi e siano e_1, e_2 due eventi distinti. $e_1 \smile e_2$ se e solo se $\lceil e_1 \rceil$ e $\lceil e_2 \rceil$ non sono compatibili.*

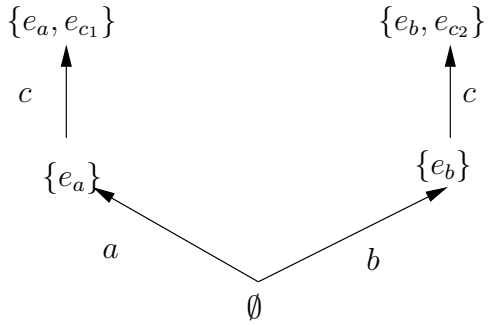
Dimostrazione. Esercizio □

La relazione di transizione da una configurazione all'altra, data come sistema di transizione etichettato è pertanto definita come segue:

Definizione 2.4 (Labelled Transition System)**.** *Sia $\mathcal{E}$ una struttura di eventi, siano $C, C' \in \mathcal{L}(\mathcal{E})$ due configurazioni e sia $a \in L$ un'etichetta. Si scrive*

$$C \xrightarrow{a} C' \iff \exists e \in \mathcal{E} \,.\, \lambda(e) = a \wedge e \notin C \wedge C' = C \cup \{e\}$$

Esempio: prendendo come riferimento il precedente esempio, il sistema di transizione etichettata indotto dalla struttura di eventi $\mathcal{E}$ è il seguente:



2.2 Una categoria di strutture ad eventi

È possibile definire una nozione di morfismo sulle strutture ad eventi, nel modo seguente:

Definizione 2.5 (morfismo sulle strutture ad eventi)**.** *Siano $\mathcal{E} = \langle E, \leq, \smile \rangle$ ed $\mathcal{E}' = \langle E', \leq', \smile' \rangle$ due strutture ad eventi. Un morfismo fra $\mathcal{E}$ ed $\mathcal{E}'$ è una qualunque mappa parziale $f : E \to E'$ sugli eventi che soddisfa le seguenti:*

- $\mathcal{C} \in \mathcal{D}(\mathcal{E}) \Rightarrow f(C) \in \mathcal{D}(\mathcal{E}')$
- $\forall C \in \mathcal{D}(\mathcal{E}) \forall e_1, e_2 \in C. f(e_1) = f(e_2) \Rightarrow e_1 = e_2$

Un morfismo $f : \mathcal{E} \to \mathcal{E}'$ tra strutture di eventi esprime come il comportamento in $\mathcal{E}$ determina il comportamento in $\mathcal{E}'$. La funzione parziale f esprime come l'occorrenza di un evento in $\mathcal{E}$ implica la simultanea occorrenza di un evento in $\mathcal{E}'$; il fatto che $f(e) = e'$ può essere interpretato come l'espressione del fatto che l'evento e' è una componente dell'evento e' e, in un certo senso, che l'occorrenza di e implica la simultanea occorrenza di e'.

I morfismi fra strutture di eventi possono essere descritti in modo più diretto in termini delle relazioni di causalità e di conflitto delle strutture ad eventi.

Proposizione 2.2. *Un morfismo di strutture ad eventi da* $\langle E, \leq, \smile \rangle$ *a* $\langle E', \leq' , \smile' \rangle$ *è una funzione parziale* $f : E \to E'$ *tale che*

- $f(e)$ *definita* $\Rightarrow \lceil f(e) \rceil \subseteq f(\lceil e \rceil)$
- $f(e), f(e')$ *entrambe definite e* $f(e) \asymp f(e') \Rightarrow e \asymp e'$

La categoria delle strutture ad eventi possiede prodotti e co-prodotti, utili per modellare la composizione parallela e la somma non deterministica

