Финальные модели спецификации

Игорь Бурдонов <igor@ispras.ru>,
Александр Косачев kos@ispras.ru

Аннотация. Работа посвящена исследованию формальных методов тестирования соответствия (конформности) исследуемой системы требованиям, заданным в форме спецификации. Такое тестирование основано на семантике взаимодействия, которая определяет тестовые возможности по управлению (заданный набор тестовых воздействий) и наблюдению действий и отказов (отсутствие действий). Допускаются также ненаблюдаемые (внутренние) действия. Семантика параметризуется семействами наблюдаемых и ненаблюдаемых отказов. Вводится разрушение – запрещённое действие, которого следует избегать при взаимодействии. Определяется понятие безопасного тестирования, при котором не возникают ненаблюдаемые отказы и разрушение, и тестовые воздействия не подаются при дивергенции (бесконечной последовательности ненаблюдаемых действий). На этой основе определяются реализационная гипотеза о безопасности и безопасная конформность, а также генерация полного набора тестов по спецификации.
В работе исследуются различные модели для описания спецификационных требований. Наиболее распространенной моделью является система помеченных переходов – LTS (Labelled Transition System). В то же время для рассматриваемой семантики взаимодействия существенны только трассы (последовательности наблюдений), но не состояния LTS. Поэтому естественной оказывается трассовая модель как множество трасс LTS. Такая семантика позволяет определять только конформности типа редукции, в отличие от конформностей типа симуляций, для проверки которых требуется дополнительная тестовая возможность – опрос состояния реализации [10],[11],[12],[19],[22]. При безопасном тестировании тесты генерируются по безопасным трассам спецификации, для прохождения которых используются только безопасные тестовые воздействия.
Целью данной работы является выделение подмножества трасс спецификации, достаточного для генерации полного набора тестов. Такое подмножество мы назвали финальной трассовой моделью спецификации. С другой стороны, LTS-модель удобна тем, что является способом конечного представления регулярных множеств трасс. Для представления финальной трассовой модели в работе предлагается разновидность LTS, названная финальной RTS (Refusal Transition System). Переходы по наблюдаемым отказам задаются явно (эти отказы входят в алфавит RTS). Такая модель обладает целым рядом полезных для генерации тестов свойств: 1) она детерминирована, 2) трасса наблюдений безопасна тогда и только тогда, когда она заканчивается в нетерминальном состоянии, где нет разрушения, 3) тестовое воздействие безопасно после трассы тогда и только тогда, когда оно безопасно в конечном состоянии трассы, то есть в этом состоянии нет дивергенции, тестовое воздействие не вызывает разрушения и ненаблюдаемых отказов .
В работе предложены алгоритмы преобразования LTS-модели в финальную RTS-модель и определены достаточные условия построения конечной RTS за конечное время.

Ключевые слова: Семантика взаимодействия, отказы, разрушение, дивергенция, конформность, безопасное тестирование, трассы, LTS, генерация тестов.

1. Теория конформности

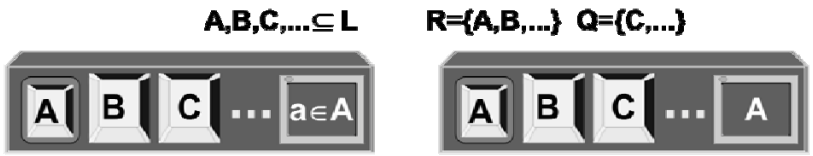
1.1. Семантика взаимодействия и безопасное тестирование

Верификация конформности понимается как проверка соответствия исследуемой системы заданным требованиям. В модельном мире система отображается в реализационную модель, требования – в спецификационную модель, а их соответствие – в бинарное отношение конформности. Если требования выражены в терминах взаимодействия системы с окружающим миром, возможно тестирование как проверка конформности в процессе тестовых экспериментов, когда тест подменяет собой окружение системы. В этом случае само отношение конформности и его тестирование основаны на той или иной модели взаимодействия.

Мы рассматриваем семантики взаимодействия, которые определяются только внешним, наблюдаемым поведением системы и не учитывают её внутреннее устройство, которое на уровне модели отображается понятием *состояния*. Мы можем наблюдать только такое поведение реализации, которое, во-первых, «спровоцировано» тестом (управление) и, во-вторых, наблюдаемо во внешнем взаимодействии. Такое взаимодействие может моделироваться с помощью, так называемой, машины тестирования [2],[4],[5],[20],[21],[23]. Она представляет собой «чёрный ящик», внутри которого находится реализация (рис. 1). Управление сводится к тому, что оператор машины, выполняя тест (понимаемый как инструкция оператору), нажимает кнопки на клавиатуре машины, «разрешая» реализации выполнять те или иные действия, которые могут им наблюдаться. Наблюдения (на «дисплее» машины) бывают двух типов: наблюдение некоторого *внешнего (наблюдаемого) действия*, разрешённого оператором и выполняемого реализацией, и наблюдение *отказа* как отсутствия каких бы то ни было наблюдаемых действий из числа тех, что разрешены нажатой кнопкой.

Подчеркнём, что при управлении оператор разрешает реализации выполнять именно множество действий, а не обязательно одно действие. Будем считать, что каждой кнопке соответствует своё множество разрешаемых действий, и оператор нажимает только одну кнопку. После наблюдения (действия или отказа) кнопка отжимается, и все внешние действия запрещаются. Далее оператор может нажать другую (или ту же самую) кнопку.

Кроме внешних действий реализация может совершать *внутренние* (*ненаблюдаемые*) действия, обозначаемые символом τ, которые считаются всегда разрешенными.



Машина тестирования

Тестовые возможности определяются тем, какие «кнопочные» множества есть в машине, а также для каких кнопок возможно наблюдение отказа. Тем самым, семантика взаимодействия определяется алфавитом внешних действий **L** и двумя наборами кнопок (соответствующих им множеств действий) машины тестирования: с наблюдением соответствующих отказов – семейство $\mathbf{R}\subseteq P(\mathbf{L})$ и без наблюдения отказа – семейство $\mathbf{Q}\subseteq P(\mathbf{L})$. Предполагается, что $\mathbf{L}\cap P(\mathbf{L})=\varnothing$ (действия и отказы «не путаются» между собой), $\mathbf{R}\cap\mathbf{Q}=\varnothing$ (отказ либо наблюдаемый, либо ненаблюдаемый), $\cup\mathbf{R}\cup\cup\mathbf{Q}=\mathbf{L}$ (каждое внешнее действие разрешается хотя бы одной кнопкой и каждая кнопка разрешает действия только из алфавита внешних действий). Такую семантику мы называем **R**/**Q**-семантикой [2],[5],[7],[9],[13],[14],[15].

Множество наблюдений (действий и отказов), разрешаемых кнопкой P, обозначим ***obs***(P) $P\cup\{P|P\in\mathbf{R}\}$. Наоборот, множество кнопок, разрешающих наблюдение u (действие и отказ), обозначим ***but***(u) $\{P\in\mathbf{R}\cup\mathbf{Q}|u\in\mathbf{obs}(P)\}$.

В качестве примера можно привести хорошо известную семантику отношения ***ioco*** [24],[25], в которой действия делятся на стимулы и реакции. Каждый стимул x разрешается одной **Q**-кнопкой {x}, а все реакции разрешаются одной **R**-кнопкой, обозначаемой δ.

Для выполнимости любого действия необходимо, чтобы оно было определено в реализации и разрешено оператором. Если этого условия также и достаточно, то есть на выполнение выбирается любое из таких действий недетерминированным образом, то говорят, что в системе нет приоритетов [21]. Здесь мы ограничимся системами без приоритетов. Тестирование систем с приоритетами рассмотрено в наших работах [6],[8].

Предполагается, что любая конечная последовательность любых действий (как внешних, так и внутренних) совершается за конечное время, а бесконечная – за бесконечное время. Также предполагается, что «передача» тестового воздействия (нажатие кнопки) от машины тестирования в реализацию и наблюдения от реализации на дисплей машины выполняются за конечное время.

Бесконечная последовательность τ-действий называется *дивергенцией* и обозначается символом Δ. Дивергенция сама по себе не опасна, но при попытке выхода из неё, когда оператор нажимает любую кнопку, получение наблюдения не гарантируется, поскольку реализация может бесконечно долго

выполнять только внутренние (ненаблюдаемые) действия. Поэтому оператор не может ни продолжать тестирование, ни закончить его.

При отсутствии дивергенции указанные выше предположения гарантируют, что если после нажатия кнопки реализация выполняет некоторое внешнее действие, разрешаемое этой кнопкой, то через конечное время это действие будет наблюдаться на дисплее машины тестирования.

Эти же предположения часто используются для реализации наблюдения **R**-отказа при отсутствии дивергенции, но в усиленном варианте: время выполнения каждого действия, разрешаемого кнопкой, вместе с возможными предшествующими ему внутренними действиями не только конечно, но и ограничено. В этом случае вводится тайм-аут, истечение которого без наблюдения действия трактуется как отказ (при условии отсутствия дивергенции). Важно отметить, что это не единственный возможный способ реализации наблюдения отказа.

В любом случае предполагается, что при отсутствии дивергенции после нажатия **R**-кнопки через конечное время оператор наблюдает или разрешенное этой кнопкой внешнее действие или соответствующий отказ. Однако при нажатии **Q**-кнопки, если в реализации возможен отказ, то, поскольку этот отказ не наблюдаем, оператор не знает, нужно ли ему ждать наблюдения внешнего действия или такого действия не будет, поскольку возник отказ. Поэтому оператор не может ни продолжать тестирование, ни закончить его.

Кроме этого мы вводим [1],[2],[3],[4],[5] специальное, всегда разрешенное действие, которое называем *разрушением* и обозначаем символом γ. Оно моделирует любое поведение реализации, которое не должно допускаться во время тестирования.

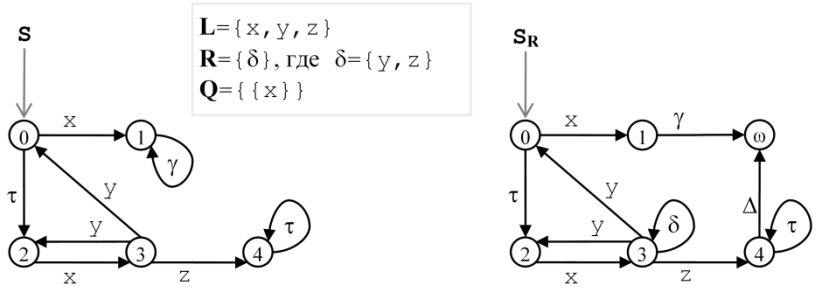
Тестирование, при котором не возникает попыток выхода из дивергенции, ненаблюдаемых отказов и разрушения, называется *безопасным* [2],[5][1].

