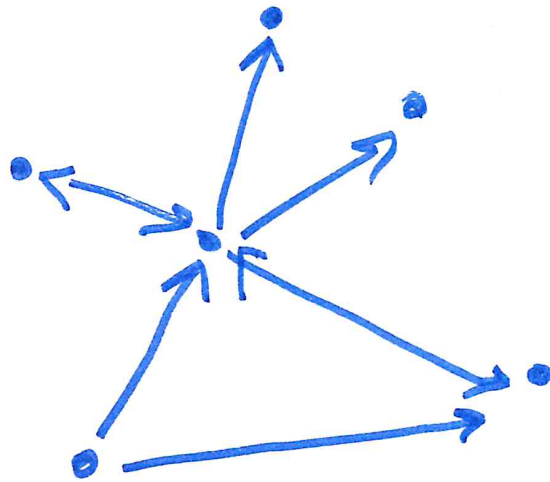


AMBIENTE DI CALCOLO DISTRIBUITO



\rightarrow
 G

• = entità

\rightarrow = link
(connessioni)

Ogni entità possiede:

- \Rightarrow — memoria locale
- capacità di calcolo
- capacità di comunicazione
- \Rightarrow — clock locale

Esempi di entità:

processori, processo, sensore, switch, ...

- Nella memoria locale:

- registri di input

$\text{valore}(x) = \text{input dell'entità } x$

- registro di stato

$\text{stato}(x) = \text{stato dell'entità } x$

↳ cambiato localmente dalla stessa entità x

- Per il clock locale è possibile :
senzare o resettare una sveglia

Proprietà delle entità:

- ① Entità sono reattive:
all'accadere di un "evento" compiono una "azione"

Eventi : - interni al sistema :

- a) ricezione di messaggi
- b) sveglia

- esterni al sistema :

impulso spontaneo ← START

Azione: sequenza finita di operazioni indivisibile

esempio di azione: nil \leftarrow azione nulla

② Le entità seguono delle regole.

Def una regola è un oggetto della forma

stato \times evento \rightarrow azione

Def Sia x una entità. $B(x)$ = l'insieme delle regole a cui è soggetta x

$B(x)$ deve essere: completo e non ambiguo

$\underbrace{\hspace{10em}}$
è praticamente il codice di x

Se E = l'insieme delle entità che cooperano Tra loro

$$B(E) = \bigcup_{x \in E} B(x) = \text{comportamento del sistema}$$

ed è importante che sia omogeneo:

$$\forall x, y \in E \text{ si ha } B(x) = B(y)$$

$B(E)$ omogeneo = protocollo per E
= algoritmo distribuito per E

Fatto: È sempre possibile ottenere $B(E)$
OMOGENEO

Proof: Idea è di utilizzare un registro locale aggiuntivo che differenzia quelle entità che alla stessa coppia (stato, evento) hanno azioni diverse:

$rulo(x)$ = registro locale di x che contiene il ruolo per x

la regola viene modificata in:

stato x evento \Rightarrow If $rulo(x) = a$ then A_a
else A_b

Proprietà della rete :

- ① La comunicazione avviene usando una etichettatura sui link

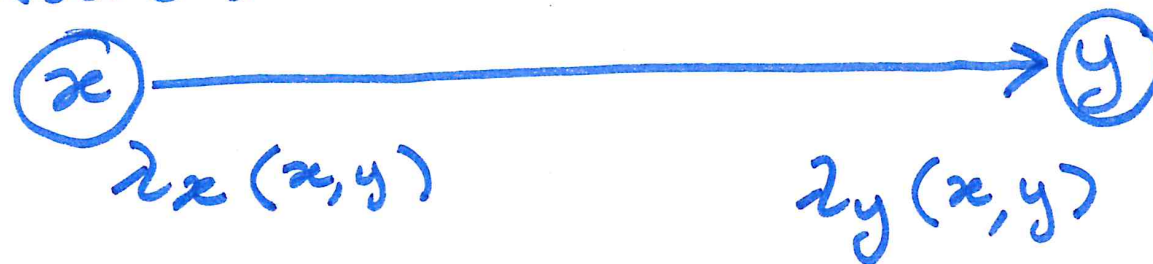
per l'entità x , l'etichettatura è denotata con λ_x

dato che x si trova in \vec{G} si indicano con:

