

Paso 3. Generar todas las combinaciones de condiciones y eliminar las combinaciones de condiciones inviables. Para cada combinación factible, se crea una columna separada en la tabla con los valores de cada condición enumerada en esta columna.

Paso 4. Identificar, para cada combinación de condiciones identificada, qué acciones y cómo deberían ocurrir. Esto da como resultado completar la parte inferior de la columna correspondiente de la tabla de decisiones.

Paso 5. Para cada columna de la tabla de decisión, diseñe un caso de prueba en el que la entrada de la prueba represente una combinación de condiciones especificadas en esta columna. La prueba se supera si, tras su ejecución, el sistema realiza las acciones descritas en la parte inferior de la tabla en la columna correspondiente. Estas entradas de acción sirven como resultado esperado para el caso de prueba.

Notación y posibles entradas en la tabla de decisiones Normalmente,

los valores de condición y acción toman la forma de valores lógicos VERDADERO o FALSO. Se pueden representar de varias maneras, como los símbolos T y F o Y y N (sí/no) o 1 y 0 o como las palabras "verdadero" y "falso". Sin embargo, los valores de las condiciones y acciones pueden ser, en general, cualquier objeto, como números, rangos de números, valores de categorías, particiones de equivalencia, etc. Por ejemplo, en nuestra tabla (Tabla 4.4), los valores de la acción "descuento otorgado" son categorías que expresan diferentes tipos de descuentos: 0%, 5% y 10%. En una misma tabla, puede haber condiciones y acciones de diferentes tipos, por ejemplo, condiciones lógicas, numéricas y categóricas pueden ocurrir simultáneamente.

La tabla de decisiones que solo tiene valores booleanos (verdadero/falso) se denomina tabla de decisiones de entrada limitada. Si para cualquier condición o acción existen entradas distintas a las booleanas, dicha tabla se denomina tabla de decisiones de entradas extendidas.

Cómo determinar todas las combinaciones de condiciones Si

necesitamos determinar manualmente combinaciones de condiciones y tememos perder algunas combinaciones, podemos usar un método de árbol muy simple para determinar sistemáticamente todas las combinaciones de condiciones. Considere el siguiente ejemplo:

Ejemplo Supongamos que una tabla de decisiones tiene tres condiciones:

- Ingresos (dos valores posibles: S, pequeño; L, grande) • Edad (tres valores posibles: Y, joven; MA, mediana edad; O, viejo) • Lugar de vida (dos valores posibles: C, ciudad; V, aldea)

Para crear todas las combinaciones de valores de triples (edad, ingresos, residencia), construimos un árbol, de cuya raíz derivamos todas las posibilidades de la primera condición (ingresos). Este es el primer nivel del árbol. A continuación, de cada vértice de este nivel derivamos todas las posibilidades de la segunda condición (edad). Obtenemos el segundo nivel del árbol. Finalmente, de cada vértice de este nivel se derivan todos los valores posibles de la tercera condición (lugar de vida). Por supuesto, si hubiera más condiciones, procederíamos de manera análoga. Nuestro árbol final se parece al de la Fig. 4.8.

Cada combinación posible de condiciones es una combinación de etiquetas de vértice en los caminos que van desde la raíz (el vértice en la parte superior del árbol) hasta cualquier vértice en la parte inferior del árbol. Por lo tanto, habrá tantas combinaciones como vértices en

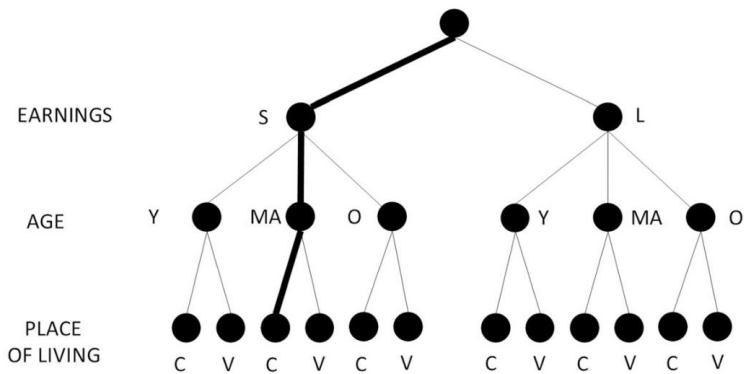


Fig. 4.8 Árbol de soporte para identificar combinaciones de condiciones

el nivel más bajo (en nuestro caso, 12; esto, por supuesto, se deriva del número de combinaciones: $2 \times 3 \times 2 = 12$). En la figura, las líneas en negrita indican el camino correspondiente a la combinación de ejemplo (S, MA, C), que significa ingresos bajos, mediana edad y ciudad como lugar de vida. Ahora podemos ingresar cada una de estas combinaciones en las columnas individuales de nuestra tabla de decisiones. Al mismo tiempo, estamos seguros de que ninguna combinación se ha quedado fuera. Las condiciones de la tabla de decisión así creada se muestran en la Tabla 4.5.