2. LTS-модель

В качестве основной модели мы используем *систему помеченных переходов* (LTS – Labelled Transition System) – ориентированный граф с выделенной начальной вершиной, дуги которого помечены некоторыми символами. Формально, LTS – это совокупность $\mathbf{S}=LTS(V_S,\mathbf{L},E_S,s_0)$, где V_S – непустое множество состояний (вершин графа), **L** – алфавит внешних действий, $E_S\subseteq V_S\times(\mathbf{L}\cup\{\tau,\gamma\})\times V_S$ – множество переходов (помеченных дуг графа), $s_0\in V_S$ – начальное состояние (начальная вершина графа).

Пример LTS (в ***ioco***-семантике) приведен на рис. 2 слева.

[1] В наших более ранних работах [1][2][3][4] опасной считалась сама дивергенция, а не попытка выхода из нее.



$$tr(S_R) = (\delta^* xy)^* (\delta^* [x[z[\Delta]]] \mid x\gamma)$$

Рис. 2. Пример LTS и ее $\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$*-трасс*[2]

Множество всех LTS в алфавите **L** обозначим ***LTS***(**L**). Также введем обозначения для наличия/отсутствия переходов:

$s{-}z{\to}s'$ $(s,z,s')\in E_S$, $s{-}z\ s'$ $\neg(s{-}z{\to}s')$,

$s{-}z{\to}$ $\exists s'\ s{-}z{\to}s'$, $s{-}z$ $\neg(s{-}z{\to})$.

Там, где это не приводит к недоразумениям, будем использовать запись $s{-}z{\to}s'$ для обозначения самого перехода (s,z,s'), а не как предикат $(s,z,s')\in E_S$.

Маршрутом LTS называется последовательность смежных переходов: конец каждого перехода, кроме последнего, совпадает с началом следующего перехода. В машине тестирования выполнение LTS сводится к прохождению маршрута, каждый переход которого разрешается нажатой кнопкой. После выполнения перехода по внешнему действию, разрешаемому нажатой кнопкой, или после возникновения **R**-отказа кнопка автоматически отжимается. При этом τ- и γ-переходы всегда разрешены.

Состояние s` *достижимо из состояния* s, если в s` заканчивается маршрут, начинающийся в состоянии s. Просто *достижимое состояние* – это состояние, достижимое из начального состояния s_0. Множество состояний, достижимых из состояния s, будем обозначать ***der***(s). Определим ***der***(**S**)≜***der***(s_0).

Состояние *терминально*, если из него не выходят никакие переходы, и *стабильно*, если из него не выходят τ- и γ-переходы. Отказ P порождается стабильным состоянием, из которого не выходят переходы по действиям из P. Состояние *дивергентно*, если в нем начинается бесконечный τ-маршрут (маршрут из τ-переходов); в противном случае состояние *конвергентно*.

[2] При записи регулярных выражений мы для краткости будем опускать знаки конкатенации «·» для последовательностей и множеств последовательностей, запятые «,», разделяющие элементы последовательности, угловые скобки «⟨» и «⟩», отмечающие начало и конец последовательности, а также фигурные скобки «{» и «}» для множеств. Например, регулярное выражение на данном рисунке, записанное в краткой форме как $(\delta^* xy)^*(\delta^*[x[z[\Delta]]] \mid x\gamma)$, в полной форме записывается так:

$(\{\langle\delta\rangle\}^* \cdot \{\langle x,y\rangle\})^* \cdot (\{\langle\delta\rangle\}^* \cdot \{[\langle x\rangle \cdot [\langle z\rangle \cdot [\langle\Delta\rangle]]]\} \mid \{\langle x,\gamma\rangle\})$.

3. Тестовые истории и трассы LTS

3.1. Тестовые истории

Для взаимодействия, основанного на наблюдениях, единственным результатом тестового эксперимента является чередующаяся последовательность кнопок (тестовых воздействий) и наблюдений, которую будем называть (тестовой) *историей*. Дивергенция и разрушение считаются условно-наблюдаемыми действиями: хотя они не должны возникать при безопасном тестировании, но для полноты моделирования должны присутствовать в историях, отмечая те их них, после которых возможны дивергенция или разрушение. Так как нас не интересует поведение системы после дивергенции или разрушения, символы Δ и γ могут быть только последними элементами историй. Любое другое наблюдение u (внешнее действие или **R**-отказ) должно разрешаться непосредственно предшествующей ему в истории кнопкой P∈***but***(u). Заметим, что по этому определению любой префикс истории является историей.

Подпоследовательность истории, состоящая только из наблюдений (включая Δ и γ), называется *трассой* (этой истории). В общем случае будем называть ***A***-трассой последовательность в алфавите ***A***. Трасса истории – это $\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$-трасса.

Для систем без приоритетов важны только трассы, поскольку возможность или невозможность появления данного наблюдения после некоторой трассы определяется только тем, что нажимаемая кнопка разрешает данное наблюдение, и не зависит от того, какие еще наблюдения она разрешает или запрещает. Для данной тестируемой системы без приоритетов множество ее историй однозначно восстанавливается по множеству ее трасс.

Как было сказано выше, в машине тестирования выполнение LTS сводится к прохождению маршрута, каждый переход которого разрешается нажатой кнопкой. При этом на дисплее машины мы наблюдаем последовательность внешних действий, которыми помечены выполняемые переходы, и возникающие **R**-отказы в стабильных состояниях. Если LTS находится в конвергентном состоянии s, когда нажимается кнопка P, то через конечное время либо выполняется переход по действию z∈***obs***(P), то есть z∈P, либо происходит отказ P. Этот отказ P наблюдается, то есть P∈***obs***(P), если P∈**R**. Поскольку τ-переходы всегда разрешены, может оказаться, что переход по

действию z выполняется не из состояния s, а из другого состояния s`, достижимого из s по цепочке τ-переходов. Аналогично, поскольку при возникновении **R**-отказа P LTS должна находиться в стабильном состоянии, это состояние совпадает с состоянием s только в том случае, когда состояние s стабильно, а в противном случае отказ происходит в стабильном состоянии s`, достижимом из s по цепочке τ-переходов. Если состояние s дивергентно, то после нажатия кнопки P никаких внешних действий и отказов может не быть, если бесконечно долго будут выполняться только τ-переходы.

Простые трассы LTS

Простой трассой LTS **S** будем называть последовательность σ пометок на переходах маршрута с пропуском символов τ. Простая трасса LTS – это **L**∪{γ}-трасса. Множество простых трасс, начинающихся (то есть маршруты которых начинаются) в состоянии s будем обозначать ***tr***(s,**S**).

Через s ***after*** σ обозначим множество состояний, в которых заканчивается простая трасса σ, начинающаяся в состоянии s, то есть заканчиваются все маршруты с простой трассой σ, начинающиеся в состоянии s. Распространим оператор ***after*** на множество A состояний обычным образом: A ***after*** σ ≜ {s ***after*** σ|s∈A}, результатом является множество множеств состояний LTS. По умолчанию, будем считать, что простая трасса начинается в начальном состоянии LTS, и обозначать **S** ***after*** σ ≜ s_0 ***after*** σ, ***tr***(**S**) ≜ ***tr***(s_0,**S**).

Введем следующие обозначения для состояний s и s` и трассы σ:

s=σ⇒s` ≜ s`∈(s ***after*** σ), s=σ⇏s` ≜ ¬(s=σ⇒s`),

s⇒s` ≜ s=$\epsilon$⇒s`, где ϵ пустая трасса, s⇏s` ≜ ¬(s⇒s`),

s=σ⇒ ≜ ∃s` s=σ⇒s`, s=σ⇏ ≜ ¬(s=σ⇒).

Распространим эти обозначения на множества состояний A и A`:

A=σ⇒A` ≜ A`=(A ***after*** σ)≠∅, A=σ⇏A` ≜ ¬(A=σ⇒A`),

A⇒A` ≜ A=$\epsilon$⇒A`, A⇏A` ≜ ¬(A⇒A`),

A=σ⇒ ≜ ∃A` A=σ⇒A`, A=σ ⇏ ≜ ¬(A=σ⇒).

LTS *детерминирована*, если каждая ее простая трасса заканчивается ровно в одном состоянии. Очевидно, что в детерминированной LTS нет τ-переходов из достижимых состояний.

L∪R∪{Δ,γ}*-трассы LTS*

Последовательность наблюдений, которая может быть получена при взаимодействии с LTS в **R**/**Q**-семантике, то есть последовательность не только действий, но и **R**-отказов, является **L∪R∪{Δ,γ}**-трассой. Для определения таких трасс LTS **S** добавим в каждом ее стабильном состоянии виртуальные петли s—R→s, помеченные **R**-отказами, порождаемыми в этом состоянии, добавим новое терминальное состояние ω, перенаправим в это состояние все γ-переходы, а также проведем в него Δ-переходы из дивергентных состояний.

Формально для любого заданного семейства множеств **R**⊆P(**L**) это преобразование **S**→$\mathbf{S_R}$ дает LTS $\mathbf{S_R}$=LTS(V_S∪{ω},**L**∪**R**∪{Δ},E_R,s_0), где ω∉V_S, а множество переходов E_R определяется как минимальное множество, порождаемое следующими правилами вывода: ∀s,s`∈V_S ∀z∈**L**∪{τ} ∀R∈**R**

s—z→s`	⊢	s—z→s`,
∀z∈R∪{τ,γ} s—z↛	⊢	s—R→s,
s—γ→	⊢	s—γ→ω,
s *дивергентно*	⊢	s—Δ→ω.

Пример LTS **S**, LTS $\mathbf{S_R}$ и множества **L∪R∪**{Δ,γ}-трасс LTS **S** приведен на рис. 2.

Формально **L∪R∪**{Δ,γ}-трассой LTS **S** будем называть простую трассу LTS $\mathbf{S_R}$. Множество ***tr***($\mathbf{S_R}$) всех **L∪R∪**{Δ,γ}-трасс LTS **S** называется *(трассовой)* **R**-*моделью*, когда по умолчанию предполагается заданным алфавит **L**, или просто *трассовой моделью*, когда по умолчанию предполагаются заданными как алфавит **L**, так и семейство **R**⊆P(**L**)). Для **R**=P(**L**) получаем *полную трассовую модель* – она содержит все отказы, порождаемые стабильными состояниями. Полная трассовая модель (как и LTS) описывает как устроена система «на самом деле», а подмножество ее **L∪R∪**{Δ,γ}-трасс, то есть трассовая **R**-модель, – это «взгляд» на систему, определяемый тестовыми возможностями по управлению и наблюдению, которые описываются **R**/**Q**-семантикой, и безопасностью тестирования (отсутствия в трассах ненаблюдаемых отказов).