$N_{in}(x)$ = vicini di ingresso ad x

$N_{out}(x)$ = " " uscita di x

$y \in N_{out}(x)$ $x \in N_{in}(y)$



② Assiomi della rete :

- Ritardo finito di comunicazione
in assenza di errori, un msg spedito
prima o poi arriverà
- Orientamento locale
ogni entità riesce a distinguere tra i
suoi vicini $N_{in}(x)$ e $N_{out}(x)$ grazie
alla conoscenza della funzione λx

Parametri di rete :

numero di entità = n

numero di link = m

diametro della rete = d

Oltre agli assiomi possiamo avere delle RESTRIZIONI sulla rete. Da dichiarare al momento della scrittura del codice - In genere sono proprietà positive della rete su cui facciamo affidamento.

Restriomi sulla comunicazione:

- Link Bidirezionali:
connessioni tra entità di tipo full-duplex

$\overrightarrow{G} \rightsquigarrow G$ graf non diretto

Def: $\forall x \quad N_{in}(x) = N_{out}(x) \quad (= N(x))$

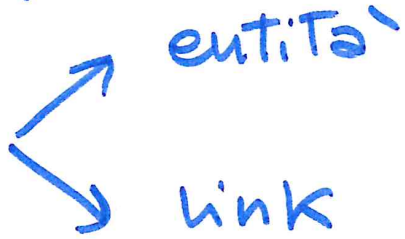
\wedge

$$\lambda_x(x, y) = \lambda_x(y, x)$$

- ordinamento dei msg:

i messaggi sullo stesso link vengono prelevati
con la politica FIFO

Restrizioni sull'affidabilità:

- rilevazione di errori 

- affidabilità parziale
non ci saranno errori in futuro

- affidabilità Totale
non ci sono stati errori e non ce ne saranno
in futuro

Restrizioni sulla Topologia di rete:

- connettività del Grafo:
 - \vec{G} è fortemente connesso
 - G è connesso

Restrizioni sul Tempo:

- Tempi di comunicazione unitari
- clock sincronizzati

NOTA: Tali restrizioni a volte vengono considerate per il calcolo delle prestazioni IDEALI del codice distribuito

Misure di complessità:

* - Tempo:
l'intervallo tra la prima entità che si attiva
e l'ultima che termina

- quantità di comunicazione:

→ n^2 di msg spediti
(se i msg sono omogenei)

→ n^2 di bit spediti

* Esecuzioni \neq dello stesso codice distribuito
pos. portare 2 Tempi diversi



Tempo ideale

Tempo misurato considerando:

- comunicazioni unitarie
- clock sincroni

Tempo causale (caso peggiore)

Tempo misurato considerando la catena più lunga di comunicazione richiesta dal codice

Definizione di un problema:

$$P = \langle P_{init}, P_{final}, \underline{R} \rangle$$

↙ ↘
predicati che descrivono
le configurazioni del
sistema all'inizio e
alla fine

↘
restrizioni del
sistema:
es: Full-duplex,
no-errori,
connettività...

Esempio di P : Broadcasting

P_{init} : una entità detiene I (info)

$\exists x \in E$ t.c. $valore(x) = I$ \wedge

$\forall x \neq y$ $valore(y) = \emptyset$

P_{final} : Tutte le entità posseggono I

$\forall x \in E$ $valore(x) = I$

R =

link bidirezionali
affidabilità totale
connettività
unico iniziatore

BL	}	R
TR		+
CN		
UI	}	I
		= RI