

П.В. Сенченко, В.С. Масляев

Динамическая модель символического выполнения параллельных программ

Рассматривается пример представления в виде сети Петри модели символического выполнения параллельных программ. Отмечается, что символическое выполнение программ может быть рассмотрено как композиция тестирования программных продуктов и доказательных методик проверки корректности.

Ключевые слова: сети Петри, верификация, валидация, контроль качества, тестирование, символическое выполнение программ.

На каждом этапе разработки программный продукт (в том числе отдельные программные системы и комплексы программ) подвергается контролю качества. Всегда следует учитывать, что не полностью протестированная программа с большой долей вероятности будет некорректно функционировать, т.е. не будет удовлетворять потребностям конечных пользователей. По этой причине тестированию (и другим методам контроля качества) во всех компаниях, занимающихся разработкой программных продуктов, уделяется особое внимание.

Тестирование и оценка работоспособности программного продукта, которые в совокупности обеспечивают подтверждение его соответствия требованиям, предъявляемым к функциональным и эксплуатационным характеристикам, а также требованиям к пользовательскому интерфейсу, называется «валидацией программного продукта» [1]. Кроме валидации, одной из составляющих частей тестирования программного продукта является его верификация. «Верификация – это процесс определения соответствия программы на каждом этапе разработки требованиям, устанавливаемым на предыдущих этапах» [Там же].

Одним из способов дополнительного контроля качества является символическое выполнение программ. Это довольно мощное инструментальное средство, результатом которого являются формулы, позволяющие выводить определенные свойства программы. Символическим выполнением называется метод верификации, который по классификации может располагаться между тестированием и анализом: это синтез экспериментального и аналитического подходов к верификации программного продукта [2]. Символическое выполнение программ можно использовать косвенно как вспомогательное средство при тестировании программного обеспечения. Применение такого метода способно помочь при выборе тестовых данных для определения пути выполнения тестирования довольно сложных программных продуктов.

Рассмотрим использование символического выполнения при анализе параллельного программного обеспечения. Для проведения анализа будем рассматривать сеть Петри, в которой меткам присвоены значения, а каждый переход определяет некое действие и условие (предикаты), определяющее момент срабатывания перехода, на основании значений вхо-

дящих меток. Будем использовать сведения, представленные в [2, 3].

Сеть Петри C может быть представлена четверкой [4]:

$$C = (P, T, I, O), \quad (1)$$

где $P = \{p_1, p_2, \dots, p_N\}$ – конечное множество позиций, $N \geq 0$; $T = \{t_1, t_2, \dots, t_M\}$ – конечное множество переходов, $M \geq 0$; $P \cap T = \emptyset$; $I: T \rightarrow P^\infty$ – входная функция, представляющая собой отображение множества переходов в комплекты позиций; $O: T \rightarrow P^\infty$ – выходная функция, представляющая собой отображение множества переходов в комплекты позиций.

Мощность множества P есть число N , а мощность множества T есть число M . Произвольный элемент T обозначается символом t_j , $j = \overline{1, M}$, а произвольный элемент P – p_i , $i = \overline{1, N}$. Позиция p_i является входной позицией перехода t_j , в том случае, если $p_i \in I(t_j)$. Позиция p_i является выходной позицией перехода t_j если $p_i \in O(t_j)$. Входы и выходы переходов представляют собой комплекты позиций.

Символическое выполнение дает возможность применять для меток некие символические значения, при этом символически оценивая предикаты переходов. В последовательной программе состояние символического интерпретатора в любой точке можно представить в виде тройки:

$$\langle \text{symbolic_variable_values}, \text{execution_path}, \text{path_condition} \rangle, \quad (2)$$

где path_condition однозначно определяет выполняемую ветвь. Как известно, в параллельных системах путь выполнения некой составной операции представляет собой последовательность элементарных шагов выполнения [2]. В проектируемой сети Петри элементарный шаг можно смоделировать в виде срабатывания перехода, получающего на вход кортеж меток из входных позиций сети.

Пусть в каждой позиции присутствует не более одной метки, тогда последовательность элементарных шагов можно смоделировать последовательностью срабатываний переходов, устраняющей неопределенность, возникающую при одновременном срабатывании нескольких переходов. Таким образом, выражение (2) можно использовать для моделирования символического состояния интер-

претатора при условии, что *execution_path* определяется как некая последовательность срабатывания.

