**Обзор перехода из состояния в состояние**

Тестирование перехода из состояния в состояние исследует, как программная система переходит из одного состояния в другое в ответ на события. Состояния - это различные состояния или режимы, в которых может находиться система. Определенные события запускают переходы и отображают перемещения между этими состояниями.

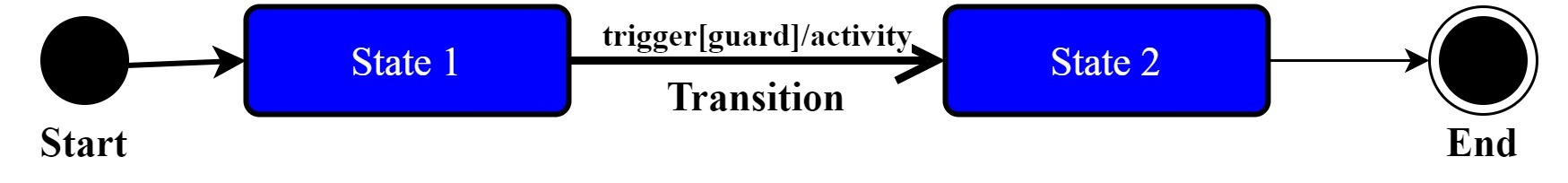
В рамках этого подхода мы анализируем поведение программного обеспечения в различных сценариях, учитывая события, вызывающие переходы, и последующие действия. **Тестирование перехода из состояния в состояние полезно для систем с четкой последовательностью состояний и событий.**Это связано с тем, что в нем систематически исследуется, как программное обеспечение реагирует на различные сценарии, обеспечивая надлежащее поведение во время переходов между состояниями, вызванных определенными событиями.

Более того, этот подход расширяет охват тестированием и помогает в выявлении потенциальных [дефектов](https://translated.turbopages.org/proxy_u/en-ru.ru.a4ac828e-658511a6-b6e7045b-74722d776562/https/www.baeldung.com/cs/software-testing-defect-bug-error-and-failure) в системах, где изменения состояния следуют в четко определенной последовательности. Допустим, у нас есть система светофоров, которая изменяет сигналы в зависимости от времени суток, интенсивности движения и присутствия пешеходов. Тестирование перехода от состояния к состоянию поможет убедиться, что система правильно реагирует на различные комбинации этих факторов.

**3. Диаграммы перехода от состояния к состоянию: визуальное представление**

Мы визуализируем тестирование перехода от состояния к состоянию с помощью диаграмм перехода от состояния к состоянию. Эти диаграммы обеспечивают**графическое представление состояний, переходов, событий и действий системы.**

Диаграммы перехода от состояния к состоянию состоят из состояний, представленных прямоугольниками, переходов, показанных стрелками между состояниями, событий, запускающих переходы, и действий, связанных с переходами:



Кружками обозначены начальное и конечное состояния.

Такие диаграммы дают нам четкое и структурированное представление о том, как наше программное обеспечение ведет себя в ответ на различные входные данные. Таким образом, они помогают нам выявлять потенциальные дефекты, проверять правильные переходы состояний и гарантировать, что программное обеспечение работает так, как задумано.

Диаграммы перехода от состояния к состоянию служат основой для разработки тестовых примеров и изучения поведения нашего программного обеспечения.

