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 $\mathbf{L}$. Кнопка – это подмножество $P \subseteq \mathbf{L}$; нажатие кнопки P моделирует воздействие на реализацию, сводящееся к разрешению выполнять любое действие из P. Наблюдаться может либо действие $a \in P$, либо (для некоторых кнопок) отсутствие таких действий, называемое отказом P. Семантика взаимодействия задается алфавитом $\mathbf{L}$ и двумя наборами кнопок: с наблюдением соответствующих отказов – семейство $\mathbf{R} \subseteq \mathcal{P}(\mathbf{L})$ и без наблюдения отказов – семейство $\mathbf{Q} \subseteq \mathcal{P}(\mathbf{L})$. Предполагается, что $\mathbf{R} \cap \mathbf{Q} = \varnothing$ и $(\cup \mathbf{R}) \cup (\cup \mathbf{Q}) = \mathbf{L}$.

При нажатии кнопки $Q \in \mathbf{Q}$ в общем случае неизвестно, нужно ли ждать наблюдения $a \in Q$, 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 $\mathbf{S}=\mathtt{LTS}(V_\mathbf{S}, \mathbf{L}, E_\mathbf{S}, s_0)$, где $V_\mathbf{S}$ – непустое множество состояний, $\mathbf{L}$ – алфавит внешних действий, $E_\mathbf{S} \subseteq V_\mathbf{S} \times (\mathbf{L} \cup \{\tau, \gamma\}) \times V_\mathbf{S}$ – множество переходов, $s_0 \in V_\mathbf{S}$ – начальное состояние. Переход из состояния s в состояние $s`$ по действию $z$ обозначается $s \text{—} z \rightarrow s`$. 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 $P \in \mathbf{R} \cup \mathbf{Q}$ порождается стабильным состоянием, из которого нет переходов по действиям из P.

Для определения трасс (с отказами из $\mathbf{R} \cup \mathbf{Q}$) LTS $\mathbf{S}$ в каждом ее стабильном состоянии добавляются виртуальные петли $s \text{—} P \rightarrow s$, помеченные порождаемыми отказами, и Δ-петли в дивергентных состояниях $s \text{—} \Delta \rightarrow s$. 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 $s, s` \in V_\mathbf{S}$, $u \in \mathbf{L} \cup \mathbf{R} \cup \mathbf{Q} \cup \{\gamma, \Delta\}$, $\sigma = \langle u_1, \ldots, u_n \rangle \in (\mathbf{L} \cup \mathbf{R} \cup \mathbf{Q} \cup \{\gamma, \Delta\})^*$:

$$s \Rightarrow s` \quad \triangleq s = s` \vee \exists s_1, \ldots, s_n \; s = s_1 \text{—}\tau\rightarrow s_2 \text{—}\tau\rightarrow \ldots \text{—}\tau\rightarrow s_n = s`,$$

$$s = \langle u \rangle \Rightarrow s` \quad \triangleq \exists s_1, s_2 \; s \Rightarrow s_1 \text{—} u \rightarrow s_2 \Rightarrow s`,$$

$$s = \sigma \Rightarrow s` \quad \triangleq \exists s_1, \ldots, s_{n+1} \; s = s_1 = \langle u_1 \rangle \Rightarrow s_2 \ldots s_n = \langle u_n \rangle \Rightarrow s_{n+1} = s`,$$

$$s = \sigma \Rightarrow \quad \triangleq \exists s` \; s = \sigma \Rightarrow s`,$$

$s{=}\sigma{\not\Rightarrow} \quad \triangleq \neg(s{=}\sigma{\Rightarrow})$,

s ***after*** $\sigma \triangleq \{s` \mid s{=}\sigma{\Rightarrow}s`\}$.

3. СЛАБАЯ СИМУЛЯЦИЯ

Общая теория трассовой конформности, основанной на трассах реализации и спецификации, развита в работах авторов [1,2,3]. 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 [5,6]) .

