

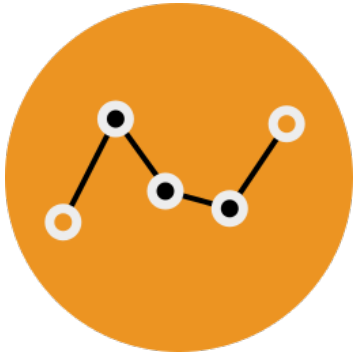
Traducción y Compilación

Paradigmas de Programación

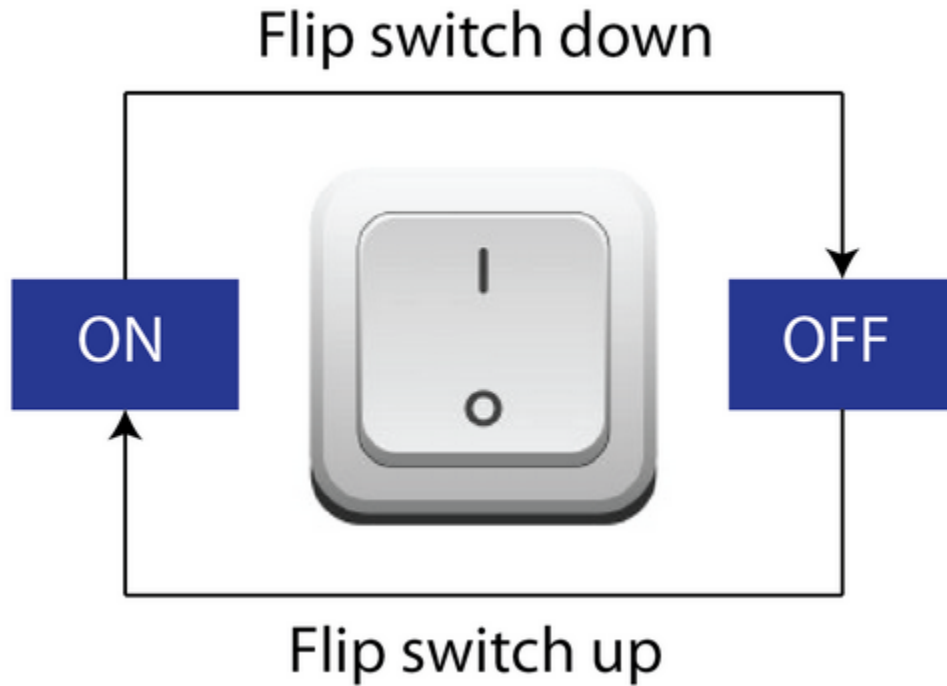


MPM. Maikol Guzmán Alán

Autómata



Modelo matemático para una máquina de estados finita, la cual es una máquina que dada una entrada de símbolos, "salta" a través de una serie de estados de acuerdo a una función de transición.



Autómatas de estado Finito (FSA)

Autómata de estados finitos (FSA)

- Máquina con un número **finito** de estados que lee una entrada **infinita**.
- Se utilizan para reconocer lenguajes de programación.



FSA tipos

- **Deterministas**, a cada estado y acción le corresponde uno y exactamente un solo estado siguiente (dado un estado y una acción, el estado siguiente está siempre bien determinado)
- **No Deterministas**, pueden existir varias transiciones a estados o varias acciones hacia diferentes estados siguientes.

Ambos términos son abstractos y se usan en forma teórica (no práctica)

Definición funcional

FSA son programas de computación que consisten en:

- **Eventos** a lo que el programa responde
- **Estados** en donde el programa espera entre eventos
- **Transiciones** entre estados en respuesta a eventos
- **Acciones** que ocurren durante las transiciones
- **Variables** que mantienen los valores que se necesitan las acciones entre eventos

Descripción funcional



FSA son utilizados en situaciones en donde el comportamiento es ocasionado por diferentes tipos de **eventos**. El **evento** puede ser externo originado desde el teclado, mouse, red, etc. Interno desde otras aplicaciones. El **estado** es la forma de recordar los **eventos** anteriores, las **transiciones** organizan los **eventos** futuros. Un **estado** puede ser inicial y quizás exista un **estado** final.