2.2.1 Co-prodotto di strutture ad eventi

Siano $\mathcal{E}_1 = \langle E_1, \leq_1, \smile_1 \rangle$ ed $\mathcal{E}_2 = \langle E_2, \leq_2, \smile_2 \rangle$ due strutture ad eventi. Il loro co-prodotto $\mathcal{E}_1 + \mathcal{E}_2$ è la struttura di eventi $\langle E_1 + E_2, \leq, \smile \rangle$ dove

$$\begin{aligned} e \leq e' \iff & \exists e_1, e_1' \in E_1. e_1 \leq_1 e_1' \wedge in_1(e_1) = e \wedge in_1(e_1') = e' \text{ or} \\ & \exists e_2, e_2' \in E_2. e_2 \leq_2 e_2' \wedge in_2(e_2) = e \wedge in_2(e_2') = e' \end{aligned}$$

e

$$\begin{aligned} e \smile e' \iff & \exists e_1, e_1' \in E_1. e_1 \smile_1 e_1' \wedge in_1(e_1) = e \wedge in_1(e_1') = e' \text{ or} \\ & \exists e_2, e_2' \in E_2. e_2 \smile_2 \mathfrak{e}_2' \wedge in_2(e_2) = e \wedge in_2(e_2') = e_2' \text{ or} \\ & \exists e_1 \in E_1, e_2 \in E_2. (in_1(e_1) = e \wedge in_2(e_2) = e') \vee (in_1(e_1) = e' \wedge in_2(e_2) = e) \end{aligned}$$

Dove $in_1 : E_1 \to E_1 + E_2$ ed $in_2 : E_2 \to E_1 + E_2$ sono rispettivamente le iniezioni di E_1 ed E_2 nelle loro unioni disgiunte.

2.2.2 Prodotto di strutture ad eventi

Siano $\mathcal{E}_1 = \langle E_1, \leq_1, \smile_1 \rangle$ ed $\mathcal{E}_2 = \langle E_2, \leq_2, \smile_2 \rangle$ due strutture di eventi.

Si aggiunga a ciascun E_i un evento fittizio $e_i^\star$ che rappresenta l'occorrenza di un'azione effettuata dall'ambiente esterno (si denota con $E_i^\star$ l'insieme degli eventi della struttura di eventi $\mathcal{E}_i$ con aggiunto l'evento fittizio $e_i^\star$).

Si consideri l'insieme $\widetilde{E}$ ottenuto come soluzione iniziale dell'equazione $X = \mathcal{P}_{fin}(X) \times E_1^\star \times E_2^\star$. I suoi elementi hanno la forma (x, e_1, e_2) con x finito, $x \subseteq \widetilde{E}$. Questo permette di definire induttivamente una nozione di *altezza*[4] di un elemento di $\widetilde{E}$.

$$\begin{aligned} h(\emptyset, e_1, e_2) &= 0 \\ h(x, e_1, e_2) &= \max\{h(e) | e \in x\} + 1 \end{aligned}$$

Si definisce $\mathcal{E}_1 \times \mathcal{E}_2 = \langle E, \leq, \smile \rangle$, con $E \subseteq \widetilde{E}$ come segue, per induzione sull'altezza di $e \in E$:

caso base: si ha che $(\emptyset, e_1, e_2) \in E$ se

1. $e_1 \in E_1, e_2 \in E_2$ e e_1 minimale in E_1 e e_2 minimale in E_2 o
2. $e_1 \in E_1, e_2 = e_2^\star$ e e_1 minimale in E_1 o
3. $e_1 = e_1^\star, e_2 \in E_2$ e e_2 minimale in E_2

