Бурдонов И.Б., Косачев А.С.

Симуляция систем с отказами и разрушением

# 1. Семантика безопасного взаимодействия

Верификация программных систем на основе формальных моделей – это проверка конформности (соответствия) реализации (модели системы) заданной спецификации (модели требований). Конформность функциональна, если она определяется через взаимодействие системы с ее окружением. Семантика взаимодействия формализуется в терминах *внешних действий* и *кнопок*. Действие – это поведение реализации, наблюдаемое в ответ на внешнее воздействие. Множество действий называется алфавитом действий и обозначается **L**. Кнопка – это подмножество P⊆**L**; нажатие кнопки P моделирует воздействие на реализацию, сводящееся к разрешению выполнять любое действие из P. Наблюдаться может либо действие a∈P, либо (для некоторых кнопок) отсутствие таких действий, называемое отказом P. Семантика взаимодействия задается алфавитом **L** и двумя наборами кнопок: с наблюдением соответствующих отказов – семейство **R**⊆(**L**) и без наблюдения отказов – семейство **Q**⊆(**L**). Предполагается, что **R**∩**Q**=∅ и (∪**R**)∪(∪**Q**)=**L**.

При нажатии кнопки Q∈**Q** в общем случае неизвестно, нужно ли ждать наблюдения a∈Q, или никакого наблюдения не будет, поскольку возник ненаблюдаемый отказ Q. При правильном взаимодействии такая кнопка нажимается только, если в реализации нет отказа.

Кроме внешних действий реализация может совершать внутренние (ненаблюдаемые) действия, обозначаемые τ. Эти действия всегда разрешены. Предполагается, что любая конечная последовательность любых действий совершается за конечное время, а бесконечная – за бесконечное время. Бесконечная последовательность τ-действий («зацикливание») называется *дивергенцией* и обозначается Δ. Дивергенция сама по себе не опасна, но при попытке выхода из нее (нажатии любой кнопки), неизвестно, нужно ли ждать наблюдения или бесконечно долго будут выполняться τ-действия. Поэтому при правильном взаимодействии кнопки нажимаются только тогда, если в реализации нет дивергенции.

Мы также вводим специальное, не регулируемое кнопками, действие, называемое *разрушением* и обозначаемое γ. Оно моделирует любое нежелательное поведение системы, в том числе и ее реальное разрушение. Семантика разрушения предполагает, что правильное взаимодействие должно его избегать.

Такое правильное взаимодействие, при котором не возникает ненаблюдаемых отказов, попыток выхода из дивергенции и разрушения, называется безопасным.

# 2. LTS-модель

В качестве модели реализации и спецификации используется LTS (Labelled Transition System), определяемая как совокупность **S**=LTS(V**S**,**L**,E**S**,s0), где V**S** – непустое множество состояний, **L** – алфавит внешних действий, E**S**⊆V**S**×(**L**∪{τ,γ})×V**S** – множество переходов, s0∈V**S** – начальное состояние. Переход из состояния s в состояние s` по действию z обозначается szs`. Маршрут – это цепочка смежных переходов: первый переход начинается в начальном состоянии, а каждый другой переход – в конце предыдущего перехода.

Состояние *дивергентно*, если в нем начинается бесконечный τ-маршрут. Состояние *стабильно*, если из него не выходят τ- и γ-переходы. Отказ P∈**R**∪**Q** порождается стабильным состоянием, из которого нет переходов по действиям из P.

Для определения трасс (с отказами из **R**∪**Q**) LTS **S** в каждом ее стабильном состоянии добавляются виртуальные петли sPs, помеченные порождаемыми отказами, и Δ-петли в дивергентных состояниях sΔs. Затем рассматриваются маршруты, не продолжающиеся после Δ- и γ-переходов, и трассой называется последовательность пометок на переходах такого маршрута с пропуском символов τ. Обозначим для s,s`∈V**S**, u∈**L**∪**R**∪**Q**∪{γ,Δ}, σ=u1,…,un∈(**L**∪**R**∪**Q**∪{γ,Δ})\*:

ss` s=s` ∨ ∃s1,…,sn s=s1τs2τ…τsn=s`,

sus` ∃s1,s2 ss1us2s`,

sσs` ∃s1,…,sn+1 s=s1u1s2…snunsn+1=s`,

sσ ∃s` sσs`,

sσ ¬(sσ),

s ***after*** σ {s`|sσs`}.