Combinaciones inviables A

veces, después de enumerar todas las combinaciones posibles de condiciones, es posible que descubra que algunas de ellas no son factibles por diversas razones. Por ejemplo, supongamos que tenemos las dos condiciones siguientes en la tabla de decisiones:

- ¿La edad del cliente es >18? (valores posibles: Sí, NO) • ¿Edad del cliente ≤18? (valores posibles: Sí, NO)

Es obvio que aunque tenemos cuatro posibles combinaciones de estas condiciones, (SI, SI), (SI, NO), (NO, SI) y (NO, NO), solo dos de ellas son factibles, (SI, NO) y (NO, SI), ya que no es posible tener más de 18 y menos de 19 años al mismo tiempo. Eso sí, en este caso podríamos sustituir estas dos condiciones por una, “edad”, y por dos valores posibles: mayor que 18 y menor o igual a 18.

A veces, la tabla de decisiones no contendrá algunas combinaciones, no por razones puramente lógicas sino por razones semánticas. Por ejemplo, si tenemos dos condiciones:

- ¿Se definió el objetivo? (Sí, NO) • ¿Se logró la meta? (Sí NO)

entonces la combinación (NO, Sí) es inviable (sin sentido), porque es imposible posible lograr una meta que no has definido previamente.

Tabla 4.5 Combinaciones de condiciones de la tabla de decisión formadas a partir del árbol de soporte

	1	2	3		4	5	6	7	8	9		10	11	12
Ganancias	S	S	S		S		S	S	L	L	L		I	
Edad	Y	Y	M	A	M	A	O	O	Y	Y	M	A	M	O
Lugar de residencia	C	V	C	V	C	V	C	V	C	V	C	V	C	V

Minimizar la tabla de decisiones

En ocasiones, algunas condiciones pueden no tener ningún efecto sobre las acciones tomadas por el sistema. Por ejemplo, si el sistema permite que sólo los clientes adultos compren un seguro, y dependiendo de si fuman o no obtienen un descuento en ese seguro, entonces Mientras el cliente sea menor de edad, el sistema no le permitirá contratar un seguro. independientemente de si el cliente fuma o no. Estos valores irrelevantes son los más a menudo marcado en la tabla de decisiones con un símbolo de guión o N/A (no aplicable). Normalmente, se utiliza un guión cuando ocurre la condición correspondiente, pero su valor es irrelevante para determinar la acción. El símbolo N/A, por otro lado, se utiliza cuando la condición no puede ocurrir. Por ejemplo, considere dos condiciones: "tipo de pago" (tarjeta o efectivo) y "¿el PIN es correcto?" (sí o no). Si el tipo de pago es efectivo, nosotros Ni siquiera puedo verificar el valor de la condición "¿Es correcto el PIN?" porque la condición no ocurre en absoluto. Entonces tenemos sólo tres combinaciones posibles de condiciones: (tarjeta, sí), (tarjeta, no) y (efectivo, N/A).

Esta minimización, o colapso, hace que la tabla de decisiones sea más compacta con menos columnas y, por lo tanto, menos casos de prueba para ejecutar. Por otra parte, para una prueba con un valor irrelevante en el caso de prueba real, tenemos que elegir algún valor específico, valor concreto para esta condición. Por lo tanto, existe el riesgo de que si se produce un defecto en algún combinación específica de valores marcados como irrelevantes en la tabla de decisión, podemos Es fácil pasarlo por alto y no probarlo.

Sin embargo, minimizar las tablas de decisión es un ejercicio de mitigación de riesgos: tal vez el La maqueta actual de la GUI no permite ciertas entradas, pero la implementación real o una API futura también podrían hacerlo.

Considere la tabla de decisiones contraída que se muestra en la Tabla 4.6. Si quisiéramos diseñar un caso de prueba concreto para la primera columna, tendríamos que decidir si el El cliente fuma o no (aunque desde el punto de vista de la especificación esto es irrelevante), porque esta información debe darse. Podemos decidir la combinación. (adulto = NO, fuma = NO), y dicha prueba pasará, pero podemos imaginar que debido debido a algún defecto en el código, el programa no funciona correctamente para la combinación (adulto = NO, fuma = SI). Esta combinación no ha sido probada por nosotros y El fallo no será detectado.

En el examen real, puede haber preguntas que involucren tablas de decisión minimizadas, pero no se requiere que el candidato sea capaz de realizar la minimización sino sólo que sea capaz de comprender, interpretar y utilizar tablas de decisión que ya están minimizadas. El Se requiere minimización en el examen de certificación de Nivel avanzado: Analista de pruebas. Por lo tanto, en este libro no presentamos un algoritmo para minimizar tablas de decisión.

Cobertura

En las pruebas de tablas de decisiones, los elementos de cobertura son las columnas individuales de la tabla, que contienen posibles combinaciones de condiciones (es decir, las llamadas columnas factibles).

Tabla 4.6 Tabla de decisiones con valores irrelevantes

CONDICIONES			
¿Adulto?	NO	SÍ	SÍ
¿Fuma?	-	SÍ	NO
COMPORTAMIENTO			
¿Subvención de seguro?	NO	SÍ	SÍ
¿Descuento de subvención?	NO	NO	SÍ

Para una tabla de decisiones determinada, una cobertura total del 100% requiere que al menos un caso de prueba correspondiente a cada columna factible sea preparada y ejecutada. Se pasa la prueba si el sistema realmente ejecuta las acciones definidas para esa columna.

