

Proyecto: Procesamiento de secuencias de eventos

Dado un conjunto de secuencias de eventos o traza) $\lambda = \{\sigma_i\}$, donde cada secuencia está formada por símbolos de un alfabeto de eventos Σ ($\sigma_i \in \Sigma^*$), obtener las siguientes estructuras y su representación gráfica (diagramas).

- Una relación de precedencia $BEF(\lambda) \subseteq (\Sigma \times \Sigma) \times \{\sigma_i\}$ (before), consistente en parejas de eventos de Σ y el nombre de una traza σ_i , tales que $((a, b), \sigma_1)$ denota a BEF b , (“ a fue observado antes que b en σ_1 de λ ”); también se escribe (a, b, σ_1) o $a < b$ (adicionalmente, se puede especificar “en σ_1 ”).

Para los primeros eventos a de una secuencia, se denotará la precedencia $(-, a)$; de igual forma, para los últimos eventos w de una secuencia se anotan $(w, -)$.

Por ejemplo, para un conjunto de trazas $\lambda = \{\sigma_1, \sigma_2, \sigma_3\}$ sobre $\Sigma = \{A, B, C, D, E, F, G\}$ con las secuencias $\sigma_1 = ABCBCD$, $\sigma_2 = AEF GD$ y $\sigma_3 = AFEGD$ se generará:

$BEF(\lambda) = \{((-, A), \sigma_1), ((A, B), \sigma_1), ((B, C), \sigma_1), ((C, B), \sigma_1), ((C, D), \sigma_1), ((D, -), \sigma_1),$
 $((-, A), \sigma_2), ((A, E), \sigma_2), ((E, F), \sigma_2), ((F, G), \sigma_2), ((G, D), \sigma_2), ((D, -), \sigma_2),$
 $((-, A), \sigma_3), ((A, F), \sigma_3), ((F, E), \sigma_3), ((E, G), \sigma_3), ((G, D), \sigma_3), ((D, -), \sigma_3)\}.$

- Un diagrama (grafo dirigido $G_{<}$) de precedencia de $BEF(\lambda)$ -todas las secuencias σ_i -, con $|\Sigma|$ vértices. Para λ del ejemplo, el grafo es el mostrado en la Figura 1.
- Dado $BEF(\lambda)$, obtener la estructura $PO(\lambda) = BEF(\lambda) \setminus \text{Conc}$ (PO: partial order); donde $\text{Conc} = \{((a, b), \sigma_i), ((b, a), \sigma_j) \in BEF(\lambda) \mid \sigma_i \neq \sigma_j\}$ (a y b se han visto en diferente orden en diferentes trazas) y obtener su diagrama de precedencia.

En el ejemplo, el diagrama de $PO(\lambda)$ es el mostrado en la Figura 2; esto pretende ser una representación de un orden parcial; los arcos entre E y F desaparecen por ser eventos concurrentes. Sin embargo, existe un ciclo entre B y C ; esto se soluciona “desdoblando” el evento B como se muestra en la Figura 3.

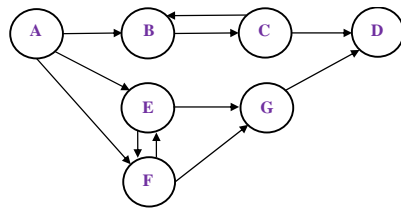


Figura 1

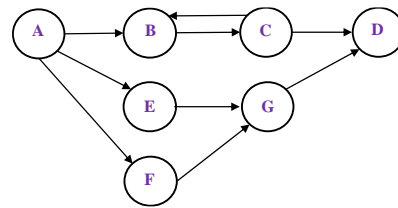


Figura 2

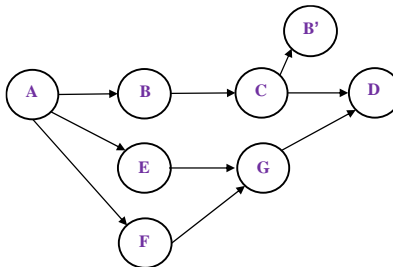


Figura 3

- A partir de λ , o de $BEF(\lambda)$, o de su diagrama $G_{<}$, obtenga un autómata finito no determinista que acepta las trazas de λ , donde $(-, a)$ es el primer evento a partir de un estado inicial; de manera similar $(y, -)$ será representado con el evento y que conduce al estado final. En este autómata solo se incluyen las transiciones que conducen a un estado de aceptación; cualquier símbolo no incluido en las transiciones de salida de un estado hace que el autómata pare (estado pozo en autómatas deterministas). Para λ , el autómata trivial se muestra en la Figura 4, el cual representa explícitamente cada traza de λ como un camino. Un autómata no determinista más compacto se obtiene realizando agrupamiento de estados por prefijos y sufijos comunes (Figura 5); esto se obtiene directamente desde $G_{<}$.

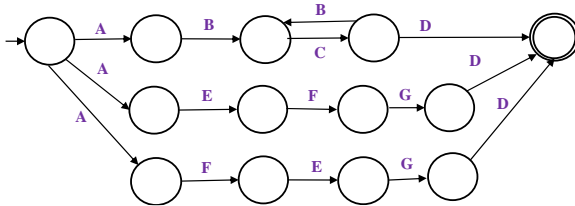


Figura 4

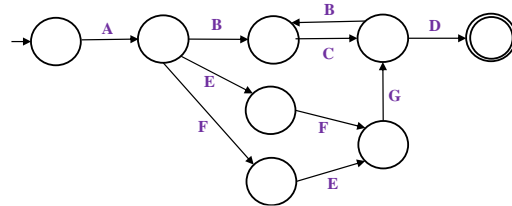


Figura 5

- Considere que los datos de entrada (λ) se dan en un archivo de texto donde cada evento es una cadena de símbolos, los cuales se separan por un espacio en la secuencia. Para el registro λ del ejemplo, puede ser:
NTR: 3 - dato opcional-
TR1: A B C B C D
TR2: A E F G D
TR3: A F E G D

De igual forma, “TRi” puede ser omitido.