# 3. Слабая симуляция

Общая теория трассовой конформности, основанной на трассах реализации и спецификации, развита в работах авторов [,2,]. Однако в литературе рассматриваются также симуляции – конформности, основанные на соответствии R состояний реализации и спецификации ([4]). Целью данной работы является распространение общего подхода, учитывающего отказы и разрушение, на симуляции. Требуется, чтобы каждое наблюдение u, возможное в реализационном состоянии i с постсостоянием i`, было возможно в каждом соответствующем ему спецификационном состоянии s, и в спецификации для s и u нашлось бы постсостояние s`, соответствующее i`. Разные симуляции отличаются друг от друга, главным образом, отношением к наблюдаемости внутренних действий (τ). В данной статье мы исходим из основного допущения о принципиальной ненаблюдаемости τ-действий: при взаимодействии невозможно различить наличие и отсутствие τ-действий как до, так и после внешнего действия. Этому соответствует слабая симуляция (weak simulation), называемая также наблюдаемой симуляцией (observation simulation). Дадим три эквивалентных определения слабой симуляции (два первых принадлежат Милнеру [,]) .

**I**≤1ws**S** ∃R⊆V**I**×V**S** (i0,s0)∈R & ∀(i,s)∈R ∀σ∈**L**\* ∀i`

(iσi` ⇒ ∃s` sσs` & (i`,s`)∈R) (рис. 1 слева).

**I**≤2ws**S** ∃R⊆V**I**×V**S** (i0,s0)∈R & ∀(i,s)∈R ∀u∈**L** ∀i`

(iτi` ⇒ ∃s` ss` & (i`,s`)∈R) &

(iui` ⇒ ∃s` sus` & (i`,s`)∈R) (рис. 1в центре).

**I**≤3ws**S** ∃R⊆V**I**×V**S** (i0,s0)∈R & ∀(i,s)∈R ∀u∈**L** ∀i`

(iui` ⇒ ∃s` sus` & (i`,s`)∈R) (рис. 1справа).

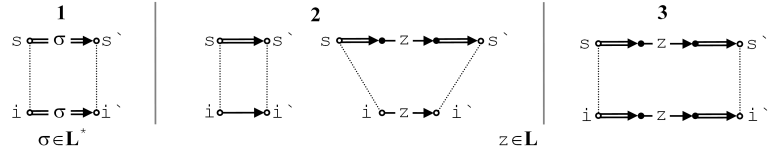


Рис. 1. Три определения слабой симуляции

Соответствие R, для которого выполнены условия слабой симуляции, называется *конформным соответствием*.

**Отказы.** Если под наблюдениями понимать не только внешние действия из **L**, но и наблюдаемые отказы из **R**, то модификация слабой симуляции с отказами выглядит так (изменения по сравнению c 3ws подчеркнуты волнистой линией):

**I**≤4ws**S** ∃R⊆V**I**×V**S** (i0,s0)∈R & ∀(i,s)∈R ∀u∈**L**∪**R** ∀i`

(iui` ⇒ ∃s` sus` & (i`,s`)∈R).

На классе реализаций без наблюдаемых отказов эти соответствия совпадают: ≤4ws = ≤3ws.

**Безопасность.** Состояние s назовем *безопасным*, если в этом состоянии не начинается γ-трасса: sγ. При безопасном взаимодействии проходятся только безопасные состояния реализации. Кнопку P∈**R**∪**Q** назовем *безопасной в (безопасном) состоянии* s, если ее можно нажимать при безопасном взаимодействии:

P ***safe*** s sγ & sΔ & (P∈**Q** ⇒ sP) & ∀z∈P sz,γ.

Наблюдение *безопасно*, если оно разрешается безопасной кнопкой. Состояние *безопасно достижимо*, если оно достижимо из начального состояния последовательностью нажатий безопасных кнопок. Модификация слабой симуляции с отказами и безопасностью выглядит так (изменения по сравнению c 4ws подчеркнуты волнистой линией):

**I**≤5ws**S** ∃R⊆V**I**×V**S** (s0γ ⇒ (i0,s0)∈R)

& ∀(i,s)∈R ∀P ***safe*** i ∀u∈P∪{P} ∀i`

(P ***safe*** s & iui` ⇒ ∃s` sus` & (i`,s`)∈R).

На классе реализаций и спецификаций, в которых все отказы наблюдаемы, нет дивергенции и разрушения, эти соответствия совпадают: ≤4ws = ≤5ws.

**Гипотеза о безопасности.** Поскольку спецификация задана, по ней можно проверять условие P ***safe*** s. Условие P ***safe*** i можно проверять, если реализация также известна. В противном случае (при тестировании) судить о безопасности кнопок в состояниях реализации мы можем только на основании некоторой *гипотезы о безопасности*. Эта гипотеза основана на соответствии H⊆V**I**×V**S** состояний реализации и спецификации, и называется H-*гипотезой*. Она предполагает 1) безопасность начального состояния i0 реализации, если безопасно начальное состояние s0 спецификации, 2) безопасность кнопки в состоянии реализации, если она безопасна хотя бы в одном соответствующем по H состоянии спецификации.