Lo importante es que la cobertura cuenta contra el número de (factibles) columnas de la tabla de decisiones, no contra el número de todas las combinaciones posibles de condiciones. Por lo general, estos dos números son iguales, pero en el caso de que ocurra de combinaciones inviables, como comentamos anteriormente, este podría no ser el caso.

Por ejemplo, para lograr una cobertura del 100% para la tabla de decisiones en Tabla 4.6, necesitamos tres (no cuatro, como sugeriría el número de combinaciones) Casos de prueba. Si tuviéramos las siguientes pruebas:

- Adulto = SI, fuma = SI.
- Adulto = NO, fuma = SI.
- Adulto = NO, fuma = NO.

entonces lograríamos una cobertura de 2/3 (o alrededor del 66%), ya que los dos últimos casos de prueba cubrir la misma primera columna de la tabla.

Tablas de decisión como técnica de prueba estática

La prueba de la tabla de decisiones es excelente para detectar problemas con los requisitos, como su ausencia o contradicción. Una vez creada la tabla de decisiones a partir de la especificación o incluso mientras aún se está creando, es muy fácil descubrir tales problemas de especificación como:

- Incompletitud: no hay acciones definidas para una combinación específica de condiciones.
- Contradicción: definir en dos lugares diferentes de especificación dos comportamientos del sistema frente a la misma combinación de condiciones
- Redundancia: definir el mismo comportamiento del sistema en dos lugares diferentes del sistema. especificación (quizás descrita de manera diferente)

4.2.4 Pruebas de transición estatal

Solicitud

La prueba de transición de estado es una técnica utilizada para comprobar el comportamiento de un componente o sistema. Por lo tanto, verifica su aspecto conductual: cómo se comporta a lo largo del tiempo y cómo cambia su estado bajo la influencia de varios tipos de eventos.

El modelo que describe este aspecto del comportamiento es la llamada transición de estado. diagrama. En la literatura, diferentes variantes de este modelo se denominan autómatas finitos, autómatas de estados finitos, máquina de estados o sistema de transición etiquetado. El

El programa de estudios utiliza el nombre “diagrama de transición de estados” para indicar la forma gráfica del modelo de transición de estados y “tabla de transición de estados” para indicar la forma tabular equivalente del modelo.

Construcción del diagrama de transición de estados Un

diagrama de transición de estados es un modelo gráfico, como se describe en el estándar UML.

Desde un punto de vista teórico, es un gráfico dirigido etiquetado. El diagrama de transición de estado consta de los siguientes elementos:

- Estados—representan posibles situaciones en las que el sistema puede estar •
- Transiciones—representan posibles (correctos) cambios de estados •
- Eventos—representan fenómenos, generalmente externos al sistema, la ocurrencia de que desencadena las transiciones correspondientes
- Acciones: acciones que el sistema puede realizar durante la transición entre estados. •
- Condiciones de guardia: condiciones lógicas asociadas con las transiciones; una transición puede ejecutarse sólo si la condición de guardia asociada es verdadera

La Figura 4.9 muestra un ejemplo de un diagrama de transición de estados. No es trivial, aunque en la práctica a menudo se utilizan modelos aún más complicados, utilizando una notación más rica que la analizada en el programa de estudios. Sin embargo, este diagrama nos permite mostrar todos los elementos esenciales del modelo de transición de estado descrito en el programa de estudios manteniendo el ejemplo práctico.

El diagrama representa un modelo del comportamiento del sistema al realizar una llamada telefónica a un usuario de teléfono celular con un número específico. El usuario marca un número de teléfono de nueve dígitos presionando las teclas correspondientes a los dígitos sucesivos del número uno por uno. Cuando se ingresa el noveno dígito, el sistema intenta realizar la llamada automáticamente.

El diagrama de transición de estados que modela este sistema consta de cinco estados (rectángulos). Las posibles transiciones entre ellos están indicadas por flechas. El sistema se inicia en la pantalla de bienvenida del estado inicial y espera un evento (etiquetado EnterDigit) en el que el usuario selecciona el primer dígito de un número de teléfono de nueve dígitos. Cuando ocurre este evento, el sistema pasa al estado Entrando y, además, durante esta transición, el sistema realiza una acción para establecer el valor de la variable x en 1. Esta variable representará la cantidad de dígitos del número de teléfono marcado. ingresado por el usuario hasta el momento.

En el estado Entrando, solo puede ocurrir el evento EnterDigit, pero dependiendo de cuántos dígitos se hayan ingresado hasta el momento, son posibles transiciones a dos estados diferentes. Mientras el usuario no haya ingresado el noveno dígito, el sistema permanece en el estado Entrando, aumentando cada vez la variable x en uno. Esto se debe a que la condición de guardia “ $x < 8$ ” es verdadera en esta situación. Justo antes de que el usuario ingrese el último noveno dígito, la variable x tiene un valor de 8. Esto significa que la condición de protección “ $x < 8$ ” es falsa y la condición de protección “ $x = 8$ ” es verdadera. Por lo tanto, al seleccionar el último noveno dígito del número se cambiará del estado Entrando bajo el evento EnterDigit al estado Conectar.