Распространим LTS-обозначения, использующие трассы, на **L∪R∪**{Δ,γ}-трассы. Такими обозначениями являются: s=σ⇒s`, а также производные от этого обозначения s=σ⇒, s=σ⇏s`, s=σ⇏, s ***after*** σ и **S** ***after*** σ. Обозначение s=σ⇒s` имеет один и тот же смысл для LTS **S** и $\mathbf{S_R}$, если трасса σ содержит только внешние действия. Если σ содержит отказы или заканчивается Δ-символом, двойная стрелка, очевидно, применяется к LTS $\mathbf{S_R}$, поскольку в LTS **S** нет переходов по Δ-символу и отказам. Двусмысленность возможна лишь в том случае, когда σ не содержит отказов и символа Δ и заканчивается разрушением γ, что происходит из-за того, что все γ-переходы в LTS $\mathbf{S_R}$ перенаправляются в состояние ω. Мы будем считать, что в этом случае имеется в виду LTS $\mathbf{S_R}$. Тем самым, мы переопределяем двойную стрелку s=σ⇒s` и производные обозначения для LTS **S** и трасс без отказов и Δ-символа, заканчивающихся разрушением.

Состояния s и s` называют *эквивалентными*, если в них начинаются одни и те же простые трассы: ***tr***(s,**S**)=***tr***(s`,**S**). Множества состояний A и A` будем

называть *эквивалентными* и обозначать $A\sim_T A`$, если в их состояниях начинаются одни и те же простые трассы: $\cup\{\boldsymbol{tr}(s,\mathbf{S})|s\in A\}=\cup\{\boldsymbol{tr}(s`,\mathbf{S})|s`\in A`\}$.

4. Трассовая модель

4.1. Сравнение LTS-модели и трассовой модели

Сравним LTS-модель и трассовую модель, определенную выше как множество $\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$-трасс LTS. LTS-модель более «наглядна», чем трассовая модель. Формально это означает, что любой граф с выделенной начальной вершиной и дугами, помеченными символами из алфавита $\mathbf{L}\cup\{\tau,\gamma\}$, является LTS-моделью в алфавите **L**. В то же время далеко не любое множество $\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$-трасс является трассовой моделью: как будет показано ниже, оно должно обладать определенным набором нетривиальных свойств. Кроме того, LTS-модель является способом конечного представления регулярных трассовых моделей, в том числе бесконечных, то есть регулярных множеств последовательностей, являющихся трассовыми моделями.

Но с другой стороны трассовая модель имеет то преимущество перед LTS-моделью, что не содержит ничего «лишнего», поскольку представляет собой множество трасс, которые можно наблюдать при работе с системой в **R**/**Q**-семантике. Поэтому трассовая модель является наиболее естественной моделью такой системы. В частности, одной и той же трассовой модели может соответствовать несколько LTS с одним и тем же множеством $\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$-трасс, и эти LTS неразличимы при взаимодействии с ними в **R**/**Q**-семантике.

В этом подразделе мы дадим независимое от LTS определение трассовой модели как множества трасс, удовлетворяющего некоторому набору свойств. Иными словами, мы покажем, что множество трасс является трассовой моделью тогда и только тогда, когда оно удовлетворяет этому набору свойств. Сначала дадим несколько определений и обозначений.

1. Свойства трасс и операции над трассами

Пусть $\mathbf{R}\subseteq P(\mathbf{L})$ произвольное семейство подмножеств алфавита.

Трассу будем называть *допустимой*, если все ее элементы, кроме, быть может, последнего, отличны от дивергенции и разрушения.

Трассу будем называть *согласованной*, если после любой непустой последовательности отказов в трассе не следует ни дивергенция, ни разрушение, ни какое-либо внешнее действие, принадлежащее какому-либо отказу, входящему в эту последовательность.

Трассу будем называть **R**-*конвергентной* во множестве трасс **Σ**, если она содержит или продолжается в **Σ** разрушением или дивергенцией, а в противном случае для каждого **R**-отказа продолжается в **Σ** этим отказом или каким-либо внешним действием, принадлежащим этому отказу.

Определим три операции над трассами.

Операция $\boldsymbol{p_{re}}$ – взятие префикса трассы: $\mu\cdot\lambda\text{—}\boldsymbol{p_{re}}\!\!\rightarrow\mu$. Введем отношение «является префиксом»: $\mu\le\sigma \triangleq \exists\lambda\ \sigma=\mu\cdot\lambda$. Множество трасс будем называть *префикс-замкнутым*, если оно вместе с каждой трассой содержит все ее префиксы, то есть все трассы, получаемые из данной по $\boldsymbol{p_{re}}$-операциям. Через $\boldsymbol{p_{re}}(\sigma)$ обозначим префикс-замыкание трассы σ как множество всех ее префиксов, то есть замыкание по операции $\boldsymbol{p_{re}}$ ($\boldsymbol{p_{re}}$-замыкание) как множество трасс, получаемых из трассы σ всеми возможными применениями операции:

$\boldsymbol{p_{re}}(\sigma) \triangleq \{\mu|\mu\le\sigma\} = \{\mu|\sigma\text{—}\boldsymbol{p_{re}}\!\!\rightarrow\mu\}$.

Распространим операцию $\boldsymbol{p_{re}}$ на множество **Σ** трасс обычным образом: $\boldsymbol{p_{re}}(\mathbf{\Sigma})\triangleq\{\boldsymbol{p_{re}}(\sigma)|\sigma\in\mathbf{\Sigma}\}$, результатом такой операции является семейство множеств трасс. Множество трасс **Σ** $\boldsymbol{p_{re}}$-замкнуто тогда и только тогда, когда $\cup\boldsymbol{p_{re}}(\mathbf{\Sigma})=\mathbf{\Sigma}$. Более того, для любого множества трасс **Σ** множество $\cup\boldsymbol{p_{re}}(\mathbf{\Sigma})$ $\boldsymbol{p_{re}}$-замкнуто и является наименьшим (по вложенности) $\boldsymbol{p_{re}}$-замкнутым множеством, содержащим **Σ**.

Операция $\boldsymbol{d}$ – удаление **R**-отказа P из трассы $\mu\cdot\langle P\rangle\cdot\lambda$: $\mu\cdot\langle P\rangle\cdot\lambda\text{—}\boldsymbol{d}\!\!\rightarrow\mu\cdot\lambda$.

Множество трасс будем называть $\boldsymbol{d}$-*замкнутым*, если оно вместе с каждой трассой содержит все трассы, получаемые из данной по $\boldsymbol{d}$-операциям. Через $\boldsymbol{d}(\sigma)$ обозначим замыкание трассы σ по операции $\boldsymbol{d}$ ($\boldsymbol{d}$-замыкание) как множество трасс, получаемых из трассы σ всеми возможными конечными цепочками применения операции $\boldsymbol{d}$:

$\boldsymbol{d}(\sigma) \triangleq \{\sigma`|\exists n\ \exists\sigma_1,\sigma_2,\ldots,\sigma_n\ \sigma_n=\sigma` \ \&\ \sigma_1\text{—}\boldsymbol{d}\!\!\rightarrow\sigma_2\text{—}\boldsymbol{d}\!\!\rightarrow\ldots\text{—}\boldsymbol{d}\!\!\rightarrow\sigma_n\}$.

Распространим операцию $\boldsymbol{d}$ на множество **Σ** трасс обычным образом: $\boldsymbol{d}(\mathbf{\Sigma})\triangleq\{\boldsymbol{d}(\sigma)|\sigma\in\mathbf{\Sigma}\}$, результатом такой операции является семейство множеств трасс. Объединение множество этого семейства обозначим $\boldsymbol{D}(\mathbf{\Sigma})=\cup\boldsymbol{d}(\mathbf{\Sigma})$. Множество трасс **Σ** $\boldsymbol{d}$-замкнуто тогда и только тогда, когда $\boldsymbol{D}(\mathbf{\Sigma})=\mathbf{\Sigma}$. Более того, для любого множества трасс **Σ** множество $\boldsymbol{D}(\mathbf{\Sigma})$ $\boldsymbol{d}$-замкнуто и является наименьшим (по вложенности) $\boldsymbol{d}$-замкнутым множеством, содержащим **Σ**.

Операция $\boldsymbol{i}_{\mathbf{R}}$ – вставка **R**-отказа P в трассу $\mu\cdot\lambda$ после трассы μ при условии, что трасса μ заканчивается каким-нибудь отказом и не продолжается во множестве трасс **Σ** дивергенцией, разрушением и действиями из P: $\mu\cdot\lambda\text{—}\boldsymbol{i}_{\mathbf{R}}\!\!\rightarrow\mu\cdot\langle P\rangle\cdot\lambda$. Множество **Σ** трасс будем называть $\boldsymbol{i}_{\mathbf{R}}$-*замкнутым*, если оно вместе с каждой трассой содержит все трассы, получаемые из данной по $\boldsymbol{i}_{\mathbf{R}}$-операциям. Через $\boldsymbol{i}_{\mathbf{R}}(\sigma)$ обозначим замыкание трассы σ по операции $\boldsymbol{i}_{\mathbf{R}}$ как множество трасс, получаемых из трассы σ с помощью всех возможных одновременных вставок любого (в том числе нулевого) числа отказов:

$\boldsymbol{i}_{\mathbf{R}}(\sigma)\triangleq \{\sigma`|\exists n\ \exists\mu_0,\ldots,\mu_n\ \exists R_1,\ldots,R_n\ \sigma=\mu_0\cdot\ldots\cdot\mu_n$

$\&\ \sigma`=\mu_0\cdot\langle R_1\rangle\cdot\mu_1\cdot\ldots\cdot\langle R_n\rangle\cdot\mu_n$

$\&\ \forall i\in[1..n]\ \sigma\text{—}\boldsymbol{i}_{\mathbf{R}}\!\!\rightarrow\mu_0\cdot\ldots\cdot\mu_{i-1}\cdot\langle R_i\rangle\cdot\mu_i\cdot\ldots\cdot\mu_n\}$.

Распространим операцию $\boldsymbol{i}_{\mathbf{R}}$ на множество $\boldsymbol{\Sigma}$ трасс обычным образом: $\boldsymbol{i}_{\mathbf{R}}(\boldsymbol{\Sigma}) \triangleq \{\boldsymbol{i}_{\mathbf{R}}(\sigma) | \sigma \in \boldsymbol{\Sigma}\}$, результатом такой операции является семейство множеств трасс.

