



Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación

# **Clase 6**

# **Cobertura de grafos aplicada**

## **IIC3745 – Testing**

Rodrigo Saffie

rasaffie@uc.cl

2 de septiembre de 2019

## 1. Clase pasada

- Cobertura aplicada en grafos
  - Elementos del diseño
  - Especificación del diseño: restricciones de secuencia

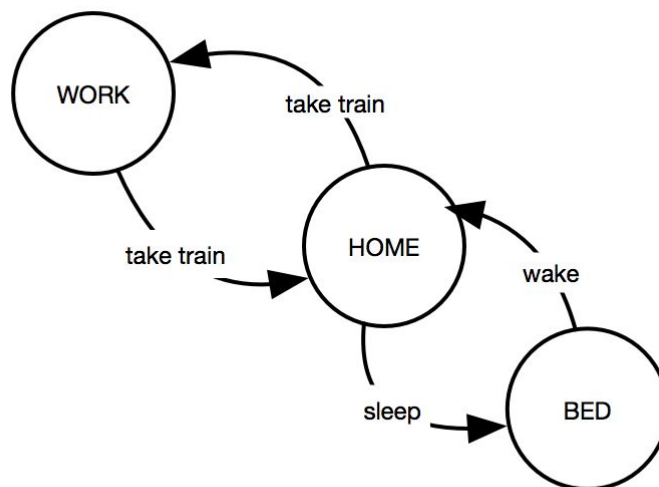
## 2. Cobertura aplicada en grafos

- Especificación del diseño: estado
- Casos de Uso

## 3. Herramientas proyecto

# Comportamiento según estado

- Utilizaremos una máquina de estados finitos (*FSM*) como descriptor del estado de las variables de un *software* y sus modificaciones en ejecución.
  - Nodos: Estados representados como un conjunto de valores de variables claves.
  - Aristas: Transiciones, posibles cambios de estado



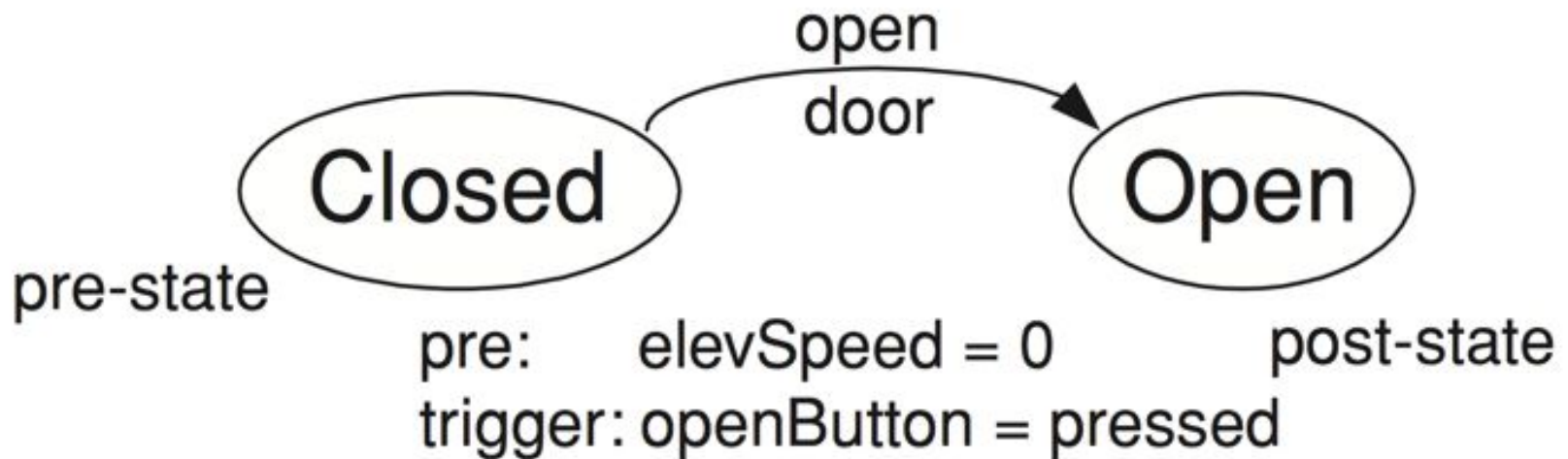
# Comportamiento según estado

- Las máquinas de estado pueden modelar distintos tipos de *software*:
  - *Software* embebidos
  - Estructuras de datos
  - Compiladores y sistemas operativos
- Diseñar *software* en base a máquinas de estado permite capturar posibles errores
- Es impracticable en *software* con múltiples estados.