Cuando la llamada tiene éxito (la ocurrencia del evento ConnectionOK), el sistema ingresa al estado de Llamada, en el que permanece hasta que el usuario finaliza la llamada, lo que será señalado por la ocurrencia del evento EndConnection. En este punto, el sistema pasa al estado final Fin y finaliza su funcionamiento. Si el sistema está en el estado Conectar y ocurre el evento ConnectionError, el sistema pasa al estado final, pero a diferencia de la transición análoga desde el estado Llamada, realizará adicionalmente la

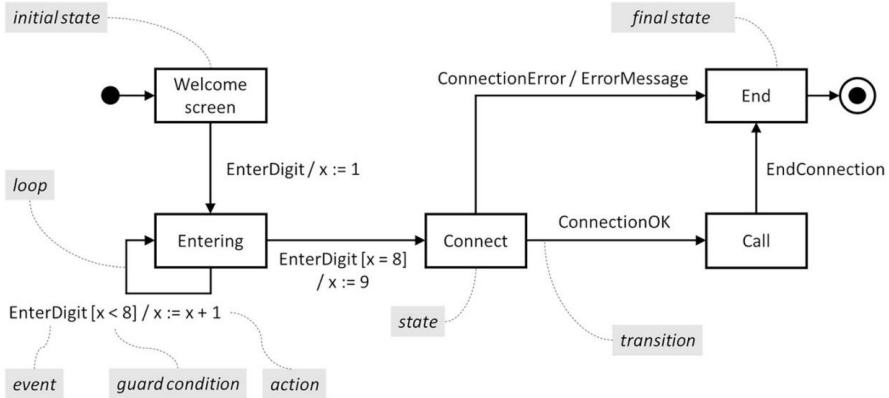


Fig. 4.9 Ejemplo de diagrama de transición de estado

Acción de ErrorMessage, que indica al usuario la imposibilidad de conectarse al número seleccionado.

El sistema en cada momento se encuentra exactamente en uno de los estados, y el cambio de estado se produce como resultado de la ocurrencia de los eventos correspondientes. El concepto de Estado es abstracto. Un estado puede denotar una situación de muy alto nivel (por ejemplo, el sistema está en una determinada pantalla de aplicación), pero también puede describir situaciones de bajo nivel (por ejemplo, el sistema ejecuta una determinada instrucción de programa). El nivel de abstracción depende del modelo adoptado, es decir, lo que el modelo realmente describe y en qué nivel de generalidad. Se supone que cuando ocurre un determinado evento, el cambio de estados es instantáneo (se puede asumir que es un evento de duración cero). Las etiquetas de transición (es decir, las flechas en el diagrama) tienen en general la forma:

evento [condición de guardia] / acción

Si, en un caso dado, la condición o acción de guardia no existe o no es relevante desde el punto de vista del evaluador, se pueden omitir. Por tanto, las etiquetas de transición también pueden adoptar una de las tres formas siguientes:

Evento

Evento/acción

Evento [condición de guardia]

Una condición de guardia para una transición determinada permite que esa transición se ejecute solo si la condición se cumple. Las condiciones de guardia nos permiten definir dos transiciones diferentes bajo el mismo evento evitando el no determinismo. Por ejemplo, en el diagrama de la Fig. 4.9 del estado Entrando, tenemos dos transiciones bajo el evento EnterDigit, pero solo una de ellas se puede ejecutar en cualquier momento porque las condiciones de protección correspondientes son independientes (ya sea x es menor que 8 o es igual a 8).

Un caso de prueba de ejemplo para el diagrama de transición de estado en la Fig. 4.9, verificando la exactitud de una conexión exitosa, podría parecerse al de la Tabla 4.7.

Tabla 4.7 Ejemplo de una secuencia de transiciones en el diagrama de transición de estados de la Fig. 4.9

Paso	Estado	Evento	Acción	siguiente estado
1	Pantalla de bienvenida	Ingresar dígito	$x := 1$	Entrando
2	Entrando	Ingresar dígito	$x := x + 1$	Entrando
3	Entrando	Ingresar dígito	$x := x + 1$	Entrando
4	Entrando	Ingresar dígito	$x := x + 1$	Entrando
5	Entrando	Ingresar dígito	$x := x + 1$	Entrando
6	Entrando	Ingresar dígito	$x := x + 1$	Entrando
7	Entrando	Ingresar dígito	$x := x + 1$	Entrando
8	Entrando	Ingresar dígito	$x := x + 1$	Entrando
9	Entrando	Ingresar dígito	$x := 9$	Conectar
10	Conectar	Conexión OK		Llamar
11	Llamar	Conexión final		Fin

El escenario se desencadena por una secuencia de 11 eventos: EnterDigit (9 veces), Conexión OK y Finalizar conexión. En cada paso, activamos el correspondiente evento y comprobamos si, después de su ocurrencia, el sistema realmente pasa al estado descrito en la columna Siguiente estado.