Заметим, что $\boldsymbol{i}_{\mathbf{R}}$-операция над трассой, в отличие от $\boldsymbol{p}_{re}$- и $\boldsymbol{d}$-операций, зависит не только от этой трассы, но и от множества трасс $\boldsymbol{\Sigma}$, которому она принадлежит. Так же, как для $\boldsymbol{p}_{re}$- и $\boldsymbol{d}$-операций, множество трасс $\boldsymbol{\Sigma}$ $\boldsymbol{i}_{\mathbf{R}}$-замкнуто тогда и только тогда, когда $\cup\boldsymbol{i}_{\mathbf{R}}(\boldsymbol{\Sigma})=\boldsymbol{\Sigma}$. Также для любого множества трасс $\boldsymbol{\Sigma}$ множество $\cup\boldsymbol{i}_{\mathbf{R}}(\boldsymbol{\Sigma})$ $\boldsymbol{i}_{\mathbf{R}}$-замкнуто, однако, вообще говоря, оно не является не только наименьшим, но и минимальным $\boldsymbol{i}_{\mathbf{R}}$-замкнутым множеством, содержащим $\boldsymbol{\Sigma}$. В качестве примера рассмотрим множество $\mathbf{A}=\cup\boldsymbol{p}_{re}(\{\sigma_1,\sigma_2\})$, где $\sigma_1=\langle A,x,A,b,\Delta\rangle$, $\sigma_2=\langle A,B,x,A\rangle$, A={a}, B={b} и $\mathbf{R}$={A,B}, $\mathbf{Q}$={{x}}. Тогда $\boldsymbol{i}_{\mathbf{R}}(\sigma_1)$ и $\boldsymbol{i}_{\mathbf{R}}(\sigma_2)$ это множества трасс, которые могут быть записаны регулярными выражениями $A(AB)^*xAA^*b$ и $A(AB)^*B(AB)^*xA(AB)^*$, соответственно. Множество $\mathbf{A}_1= \cup\boldsymbol{p}_{re}(\boldsymbol{i}_{\mathbf{R}}(\sigma_1))= [A(AB)^*[x[AA^*[b[\Delta]]]]]$ содержит множество $\mathbf{A}$ и $\boldsymbol{i}_{\mathbf{R}}$-замкнуто, но не содержит трассы ABxAB, принадлежащей $\boldsymbol{i}_{\mathbf{R}}(\sigma_2)$ и, следовательно, принадлежащей $\cup$ $\boldsymbol{i}_{\mathbf{R}}(\mathbf{A})$. Тем самым $\cup\boldsymbol{i}_{\mathbf{R}}(\mathbf{A})\not\subseteq\mathbf{A}_1$.

Можно было бы выбрать другой, второй, способ определения $\boldsymbol{i}_{\mathbf{R}}$-замыкания трассы σ как множество трасс, получаемых из трассы σ с помощью всех возможных одиночных вставок отказов. Однако в этом случае множество $I_1=\cup\boldsymbol{i}_{\mathbf{R}}(\boldsymbol{\Sigma})$, хотя и содержит множество $\boldsymbol{\Sigma}$, но может быть не $\boldsymbol{i}_{\mathbf{R}}$-замкнуто. Например, если $\mathbf{R}$={{a}} и $\boldsymbol{\Sigma}$={⟨{a}⟩}, то одиночные вставки дадут только одну новую трассу ⟨{a},{a}⟩. Полученное множество, состоящее из двух трасс ⟨{a}⟩ и ⟨{a},{a}⟩, не $\boldsymbol{i}_{\mathbf{R}}$-замкнуто, поскольку не содержит трассы ⟨{a},{a},{a}⟩, хотя одновременная вставка нескольких отказов дает такую трассу (как и трассы с любым ненулевым числом отказов {a}).

Поэтому при одиночной вставке отказов нужно снова взять $\boldsymbol{i}_{\mathbf{R}}$-замыкание $I_2=\cup\boldsymbol{i}_{\mathbf{R}}((\cup\boldsymbol{i}_{\mathbf{R}}(\boldsymbol{\Sigma})))$, и так далее. При этом мы получим, вообще говоря, бесконечную последовательность расширяющихся множеств $I_1\subseteq I_2\subseteq\ldots$. Для того, чтобы окончательно получить $\boldsymbol{i}_{\mathbf{R}}$-замкнутое множество, содержащее исходное множество $\boldsymbol{\Sigma}$, нужно взять предел этой последовательности, то есть объединение $\cup\{I_n|n=1,2\ldots\}$. Результат будет тот же, что при первом способе, то есть при однократном $\boldsymbol{i}_{\mathbf{R}}$-замыкании множества $\boldsymbol{\Sigma}$ с одновременной вставкой нескольких отказов. Понятно, что первый способ предпочтительнее, поэтому мы его и выбрали.

Множество всех отказов в конце трассы σ (то есть отказов, после которых в σ следуют только отказы) будем обозначать $\boldsymbol{Ip}(\sigma)$.

2. Характеристические свойства трассовой модели

Теорема 1: Множество $\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$-трасс $\boldsymbol{\Sigma}$ является $\mathbf{R}$-моделью (то есть множеством $\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$-трасс некоторой LTS в алфавите $\mathbf{L}$) тогда и только тогда, когда оно не пусто, префикс-замкнуто $\cup\boldsymbol{p}_{re}(\boldsymbol{\Sigma})=\boldsymbol{\Sigma}$ и удовлетворяет следующим требованиям:

- допустимость: все трассы $\boldsymbol{\Sigma}$ допустимы,
- согласованность: все трассы $\boldsymbol{\Sigma}$ согласованы,
- $\mathbf{R}$-конвергентность: все трассы $\boldsymbol{\Sigma}$ $\mathbf{R}$-конвергентны в $\boldsymbol{\Sigma}$,
- замкнутость: $\boldsymbol{\Sigma}$ $\boldsymbol{d}$-замкнуто: $\boldsymbol{D}(\boldsymbol{\Sigma})=\boldsymbol{\Sigma}$,
- $\mathbf{R}$-полнота: $\boldsymbol{\Sigma}$ $\boldsymbol{i}_{\mathbf{R}}$-замкнуто: $\cup\boldsymbol{i}_{\mathbf{R}}(\boldsymbol{\Sigma})=\boldsymbol{\Sigma}$.

□Доказательство этого утверждения см. в [5], теорема 14.

В дальнейшем для заданной $\mathbf{R}/\mathbf{Q}$-семантики в качестве трассовой модели спецификации мы будем использовать $\mathbf{R}$-модель, а в качестве трассовой модели реализации мы будем использовать $\mathbf{R}\cup\mathbf{Q}$-модель.

5. Безопасное тестирование и безопасная конформность

При безопасном тестировании будут проходиться только $\mathbf{L}\cup\mathbf{R}$-трассы, причем нажиматься будут только кнопки, которые считаются безопасными после них. Это предполагает формальное определение на уровне модели отношения безопасности «кнопка P безопасна во множестве трасс $\mathbf{N}$ после $\mathbf{L}\cup\mathbf{R}$-трассы σ».

5.1. Произвольное отношение безопасности

Пусть заданы $\mathbf{R}/\mathbf{Q}$-семантика, непустое префикс-замкнутое множество трасс $\mathbf{N}$ и произвольное отношение $\boldsymbol{rel}\subseteq(\mathbf{R}\cup\mathbf{Q})\times(\mathbf{N}\cap(\mathbf{L}\cup\mathbf{R})^*)$, связывающее кнопки с $\mathbf{L}\cup\mathbf{R}$-трассами из $\mathbf{N}$, которое будем называть отношением безопасности. Запись P ***rel*** $\mathbf{N}$ ***after*** σ читается как «кнопка $P\in\mathbf{R}\cup\mathbf{Q}$ *безопасна по отношению* ***rel*** в $\mathbf{N}$ после трассы σ.

Будем говорить, что отказ $P\in\mathbf{R}$ *безопасен по отношению* ***rel*** в $\mathbf{N}$ после трассы σ и писать P ***rel*** $\mathbf{N}$ ***after*** σ, если кнопка P безопасна по отношению ***rel*** в $\mathbf{N}$ после σ.

Будем говорить, что действие $z\in\mathbf{L}$ *безопасно по отношению* ***rel*** в $\mathbf{N}$ после трассы σ и писать z ***rel*** $\mathbf{N}$ ***after*** σ, если оно разрешается такой кнопкой $P\in\boldsymbol{but}(z)$, что P ***rel*** $\mathbf{N}$ ***after*** σ.

Соответственно, будем говорить, что кнопка или наблюдение $u\in\mathbf{L}\cup\mathbf{R}\cup\mathbf{Q}$ опасны по отношению ***rel*** в $\mathbf{N}$ после $\mathbf{L}\cup\mathbf{R}$-трассы $\sigma\in\mathbf{N}$ и писать u ***rel*** $\mathbf{N}$ ***after*** σ, если они не являются безопасными по отношению ***rel*** в $\mathbf{N}$ после σ.

Будем считать, что, кроме отношения ***rel***, указано, является пустая трасса безопасной или опасной. Тогда множество всех **L**∪**R**-трасс по отношению ***rel*** разбивается на три множества трасс: безопасных, опасных и остальных.

L∪**R**-трассу σ∈**N** будем называть *безопасной по отношению* ***rel***, если пустая трасса безопасна и для каждого префикса μ·⟨u⟩≤σ наблюдение u ***rel*** **N** ***after*** μ.

Множество безопасных по отношению ***rel*** трасс, как правило, будем обозначать ***Rel***(**N**). Это множество трасс, очевидно, префикс-замкнуто и пусто, если пустая трасса опасна.

L∪**R**-трассу σ будем называть *опасной по отношению* ***rel***, если пустая трасса опасна или трасса σ имеет вид σ=μ·⟨u⟩·λ, где μ∈***Rel***(**N**), а u ***rel*** **N** ***after*** μ.

Заметим, что по этому определению опасная трасса может как принадлежать, так и не принадлежать множеству **N**. Множество опасных по отношению ***rel*** трасс, как правило, будем обозначать ***UnRel***(**N**).

Если пустая трасса опасна, то все трассы опасны. В противном случае могут быть еще трассы, которые не безопасные и не опасные, такие трассы имеют вид μ·⟨u⟩·λ, где трасса μ∈***Rel***(**N**), u ***rel*** **N** ***after*** μ, но μ·⟨u⟩∉**N**.

5.2. Отношение неразрушаемости

При тестировании безопасная кнопка – это, прежде всего, *неразрушающая* кнопка, под которой мы понимаем кнопку, нажатие которой после **L**∪**R**-трассы не может привести к попытке выхода из дивергенции или к разрушению после выполнения разрешаемого этой кнопкой действия. Это отношение безопасности обозначим через $\boldsymbol{safe}_{\gamma\Delta}$ и определим его формально для любого непустого префикс-замкнутого множества трасс **N**:

∀P∈**R**∪**Q** ∀σ∈**N**∩(**L**∪**R**)*

P $\boldsymbol{safe}_{\gamma\Delta}$ **N** ***after*** σ ≙ σ·⟨Δ⟩∉**N** & ∀z∈P σ·⟨z,γ⟩∉**N**.