Gli elementi di altezza 0 non sono confrontabili.
Infine, per il conflitto, si ha $(\emptyset, e_1, e_2) \asymp (\emptyset, d_1, d_2)$ se $e_1 \asymp_1 d_1$ o $e_2 \asymp_2 d_2$

passo induttivo: si assuma che tutti gli elementi in E di altezza $\leq n$ siano stati definiti e su di essi siano definiti ordine e conflitto. Sia (x, e_1, e_2) un elemento di altezza $n+1$. Sia y l'insieme degli elementi massimali in x. Siano $y_1 = \{d_1 \in E_1 | (z, d_1, d_2) \in y\}$ e $y_2 = \{d_2 \in E_2 | (z, d_1, d_2) \in y\}$ le proiezioni di y sulle due componenti. Abbiamo che$(x, e_1, e_2) \in E$ se x è chiuso verso il basso e senza conflitti e inoltre:

1. Si supponga $e_1 \in E_1, e_2 = e_2^\star$. Allora deve valere che $y_1 = parents(e_1)$
2. Si supponga $e_2 \in E_2, e_1 = e_1^\star$. Allora deve valere che $y_2 = parents(e_2)$
3. Si supponga $e_1 \in E_1, e_2 \in E_2$. Allora

[4]Gran parte delle prove che coinvolgeranno il prodotto categoriale di strutture di eventi saranno per induzione sull'altezza degli elementi

- se $(z, d_1, d_2) \in y$ allora $d_1 \in parents(e_1)$ o $d_2 \in parents(e_2)$
- per ogni $d_1 \in parents(e_1)$, esiste un $(z, d_1, d_2) \in x$
- per ogni $d_2 \in parents(e_2)$, esiste un $(z, d_1, d_2) \in x$

4. Sia $x_1 = \{d_1 \in E_1 | (z, d_1, d_2) \in x\}$ e $x_2 = \{d_2 \in E_2 | (z, d_1, d_2) \in x\}$. Allora per nessun $d_1 \in x_1, d_1 \asymp_1 e_1$ e per nessun $d_2 \in x_2, d_2 \asymp_2 e_2$.

L'ordine parziale è esteso con $e \leq (x, e_1, e_2)$ se $e \in x$ o $e = (x, e_1, e_2)$. Si noti che se $e < e'$ allora $h(e) < h(e')$.
Infine, per i conflitti si prenda $e = (x, e_1, e_2)$ e $d = (z, d_1, d_2)$ dove o $h(e) = n + 1$, o $h(d) = n + 1$ o entrambi. Allora si ha $e \smile d$ se vale una delle seguenti:

- $e_1 \asymp_1 d_1$ o $e_2 \asymp_2 d_2$ con $e \neq d$
- esiste un $e' = (x', e'_1, e'_2) \in x$ tale che $e'_1 \asymp_1 d_1$ o $e'_2 \asymp_2 d_2$ e $e' \neq d$
- esiste un $d' = (z', d'_1, d'_2) \in z$ tale che $d'_1 \asymp_1 e_1$ o $d'_2 \asymp_2 e_2$ e $e \neq d'$
- esiste un $e' \in x$, $d' \in z$ tali che $e' \smile d'$

2.3 Domini di configurazioni

Vedendo gli stati della computazione come configurazioni, il progresso in una computazione è misurato dall'occorrenza di più eventi. Data una struttura di eventi $\mathcal{E}$, siano $x, y \in \mathcal{D}(\mathcal{E})$. Se $x \subseteq y$ allora x può essere visto come un sottocomportamento di y. La relazione di inclusione tra configurazioni è un ordine di informazioni comune nell'ambito della semantica denotazionale, ma speciale nel senso che più informazione corrisponde a più eventi occorsi. È facile vedere che l'ordine $(\mathcal{D}(\mathcal{E}), \subseteq)$ ha un l.u.b. dato dall'unione e che l'ordine è un cpo dove il minimo elemento è l'insieme vuoto.

La più semplice caratterizzazione dei domini rappresentati da strutture di eventi prime comincia osservando che un evento e in una struttura di eventi corrisponde alla configurazione $\lceil e \rceil$. Tali elementi sono caratterizzati come essere *primi completi* e i domini di configurazione hanno la proprietà che ogni elemento è il l.u.b. di questi elementi speciali.