$\mathbf{I}{\le}^1_{ws}\mathbf{S} \triangleq \exists R{\subseteq}V_{\mathbf{I}}{\times}V_{\mathbf{S}}\ (i_0,s_0){\in}R\ \&\ \forall(i,s){\in}R\ \forall\sigma{\in}\mathbf{L}^*\ \forall i`$

$\quad (i{=}\sigma{\Rightarrow}i` \Rightarrow \exists s`\ s{=}\sigma{\Rightarrow}s`\ \&\ (i`,s`){\in}R)$ (рис. 1 слева).

$\mathbf{I}{\le}^2_{ws}\mathbf{S} \triangleq \exists R{\subseteq}V_{\mathbf{I}}{\times}V_{\mathbf{S}}\ (i_0,s_0){\in}R\ \&\ \forall(i,s){\in}R\ \forall u{\in}\mathbf{L}\ \forall i`$

$\quad (i{-}\tau{\rightarrow}i` \Rightarrow \exists s`\ s{\Rightarrow}s`\ \&\ (i`,s`){\in}R)\ \&$

$\quad (i{-}u{\rightarrow}i` \Rightarrow \exists s`\ s{=}\langle u\rangle{\Rightarrow}s`\ \&\ (i`,s`){\in}R)$ (рис. 1в центре).

$\mathbf{I}{\le}^3_{ws}\mathbf{S} \triangleq \exists R{\subseteq}V_{\mathbf{I}}{\times}V_{\mathbf{S}}\ (i_0,s_0){\in}R\ \&\ \forall(i,s){\in}R\ \forall u{\in}\mathbf{L}\ \forall i`$

$\quad (i{=}\langle u\rangle{\Rightarrow}i` \Rightarrow \exists s`\ s{=}\langle u\rangle{\Rightarrow}s`\ \&\ (i`,s`){\in}R)$ (рис. 1справа).

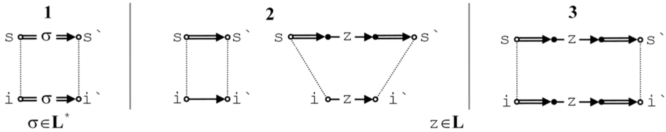

Рис. 1. Три определения слабой симуляции

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 с $\leqslant^3_{ws}$ подчеркнуты волнистой линией):

$\mathbf{I}\le^4_{ws}\mathbf{S} \triangleq \exists R\subseteq V_{\mathbf{I}}\times V_{\mathbf{S}}\ (i_0,s_0)\in R\ \&\ \forall (i,s)\in R\ \forall u\in \mathbf{L}\cup\mathbf{R}\ \forall i`$

$(i=\langle u\rangle\Rightarrow i`\ \Rightarrow \exists s`\ s=\langle u\rangle\Rightarrow s`\ \&\ (i`,s`)\in R)$.

На классе реализаций без наблюдаемых отказов эти соответствия совпадают: $\le^4_{ws} = \le^3_{ws}$.

Безопасность. Состояние s назовем *безопасным*, если в этом состоянии не начинается γ-трасса: $s=\langle\gamma\rangle\not\Rightarrow$. При безопасном взаимодействии проходятся только безопасные состояния реализации. Кнопку $P\in\mathbf{R}\cup\mathbf{Q}$ назовем *безопасной в (безопасном) состоянии* s, если ее можно нажимать при безопасном взаимодействии:

$P\ \textbf{\textit{safe}}\ s \triangleq s=\langle\gamma\rangle\not\Rightarrow\ \&\ s=\langle\Delta\rangle\not\Rightarrow\ \&\ (P\in\mathbf{Q}\Rightarrow s=\langle P\rangle\not\Rightarrow)\ \&\ \forall z\in P\ s=\langle z,\gamma\rangle\not\Rightarrow$.

