* 1. Definiendo modelos DEVS abstractos con

CML-DEVS **17**

atomic *cola* is *<* X*,* Y*,* S*, δ*int*, δ*ext*, λ,* ta *>* where X is

*in* : R *∪ {emitir}*;

end X

Y is

*out* : R *∪ {∅}*;

end Y

S is

*s* : List R *×* Time;

end S

*δ*int((*xs, σ*)) is

defcases

case *s* = (tail *xs, ∞*); if (*xs ƒ*= *{}*)

case *s* = (*xs, ∞*); if (*xs* = *{}*)

end defcases

end *δ*int

*δ*ext((*xs, σ*)*, e,* (*in, x* )) is defcases

case *s* = (*xs* ^ *x, ∞*); if (*x ∈* R)

case *s* = (*xs,* 0); if (*x* = *emitir* )

end defcases end *δ*ext

*λ*((*xs, σ*)) is

defcases

case *out* = head *xs*; if (*xs ƒ*= *{}*)

case *out* = *∅*; if (*x* = *{}*)

end defcases

end *λ*

ta((*xs, σ*)) is

*σ*;

end ta

end atomic

**Figura 2.2:** Modelo de la Cola de Procesamiento en CML-DEVS

* + 1. **Estructura de un Modelo At´omico**

Un modelo DEVS at´omico en CML-DEVS consiste en varias subestructuras o com- ponentes, cada una de ellas representando un componente de un modelo DEVS ato´mico. El orden en el que se declaran estas estructuras es indistinto, y tampoco es necesario u obligatorio que se declaren en caso de que no sean usadas. Es decir, si un modelo no recibe eventos, puede omitirse la declaracio´n de X y de *δ*ext. Si estas no se definen, tampoco deben declararse en el vector que define el modelo: *<* X*,* Y*,* S*, δ*int*, δ*ext*, λ,* ta *>*. Es decir, todas las componentes que all´ı se declaren, deben luego estar definidas.

Adem´as de las estructuras que se pueden observar en la Figura 2.2, del ejemplo an- terior, existen dos ma´s que son opcionales. Las mismas no forman parte del formalismo DEVS original pero son utilizadas en casi todas las implementaciones de modelos de simulacio´n. Estas permiten la definicio´n de *par´ametros* o *valores constantes* del modelo y *funciones auxiliares*, que simplifican la descripci´on y mantenimiento de los modelos. Como en todo lenguaje, existen palabras claves o reservadas, en el ejemplo de la