Определим соответствие H рекурсивно. Начальные состояния соответствуют друг другу, если они оба безопасны; тогда соответствуют друг другу любые два состояния, достижимые из начальных состояний по пустой трассе. Состояния i` и s` соответствуют друг другу, если они достижимы из соответствующих друг другу состояний i и s по наблюдению u, разрешаемому кнопкой P, безопасной в обоих состояниях i и s. Соответствие H – это минимальное соответствие, порождаемое следующими правилами вывода:

∀i,i`∈V**I** ∀s,s`∈V**S** ∀P∈**R**∪**Q** ∀u∈P∪{P}

s0γ & i0γ & i0i & s0s (i,s)∈H,

(i,s)∈H & P ***safe*** i & P ***safe*** s & iui` & sus` (i`,s`)∈H.

Кнопку P будем называть H-*безопасной* в реализационном состоянии i, если она безопасна хотя бы в одном соответствующем i спецификационном состоянии s:

P H-***safe*** i ∃s (i,s)∈H & P ***safe*** s.

Теперь дадим формальное определение H-гипотезы:

**I** H-***safe*** **S**

(s0γ ⇒ i0γ) & ∀i∈V**I** ∀P∈**R**∪**Q** (P H-***safe*** i ⇒ P ***safe*** i).

**Безопасная симуляция.** Соединив H-гипотезу о безопасности и слабую симуляцию, получаем вариант слабой симуляции с отказами и безопасностью (изменения по сравнению c 5ws подчеркнуты волнистой линией), которую будем называть *безопасной симуляцией* и обозначать ***ss***:

**I** ***ss*** **S** **I** H-***safe*** **S** & ∃R⊆V**I**×V**S** (s0γ ⇒ (i0,s0)∈R)

& ∀(i,s)∈R ∀P H-***safe*** i ∀u∈P∪{P} ∀i`

(P ***safe*** s & iui` ⇒ ∃s` sus` & (i`,s`)∈R).

Отношение ***ss*** транзитивно и на классе спецификаций, удовлетворяющих собственной H-гипотезе, рефлексивно, то есть является предпорядком.

Если реализация задана явно, то можно аналитически проверять как H-гипотезу, так и безопасную симуляцию. Когда реализация неизвестна, требуется тестирование, а H-гипотеза становится предусловием безопасности тестирования. Если s0γ, то H=∅, безопасное тестирование невозможно, но и не нужно, так как любая реализация конформна (при любом R). Если s0γ, то тестирование заключается в проверке *тестируемого условия* (нижние две строки определения ***ss***). Нажимается каждая кнопка P H-***safe*** i, и полученные наблюдение u и постсостояние i` верифицируются по спецификации: наблюдение u должно быть в каждом состоянии спецификации s, которое соответствует по R состоянию i, и в котором кнопка P безопасна, а среди постсостояний s` хотя бы одно должно соответствовать i` по R.

Для класса спецификаций без ненаблюдаемых отказов, дивергенции и разрушения, имеем: H-***safe*** ∩ ≤5ws = ***ss***, а на поддомене безопасных реализаций ≤5ws = ***ss***.

Для конформного по ***ss*** соответствия R соответствие R∩H тоже конформно. Мы можем переформулировать определение безопасной симуляции следующим образом:

**I** ***ss*** **S** **I** H-***safe*** **S** & ∃R⊆Н (s0γ ⇒ (i0,s0)∈R)

& ∀(i,s)∈R ∀P ***safe*** s ∀u∈P∪{P} ∀i`

(iui` ⇒ ∃s` sus` & (i`,s`)∈R).

Мы можем ограничиться такими соответствиями R, которые вложены в H. Объединение конформных по ***ss*** соответствий конформно, что дает два естественных конформных соответствия: R1 – объединение всех конформных соответствий, и R2=R1∩H.

# Список литературы

1. **Бурдонов И.Б., Косачев А.С., Кулямин В.В.** Формализация тестового эксперимента // Программирование, 2007, No. 5.
2. **Бурдонов И.Б.** Теория конформности для функционального тестирования программных систем на основе формальных моделей. Диссертация на соискание ученой степени д.ф.-м.н., Москва, 2008.

http://www.ispras.ru/~RedVerst/RedVerst/Publications/TR-01-2007.pdf

1. **Бурдонов И.Б., Косачев А.С.** Полное тестирование с открытым состоянием ограниченно недетерминированных систем // Программирование, 2009, No. 6.
2. **van Glabbeek R.J.** The linear time - branching time spectrum II; the semantics of sequential processes with silent moves. Proceedings CONCUR ’93, Hildesheim, Germany, August 1993 (E. Best, ed.), LNCS 715, Springer-Verlag, 1993, pp. 66-81.
3. **Milner R.** Lectures on a calculus for communicating systems. Seminar on Concurrency, LNCS 197, Springer-Verlag, pp. 197-220.
4. **Milner R.** Communication and Concurrency, Prentice-Hall International, Englewood Cliffs, 1989.