Sia $(D, \sqsubseteq)$ un insieme parzialmente ordinato in cui si denota il l.u.b. di sottoinsiemi X con $\bigsqcup X$.

Si dice che D è *bounded complete* sse tutti i sottoinsiemi $X \subseteq D$ che ammettono un upper-bound in D, hanno un l.u.b. $\bigsqcup X$ in D.

Si dice che D è *coerente* sse tutti i sottoinsiemi $X \subseteq D$ tali che per ogni $d_1, d_2 \in X$ esiste un upper-bound $d \in D$ ammettono l.u.b $\bigsqcup X$ in D.

Un elemento $p \in D$ si dice primo completo se per ogni $X \subseteq D$ che ammette l.u.b. in D vale

$$p \sqsubseteq \bigsqcup X \Rightarrow \exists x \in X. p \sqsubseteq x$$

D si dice *primo algebrico* se per ogni $x \in D$

$$x = \bigsqcup \{p \sqsubseteq x | p \text{ è un primo completo}\}$$

Se inoltre gli insiemi $\{p \sqsubseteq q | p \text{ è un primo completo}\}$ sono sempre finiti quanto q è un primo completo, allora D è detto *finitario*.

Se D è bounded complete e primo algebrico, allora è detto *dominio primo algebrico*

Teorema 2.1. *Sia $\mathcal{E}$ una struttura di eventi. Allora $(\mathcal{D}(\mathcal{E}), \subseteq)$ è un dominio coerente, finitario e primo algebrico; i primi completi sono gli insiemi $\{\lceil e \rceil \, | e \in \mathcal{E}\}$*

Conversamente, ad ogni dominio coerente, finitario, primo algebrico è associato con una struttura di eventi in cui gli eventi sono i suoi primi completi.

Teorema 2.2. *Sia $(D, \sqsubseteq)$ un dominio coerente finitario primo algebrico. Si definisca $\mathcal{E} = \langle P, \leq, \smile \rangle$ dove P sono i primi completi di D, dove*

- $p \leq p' \iff p \sqsubseteq p'$
- $p \smile p' \iff p$ *e* p' *non ammettono upper bound.*

Allora $\mathcal{E}$ è una struttura di eventi.

Da questi due risultati si evince che le strutture di eventi e i domini finitari primi algebrici sono equivalenti; uno può essere usato per rappresentare l'altro. Questi domini sono esattamente i dI-domain di Berry [**?**].

3 Semantica di CCS nelle strutture di eventi

In questa sezione daremo una semantica denotazionale al linguaggio dei processi $Proc_L$ basata sul lavoro di Winskel [Win82]. Questo linguaggio é un linguaggio equivalente a CCS, eccezion fatta per l'operatore di ricorsione μ utilizzato al posto delle usuali equazioni ricorsive di CCS.

3.1 Algebra di sincronizzazione

In questa sezione analizziamo le etichette che utilizzeremo per etichettare gli eventi dei processi. Le possibili sincronizzazioni tra due processi che eseguono in parallelo sono determinate da un'algebra di sincronizzazione. Un'algebra di sincronizzazione specifica come, in base alle loro etichette, coppie di eventi siano combinate per formare eventi sincronizzati e quali etichette una tale combinazione genera.

Formalmente un'algebra di sincronizzazione non è altro che un operatore binario parziale sulle etichette. All'interno dell'insieme delle etichette L distinguiamo una costante speciale $\star$. Questa costante rappresenta un'azione indefinita. Nessun evento reale è etichettato con $\star$. Comunque quando due processi eseguono in parallelo, un evento di un processo può occorrere asincronicamente, senza sincronizzarsi con nessun evento dell'altro processo. Per questo motivo è enormemente conveniente pretendere, matematicamente, che quell'evento si sincronizzi con l'evento irreale etichettato con $\star$ (rappresentante il contesto), esattamente come si è fatto nella costruzione del prodotto categoriale.