Representación

Grafos dirigidos: los nodos representan estados, las flechas las transiciones y las aristas los eventos y las acciones.

Tablas de dos dimensiones: filas y columnas representan eventos y estados, las celdas contienen acciones y transiciones

Grafo Dirigido

Ejemplo tooltip en HTML

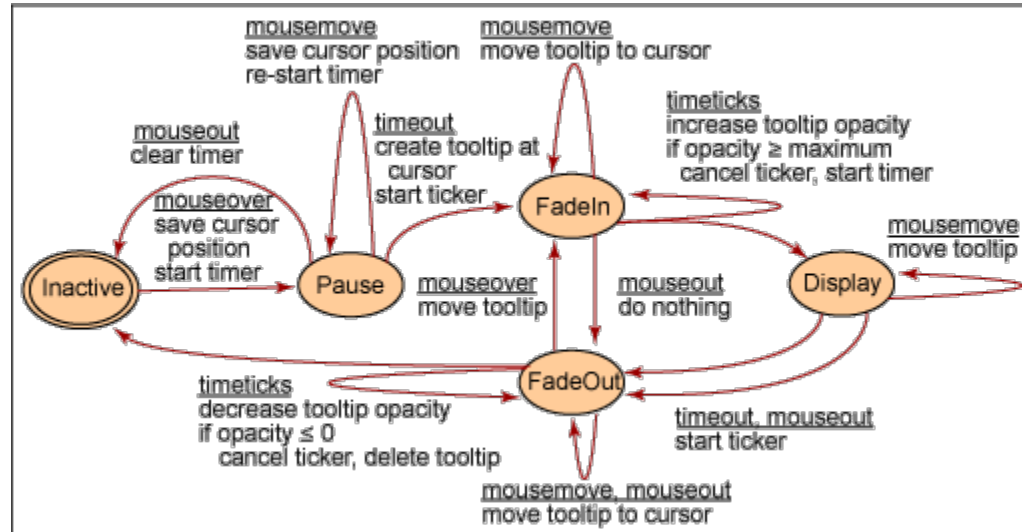


Tabla de dos dimensiones

Ejemplo tooltip en HTML

| states ► events ▼ | Inactive | Pause | FadeIn | Display | FadeOut |
|----------------------|---|--|--|---------------------------------------|--|
| mouseover | cancel timer save cursor position start time next state is Pause | | | | move tooltip to cursor next state is FadeIn |
| mousemove | | do [Inactive, mouseover] | move tooltip to cursor | do [FadeIn, mousemove] | do [FadeIn, mousemove] |
| mouseout | | cancel timer next state is Inactive | next state is FadeOut | do [Display, timeout] | do nothing |
| timeout | | create tooltip at cursor start ticker next state is FadeIn | | start ticker next state is FadeOut | |
| timetick | | | Increase opacity If opacity \geq maximum cancel ticker start timer next state is Display | | Decrease opacity If opacity \leq 0 cancel ticker delete tooltip next state is Inactive |



GitHub

Instrucciones

1. Clonar el repositorio

<https://github.com/mikeavantica/UNA-EIF400-Paradigmas.git>

2. Checkout los diferentes ejemplos



Demo

Ejemplo tooltip en HTML

```
git checkout -f 01_automatas
```

Bibliografía

- Chapters 21 "Protocol Representation with Finite State Models" by Andre A. S. Danthine, and 25 "Executable Representation and Validation of SNA" by Gary D. Schultz, et. al. in Computer Network Architectures and Protocols (edited by Paul E. Green, Jr., Plenum Press, 1982): Read historic examples of finite state machines applied to computer network protocols.
- Chapter 3.5 "Finite Automata" in Compilers: Principles, Techniques, ad Tools (Alfred V. Aho et. al., Addison-Welsley, 1986): Read a description of how finite state machines are applied to computer language compilers.