Modelos de Pruebas

Luis Daniel Benavides Navarro, Ph.D.

La clase pasada

Cuatro conceptos clave

- Dynamic Lookup
- Abstraction
- Subtyping
- Inheritance

- Nota: No todos los lenguajes a objetos usan clases, e.g., Self (Prototype based languages)

Patrones de diseño

- Solución general a un problema derivada de la repetición (Heurística)
- Un buen patrón hace lo siguiente:
  - Soluciona un problema
  - Es un concepto probado (heurística)
  - La solución no es obvia
  - Describe una relación
  - Tiene un significado para los humanos

Modelos de falla para OO

- Algunas características esenciales presentan peligros potenciales de faltas
  - Dynamic binding
  - Iterface pollution
  - Estado almacenado en Objetos pero el flujo de control esta distribuido sobre todo el programa

Efectos de Borde del paradigma

- Que puede salir mal
- Encapsulación
- Herencia
- Polimorfismo
- Secuencias de mensajes y errores relacionados al estado
- Conclusión

La clase de hoy

- Resumen de la clase anterior (10 min)
- Comprobación de lectura (20 min)
- Modelos de pruebas para OO (30 min)
- Taller para entregar (90) min

Modelos de pruebas

¿Por qué usar modelos?

- No se puede probar sin saber que debe hacer la implementación bajo pruebas
- La complejidad del software requieres modelos para soportar pruebas
- Un Ing. de pruebas debe aumentar y revisar los modelos disponibles
- Muy frecuentemente se necesitan modelos de pruebas adicionales

Pruebas basadas en modelos

- Pruebas es un proceso de búsqueda de bugs (sobre una infinidad de posibles casos)
- La búsqueda debe ser sistemática, enfocada y automatizada
- Modelos
  - Soportan la enumeración sistemática de entradas y salidas
  - Se puede enfocar en los casos que pueden generar errores (generalmente un educated guess)
  - Se puede usar como entrada para la automatización

El rol de los modelos en las pruebas

- *Validación de modelos:* busca establecer la equivalencia entre un modelo y los requerimientos
- *Verificación:* busca verificar que la implementación es correcta con respecto a su representación
- *Evaluación de consistencia:* verificar la validez de un modelo contra el meta-modelo
- *Pruebas basada en responsabilidad:* Evalúa el comportamiento observable contra la representación (modelo)
- *Pruebas basadas en implementación:* Evalúa el comportamiento observable contra la representación derivada de la implementación (modelo)
- *Validación de producto:* Evalúa el comportamiento observable contra la requerimientos

Caricatura vs modelos

- La mayoría de especificaciones hechas en el procesos de desarrollo de software son caricaturas
- Los modelos de pruebas deben ser completos y no-ambiguos
- El ingeniero de prueba debe completarlos y aumentarlos
- Ejemplo: Herramientas de UML permiten diagramas incompletos y ambiguos