**4. Применение тестирования перехода из состояния в состояние**

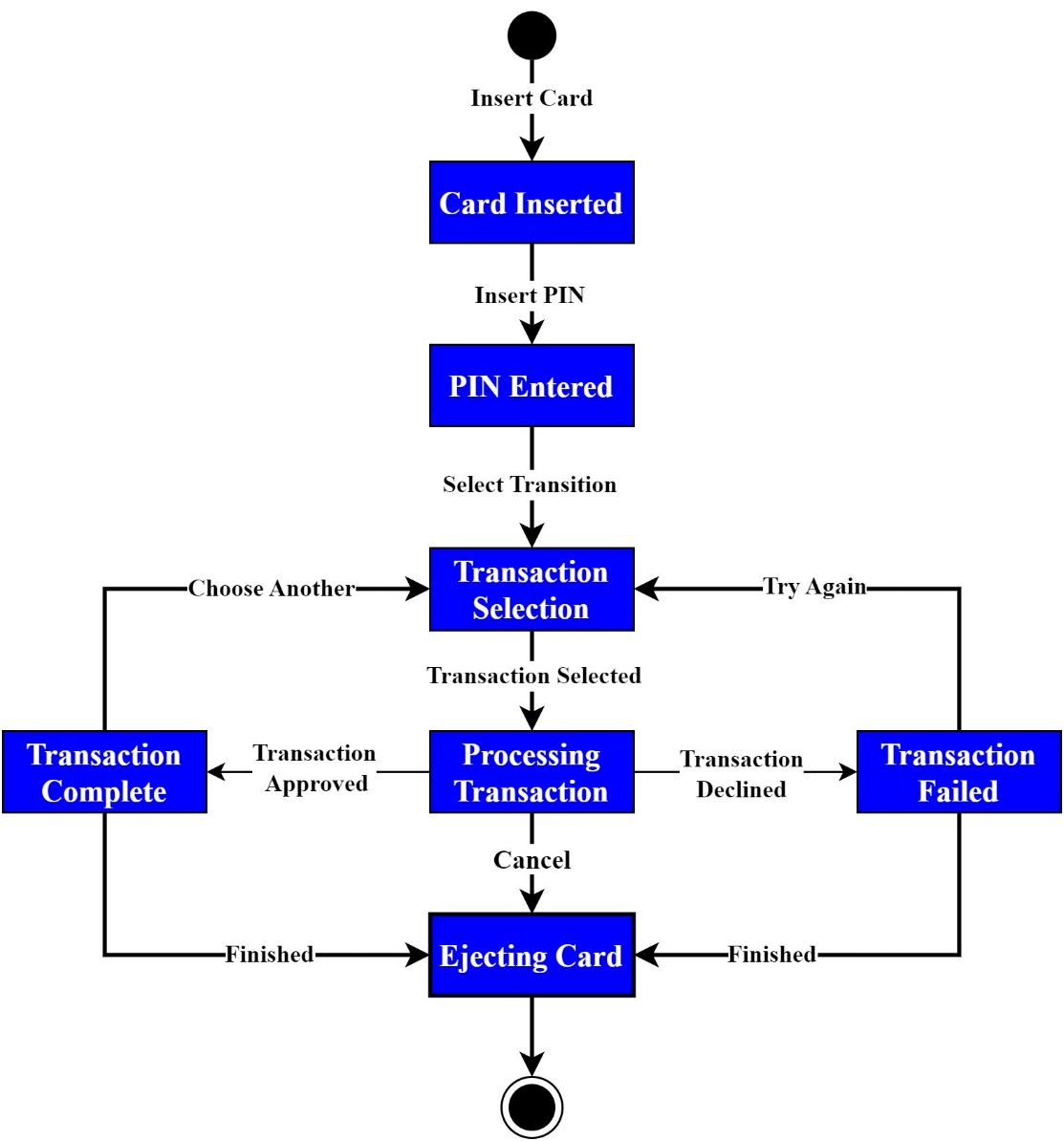
Чтобы эффективно применять тестирование перехода от состояния к состоянию, мы выполняем следующие действия:

1. **Определение состояний и событий:** Мы начинаем с понимания различных состояний системы и событий, которые запускают переходы состояний. Это включает анализ [системных требований](https://translated.turbopages.org/proxy_u/en-ru.ru.a4ac828e-658511a6-b6e7045b-74722d776562/https/www.baeldung.com/cs/requirements-functional-vs-non-functional) и спецификаций
2. **Создание диаграмм перехода от состояния к состоянию:** Мы разрабатываем диаграммы перехода от состояния к состоянию для визуализации состояний, переходов, событий и действий. Эти диаграммы дают четкий обзор поведения программного обеспечения.
3. **Вывод тестовых примеров:** Мы выводим тестовые примеры из диаграмм перехода состояний, которые охватывают все возможные переходы состояний и комбинации событий. Для каждого тестового примера **мы указываем начальное состояние, инициирующее событие и ожидаемый результат**
4. **Выполнение тестов:** Мы выполняем производные тестовые примеры в программном обеспечении и наблюдаем, как система переходит из состояния в состояние. Этот шаг определяет, точно ли выполняются ожидаемые действия
5. **Анализ результатов:** На заключительном этапе мы оцениваем результаты тестирования, чтобы убедиться, что программное обеспечение работает так, как задумано. Наконец, мы документируем и устраняем любые несоответствия или отклонения от ожидаемого поведения.

**4.1. Пример**

Давайте рассмотрим реальный пример, чтобы лучше понять тестирование перехода из состояния в состояние. Мы рассмотрим, как тестирование перехода из состояния в состояние может быть применено к системе банкоматов.

Система ATM претерпевает различные состояния и переходы в зависимости от взаимодействия пользователя:



Сначала мы определяем состояния в системе банкоматов, такие как “Карта вставлена”, "Введен PIN-код” и ”Выбор транзакции". После идентификации состояний мы определим события, которые запускают переходы между этими состояниями.