Definizione 3.1. *Un'algebra di sincronizzazione è una terna $\langle L, \star, \cdot \rangle$, dove L è un insieme di etichette contenente $\star$ e $\cdot$ è un operatore parziale commutativo associativo sulle etichette avente come elemento neutro $\star$.*

Un esempio di algebra di sincronizzazione abbastanza comune è dato dalla seguente. Posto N un insieme di nomi e $\overline{N} = \{\overline{a} | a \in N\}$ abbiamo

- $L = N \cup \overline{N} \cup \{\tau, \star\}$
- $\cdot$ è un operatore parziale commutativo associativo con elemento neutro $\star$, soddisfacente $a \cdot \overline{a} = \tau$

L'algebra che abbiamo qui sopra presentato è un algebra utilizzata comunemente ed è la base del calcolo dei processi, e nel seguito sarà l'algebra di riferimento per il linguaggio $Proc_L$

3.2 Il linguaggio $Proc_L$

Sia $\langle L, \star, \cdot \rangle$ un'algebra di sincronizzazione e siano $\alpha, \beta, \gamma \ldots$ etichette differenti da $\star$. Si assuma inoltre un insieme numerabile di variabili di processo $X, Y, \ldots$. La sintassi di $Proc_L$ è generata dalla seguente grammatica:

$$P ::= 0 \mid X \mid \alpha.P \mid P + P \mid P|P \mid \ P\langle S \rangle \mid P \setminus \alpha \mid \mu X.P$$

dove S è un endomorfismo di L

3.2.1 Semantica operazionale

La semantica operazione di $Proc_L$ è data in termini di un sistema di transizione etichettato ed è la seguente:

$$\frac{}{\alpha.P \xrightarrow{\alpha} P}\ p \quad \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}\ sum - l \quad \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}\ par - l$$

$$\frac{P \xrightarrow{\alpha} P' \quad Q \xrightarrow{\beta} Q'}{P|Q \xrightarrow{\alpha \cdot \beta} P'|Q'}\ comm \quad \frac{P \xrightarrow{\alpha} P'}{P\langle S \rangle \xrightarrow{S(\alpha)} P'\langle S \rangle}\ sust \quad \frac{P \xrightarrow{\beta} P' \quad \alpha \neq \beta}{P \setminus \alpha \xrightarrow{\beta} P' \setminus \alpha}\ res$$

$$\frac{P[X := \mu X.P] \xrightarrow{\alpha} P'}{\mu X.P \xrightarrow{\alpha} P'}\ mu$$

È possibile dare, a partire dalla semantica etichettata, l'usuale definizione di **bisimulazione forte** ($\sim$) che è la più grande congruenza tra processi che rispetta la seguente proprietà:

- nel caso $P \sim Q$ e $P \xrightarrow{\alpha} P'$ allora $Q \xrightarrow{\alpha} Q'$ e $P' \sim Q'$.

3.3 Modello di $Proc_L$ in strutture di eventi

In questa sezione daremo un modello del linguaggio CCS in struture di eventi etichettate tratto da [Win82]. A questo scopo definiamo anzitutto alcune operazioni sulle strutture di eventi. Nel seguito assumiamo un algebra di sincronizzazione sulle etichette.

3.3.1 Operazioni sulle strutture di eventi

Prefissamento Sia $\mathcal{E}$ una struttura di eventi etichettata e sia $\alpha \in L \setminus \{\star\}$. Definiamo la struttura di eventi $\alpha.\mathcal{E} = \langle E, \leq, \smile, \lambda \rangle$ essere la struttura di eventi ottenuta aggiungendo ad $\mathcal{E}$ un nuovo evento minimo etichettato α.