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 с $\leqslant^4_{ws}$ подчеркнуты волнистой линией):

$\mathbf{I}\le^5_{ws}\mathbf{S} \triangleq \exists R\subseteq V_{\mathbf{I}}\times V_{\mathbf{S}}\ (s_0=\langle\gamma\rangle\not\Rightarrow\ \Rightarrow (i_0,s_0)\in R)$

$\&\ \forall (i,s)\in R\ \forall P\ \textbf{\textit{safe}}\ i\ \forall u\in P\cup\{P\}\ \forall i`$

$(P\ \textbf{\textit{safe}}\ s\ \&\ i=\langle u\rangle\Rightarrow i`\ \Rightarrow \exists s`\ s=\langle u\rangle\Rightarrow s`\ \&\ (i`,s`)\in R)$.

На классе реализаций и спецификаций, в которых все отказы наблюдаемы, нет дивергенции и разрушения, эти соответствия совпадают: $\le^4_{ws} = \le^5_{ws}$.

Гипотеза о безопасности. Поскольку спецификация задана, по ней можно проверять условие $\mathtt{P}$ ***safe*** $\mathtt{s}$. Условие $\mathtt{P}$ ***safe*** $\mathtt{i}$ 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 $\mathtt{H}\subseteq \mathtt{V}_{\mathbf{I}}\times\mathtt{V}_{\mathbf{S}}$ состояний реализации и спецификации, и называется $\mathtt{H}$-*гипотезой*. Она предполагает 1) безопасность начального состояния $\mathtt{i}_0$ реализации, если безопасно начальное состояние $\mathtt{s}_0$ спецификации, 2) безопасность кнопки в состоянии реализации, если она безопасна хотя бы в одном соответствующем по $\mathtt{H}$ состоянии спецификации.

Определим соответствие $\mathtt{H}$ 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 $\mathtt{i'}$ и $\mathtt{s'}$ соответствуют друг другу, если они достижимы из соответствующих друг другу состояний $\mathtt{i}$ и $\mathtt{s}$ по наблюдению $\mathtt{u}$, разрешаемому кнопкой $\mathtt{P}$, безопасной в обоих состояниях $\mathtt{i}$ и $\mathtt{s}$. Соответствие $\mathtt{H}$ – это минимальное соответствие, порождаемое следующими правилами вывода:

$$\forall \mathtt{i},\mathtt{i'}\in \mathtt{V}_{\mathbf{I}}\ \forall \mathtt{s},\mathtt{s'}\in \mathtt{V}_{\mathbf{S}}\ \forall \mathtt{P}\in\mathbf{R}\cup\mathbf{Q}\ \forall \mathtt{u}\in \mathtt{P}\cup\{\mathtt{P}\}$$

$$\mathtt{s}_0=\langle\gamma\rangle\not\Rightarrow\ \&\ \mathtt{i}_0=\langle\gamma\rangle\not\Rightarrow\ \&\ \mathtt{i}_0\Rightarrow \mathtt{i}\ \&\ \mathtt{s}_0\Rightarrow \mathtt{s} \qquad \vdash (\mathtt{i},\mathtt{s})\in \mathtt{H},$$

$$(\mathtt{i},\mathtt{s})\in \mathtt{H}\ \&\ \mathtt{P}\ \textbf{\textit{safe}}\ \mathtt{i}\ \&\ \mathtt{P}\ \textbf{\textit{safe}}\ \mathtt{s}\ \&\ \mathtt{i}=\langle \mathtt{u}\rangle\Rightarrow \mathtt{i'}\ \&\ \mathtt{s}=\langle \mathtt{u}\rangle\Rightarrow \mathtt{s'}\ \vdash (\mathtt{i'},\mathtt{s'})\in \mathtt{H}.$$

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

$$\mathtt{P}\ \mathtt{H}\text{-}\textbf{\textit{safe}}\ \mathtt{i} \triangleq \exists \mathtt{s}\ (\mathtt{i},\mathtt{s})\in \mathtt{H}\ \&\ \mathtt{P}\ \textbf{\textit{safe}}\ \mathtt{s}.$$