Diagrama de formas equivalentes de transición de estado

Existen al menos dos formas equivalentes de representación de la transición estatal. diagrama. Considere una máquina simple de cuatro estados con los estados S1, S2, S3 y S4 y eventos A, B y C. El diagrama de transición de estados se muestra en la figura 4.10a.

Sin embargo, de manera equivalente, se puede presentar en forma de tabla de estados, donde Las filas individuales (respectivamente, columnas) de la tabla representan estados sucesivos. (respectivamente, eventos, junto con las condiciones de guardia si existen), y en la celda en la intersección de la columna y la fila correspondientes al estado especificado S y El evento E se escribe como el estado objetivo al que el sistema debe hacer la transición si, estando en estado S, ocurre el evento E. Por ejemplo, si el sistema se encuentra actualmente en el estado S2 y el evento B ha ocurrido, el sistema debe pasar al estado S1. Si nuestro diagrama usara acciones, Habría que anotarlos en las celdas correspondientes de las tablas de la Fig. 4.10b, c.

Una forma más de representar la máquina de estados es mediante el uso de la transición completa. tabla que se muestra en la figura 4.10c. Aquí la tabla representa todas las combinaciones posibles de estados, eventos y condiciones de guardia (si existen). Como tenemos cuatro estados, tres eventos diferentes y sin condiciones de guardia, la tabla contendrá $4 * 3 = 12$ filas. El La última columna contiene el estado objetivo al que la máquina debe realizar la transición si, cuando se en el estado definido en la primera columna de la tabla, el evento descrito en la segunda ocurre la columna de la tabla. Si la transición en cuestión no está definida, el estado Siguiente La columna indica esto, por ejemplo, con un guión u otro símbolo fijo.

La ausencia de una transición (es decir, la ausencia de una combinación estado/evento) en el El diagrama de transición de estado se representa simplemente por la ausencia del correspondiente flecha. Por ejemplo, al estar en el estado S1, la máquina no tiene un comportamiento definido para el ocurrencia del evento B. Por lo tanto, no hay ninguna flecha saliente desde S1 etiquetada por B.

Tanto la tabla de estado como la tabla de transición completa nos permiten mostrar directamente los llamados transiciones no válidas, que también pueden probarse, y en algunas situaciones deben probarse. Las transiciones no válidas (las "flechas que faltan" en el diagrama de transición de estado) son

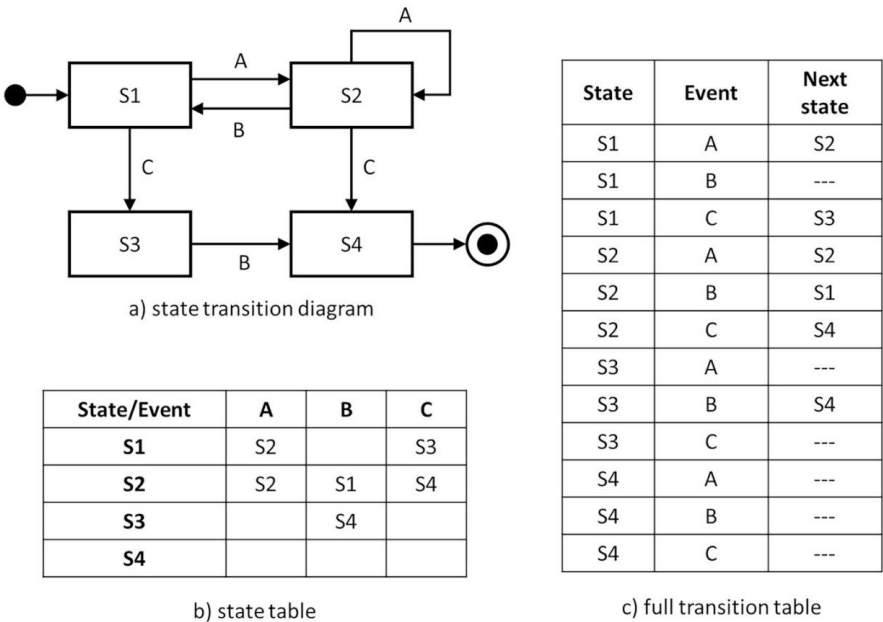


Fig. 4.10 Diferentes formas de presentación de máquinas de estados

representado por celdas vacías en las tablas. En otras palabras, una transición no válida es cualquier combinación de estado y evento que no aparece en el diagrama de transición de estado. Por ejemplo, al diagrama de la figura 4.10 le faltan seis transiciones: (S1, B), (S3, A), (S3, C), (S4, A), (S4, B) y (S4, C). . Por lo tanto, tenemos un total de seis transiciones no válidas, correspondientes a seis celdas vacías en la tabla de la figura 4.10b o seis celdas vacías en la columna "Siguiente estado" de la tabla de la figura 4.10c.