Somma Siano $\mathcal{E}_1$ ed $\mathcal{E}_2$ due strutture di eventi etichettate. Definiamo $\mathcal{E}_1 + \mathcal{E}_2$ essere il coprodotto categoriale di $\mathcal{E}_1$ ed $\mathcal{E}_2$

Rietichettamento Sia $\mathcal{E}$ una struttura di eventi etichettata in L ed $f : L \to L'$. Definiamo $\mathcal{E}[f]$ essere la struttura di eventi ottenuta rietichettando $\mathcal{E}$ secondo f. Più formalmente, se $\lambda : \mathcal{E} \to L$ è la funzione di etichettatura di $\mathcal{E}$, allora la funzione di etichettatura di $\mathcal{E}[f]$ è $f \circ \lambda$.

Restrizione Sia $\mathcal{E}$ una struttura di eventi e sia $X \subseteq L$. Definiamo $\mathcal{E} \setminus X$ la struttura di eventi ottenuta da $\mathcal{E}$ togliendo tutti gli eventi etichettati con α, per ogni $\alpha \in X$ e tutti gli eventi maggiori di questi ultimi.

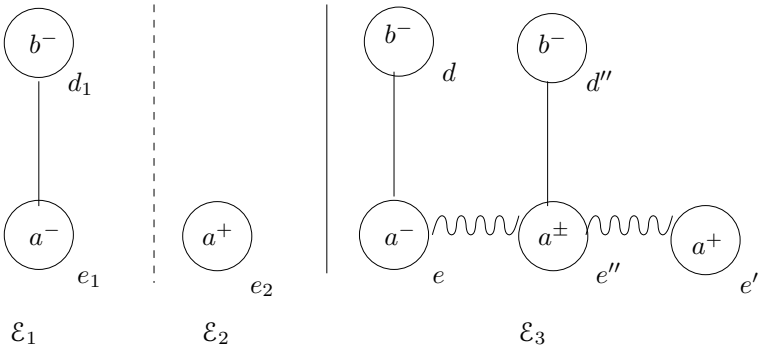
Composizione parallela Siano $\mathcal{E}_1$ ed $\mathcal{E}_2$ due strutture ad eventi. Definiamo $\mathcal{E}_1 | \mathcal{E}_2 = (\mathcal{E}_1 \times \mathcal{E}_2 \setminus X)[f]$ dove

- $\mathcal{E}_1 \times \mathcal{E}_2$ è il prodotto categoriale di $\mathcal{E}_1$ ed $\mathcal{E}_2$, con la funzione di etichettatura $\lambda : \mathcal{E}_1 \times \mathcal{E}_2 \to L \times L$ così definita

$$\lambda(x, e_1, e_2) = \begin{cases} (\star, \lambda_2(e_2)) & \text{se } e_1 = e_1^\star \\ (\lambda_1(e_1), \star) & \text{se } e_2 = e_2^\star \\ (\lambda_1(e_1), \lambda_2(e_2)) & \text{altrimenti} \end{cases}$$

- $X = \{(l_1, l_2) \in L \times L | \, l_1 \cdot l_2 \text{ è indefinito}\}$
- $f : L \times L \to L$ definita come $f(l_1, l_2) = l_1 \cdot l_2$

Esempio:



Si consideri la figura sovrastante: siano $\mathcal{E}_1, \mathcal{E}_2$ due strutture di eventi con $E_1 = \{e_1, d_1\}$ ed $E_2 = \{e_2\}$. I conflitti e l'ordine come da figura, e $\lambda(e_1) = a^-, \lambda(d_1) = b^-, \lambda(e_2) = a^+$. La struttura di eventi $\mathcal{E}_3 = \mathcal{E}_1|\mathcal{E}_2$ è così definita:

- $E_3 = \{e := (\emptyset, e_1, \star), e' := (\emptyset, \star, e_2), e'' := (\emptyset, e_4, e_5), d := (\{e\}, d_4, \star), d'' := (\{e''\}, d_4, \star)\}$
- l'ordine è definito con $e \leq d, e'' \leq d''$
- i conflitti sono $e \smile e'', e' \smile e''$ e tutti quelli ereditati da loro.