Теперь дадим формальное определение $\mathtt{H}$-гипотезы:

$$\mathbf{I}\ \mathtt{H}\text{-}\textbf{\textit{safe}}\ \mathbf{S} \triangleq$$

$$(\mathtt{s}_0=\langle\gamma\rangle\not\Rightarrow\ \Rightarrow\ \mathtt{i}_0=\langle\gamma\rangle\not\Rightarrow)\ \&\ \forall \mathtt{i}\in \mathtt{V}_{\mathbf{I}}\ \forall \mathtt{P}\in\mathbf{R}\cup\mathbf{Q}\ (\mathtt{P}\ \mathtt{H}\text{-}\textbf{\textit{safe}}\ \mathtt{i}\Rightarrow \mathtt{P}\ \textbf{\textit{safe}}\ \mathtt{i}).$$

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

$$\mathbf{I}\ \textbf{\textit{ss}}\ \mathbf{S} \triangleq \underline{\mathbf{I}\ \mathtt{H}\text{-}\textbf{\textit{safe}}\ \mathbf{S}\ \&}\ \exists \mathtt{R}\subseteq \mathtt{V}_{\mathbf{I}}\times \mathtt{V}_{\mathbf{S}}\ (\mathtt{s}_0=\langle\gamma\rangle\not\Rightarrow\ \Rightarrow\ (\mathtt{i}_0,\mathtt{s}_0)\in \mathtt{R})$$

$$\&\ \forall(\mathtt{i},\mathtt{s})\in \mathtt{R}\ \forall \mathtt{P}\ \underline{\mathtt{H}\text{-}\textbf{\textit{safe}}}\ \mathtt{i}\ \forall \mathtt{u}\in\underline{\mathtt{P}\cup\{\mathtt{P}\}}\ \forall \mathtt{i'}$$

$(P\ \textbf{\textit{safe}}\ s\ \&\ i=\langle u\rangle\Rightarrow i` \Rightarrow \exists s`\ s=\langle u\rangle\Rightarrow s`\ \&\ (i`,s`)\in R)$.

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 $s_0=\langle\gamma\rangle\Rightarrow$, то $H=\varnothing$, безопасное тестирование невозможно, но и не нужно, так как любая реализация конформна (при любом R). Если $s_0=\langle\gamma\rangle\not\Rightarrow$, 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*** $\cap \le^5_{ws}$ = ***ss***, а на поддомене безопасных реализаций $\le^5_{ws}$ = ***ss***.

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

$$\mathbf{I}\ \textbf{\textit{ss}}\ \mathbf{S} \triangleq \mathbf{I}\ H\text{−}\textbf{\textit{safe}}\ \mathbf{S}\ \&\ \exists R\subseteq H\ (s_0=\langle\gamma\rangle\not\Rightarrow\ \Rightarrow (i_0,s_0)\in R)$$
$$\&\ \forall(i,s)\in R\ \forall P\ \textbf{\textit{safe}}\ s\ \forall u\in P\cup\{P\}\ \forall i`$$
$$(i=\langle u\rangle\Rightarrow i` \Rightarrow \exists s`\ s=\langle u\rangle\Rightarrow s`\ \&\ (i`,s`)\in R).$$

Мы можем ограничиться такими соответствиями R, которые вложены в H. Объединение конформных по ***ss*** соответствий конформно, что дает два естественных конформных соответствия: R_1 – объединение всех конформных соответствий, и $R_2=R_1\cap H$.

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
3. **Бурдонов И.Б., Косачев А.С.** Полное тестирование с открытым состоянием ограниченно недетерминированных систем // Программирование, 2009, No. 6.
4. **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.
5. **Milner R.** Lectures on a calculus for communicating systems. Seminar on Concurrency, LNCS 197, Springer-Verlag, pp. 197-220.
6. **Milner R.** Communication and Concurrency, Prentice-Hall International, Englewood Cliffs, 1989.