Пустую трассу будем считать неразрушающей (безопасной по $\boldsymbol{safe}_{\gamma\Delta}$), если разрушение невозможно с самого начала, то есть ⟨γ⟩∉**N**. Кнопку, наблюдение или трассу будем называть *неразрушающими* (*разрушающими*), если они безопасны (опасны) по отношению $\boldsymbol{safe}_{\gamma\Delta}$. Множества неразрушающих и разрушающих трасс обозначим $\boldsymbol{Safe}_{\gamma\Delta}$(**N**) и $\boldsymbol{UnSafe}_{\gamma\Delta}$(**N**), соответственно.

5.3. Отношение безопасности в реализации

Теперь определим отношение безопасности кнопок для реализации. Такая кнопка, во-первых, должна быть неразрушающей и, во-вторых, ее нажатие не должно приводить к возникновению ненаблюдаемого отказа. Для любого непустого префикс-замкнутого множества трасс **I** мы определим безопасность кнопок только после неразрушающих трасс этого множества, которые по определению являются **L**∪**R**-трассами:

∀P∈**R**∪**Q** ∀σ∈$\boldsymbol{Safe}_{\gamma\Delta}$(**I**)

P ***safe in*** **I** ***after*** σ ≙ P $\boldsymbol{safe}_{\gamma\Delta}$ **I** ***after*** σ & (P∈**Q** ⇒ σ·⟨P⟩∉**I**).

Пустую трассу будем считать безопасной по ***safe in***, если она неразрушающая (безопасна по $\boldsymbol{safe}_{\gamma\Delta}$), то есть ⟨γ⟩∉**I**. Множества безопасных и опасных по ***safe in*** трасс обозначим ***SafeIn***(**I**) и ***UnSafeIn***(**I**), соответственно. Очевидно, ***SafeIn***(**I**)⊆$\boldsymbol{Safe}_{\gamma\Delta}$(**I**), но, вообще говоря, эти множества не равны. Это объясняется тем, что действие может быть неразрушающим, но опасным по ***safe in***, если оно разрешается только опасными по ***safe in*** **Q**-кнопками и хотя бы одна из них неразрушающая, то есть опасная по ***safe in*** из-за **Q**-отказа, продолжающего трассу.

Для заданной **R**/**Q**-семантики в качестве модели реализации будем рассматривать трассовую **R**∪**Q**-модель, которая по определению непуста и префикс-замкнута, то есть на ней можно рассматривать отношение ***safe in***.

5.4. Отношение безопасности в спецификации

Теперь определим отношение безопасности кнопок для спецификации. Для произвольного непустого префикс-замкнутого множества трасс **Σ** мы будем определять безопасность кнопок только после неразрушающих трасс этого множества, которые по определению являются **L**∪**R**-трассами. Такое отношение безопасности обозначается ***safe by*** и может быть определено, вообще говоря, различным образом. Мы сформулируем только основное условие, которому оно должно удовлетворять: A) ***safe by*** совпадает с $\boldsymbol{safe}_{\gamma\Delta}$ для **R**-кнопок, то есть ***safe by*** отличается от ***safe in*** только для **Q**-кнопок, B) множество безопасных по ***safe by*** трасс, принадлежащих **Σ**, должно совпадать с множеством неразрушающих трасс, принадлежащих **Σ**,[3] C) если кнопка безопасна по ***safe by*** после трассы, то она должна разрешать некоторое наблюдение, имеющееся в **Σ** после этой трассы.

Подусловие A объясняется тем, что, поскольку **R**-отказы наблюдаемы, при нажатии **R**-кнопки опасность могут представлять собой только дивергенция и разрушение после действия, разрешаемого кнопкой. Подусловие B объясняется тем, что мы хотим использовать спецификацию «по максимуму»: если ее трасса разрушающая, то, конечно, она не может быть пройдена при безопасном тестировании; но любая неразрушающая трасса, принадлежащая спецификации, может быть пройдена при безопасном тестировании. Множества безопасных и опасных по ***safe by*** трасс обозначим ***SafeBy***(**Σ**) и ***UnSafeBy***(**Σ**), соответственно. Тогда ***SafeBy***(**Σ**)=$\boldsymbol{Safe}_{\gamma\Delta}$(**Σ**), но, вообще говоря, ***UnSafeBy***(**Σ**)≠$\boldsymbol{UnSafe}_{\gamma\Delta}$(**Σ**), поскольку наблюдение, неразрушающее после трассы, но отсутствующее в **Σ**, может быть опасным по ***safe by***. Подусловие C нужно для того, чтобы безопасность по ***safe by*** кнопки гарантировала возможность получения хотя бы какого-нибудь конформного наблюдения, а

[3] В наших предыдущих работах [8][15][17] этот факт не отмечался, хотя формальное определение требований к отношению ***safe by*** было тем же.

такими наблюдениями, как будет определено ниже, являются только те наблюдения, которые имеются в спецификации.

Для выполнения этого основного условия отношения ***safe by*** необходимо и достаточно потребовать выполнения следующих свойств отношения. 1) **R**-кнопка безопасна по ***safe by*** тогда и только тогда, когда она неразрушающая, что совпадает с отношениями ***safe in*** и $\boldsymbol{safe}_{\gamma\Delta}$ для **R**-кнопок. 2) Если действие разрешается хотя бы одной неразрушающей кнопкой, то оно должно разрешаться какой-нибудь безопасной по ***safe by*** кнопкой. Если это неразрушающая **R**-кнопка, то она же и безопасна по ***safe by***. Но если все неразрушающие кнопки, разрешающие действие, являются **Q**-кнопками, то хотя бы одна из них должна быть объявлена безопасной по ***safe by***. 3) Если кнопка безопасна по ***safe by*** после трассы, то трасса продолжается в спецификации хотя бы одним наблюдением, разрешаемым этой кнопкой. Кроме того, пустая трасса считается безопасной по ***safe by***, если она неразрушающая (безопасная по $\boldsymbol{safe}_{\gamma\Delta}$), то есть $\langle\gamma\rangle\notin\boldsymbol{\Sigma}$:

$\epsilon\in\boldsymbol{SafeBy}(\boldsymbol{\Sigma}) \Leftrightarrow \epsilon\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$.

Запишем эти требования формально: $\forall R\in\mathbf{R}\ \forall z\in\mathbf{L}\ \forall P\in\mathbf{R}\cup\mathbf{Q}\ \forall\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$

1) R ***safe by*** $\boldsymbol{\Sigma}$ ***after*** $\sigma \Leftrightarrow R$ $\boldsymbol{safe}_{\gamma\Delta}$ $\boldsymbol{\Sigma}$ ***after*** σ,

2) $\sigma\cdot\langle z\rangle\in\boldsymbol{\Sigma}\ \&\ \exists A\in\boldsymbol{but}(z)\ A$ $\boldsymbol{safe}_{\gamma\Delta}$ $\boldsymbol{\Sigma}$ ***after*** $\sigma \Rightarrow \exists B\in\boldsymbol{but}(z)\ B$ ***safe by*** $\boldsymbol{\Sigma}$ ***after*** σ,

3) P ***safe by*** $\boldsymbol{\Sigma}$ ***after*** $\sigma \Rightarrow P$ $\boldsymbol{safe}_{\gamma\Delta}$ $\boldsymbol{\Sigma}$ ***after*** $\sigma\ \&\ \exists v\in\boldsymbol{obs}(P)\ \sigma\cdot\langle v\rangle\in\boldsymbol{\Sigma}$.

Такое отношение ***safe by*** существует тогда и только тогда, когда все неразрушающие трассы $\boldsymbol{\Sigma}$ **R**-конвергентны. Действительно, если есть неразрушающая не-**R**-конвергентная трасса, то для некоторой **R**-кнопки P трасса не продолжается наблюдениями из ***obs***(P), но P неразрушающая и, следовательно, безопасная по ***safe by*** после трассы. Тем самым, не может быть выполнено третье правило отношения ***safe by***. Обратно, если все трассы $\boldsymbol{\Sigma}$ **R**-конвергентны, то неразрушающая трасса продолжается каким-нибудь наблюдением из каждой **R**-кнопки. Поэтому достаточно объявить безопасной по ***safe by*** каждую неразрушающую **Q**-кнопку, разрешающую действие, продолжающее эту трассу. Однако в целом указанные требования неоднозначно определяют отношение ***safe by***, и при задании спецификации указывается конкретное отношение.

Для заданной **R**/**Q**-семантики в качестве модели спецификации будем рассматривать трассовую **R**-модель. Эта модель по определению непуста, префикс-замкнута и все ее трассы **R**-конвергентны, поэтому на ней может быть задано отношение ***safe by***. Кроме того, в силу **R**-конвергентности **R**-модели третье правило ***safe by*** достаточно сформулировать только для **Q**-кнопок: $\forall Q\in\mathbf{Q}\ \forall\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$

4) Q ***safe by*** $\boldsymbol{\Sigma}$ ***after*** $\sigma \Rightarrow Q$ $\boldsymbol{safe}_{\gamma\Delta}$ $\boldsymbol{\Sigma}$ ***after*** $\sigma\ \&\ \exists v\in Q\ \sigma\cdot\langle v\rangle\in\boldsymbol{\Sigma}$.

Для заданной **R**/**Q**-семантики в алфавите $\mathbf{L}=\cup\mathbf{R}\cup\cup\mathbf{Q}$ спецификация полностью задается трассовой **R**-моделью и отношением безопасности кнопок ***safe by***.

3. Замечание о пустых кнопках

Пустая **Q**-кнопка опасна как после любой безопасной по отношению ***safe in*** трассы любой реализации, так и после любой безопасной по любому отношению ***safe by*** трассы любой спецификации. Такую кнопку никогда нельзя нажимать при безопасном тестировании: поскольку **Q**-отказы ненаблюдаемы, а пустая **Q**-кнопка не разрешает никаких внешних действий, нажатие **Q**-кнопки либо означает попытку выхода из дивергенции, либо приводит к возникновению ненаблюдаемого отказа. Поэтому в дальнейшем будем считать, что такой **Q**-кнопки в семантике нет: $\varnothing\notin\mathbf{Q}$.

В то же время пустая **R**-кнопка имеет смысл: она безопасна как по ***safe in***, так и по ***safe by***, после любой неразрушающей трассы, не продолжающейся дивергенцией, а наблюдение пустого **R**-отказа означает, что реализация оказалась в стабильном состоянии[4].