# Anotaciones en *FSM*

- Una *FSM* puede ser anotada con distintas acciones:
  - Transiciones
  - Entrada de nodos
  - Salidas de nodos
- Una acción puede expresar cambios en variables o condiciones en las mismas
- Consideraremos las siguientes anotaciones:
  - Precondiciones: condiciones que deben ser verdaderas para que se realice una transición.
  - Gatilladores: Cambios en variables que causan transiciones.

# Ejemplo Ascensor



# Cobertura en *FSM*

- Estructural
  - ***State coverage***: ejecutar cada estado
  - ***Transition coverage***: ejecutar cada transición
  - ***Transition-pair***: ejecutar cada par de transiciones
- Flujo de información:
  - Nodos a menudo no incluyen ***defs*** ni ***uses***
  - Usualmente sólo modelan un subconjunto de variables.
  - Se han hecho pocos intentos de aplicar criterios de flujo de información en *FSM*.

# Derivar diagrama *FSM*

- En general es más difícil generar un *FSM* que cubrirlo.
- Se puede generar un *FSM* en la etapa de diseño, pero es deber del *tester* revisar que se encuentre al día respecto con la implementación.
- También se suelen utilizar estrategias (algunas incorrectas) para derivar *FSM* del software:
  - a. Combinar grafos de control de flujo
  - b. Basarse en la estructura del *software*
  - c. Modelar variables de estado
  - d. Utilizar especificaciones explícitas o implícitas



# Cobertura con *FSM*

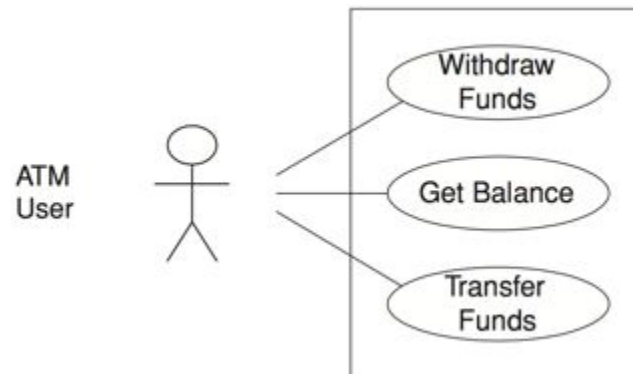
- Ventajas
  - a. Las pruebas pueden ser diseñadas antes de la implementación
  - b. Analizar un *FSM* es más simple que analizar el código
- Desventajas
  - a. Algunas decisiones de implementación no pueden ser modeladas como *FSM*
  - b. Hay variaciones debido a la naturaleza subjetiva de derivar *FSM*
  - c. Las pruebas deben ser mapeadas a *inputs* reales del programa

# Cobertura de grafos aplicada

- Código fuente
- Elementos de diseño
- Especificación de diseño
- Casos de uso

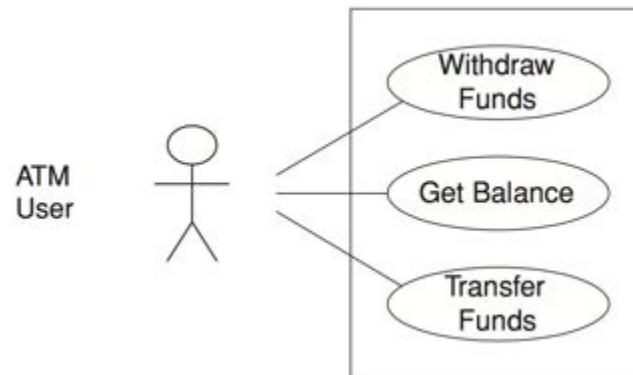
# Casos de Uso

- Son utilizados generalmente para expresar requerimientos de *software*.
- Ayudan a expresar el flujo de la aplicación.