3.3.2 Un cpo di strutture di eventi

Diciamo che una struttura di eventi $\mathcal{E}$ è un prefisso di una struttura di eventi $\mathcal{E}'$ (scritto $\mathcal{E} \leq \mathcal{E}'$) se esiste una struttura di eventi $\mathcal{E}''$ isomorfa ad $\mathcal{E}'$ tale che $E \subseteq E''$ e nessun evento di $E'' \setminus E$ è minore di un qualunque evento di E. È possibile dimostrare [Win82] che la classe delle strutture ad eventi con la relazione $\leq$ forma un cpo, per cui il l.u.b. di catene crescenti esiste. È possibile inoltre dimostrare che tutti gli operatori di cui sopra sono continui.

3.3.3 Semantica denotazionale

Denotiamo con $\mathcal{EV}$ l'insieme delle strutture di eventi etichettate. Si definisce un **ambiente** una qualunque funzione $\rho : X \to \mathcal{EV}$ da variabili di processo a strutture di eventi etichettate. Da un termine P e un ambiente ρ definiamo $[\![P]\!]_\rho$ essere la struttura di eventi corripondente a P secondo l'ambiente ρ, per induzione strutturale su P nel modo seguente:

- $[\![0]\!]_\rho = \langle \emptyset, \emptyset, \emptyset \rangle$
- $[\![x]\!]_\rho = \rho(x)$
- $[\![\alpha.P]\!]_\rho = \alpha.[\![P]\!]_\rho$
- $[\![P_1 + P_2]\!]_\rho = [\![P_1]\!]_\rho + [\![P_2]\!]_\rho$
- $[\![P \setminus \alpha]\!]_\rho = [\![P]\!]_\rho \setminus \{\alpha\}$
- $[\![P_1|P_2]\!]_\rho = [\![P_1]\!]_\rho|[\![P_2]\!]_\rho$
- $[\![P\langle S \rangle]\!]_\rho = [\![P]\!]_\rho[S]$
- $[\![\mu x.P]\!]_\rho = fix\, F$ dove $F : \mathcal{EV} \to \mathcal{EV}$ definito come $F(\mathcal{E}) = [\![P]\!]_{\rho[x \leftarrow E]}$

Abbiamo il seguente:

Teorema 3.1. $[\![\cdot]\!]$ *è ben definito ovvero dato un termine* P *e un ambiente* ρ, $[\![P]\!]_\rho$ *è una struttura di eventi etichettata.*

È possibile dimostrare l'asserto di cui sopra tramite una semplice induzione sulla struttura del termine P. Il caso più difficile è dato dal mostrare la buona definizione dell'operatore di ricorsione, ma ciò segue dalle considerazioni fatte nel precedente paragrafo.

Il modello è inoltre corretto e completio rispetto alla semantica operazionale ovvero vale il seguente

Teorema 3.2. $P \sim Q \iff [\![P]\!] = [\![Q]\!]$

La dimostrazione è data in [Win82].

Riferimenti bibliografici

[NPW81] M. Nielsen, G. Plotkin, and G. Winskel. Event structures and domains 1. *Theoretical Computer Science*, 13:85–108, 1981.

[Win80] G. Winskel. *Events in computation.* PhD thesis, Dept. of Computer Science, University of Edinburgh, 1980.

[Win82] G. Winskel. Event structure semantics for ccs and related languages. In *Proceedings of 9th ICALP*, volume 140 of *LNCS*, pages 561–576. Springer, 1982.

[Win87] G. Winskel. Event structures. *Advances in Petri Nets 1986, Part II*, volume 140 of LNCS:561–576, 1987.

[WN95] G. Winskel and M. Nielsen. *Handbook of Logic in Computer Science*, volume 4, chapter Models for concurrency. Clarendon Press, 1995.