На рис. 1 представлен пример спецификации параллельной системы [2]. Предложенная сеть Петри представляет сообщения, которые можно получать и отправлять на один из трех каналов. Оператор $m := f()$ моделирует генерацию сообщения в почтовом ящике. Функцию f без аргумента можно представить как генератор случайных чисел, определяющий, что сообщения поступили в систему из внешней среды без взаимосвязи с внутренним состоянием системы.

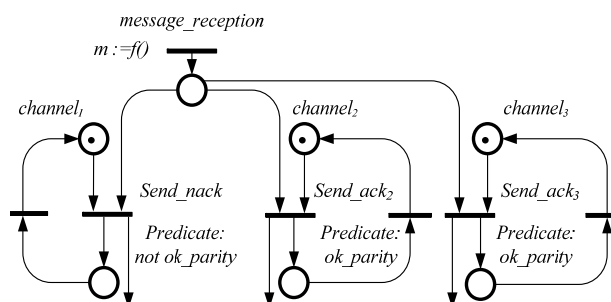


Рис. 1. Пример сети Петри для описания канала передачи информации

При символическом выполнении сети Петри определим начальное значение True для условия пути. Переход может сработать в том случае, если сопоставляемый ему предикат подразумевается условием пути. Если определено, что данный переход должен сработать, выполняемый путь (т.е. последовательность срабатывания) обновляется.

Определим также, что переход может сработать и в том случае, если условие пути не подразумевает ни истинности, ни ложности связанного с ним предиката. В этом случае при решении о срабатывании данного перехода обновляется не только путь выполнения, но и условие пути. Обновление происходит конъюнкцией оцениваемого предиката. Рассмотрим условие пути:

$M_1.ok_parity$ and not $M_2.ok_parity$ and $M_3.ok_parity$,

где M_1, M_2, M_3 – символические значения полученных сообщений (метки, сгенерированные переходом *message_reception*); $M_i.ok_parity$ ($i = 1, 2, 3$) указывает на корректность контрольного разряда четности M_i . Таким образом, условие пути делает допустимой следующую возможность срабатывания:

<message_reception, send_ack₂, message_reception, send_nack, message_reception, send_ack₃>.

В том случае если условие пути содержит $M_i.ok_parity$, могут срабатывать два перехода, поскольку связанный с ним предикат истинный. Последовательность срабатывания определяет выбор между этими переходами. Следовательно, для явной характеристики выполнения пути необходимы условие пути и последовательность срабатывания.

Далее можно использовать пару <firing sequence, path condition> для осуществления процесса тестирования. Например, если рассматривать сеть

Петри (см. рис. 1) как спецификацию системы коммуникаций, то для установки ограничений на входные сообщения, передаваемые системе в момент ее тестирования, можно использовать условие пути, а затем проверить, выполняет ли система операции в той же последовательности, которая указана в заданной паре.

Однако из-за недетерминированности системы другие последовательности действий также будут считаться действительными, и тогда необходимо выполнить дополнительную проверку – на самом ли деле рассматриваемая последовательность является действительной в соответствии с описанием данной сети [2].

С другой стороны, в случае если необходимо воспроизвести точно такую же последовательность действий, как указано в ранее приведенной паре, то могут возникнуть дополнительные проблемы, поскольку недетерминированность нужно разрешать в полном соответствии с тем, как она изложена в элементе пары, отвечающем за последовательность срабатывания. К сожалению, если какие-то действия выполняются внешней средой непредсказуемо (как в случае с получением сообщений почтовым ящиком), добиться этого достаточно сложно. Несмотря на отмеченные недостатки, символическое выполнение может в значительной степени систематизировать выбор тестовых данных, а в некоторых случаях на основе сети Петри возможно построение алгоритма выполнения довольно сложного процесса.

Подводя итог, необходимо еще раз отметить, что тестирование программ относится к экспериментальным методам контроля качества программного обеспечения, а критический анализ и доказательства корректности – к аналитическим. Соответственно символическая интерпретация – композиция этих понятий. Использование математического аппарата сетей Петри для моделирования символического выполнения способно облегчить проведение анализа параллельно выполняющихся программ, являясь вспомогательным инструментарием для специалистов по контролю качества программных систем. При этом запуск сети в среде имитационного моделирования (например, Dedign/CPN) способен продемонстрировать поведение программы при различных состояниях.

Работа выполнена в рамках государственного задания Министерства образования и науки РФ, проект № 8.8184.2017/8.9 «Методология создания систем энергогенерирующих и энергопреобразующих устройств для наземных и бортовых комплексов наземного, космического и подводного базирования».