4. Гипотеза о безопасности и безопасная конформность

Требование безопасности тестирования выделяет класс *безопасно-тестируемых* реализаций, то есть таких, которые могут быть безопасно протестированы для проверки их конформности заданной спецификации. Этот класс определяется следующей *гипотезой о безопасности*: для спецификационной **R**-модели $\boldsymbol{\Sigma}$ с заданным отношением ***safe by*** реализационная $\mathbf{R}\cup\mathbf{Q}$-модель $\mathbf{I}$ *безопасно-тестируема*, если выполнены следующие условия:

1) если пустая трасса безопасна по ***safe by*** в спецификации, то она безопасна по ***safe in*** в реализации, что эквивалентно: в реализации нет разрушения с самого начала, если этого нет в спецификации;
2) после общей безопасной трассы реализации и спецификации любая кнопка, безопасная по ***safe by*** в спецификации, безопасна по ***safe in*** после этой трассы в реализации:

$\mathbf{I}$ ***safe for*** $\boldsymbol{\Sigma} \triangleq (\langle\gamma\rangle\notin\boldsymbol{\Sigma} \Rightarrow \langle\gamma\rangle\notin\mathbf{I})\ \&\ \forall\sigma\in\boldsymbol{SafeBy}(\boldsymbol{\Sigma})\cap\mathbf{I}\ \forall P\in\mathbf{R}\cup\mathbf{Q}$

$(P$ ***safe by*** $\boldsymbol{\Sigma}$ ***after*** $\sigma \Rightarrow P$ ***safe in*** $\mathbf{I}$ ***after*** $\sigma)$.

Обозначим класс безопасно-тестируемых реализаций для данной спецификации $\boldsymbol{\Sigma}$ и данного отношения ***safe by***:

$\boldsymbol{SafeImp}(\boldsymbol{\Sigma},$***safe by***$) = \{\mathbf{I} | \mathbf{I}$ ***safe for*** $\boldsymbol{\Sigma}\}$.

Теперь можно определить отношение (безопасной) *конформности*: для спецификации $\boldsymbol{\Sigma}$ с заданным отношением ***safe by*** реализация $\mathbf{I}$ *безопасно конформна* (или просто *конформна*), если она безопасно-тестируема и

[4] В [21] такое наблюдение обозначено символом S.

выполнено *тестируемое условие*: любое наблюдение, возможное в реализации в ответ на нажатие безопасной в спецификации по ***safe by*** кнопки, разрешается спецификацией:

I ***saco*** **Σ** ≜ **I** ***safe for*** **Σ**
& ∀σ∈***SafeBy***(**Σ**)∩**I** ∀P ***safe by*** **Σ** ***after*** σ
obs(σ,P,**I**)⊆***obs***(σ,P,**Σ**),

где ***obs***(σ,P,**M**) ≜ {u|σ·⟨u⟩∈**M**} ∩ ***obs***(P) множество наблюдений, которые можно получить над множеством трасс **M** при нажатии кнопки P после трассы σ.

Обозначим класс конформных реализаций для данной спецификации **Σ** и данного отношения ***safe by***:

ConfImp(**Σ**,***safe by***) = {**I**|**I** ***saco*** **Σ**}.

Заметим, что, если **I** ***safe for*** **Σ**, σ∈***SafeBy***(**Σ**)∩**I**, P ***safe by*** **Σ** ***after*** σ, то ***obs***(σ,P,**I**)∩**Q**=∅. Поэтому, если u∈***obs***(σ,P,**I**) и u=P, то P∈**R**.

Следует отметить, что гипотеза о безопасности не проверяема при тестировании и является его предусловием; тестирование проверяет тестируемое условие конформности.

Определенные выше гипотеза о безопасности и конформность называются трассовыми, поскольку они основаны только на трассах наблюдений.[5] Если модели реализации и спецификации заданы в виде LTS **I** и **S**, то используются множества ***tr***($\mathbf{I}_{\mathbf{R}\cup\mathbf{Q}}$) и ***tr***($\mathbf{S}_{\mathbf{R}}$) трасс этих LTS.

6. Стандартная генерация тестов

Тестирование основано на, так называемой, *тестовой гипотезе* [18], согласно которой у реализации есть модель, адекватно описывающая поведение реализации в машине тестирования. При этом сама модель реализации может быть неизвестна, предполагается лишь ее существование. Для безопасного тестирования мы дополнительно предполагаем выполнение гипотезы о безопасности.

Для спецификации **Σ** с заданным отношением ***safe by*** *тестовой* трассой будем называть безопасную трассу, продолженную безопасным после нее наблюдением, то есть трассу σ·⟨u⟩, где σ∈***SafeBy***(**Σ**) & u ***safe by*** **Σ** ***after*** σ. При безопасном тестировании проходятся только тестовые трассы.

Безопасные трассы тестовые, но могут быть и другие тестовые трассы, если σ·⟨u⟩∉**Σ**. Такие тестовые трассы, отсутствующие в спецификации, будем называть *ошибками* (*ошибочными трассами*). Будем говорить, что *в*

[5] Другим видом конформности является симуляция, основанная, кроме того, на соответствии состояний реализации и спецификации. Вопросы безопасной симуляции и ее тестирования рассмотрены в наших работах [10][11][12].

реализации есть ошибка σ·⟨u⟩, если спецификация определяет ошибку σ·⟨u⟩, а в реализации есть трасса σ·⟨u⟩. Очевидно, безопасно-тестируемая реализация конформна тогда и только тогда, когда в ней нет ошибок.

В терминах машины тестирования тест – это инструкция оператору машины. В каждом пункте инструкции указывается кнопка, которую оператор должен нажимать, и для каждого наблюдения – пункт инструкции, который должен выполняться следующим, или вердикт (***pass*** или ***fail***), если тестирование нужно закончить. Тест можно понимать как префикс-замкнутое множество конечных историй с назначенными вердиктами, в котором:

1) каждая максимальная история заканчивается наблюдением, и ей приписан вердикт, немаксимальным историям вердикты не приписаны;
2) каждая немаксимальная история, заканчивающаяся кнопкой, продолжается во множестве только теми наблюдениями, которые разрешаются этой кнопкой (это следует из определения истории в п.3);
3) каждая немаксимальная история, заканчивающаяся кнопкой, обязательно продолжается во множестве всеми наблюдениями, которые могут встречаться в безопасно-тестируемых реализациях после трассы этой истории.

Тест безопасен тогда и только тогда, когда трассы всех его историй являются тестовыми. Реализация *проходит* тест, если её тестирование с помощью этого теста всегда заканчивается с вердиктом ***pass***. Реализация проходит набор тестов, если она проходит каждый тест из набора. Набор тестов *значимый*, если каждая конформная реализация его проходит; *исчерпывающий*, если каждая неконформная реализация его не проходит; *полный*, если он значимый и исчерпывающий. Для определения конформности или неконформности любой безопасно-тестируемой реализации ставится задача генерации полного набора тестов по спецификации.

Строгим тестом будем называть такой тест, в котором вердикт ***pass*** назначается максимальной истории, если ее трасса не является ошибкой (имеется в спецификации), а вердикт ***fail*** – если ее трасса является ошибкой (отсутствует в спецификации). Такие тесты, во-первых, значимые (не фиксируют ложных ошибок) и, во-вторых, не пропускают обнаруженных ошибок. Мы будем рассматривать только строгие тесты.

Полный набор тестов всегда существует, в частности, им является набор всех *примитивных* тестов [5]. Примитивный тест строится по одной немаксимальной безопасной трассе спецификации σ=⟨u_1,u_2,…,u_n⟩. Для этого в трассу вставляются кнопки, которые оператор должен нажимать: перед каждым наблюдением u_i вставляется какая-нибудь безопасная после префикса ⟨u_1,u_2,…,u_{i-1}⟩ кнопка P_i∈***but***(u_i): если u_i∈**R**, то P_i=u_i, иначе u_i∈P_i. После всей трассы σ вставляется любая безопасная после нее кнопка P_{n+1}. В результате получается тестовая история T=⟨P_1,u_1,P_2,u_2,…,P_{n+1}⟩. Любая другая история T`

теста – это строгий префикс истории T или имеет вид $T`=\langle P_1,u_1,P_2,u_2,\ldots,P_i,u_i`\rangle$, где $u_i`$ наблюдение, продолжающее трассу σ при i=n+1, или ответвляющееся от трассы при $i\le n$ и $u_i`\neq u_i$. Безопасность трассы σ для каждого отказа $u_i\in\mathbf{R}$ гарантирует безопасность кнопки $P_i=u_i$, и для каждого действия $u_i\in\mathbf{L}$ гарантирует наличие разрешающей его безопасной кнопки $P_i\in\boldsymbol{but}(u_i)$, а немаксимальность безопасной трассы σ гарантирует наличие последней кнопки P_{n+1}. Выбор кнопок P_i для $u_i\in\mathbf{L}$ и кнопки P_{n+1} может быть неоднозначным: по одной безопасной трассе спецификации можно сгенерировать, вообще говоря, несколько разных примитивных тестов.

Любой строгий тест как множество историй равен объединению некоторого множества примитивных тестов (в объединении максимальные истории сохраняют те вердикты, которые были им приписаны в объединяемых примитивных тестах). Поэтому такой строгий тест обнаруживает те же самые ошибки, что и соответствующее множество примитивных тестов. Поэтому обычно рассматриваются только примитивные тесты.

7. Финальные модели спецификации

Из определения отношений ***safe by***, ***safe for*** и ***saco*** видно, что не все трассы спецификации нужны для определения безопасно-тестируемости и конформности. Кроме того, отношение ***safe by*** приходится задавать отдельно от спецификационной модели.

В этом разделе мы определим такое множество трасс для спецификации, которого достаточно для определения этих отношений и которое в некотором смысле «минимально». Это множество будем называть финальной трассовой моделью спецификации. Разным отношениям ***safe by*** для одной и той же **R**-модели спецификации будут соответствовать разные финальные модели спецификации, на которых эти отношения ***safe by*** совпадают с отношением ***safe in***. Иными словами, поскольку ***safe in*** определяется однозначно, финальная модель спецификации одновременно задает как нужное множество трасс спецификации, так и отношение ***safe by***.

Мы определим преобразования **R**-модели спецификации в финальную трассовую модель спецификации. Также определим характеристические свойства финальной модели, то есть набор свойств, которых необходимо и достаточно для того, чтобы множество трасс было результатом такого преобразования.