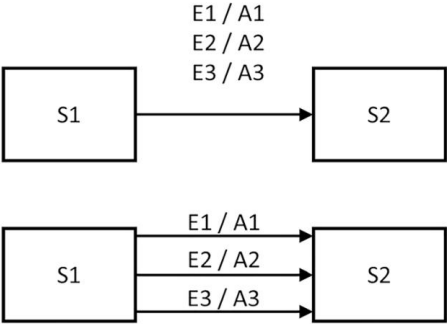
A veces, por simplicidad, solo se dibuja una flecha entre dos estados en el diagrama de transición de estados, incluso si hay más transiciones paralelas entre estos estados. Es importante prestar atención a esto, porque a veces la cuestión del número de transiciones en el diagrama es muy importante. La Figura 4.11 muestra un ejemplo de una forma "simplificada" de transiciones de dibujo y una forma "completa" equivalente.

Diseño de la prueba: Cobertura

Para las pruebas de transición estatales, existen muchos criterios de cobertura diferentes. Estos son los tres criterios más populares, descritos en el programa de estudios de Foundation Level:

- Cobertura de todos los estados: el criterio de cobertura más débil. Los elementos de cobertura son los estados. Por lo tanto, la cobertura de todos los estados requiere que el sistema haya estado en cada estado al menos una vez.
- Cobertura de transiciones válidas (también llamada cobertura de cambio 0 o cobertura de Chow): el criterio de cobertura más popular. Los elementos de cobertura son transiciones entre estados. Por lo tanto, la cobertura de transiciones válidas requiere que cada transición definida en el diagrama de transición de estado se ejecute al menos una vez.

Fig. 4.11 Formas equivalentes de representación gráfica de transiciones paralelas



- Cobertura de todas las transiciones: en la que los elementos de cobertura son todas las transiciones indicadas en la tabla estatal. Por lo tanto, este criterio requiere que se cubran todas las transiciones válidas y, además, se intente la ejecución de cada transición no válida. Es una buena práctica probar uno de estos eventos por prueba (para evitar el enmascaramiento de defectos).

Se pueden definir otros criterios de cobertura, como la cobertura de pares de transiciones (también denominada cobertura de 1 conmutador), que requiere que se prueben todas las secuencias posibles de dos transiciones válidas consecutivas. Los criterios de este tipo se pueden generalizar: podemos exigir la cobertura de las tres transiciones, de las cuatro transiciones, etc. En general, podemos definir toda una familia de criterios de cobertura con respecto a N+1 transiciones consecutivas, para cualquier no negativo. N (llamada cobertura de conmutador N). También es posible considerar la cobertura de algunos caminos particulares, la cobertura de circuitos, etc. (todos estos criterios de cobertura no están dentro del alcance del programa de estudios de nivel básico).

Por lo tanto, en el caso de las pruebas de transición de estados, estamos tratando con un número potencialmente infinito de posibles criterios de cobertura. Desde un punto de vista práctico, los criterios más utilizados son la cobertura de transiciones válidas y la cobertura de todas las transiciones, ya que lo principal que normalmente nos preocupa al probar el diagrama de transición de estados es la verificación de la implementación válida de las transiciones entre estados.

La cobertura se define como el número de elementos cubiertos por las pruebas en relación con el número de todos los elementos de cobertura definidos por el criterio. Por ejemplo, para el criterio de cobertura de todos los estados aplicado al diagrama de transición de la figura 4.10, tenemos cuatro elementos que cubrir: los estados S1, S2, S3 y S4; para el criterio de cobertura de transiciones válido, tenemos tantos elementos como transiciones entre estados (seis).

El requisito habitual es diseñar el menor número posible de casos de prueba que sean suficientes para lograr una cobertura total. Veamos cómo se pueden realizar diferentes tipos de cobertura para el diagrama de transición de estados que se muestra en la figura 4.10a.

Para el criterio de cobertura de todos los estados, tenemos cuatro estados para cubrir: S1, S2, S3 y S4. Tenga en cuenta que esto se puede lograr dentro de un único caso de prueba, por ejemplo:

S1 (A) S2 (B) S1 (C) S3 (B) S4.

La convención utilizada anteriormente describe la secuencia de transiciones entre estados bajo la influencia de eventos (indicados entre paréntesis). La notación "S (E) T" denota la transición del estado S bajo la influencia del evento E al estado T. En el

Tabla 4.8 Escenario de prueba para pruebas de transición de estado

Paso	Condición inicial	Evento	Resultado Esperado
1	T1	A	Transición a S2
2	T2	B	Transición a S1
3	T1	C	Transición a S3
4	T3	B	Transición a S4
5	T4		

En el ejemplo anterior, pasamos por los cuatro estados dentro de un caso de prueba, por lo que este único El caso de prueba logró una cobertura estatal del 100%. En la práctica, el resultado esperado de la prueba es el paso real de los estados uno por uno como lo describe el modelo. El escenario de prueba correspondiente al caso de prueba:

S1 (A) S2 (B) S1 (C) S3 (B) S4

podría verse así, como se describe en la Tabla 4.8.

Tenga en cuenta que un caso de prueba no se equipara con una condición de prueba. En nuestro ejemplo, la prueba Las condiciones eran estados individuales, pero un solo caso de prueba podría cubrirlos a todos. Una prueba El caso es una secuencia de transiciones entre estados, comenzando con el estado inicial y terminando con el estado final (posiblemente interrumpiendo este viaje antes si es necesario). Por lo que Es posible cubrir más de una condición de prueba dentro de un solo caso de prueba.

