

# Programación Distribuida y Tiempo Real

## Estados Globales

### 1. Estado/s global/es

1.1. Definición: procesos + mensajes

1.2. Utilidad: garbage collection (compactación), interbloqueo, terminación

1.2.1. Cou-11, p. 9

1.3. Evaluación formal (Coulouris, p. 383...):

historia( $p_i$ ) =  $h_i = \langle e_i^0, e_i^1, e_i^2, \dots \rangle$ , ( $p_i$ , proceso  $i$ ,  $i=1, \dots, N$ )

$h_i^k = \langle e_i^0, e_i^1, \dots, e_i^k \rangle$  (prefijo,  $k$  eventos)

$C = h_1^{c^1} \cup h_2^{c^2} \cup \dots \cup h_N^{c^N}$  (*corte* consistente o inconsistente)

Consistente:  $\forall e \in C, f \rightarrow e \Rightarrow f \in C$ , ( $f \rightarrow e$  "antes de" - Lamport)

Estado Global Consistente *corresponde a un* Corte Consistente

Ejecución:  $S_0 \rightarrow S_1 \rightarrow S_2 \dots$  ( $\rightarrow$  un evento: msg o interno)

¿Qué pasaría con los eventos concurrentes?

Dos eventos simultáneos/concurrentes: ordenados según pid, por ej.

Ejecución: *orden total*

Predicado de Estado Global: *propiedad* de estado V o F

Predicados estables y transitorios o no estables

1.4. Distributed Snapshot (Chandy-Lamport): ¡Mensajes!

Estado que podría haber sido

Consistente (global)

Requerimientos: TCP, conectados de a dos, cualquiera-continúa

Def. Cou 11, Ej. Tan 5

Nota 1: la 1ra. vez se inicializan todos los de entrada

Nota 2: ¿Termina?

Nota 3: Marcadores con *tags* de procesos que inician el alg.

1.5. Relación con reloj global (o falta de)

1.5.1. Todos los eventos con marca de tiempo + recolección

1.5.2. Algoritmo: solamente en los momentos en que se necesita