В этом подразделе будем считать, что заданы **R**/**Q**-семантика в алфавите **L** и **R**-модель спецификации **Σ**. Финальная модель будет множеством трасс $\boldsymbol{\Omega}\subseteq(\mathbf{L}\cup\mathbf{R}\cup\mathbf{Q}\cup\{\Delta,\gamma\})^*$. Будем обозначать подмножество его трасс без **Q**-отказов через $\boldsymbol{\Omega}_\mathbf{R}=\boldsymbol{\Omega}\cap(\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\})^*$.

Мы определим два вида трассовых моделей спецификации: предфинальную (трассовую) модель, которую будем называть p-моделью, и финальную (трассовую модель), которую будем называть f-моделью. В последнем разделе

рассмотрим представление этих моделей в виде детерминированных LTS в алфавите $\mathbf{L}\cup\mathbf{R}\cup\mathbf{Q}\cup\{\Delta\}$.

8. Финальные трассовые модели

8.1. Финальные суффиксы и продолжения трасс

Пусть $\boldsymbol{\Psi}\subseteq(\mathbf{L}\cup\mathbf{R}\cup\mathbf{Q}\cup\{\Delta,\gamma\})^*$ непустое префикс-замкнутое множество трасс. *Финальным суффиксом* трассы $\mu\in\boldsymbol{\Psi}$ во множестве **Ψ** будем называть такую трассу λ, что $\mu\cdot\lambda\in\boldsymbol{\Psi}$ и 1) $\lambda=\epsilon$, 2) $\lambda=\langle\Delta\rangle$, 3) $\lambda=\langle z,\gamma\rangle$, 4) $\lambda=\langle z\rangle$ и $\mu\cdot\langle z,\gamma\rangle\in\boldsymbol{\Psi}$, или 5) $\lambda=\langle Q\rangle$, где $Q\in\mathbf{Q}$ & $Q\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}\ \boldsymbol{after}\ \mu$. В случае 5 будем называть λ **Q**-финальным суффиксом. Множество финальных суффиксов трассы μ во множестве **Ψ** обозначим $\boldsymbol{fs}(\boldsymbol{\Psi},\mu)$.

Финальным продолжением трассы $\mu\in\boldsymbol{\Psi}$ во множестве **Ψ** будем называть продолжение трассы каким-либо ее финальным суффиксом, то есть такую трассу $\mu\cdot\lambda$, что $\lambda\in\boldsymbol{fs}(\boldsymbol{\Psi},\mu)$. Продолжение трассы ее **Q**-финальным суффиксом будем называть **Q**-финальным продолжением или **Q**-финальной трассой. Множество финальных продолжений трассы μ обозначим $\boldsymbol{fx}(\boldsymbol{\Psi},\mu)$. Финальные суффиксы и продолжения трассы, очевидно, связаны следующим образом:

$\boldsymbol{fx}(\boldsymbol{\Psi},\mu)=\{\mu\cdot\lambda|\lambda\in\boldsymbol{fs}(\boldsymbol{\Psi},\mu)\}$ и $\boldsymbol{fs}(\boldsymbol{\Psi},\mu)=\{\lambda|\mu\cdot\lambda\in\boldsymbol{fx}(\boldsymbol{\Psi},\mu)\}$.

5. Предфинальные трассы и предфинальные множества

Трассу из непустого префикс-замкнутого множества $\boldsymbol{\Psi}\subseteq(\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\})^*$ будем называть *предфинальной*, если это пустая трасса ϵ, трасса $\langle\gamma\rangle$ или финальное продолжение неразрушающей трассы. Заметим, что в таком множестве **Ψ** нет **Q**-отказов, поэтому у его трасс нет **Q**-финальных суффиксов. Формально множество предфинальных трасс определяется так:

$\boldsymbol{ptr}(\boldsymbol{\Psi})\triangleq\{\epsilon,\langle\gamma\rangle\}\cap\boldsymbol{\Psi}\cup\cup\{\boldsymbol{fx}(\boldsymbol{\Psi},\sigma)|\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})\}$.

<u>*Теорема 2:*</u> Пусть множество $\boldsymbol{\Psi}\subseteq(\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\})^*$ непусто и префикс-замкнуто. Тогда:

$\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Psi}))$,

$\forall\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})\ \forall P\in\mathbf{R}\cup\mathbf{Q}\ (P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}\ \boldsymbol{after}\ \sigma\Leftrightarrow P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{ptr}(\boldsymbol{\Psi})\ \boldsymbol{after}\ \sigma)$.

□264

Очевидно, что $\boldsymbol{ptr}(\boldsymbol{\Psi})\subseteq\boldsymbol{\Psi}$. Будем говорить, что множество **Ψ** *предфинально*, если оно совпадает с подмножеством его предфинальных трасс: $\boldsymbol{ptr}(\boldsymbol{\Psi})=\boldsymbol{\Psi}$.

<u>*Теорема 3:*</u> Если множество **Ψ** *предфинально*, то подмножество его неразрушающих трасс совпадает с подмножеством его трасс, не содержащих дивергенции и разрушения и не продолжающихся разрушением:

$\boldsymbol{ptr}(\boldsymbol{\Psi})=\boldsymbol{\Psi}\Rightarrow\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=\{\sigma\in\boldsymbol{\Psi}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Psi}\}$.

Если трассы из $\boldsymbol{\Psi}$ допустимы и после отказа в них нет разрушения (часть условия согласованности), то:

$\boldsymbol{ptr}(\boldsymbol{\Psi})=\boldsymbol{\Psi} \Leftarrow \langle\gamma\rangle\notin\boldsymbol{\Psi}\ \&\ \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=\{\sigma\in\boldsymbol{\Psi}\cap(\mathbf{L}\cup\mathbf{R})^{*}\mid\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Psi}\}\vee\boldsymbol{\Psi}=\{\epsilon,\langle\gamma\rangle\}.$

□265

Из этой теоремы следует, что, если трассы из $\boldsymbol{\Psi}$ допустимы и после отказа в них нет разрушения (часть условия согласованности), то:

$\boldsymbol{ptr}(\boldsymbol{\Psi})=\boldsymbol{\Psi} \Leftrightarrow \langle\gamma\rangle\notin\boldsymbol{\Psi}\ \&\ \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=\{\sigma\in\boldsymbol{\Psi}\cap(\mathbf{L}\cup\mathbf{R})^{*}\mid\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Psi}\}\vee\boldsymbol{\Psi}=\{\epsilon,\langle\gamma\rangle\}.$

6. *Незамкнутость и финально-замкнутость*

Для заданных алфавита $\mathbf{L}$ и семейства $\mathbf{R}\subseteq P(\mathbf{L})$ множество, обладающее всеми свойствами $\mathbf{R}$-модели, кроме замкнутости, будем называть *незамкнутой (трассовой)* $\mathbf{R}$-*моделью*.

Теорема 4: Если $\boldsymbol{\Sigma}$ незамкнутая $\mathbf{R}$-модель, то множество $\boldsymbol{D}(\boldsymbol{\Sigma})$ является (замкнутой) $\mathbf{R}$-моделью.

□267

Множество $\boldsymbol{\Psi}$ будем называть *финально-замкнутым*, если после удаления отказов из неразрушающей трассы получается либо разрушающая трасса, либо неразрушающая трасса с сохранением финальных суффиксов:

$\forall\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega}_{\mathbf{R}})\ \forall\mu\in\boldsymbol{d}(\sigma)$

$(\mu\in\boldsymbol{UnSafe}_{\gamma\Delta}(\boldsymbol{\Omega}_{\mathbf{R}})\vee\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega}_{\mathbf{R}})\ \&\ \boldsymbol{fs}(\boldsymbol{\Omega}_{\mathbf{R}},\sigma)\subseteq\boldsymbol{fs}(\boldsymbol{\Omega}_{\mathbf{R}},\mu))$.

Из финально-замкнутости следует, что, если $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega}_{\mathbf{R}})$, $\mu\in\boldsymbol{d}(\sigma)$ и $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega}_{\mathbf{R}})$, то для каждой кнопки P

P $\boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}$ *after* $\sigma \Rightarrow$ P $\boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}$ *after* μ.

***P*-модель**

Незамкнутую $\mathbf{R}$-модель $\boldsymbol{\Psi}$ будем называть *p-моделью*, если она предфинальна и финально-замкнута. Иными словами, p-модель – это непустое, префикс-замкнутое множество $\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$-трасс, обладающее свойствами допустимости, согласованности, $\mathbf{R}$-конвергентности, финально-замкнутости, $\mathbf{R}$-полноты и предфинальности.

Теорема 5: Множество $\boldsymbol{ptr}(\boldsymbol{\Sigma})$ предфинальных трасс $\mathbf{R}$-модели $\boldsymbol{\Sigma}$ является p-моделью.

□268

Теорема 6: $\boldsymbol{d}$-замыкание p-модели $\boldsymbol{\Psi}$ является $\mathbf{R}$-моделью, множество предфинальных трасс которой совпадает с исходной p-моделью: $\boldsymbol{ptr}(\boldsymbol{D}(\boldsymbol{\Psi}))=\boldsymbol{\Psi}$.

□271

7. *F-модель*

Из теоремы 2 следует, что множество $\boldsymbol{ptr}(\boldsymbol{\Sigma})$ предфинальных трасс $\mathbf{R}$-модели спецификации $\boldsymbol{\Sigma}$ однозначно определяет отношение $\boldsymbol{safe}_{\gamma\Delta}$ и множество неразрушающих трасс $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})=\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$. Множества $\boldsymbol{ptr}(\boldsymbol{\Sigma})$ также

достаточно для определения отношения ***safe by***. По теореме 5 множество $\boldsymbol{ptr}(\boldsymbol{\Sigma})$ является p-моделью.

Мы будем рассматривать отношение ***safe by*** на произвольной p-модели $\boldsymbol{\Psi}$. Оно должно удовлетворять правилам отношения ***safe by***:

$\forall\mathrm{R}\in\mathbf{R}\ \forall\mathrm{z}\in\mathbf{L}\ \forall\mathrm{P}\in\mathbf{R}\cup\mathbf{Q}\ \forall\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$

1) R ***safe by*** $\boldsymbol{\Sigma}$ ***after*** $\sigma \Leftrightarrow$ R $\boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Sigma}$ ***after*** σ,

2) $\sigma\cdot\langle\mathrm{z}\rangle\in\boldsymbol{\Psi}\ \&\ \exists\mathrm{A}\in\boldsymbol{but}(\mathrm{z})$ A $\boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}$ ***after*** $\sigma \Rightarrow \exists\mathrm{B}\in\boldsymbol{but}(\mathrm{z})$ B ***safe by*** $\boldsymbol{\Psi}$ ***after*** σ,

3) P ***safe by*** $\boldsymbol{\Psi}$ ***after*** $\sigma \Rightarrow$ P $\boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}$ ***after*** $\sigma\ \&\ \exists\mathrm{v}\in\boldsymbol{obs}(\mathrm{P})\ \sigma\cdot\langle\mathrm{v}\rangle\in\boldsymbol{\Psi}$.