Para el criterio de cobertura de transiciones válido, tenemos seis transiciones que cubrir. Dejar los denotamos como T1–T6:

- T1: S1 (A) S2
- T2: T1 (C) S3
- T3: S2 (A) S2
- T4: S2 (B) S1
- T5: S2 (C) S4
- T6: S3 (B) S4

Queremos diseñar la menor cantidad de pruebas posible para cubrir estas seis transiciones. El La estrategia es tratar de cubrir tanto como sea posible los elementos previamente descubiertos dentro de cada prueba sucesiva. Por ejemplo, si comenzamos el primer caso de prueba con la secuencia S1 (A) S2, en este punto no valdría la pena activar el evento C y pasar a la fase final. estado S4, cuando todavía podemos cubrir varias otras transiciones, como S2 (A) S2 (B) S1.

Tenga en cuenta que es imposible cubrir, dentro de un solo caso de prueba, las transiciones S2 (C) S4 y S3 (B) S4, porque tan pronto como se alcanza S4, el caso de prueba debe finalizar. Entonces, Necesitaremos al menos dos casos de prueba para cubrir todas las transiciones. De hecho, dos casos son suficiente para lograr una cobertura completa de las transiciones válidas. Un conjunto de ejemplo de estas dos pruebas. casos se muestra en la Tabla 4.9.

En esta tabla, la primera columna proporciona la secuencia de transiciones y la segunda La columna proporciona las transiciones cubiertas correspondientes. Transiciones cubiertas por primera vez. El tiempo se indica en **negrita** (por ejemplo, en la segunda prueba, cubrimos T1, que ya estaba cubierto previamente con el primer caso de prueba).

Tabla 4.9 Casos de prueba que logran una cobertura total de transiciones válidas

Caso de prueba	Transiciones cubiertas
S1 (A) S2 (A) S2 (B) S1 (C) S3 (B) S4	T1, T3, T4, T2, T6
S1 (A) S2 (C) S4	T1, T5

Para lograr la cobertura de todas las transiciones, debemos proporcionar casos de prueba que cubran todas transiciones válidas (ver arriba) y, además, intentar ejercer cada transición no válida. De acuerdo con las buenas prácticas descritas anteriormente, probaremos cada uno de estos transición en un caso de prueba separado. Por lo tanto, el número de casos de prueba será igual a el número de casos de prueba que cubren todas las transiciones válidas más el número de casos no válidos transiciones. En nuestro modelo, estas serán las siguientes seis transiciones no válidas (puede Escribalos rápidamente analizando la tabla de estado completa (consulte la figura 4.10c):

- S1(B)?
- T3 (A)?
- T3 (C)?
- T4 (A)?
- T4 (B)?
- T4 (C)?

Recuerde que cada caso de prueba comienza con un estado inicial. Así, por cada inválido transición, primero debemos llamar a la secuencia de eventos que alcanza el estado dado y luego, una vez que estemos en él, intente llamar a la transición no válida. Para las seis transiciones inválidas descrito anteriormente, los seis casos de prueba correspondientes podrían verse como en la Tabla 4.10.

Si logramos desencadenar un evento que no está definido en el modelo, podemos interpretar hacerlo al menos de dos maneras:

- Si el sistema cambia de estado, esto debe considerarse una falla, ya que desde el modelo no permite dicha transición, no debería ser posible desencadenarla.
- Si el sistema no cambió su estado, entonces esto puede considerarse un comportamiento correcto. (ignorando el evento). Sin embargo, debemos estar seguros de que semánticamente se trata de una situación aceptable.

También existen al menos dos posibles soluciones a situaciones tan problemáticas:

- Arreglar el sistema para que el evento no sea posible (la transición no válida es imposible que se va a activar).

Tabla 4.10 Casos de prueba que cubren transiciones no válidas

Caso de prueba	Transición incorrecta cubierta
S1(B)?	S1 (B)?
S1 (C) S3 (A)?	T3 (A)?
S1 (C) S3 (C) ?	T3 (C)?
S1 (C) S3 (B) S4 (A)?	T4 (A)?
S1 (C) S3 (B) S4 (B) ?	T4 (B)?
S1 (C) S3 (B) S4 (C) ?	T4 (C)?

- Agregar una transición bajo la influencia de este evento al modelo para que modele el “ignorar” el evento por parte del sistema (por ejemplo, un bucle al mismo estado).

Volvamos al ejemplo práctico del diagrama de transición de estados de la figura 4.9 que modela una llamada telefónica. Un ejemplo de una transición no válida es la transición del estado Conectar en respuesta al evento EnterDigit. Esta situación se desencadena de forma muy sencilla: en el momento de establecer una llamada, simplemente pulsamos una de las teclas que representan dígitos. Si el sistema no reacciona de ninguna manera, consideramos que la prueba ha pasado.

Finalmente, analicemos uno de los criterios de cobertura no descritos en el programa: la cobertura de pares de transiciones válidas (cobertura de 1 cambio). Este es un material adicional, no examinable en el examen de certificación de nivel básico.