Modelos combinatorios

Cuando son aplicables

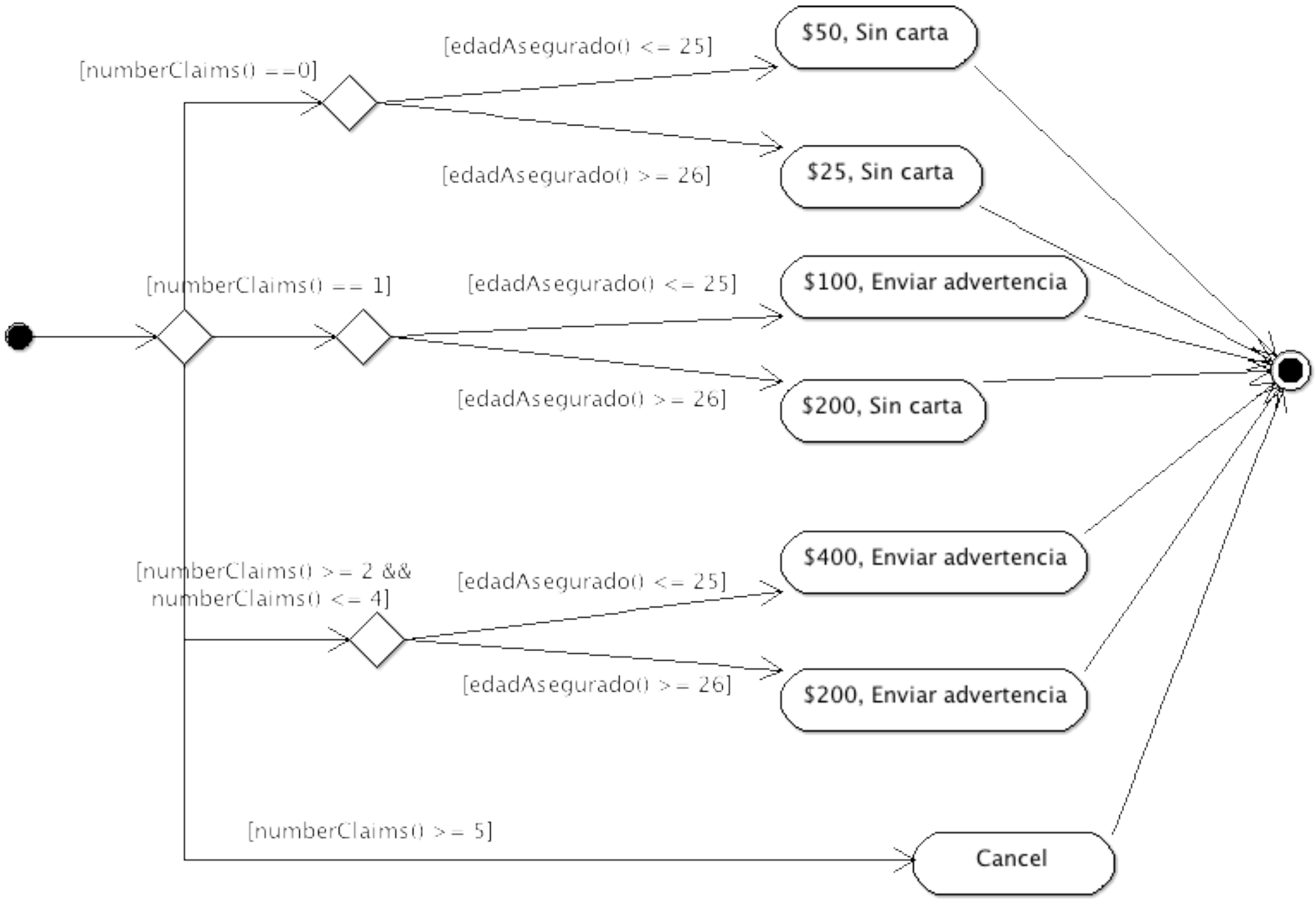
- Cuando la salida depende de la combinación de las entradas
- La salida no depende del estado anterior

Como soportan los modelos combinatorios las pruebas

- Usa tablas decisión para representar la implementación bajo test (IUT)
- Ejemplo:

	Sección de condiciones		Sección de acciones		
Variante	Numero de reclamos	Edad del asegurado	Incremento de la prima ($)	Enviar advertencia	Cancelar
1	0	25 o menos	50	No	No
2	0	26 o más	25	No	No
3	1	25 o menos	100	Si	No
4	1	26 o más	50	No	No
5	2 a 4	25 o menos	400	Si	No
6	2 a 4	26 o más	200	Si	No
7	5 o más	Cualquiera	0	No	Si

Representación de una tabla de decisión como diagrama de actividad de UML



Máquinas de Estados

¿Qué es una máquina de estado?

- Un sistema cuya salida depende de las entradas pasada y actuales
- El efecto de entradas pasadas esta representado en el estado
- Son una aplicación de ingeniería de los modelos matemáticos de autómatas finitos
- Un máquina de estado esta construida con:
  - Estados, transiciones, Eventos, Acciones

Ejemplo de máquina de Estado

- Un estado inicial
- Recibe eventos
- Si no es aceptado en el estado actual el evento es ignorado
- Si el evento es aceptado, la transición correspondiente y la acción son ejecutadas
- El ciclo se repite