Мы изображаем переходы с помощью диаграммы, иллюстрирующей переход из одного состояния в другое при возникновении события. Например, мы переходим от “Вставлена карта" к “Введен PIN-код” при возникновении события “Вставить PIN-код”.

Во всех этих транзакциях мы включаем условия, влияющие на эти переходы, такие как условие “Если PIN-код действителен" для перехода от “PIN-код введен” к “Выбору транзакции”.

**4.2. Тестовые примеры**

**Мы формулируем тестовые примеры на основе диаграммы перехода состояния,** охватывающей такие сценарии, как вставка карты без последующего ввода PIN-кода, ввод неправильного PIN-кода, выбор различных типов транзакций, отклонение транзакции и успешное завершение транзакции.

Более того, мы тестируем систему ATM, используя заранее спланированные сценарии, чтобы убедиться, что взаимодействие с пользователем приводит к плавному переходу из состояния в состояние. После завершения тестов мы анализируем результаты, чтобы подтвердить, что фактические переходы состояний соответствуют ожидаемым, как показано на диаграмме перехода состояний. Наконец, мы документируем и сообщаем о любых выявленных несоответствиях и потенциальных дефектах в системе.

**5. Преимущества и ограничения тестирования перехода от состояния к состоянию**

Тестирование с переходом в состояние имеет преимущества и ограничения, которые влияют на его применение и эффективность при тестировании программного обеспечения.

**5.1. Преимущества**

Мы используем тестирование перехода из состояния в состояние для изучения поведения программного обеспечения в различных сценариях. Таким образом, мы обеспечиваем систематический охват различных условий путем отображения состояний, переходов и событий. Таким образом, этот подход позволяет нам идентифицировать критические пути, включающие переходы состояний и события.

**Диаграммы перехода от состояния к состоянию обеспечивают четкое и понятное описание сложных действий программного обеспечения.**Такая ясность улучшает понимание и способствует эффективной коммуникации между нашей командой тестирования и заинтересованными сторонами.

Кроме того, в нашем процессе разработки тестовых примеров используются диаграммы перехода из состояния в состояние, которые помогают нам определять соответствующие тестовые сценарии и обеспечивают систематический охват этих сценариев. Это приводит к созданию всеобъемлющих и хорошо структурированных тестовых примеров.

**5.2. Ограничения**

Тестирование перехода из состояния в состояние имеет определенные ограничения. **Управление сложностью может отнимать много времени у сложных систем с многочисленными состояниями, переходами и событиями.**

Тщательность нашего тестирования перехода из состояния в состояние зависит от полной модели. Следовательно, **отсутствующие состояния или переходы могут привести к недостаточному охвату тестированием,** оставляя дефекты необнаруженными.

Кроме того,**диаграммы перехода из состояния в состояние в основном отражают статическое поведение программного обеспечения.**Динамические элементы, такие как синхронизация или [параллельные](https://translated.turbopages.org/proxy_u/en-ru.ru.a4ac828e-658511a6-b6e7045b-74722d776562/https/www.baeldung.com/cs/concurrency-vs-parallelism) события, могут быть представлены неадекватно, что ограничивает охват нашего тестирования.

Во время тестирования перехода из состояния в состояние мы фокусируемся только на явных переходах, которые происходят из-за определенных событий. Однако, **такой подход может привести к игнорированию неявных переходов** и непредвиденных взаимодействий между состояниями, создавая слепые зоны в процессе тестирования.

**6. Заключение**

**В этой статье мы познакомились с концепциями тестирования перехода из состояния в состояние, которое предлагает системный подход к проверке поведения программного обеспечения.**Этот метод помогает нам выявлять дефекты, проверять правильность переходов и повышать общую надежность программного обеспечения.

Мы можем создавать эффективные диаграммы перехода от состояния к состоянию и выводить значимые тестовые примеры, понимая состояния, события и действия системы. Включение тестирования на новое состояние в наш процесс тестирования способствует созданию [высококачественного программного обеспечения,](https://translated.turbopages.org/proxy_u/en-ru.ru.a4ac828e-658511a6-b6e7045b-74722d776562/https/www.baeldung.com/cs/software-quality)соответствующего ожиданиям пользователей.

Однако мы также должны знать о его ограничениях и дополнять его другими методами тестирования, такими как [анализ граничных значений](https://translated.turbopages.org/proxy_u/en-ru.ru.a4ac828e-658511a6-b6e7045b-74722d776562/https/www.baeldung.com/cs/bva) и [разделение эквивалентности](https://translated.turbopages.org/proxy_u/en-ru.ru.a4ac828e-658511a6-b6e7045b-74722d776562/https/www.baeldung.com/cs/software-testing-equivalence-partitioning), чтобы обеспечить комплексную и эффективную стратегию тестирования.