Consideramos que este ejemplo muestra que cuanto más fuerte es el criterio que adoptamos, más difícil es y generalmente requiere diseñar más casos de prueba que para un criterio más débil (en este caso, el criterio más débil es la cobertura de transiciones válidas).

Para satisfacer la cobertura de 1 interruptor, debemos definir todos los pares permitidos de transiciones válidas. Podemos hacer esto de la siguiente manera: para cada transición única y válida, consideramos todas sus posibles continuaciones en la forma de la siguiente transición única y válida. Por ejemplo, para la transición S1 (A) S2, las posibles continuaciones son todas las transiciones que salen de S2, es decir, S2 (A) S2, S2 (B) S1 y S2 (C) S4. Por lo tanto, todos los pares posibles de transiciones válidas se ven así:

- PP1: S1 (A) S2 (A) S2
- PP2: T1 (A) S2 (B) S1
- PP3: S1 (A) S2 (C) S4
- PP4: T1 (C) S3 (B) T4
- PP5: T2 (A) S2 (A) S2
- PP6: S2 (A) S2 (B) S1
- PP7: S2 (A) S2 (C) S4
- PP8: T2 (B) S1 (A) S2
- PP9: T2 (B) S1 (C) T3

Tenga en cuenta (análogo al caso de cobertura de transiciones válidas) que los pares de transiciones PP3, PP4 y PP7 terminan en el estado final, por lo que no pueden ocurrir dos de ellos en un solo caso de prueba, ya que la ocurrencia de cualquiera de estos pares de transiciones resulta en la terminación del caso de prueba. Por lo tanto, necesitamos al menos tres casos de prueba para cubrir pares de transiciones válidas y, de hecho, tres casos de prueba son suficientes. En la Tabla 4.11 se proporciona un conjunto de pruebas de ejemplo (los pares cubiertos por primera vez se indican en negrita). Por lo tanto, necesitamos un caso de prueba más en comparación con el caso de cobertura de transiciones válidas.

Tabla 4.11 Casos de prueba que cubren pares de transiciones válidas

Caso de prueba	Pares cubiertos de transiciones válidas.
S1 (A) S2 (A) S2 (A) S2 (B) S1 (A) S2 (B) S1 (C) S3 (B) T4	PP1, PP5, PP6, PP8, PP2, PP9, PP4
S1 (A) S2 (C) S4	PP3
S1 (A) S2 (A) S2 (C) S4	PP1, PP7

4.2.5 (*) Pruebas de casos de uso

Aplicación Un

caso de uso es un documento de requisitos que describe la interacción entre los llamados actores, que suelen ser el usuario y el sistema.

Un caso de uso es una descripción de una secuencia de pasos que realizan el usuario y el sistema, que en última instancia lleva a que el usuario obtenga algún beneficio. Cada caso de uso debe describir un escenario único y bien definido. Por ejemplo, si documentamos los requisitos para un cajero automático (un cajero automático), los casos de uso podrían, entre otras cosas, describir los siguientes escenarios:

- Rechazar una tarjeta de cajero automático no válida
- Iniciar sesión en el sistema ingresando el PIN correcto
- Retirar dinero correctamente de un cajero automático
- Intento fallido de retiro de dinero debido a fondos insuficientes en la cuenta
- Bloquear la tarjeta ingresando el PIN incorrecto tres veces

Para cada caso de uso, el evaluador puede construir un caso de prueba correspondiente, así como un conjunto de casos de prueba para verificar la ocurrencia de eventos inesperados durante la ejecución del escenario.

Construcción de casos de uso

Un caso de uso correctamente construido, además de información "técnica", como un número y un nombre únicos, debe consistir en:

- Condiciones previas (incluidos los datos de entrada)
- Exactamente un escenario principal secuencial
- (Opcional) Marcas de las ubicaciones de los llamados flujos alternativos y excepciones
- Condiciones posteriores (que describen el estado del sistema después de la ejecución exitosa de el escenario y el beneficio obtenido para el usuario)

Desde el punto de vista del evaluador, el propósito principal del caso de uso es probar el escenario principal, es decir, verificar que realizar realmente todos los pasos como describe el escenario conduce al beneficio definido para el usuario. Sin embargo, al hacerlo, también es necesario probar el comportamiento del sistema en busca de eventos "insospechados", es decir, flujos alternativos y excepciones. La diferencia entre estos dos es la siguiente:

- Un flujo alternativo provoca que ocurra un evento inesperado pero permite al usuario completar el caso de uso, es decir, le permite regresar al escenario principal y completarlo felizmente.
- Una excepción interrumpe la ejecución del escenario principal; si se produce una excepción, no es posible completar el escenario principal correctamente. El usuario no obtiene ningún beneficio y el caso de uso finaliza con un mensaje de error o se cancela. La aparición de la excepción en sí se puede controlar mediante un escenario definido en un caso de uso independiente.

Un caso de uso no debe contener lógica de negocios, es decir, el escenario principal debe ser lineal y no contener ramificaciones. La existencia de tal ramificación sugiere que