# Casos de Uso

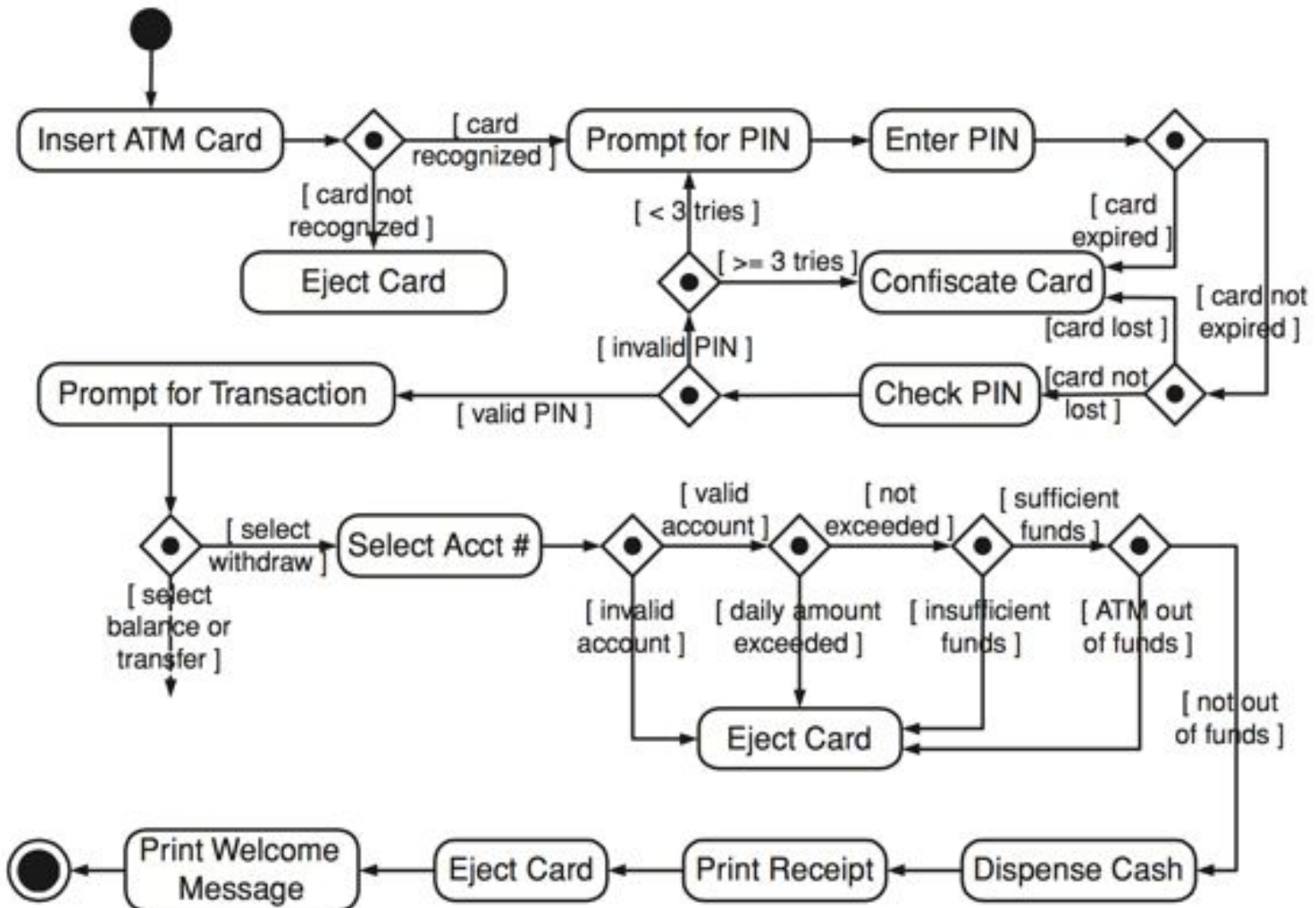
- *Node Coverage*: cubrir cada caso de uso
- No es muy útil analizar estos diagramas



# Diagramas de actividad (o flujo)

- Indican el flujo entre las actividades
- Actividades deben modelar pasos a nivel del usuario
- 2 tipos de nodos:
  - Estados de acción
  - Ramificaciones
- En general estos diagramas tienen características deseables:
  - Pocos *loops*
  - Predicados simples

# Giro en cajero automático



# Cobertura en diagramas de actividad

- Flujos de datos no aplica
- Estructural:
  - Escenario: un camino completo a través del diagrama
    - Debe tener sentido semántico para el usuario
    - Número de caminos es finito a menudo
  - *Node Coverage*
  - *Edge Coverage*
  - Cobertura de camino específico (SPC)

## 1. Clase pasada

- Cobertura aplicada en grafos
  - Elementos del diseño
  - Especificación del diseño: restricciones de secuencia

## 2. Cobertura aplicada en grafos

- Especificación del diseño: estado
- Casos de Uso

## 3. Herramientas proyecto





Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación

# **Clase 6**

# **Cobertura de grafos aplicada**

## **IIC3745 – Testing**

Rodrigo Saffie

rasaffie@uc.cl

2 de septiembre de 2019