Литература

1. Сенченко П.В. Надежность, эргономика и качество АСОИУ: учеб. пособие. – Томск: ТУСУР, 2006. – 185 с.
2. Гецци К. Основы инженерии программного обеспечения: пер. с англ. / К. Гецци, М. Джажайери, Д. Мандриоли. – 2-е изд. – СПб.: БХВ-Петербург, 2005. – 832 с.

3. Ехлаков Ю.П. Динамические модели бизнес-процессов. Теория и практика реинжиниринга / Ю.П. Ехлаков, В.Ф. Тарасенко, О.И. Жуковский, П.В. Сенченко, Ю.Б. Грищенко; под ред. Ю.П. Ехлакова. – Томск: ТУСУР, 2014. – 203 с.

4. Питерсон Дж. Теория сетей Петри и моделирование систем: пер. с англ. – М.: Мир, 1984. – 264 с.

Сенченко Павел Васильевич

Канд. техн. наук, доцент каф. АОИ ТУСУРа

Эл. почта: pvs@tusur.ru

Масляев Владимир Сергеевич

Ст. преп. каф. АОИ ТУСУРа

Эл. почта: mavose@rambler.ru

УДК 519.862

М.Г. Тиндова

Теоретическая разработка и практическая реализация приложения для оценки машин и оборудования химической отрасли

Рассматриваются вопросы теоретической разработки и программного применения мобильного приложения для оценки стоимости машин и оборудования на примере химической промышленности. На первом шаге проводится анализ особенностей страхования машин и оборудования, особенностей их оценки; проводится исследование методов оценки в рамках различных подходов, а также анализируются возможности программной реализации каждого из них. На следующем этапе разрабатывается архитектура приложения, проводится тестирование. В результате на конкретном примере показана работа приложения в целях оценки стоимости и страхования конкретного оборудования химической промышленности.

Ключевые слова: разработка мобильных приложений, страхование, стоимость машин и оборудования.

Одним из обязательных условий функционирования любого предприятия является страхование его имущества, в частности машин и оборудования, которое, согласно ГК РФ, осуществляется в основном в форме добровольного страхования, за исключением государственного имущества, передаваемого в аренду. Обязательными условиями имущественного страхования являются, во-первых, непревышение страховой суммой действительной стоимости имущества, т.е. восстановительной (балансовой) стоимости на момент заключения договора. А во-вторых, наличие франшизы – неоплачиваемой части ущерба, размер которой примерно равен затратам страховщика на определение суммы ущерба [1]. Франшиза может быть условной и безусловной. Условная франшиза определяет неоплачиваемую часть ущерба в том случае, если размер ущерба меньше ее значения. Если размер ущерба превышает размер условной франшизы, то она не учитывается при определении размера ущерба. Безусловная франшиза определяет неоплачиваемую часть ущерба независимо от его размера. Франшиза устанавливается в процентах к страховой сумме или в абсолютном значении [2].

Особенности страхования машин и оборудования

Кроме обязательных свойств, страхование машин и оборудования имеет ряд особенностей, связанных с тем, что данный вид собственности является движимым имуществом и имеет обширную классификацию. Во-первых, страхование может осуществляться по трём направлениям:

- огневое страхование;
- страхование от поломок;
- страхование рисков новой техники и технологий.

В первом случае осуществляется защита всего имущества предприятия от комплексного риска, который предполагает в качестве страхового случая пожар, удар молнии, взрыв, падение пилотируемого летательного аппарата, его частей или груза. Страховые компании возмещают убытки, которые возникли в результате непосредственного воздействия огня (пожара) на застрахованное имущество, воздействия побочных явлений (дым, тепло, давление газа или воздуха), а также убытки, которые могут быть нанесены застрахованному имуществу в результате мер, принятых для тушения пожара [3].

Во втором случае страховая защита предоставляется собственникам (пользователям) машин, механического оборудования и установок на случай их внезапного и непредвиденного выхода из строя вследствие механических поломок во время эксплуатации (а также демонтажа с целью очистки или проведения технического обслуживания, плановых, текущих, средних и капитальных ремонтов и их последующего повторного монтажа).

В третьем случае происходит страхование от непредвиденных, неблагоприятных последствий, вызванных внедрением технических и технологических новинок. Здесь имеют место опосредованные (косвенные) убытки в виде дополнительных расходов и неполученной прибыли [4]. Страхование новой техники и технологии проводится от рисков, связанных с их использованием. Особенность данного страхования заключается в возмещении потерь, возникших вследствие случайной ошибки или непреднамеренных действий лиц, которые имели необходимую профессиональную подготовку для работы с новой техникой и технологией. Если же к