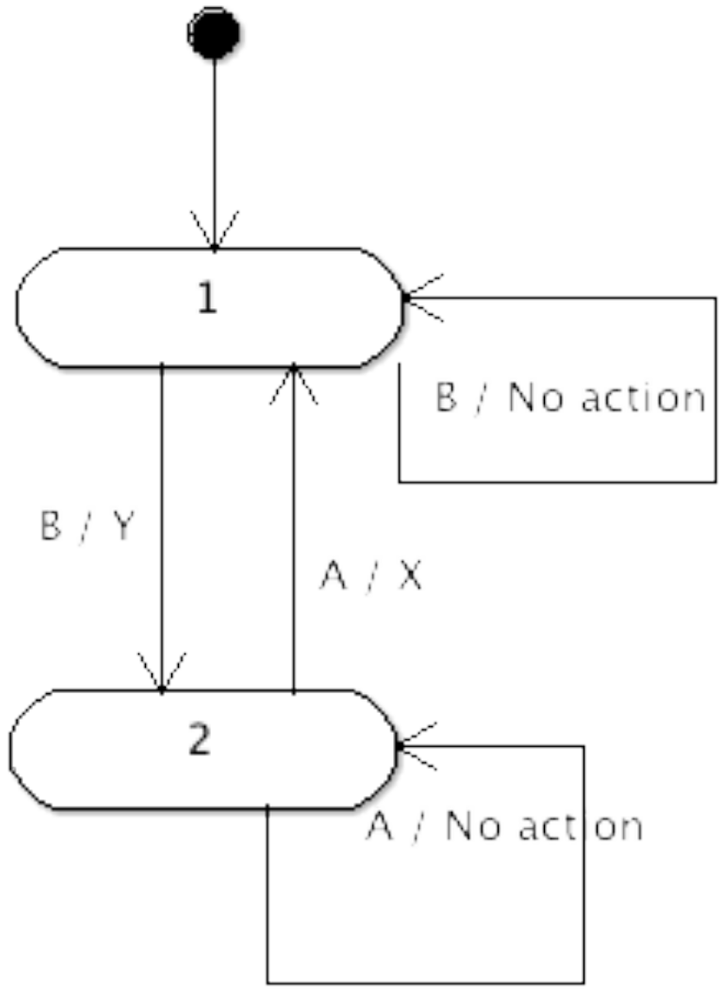


Diagrama de transición de estados

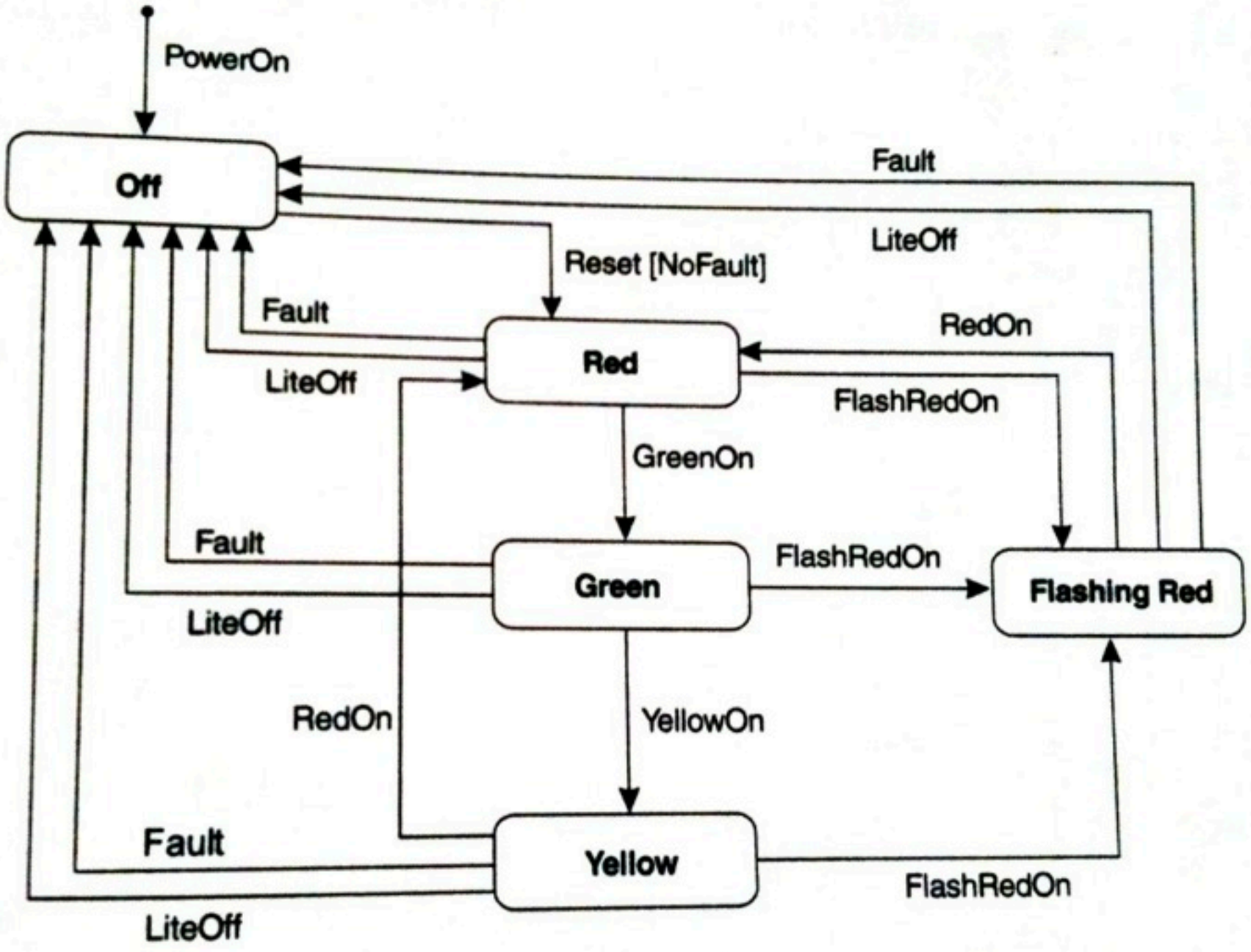
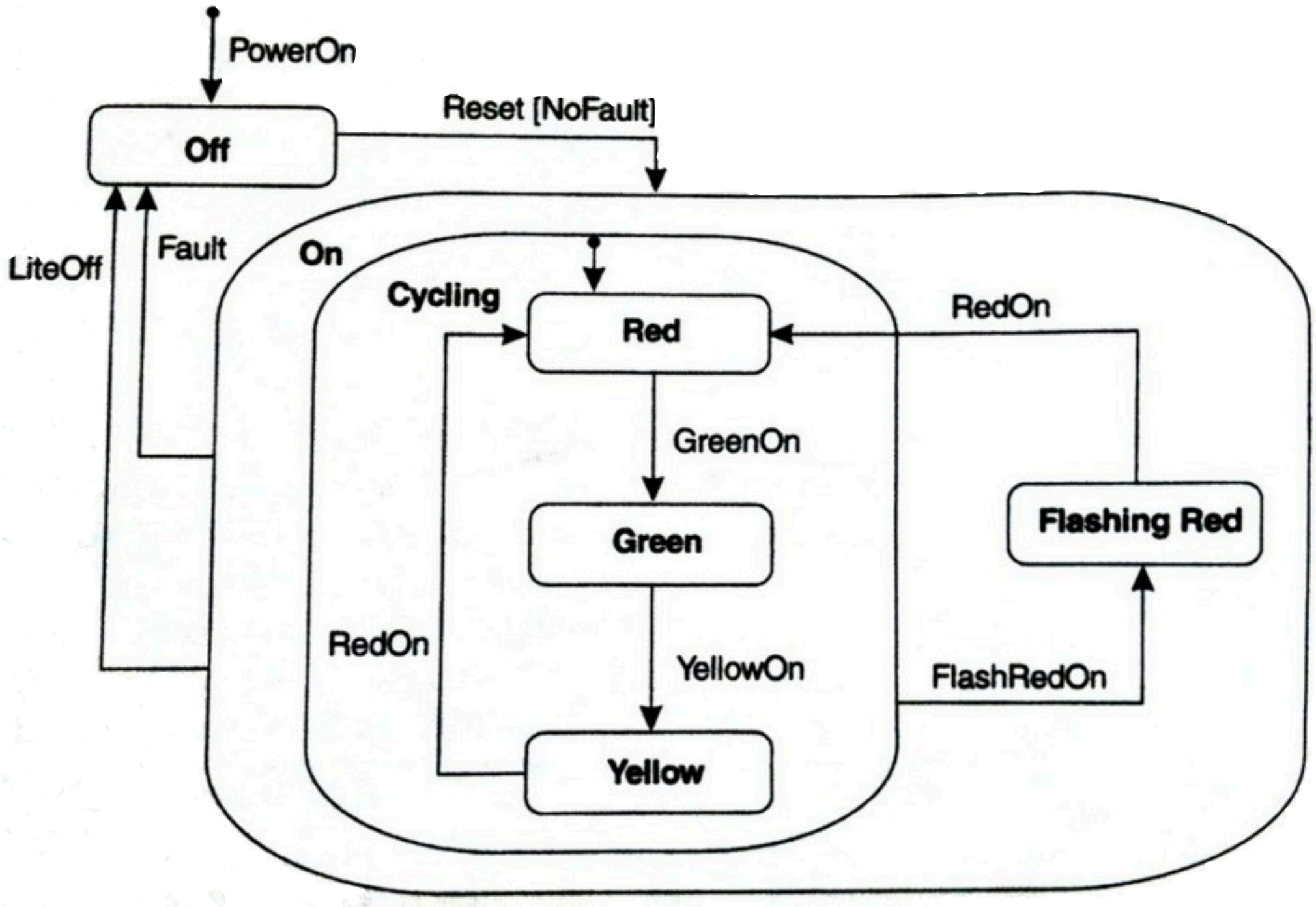
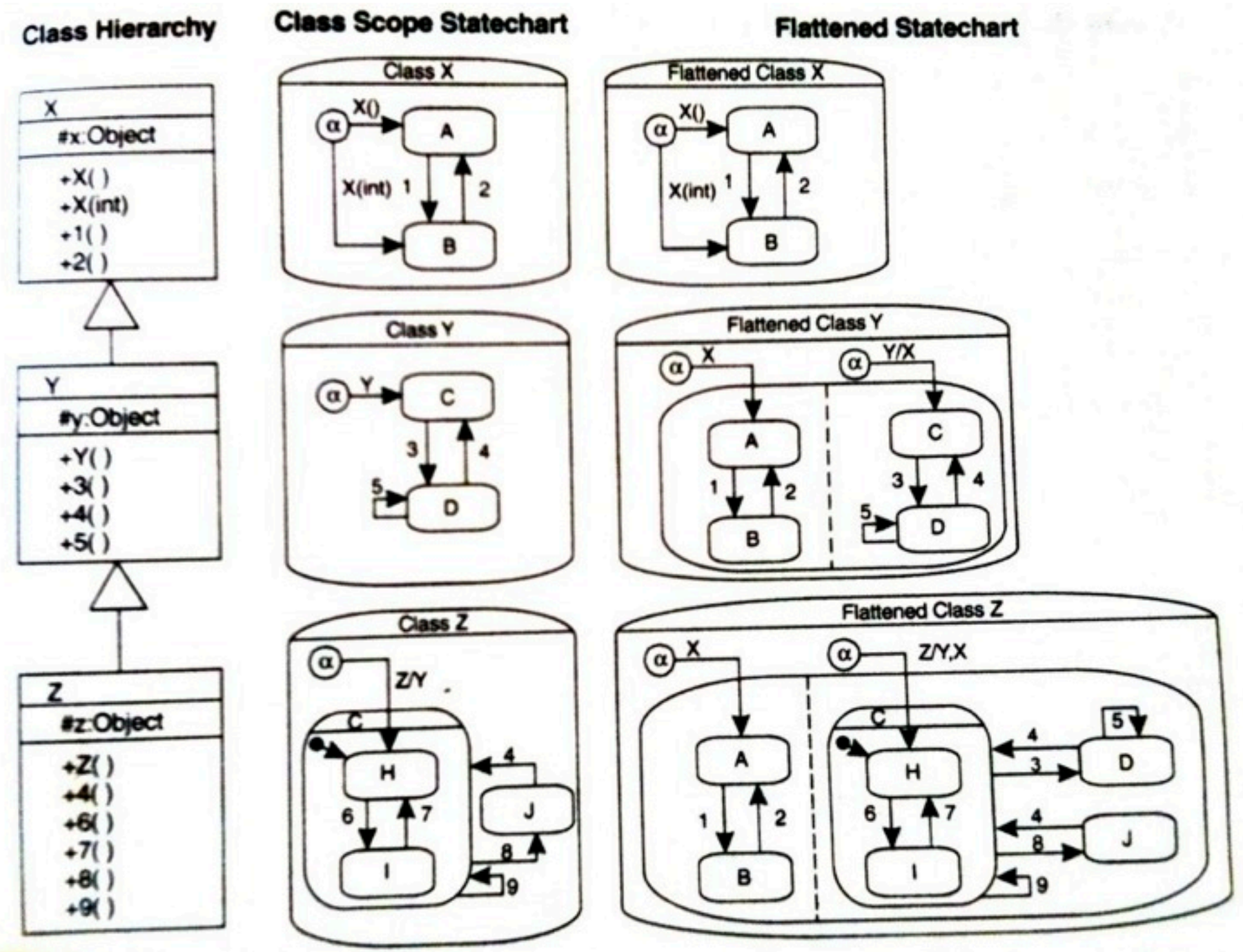


Gráfico de estados



Herencia y aplanamiento de clases



UML para pruebas

UML como un modelo de pruebas

- UML notación para expresar modelos Orientados a Objetos
- Estándar de facto para modelos OO
- La notación impone pocas restricciones.
- Permite modelos precisos e imprecisos

Estrategia de pruebas

- Interpretar los diagramas y generar los casos de prueba sobre la aplicación bajo pruebas
- Utilizar los patrones propuestos en [Binder2000]

Taller para entregar

- Realice un pequeño ensayo que documente el patrón “Class Association Test”
- Resuelva el ejercicio propuesto
- Divida el trabajo entre los miembros del grupo
- Realice una revisión de pares antes de entregar