Поскольку это отношение однозначно определяется для $\mathbf{R}$-кнопок и разрушающих $\mathbf{Q}$-кнопок, достаточно доопределить это отношение только для неразрушающих $\mathbf{Q}$-кнопок после неразрушающих трасс. Оно должно удовлетворять второму и третьему правилам отношения ***safe by***, которые можно записать в следующем виде: $\forall\mathrm{z}\in\mathbf{L}\ \forall\mathrm{Q}\in\mathbf{Q}\ \forall\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$

2a) $\sigma\cdot\langle\mathrm{z}\rangle\in\boldsymbol{\Psi}\ \&\ \exists\mathrm{A}\in\boldsymbol{but}(\mathrm{z})$ A $\boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}$ ***after*** $\sigma \Rightarrow \exists\mathrm{B}\in\boldsymbol{but}(\mathrm{z})$ B ***safe by*** $\boldsymbol{\Psi}$ ***after*** σ,

3a) Q $\boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}$ ***after*** σ & Q ***safe by*** $\boldsymbol{\Psi}$ ***after*** $\sigma \Rightarrow \exists\mathrm{v}\in\mathrm{Q}\ \sigma\cdot\langle\mathrm{v}\rangle\in\boldsymbol{\Psi}$.

Для того чтобы изобразить теперь выбранное отношение ***safe by*** в модели добавим в p-модель продолжение каждой неразрушающей трассы σ каждым неразрушающим, но опасным после нее по ***safe by*** $\mathbf{Q}$-отказом Q, то есть добавим трассу $\sigma\cdot\langle\mathrm{Q}\rangle$. Добавленная трасса $\sigma\cdot\langle\mathrm{Q}\rangle$ – это $\mathbf{Q}$-финальная трасса, то есть неразрушающая трасса σ, продолженная $\mathbf{Q}$-финальным суффиксом $\langle\mathrm{Q}\rangle$.

Далее *финальными трассами* будем называть предфинальные и $\mathbf{Q}$-финальные трассы. Для спецификации, заданной в виде $\mathbf{R}$-модели $\boldsymbol{\Sigma}$ или p-модели $\boldsymbol{\Psi}$, и отношения ***safe by***, множество финальных трасс обозначим $\boldsymbol{ftr}(\boldsymbol{\Sigma},\boldsymbol{safe\ by})$ и $\boldsymbol{ftr}(\boldsymbol{\Psi},\boldsymbol{safe\ by})$, соответственно.

Такое множество финальных трасс определяется так:

$\boldsymbol{ftr}(\boldsymbol{\Psi},\boldsymbol{safe\ by}) \triangleq \boldsymbol{\Psi}$

$\cup\ \{\sigma\cdot\langle\mathrm{Q}\rangle\mid\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})\ \&\ \mathrm{Q}\in\mathbf{Q}\ \&\ \mathrm{Q}\ \boldsymbol{safe\ by}\ \boldsymbol{\Psi}\ \boldsymbol{after}\ \sigma\ \&\ \mathrm{Q}\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}\ \boldsymbol{after}\ \sigma\}$,

$\boldsymbol{ftr}(\boldsymbol{\Sigma},\boldsymbol{safe\ by}) \triangleq \boldsymbol{ftr}(\boldsymbol{ptr}(\boldsymbol{\Sigma}),\boldsymbol{safe\ by})$.

Для p-модели $\boldsymbol{\Psi}$ обозначим $\boldsymbol{\Omega}=\boldsymbol{ftr}(\boldsymbol{\Psi},\boldsymbol{safe\ by})$. Очевидно, $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})=\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$. Второе и третье правила отношения ***safe by*** теперь можно понимать как требования к добавлению $\mathbf{Q}$-отказов после неразрушающих трасс: $\forall\mathrm{z}\in\mathbf{L}\ \forall\mathrm{Q}\in\mathbf{Q}\ \forall\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})$

2b) $\sigma\cdot\langle\mathrm{z}\rangle\in\boldsymbol{\Omega}\ \&\ \sigma\cdot\langle\mathrm{z},\gamma\rangle\notin\boldsymbol{\Omega}$

$\Rightarrow \exists\mathrm{P}\in\boldsymbol{but}(\mathrm{z})$ P $\boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Omega}$ ***after*** $\sigma\ \&\ (\mathrm{P}\in\mathbf{Q} \Rightarrow \sigma\cdot\langle\mathrm{P}\rangle\notin\boldsymbol{\Omega})$,

3b) Q $\boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Omega}$ ***after*** $\sigma\ \&\ \sigma\cdot\langle\mathrm{Q}\rangle\notin\boldsymbol{\Omega} \Rightarrow \exists\mathrm{v}\in\mathrm{Q}\ \sigma\cdot\langle\mathrm{v}\rangle\in\boldsymbol{\Omega}$.

Далее запишем формально правило, согласно которому мы добавляем только неразрушающие **Q**-отказы только после неразрушающих трасс и после **Q**-отказов ничего не добавляем:

4b) $\forall P\in\mathbf{Q}\ \forall\mu\cdot\langle P\rangle\cdot\lambda\in\boldsymbol{\Omega}\ \mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})\ \&\ P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Omega}\ \boldsymbol{after}\ \mu\ \&\ \lambda=\epsilon$.

Наконец, запишем условие того, что все добавляемые трассы содержат **Q**-отказы: $\boldsymbol{\Psi}=\boldsymbol{\Omega}_{\mathbf{R}}$, где $\boldsymbol{\Omega}_{\mathbf{R}}=\boldsymbol{\Omega}\cap(\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\})^*$.

Итак, если $\boldsymbol{\Omega}\subseteq(\mathbf{L}\cup\mathbf{R}\cup\mathbf{Q}\cup\{\Delta,\gamma\})^*$ и $\boldsymbol{\Psi}=\boldsymbol{\Omega}_{\mathbf{R}}$ p-модель, то условия 2b), 3b), 4b) эквивалентны $\boldsymbol{\Omega}=\boldsymbol{ftr}(\boldsymbol{\Psi},\boldsymbol{safe\ by})$.

Пусть множество трасс $\boldsymbol{\Omega}\subseteq(\mathbf{L}\cup\mathbf{R}\cup\mathbf{Q}\cup\{\Delta,\gamma\})^*$ непусто и префикс-замкнуто. Будем говорить, что $\boldsymbol{\Omega}$ *финально*, если

$\langle\gamma\rangle\notin\boldsymbol{\Omega}\ \&\ \boldsymbol{SafeIn}(\boldsymbol{\Omega})=\{\sigma\in\boldsymbol{\Omega}\cap(\mathbf{L}\cup\mathbf{R})^*\mid\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Omega}\}\ \vee\ \boldsymbol{\Omega}=\{\epsilon,\langle\gamma\rangle\}$.

Будем говорить, что $\boldsymbol{\Omega}$ **Q**-*допустимо*, если

$\forall P\in\mathbf{Q}\ \forall\mu\cdot\langle P\rangle\cdot\lambda\in\boldsymbol{\Omega}\ \mu\in\boldsymbol{SafeIn}(\boldsymbol{\Omega})\ \&\ P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Omega}\ \boldsymbol{after}\ \mu\ \&\ \lambda=\epsilon$.

Теорема 7: Пусть $\boldsymbol{\Omega}\subseteq(\mathbf{L}\cup\mathbf{R}\cup\mathbf{Q}\cup\{\Delta,\gamma\})^*$ и $\boldsymbol{\Psi}=\boldsymbol{\Omega}_{\mathbf{R}}$ p-модель. Тогда условия 2b), 3b), 4b) эквивалентны следующим трем условиям:

2c) $\boldsymbol{\Omega}$ финально,
3c) трассы из $\boldsymbol{\Omega}_{\mathbf{R}}$ **Q**-конвергентны в $\boldsymbol{\Omega}$,
4c) $\boldsymbol{\Omega}$ **Q**-допустимо.

□274

Очевидно, $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})=\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega}_{\mathbf{R}})$. Очевидно также, что, если $\boldsymbol{\Psi}$ p-модель и $\boldsymbol{\Omega}=\boldsymbol{ftr}(\boldsymbol{\Psi},\boldsymbol{safe\ by})$, то $\boldsymbol{\Psi}=\boldsymbol{\Omega}_{\mathbf{R}}$.

Множество трасс $\boldsymbol{\Omega}\subseteq(\mathbf{L}\cup\mathbf{R}\cup\mathbf{Q}\cup\{\Delta,\gamma\})^*$ будем называть *f-моделью*, если это непустое, префикс-замкнутое множество трасс, обладающее свойствами допустимости, **Q**-допустимости, согласованности, $\mathbf{R}\cup\mathbf{Q}$-конвергентности трасс из $\boldsymbol{\Omega}_{\mathbf{R}}$ в $\boldsymbol{\Omega}$, финально-замкнутости $\boldsymbol{\Omega}_{\mathbf{R}}$, **R**-полноты $\boldsymbol{\Omega}_{\mathbf{R}}$ и финальности.

Теорема 8: Пусть на p-модели $\boldsymbol{\Psi}$ задано отношение ***safe by***. Тогда множество $\boldsymbol{\Omega}=\boldsymbol{ftr}(\boldsymbol{\Psi},\boldsymbol{safe\ by})$ является f-моделью. При этом

$\boldsymbol{SafeIn}(\boldsymbol{\Omega})=\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})=\boldsymbol{SafeBy}(\boldsymbol{\Omega})=\boldsymbol{SafeBy}(\boldsymbol{\Psi})=\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=\boldsymbol{SafeIn}(\boldsymbol{\Psi})$.

□275

Теорема 9: Пусть $\boldsymbol{\Omega}$ f-модель. Тогда $\boldsymbol{\Omega}_{\mathbf{R}}$ p-модель, отношение ***safe in*** на $\boldsymbol{\Omega}$, взятое для трасс из $\boldsymbol{\Omega}_{\mathbf{R}}$, удовлетворяет правилам отношения ***safe by***, и $\boldsymbol{\Omega}=\boldsymbol{ftr}(\boldsymbol{\Omega}_{\mathbf{R}},\boldsymbol{safe\ in})$.

□276

Обозначим класс безопасно-тестируемых реализаций для f-модели $\boldsymbol{\Omega}$:

$\boldsymbol{SafeImp}(\boldsymbol{\Omega})=\boldsymbol{SafeImp}(\boldsymbol{\Sigma},\boldsymbol{safe\ by})$, где $\boldsymbol{\Omega}=\boldsymbol{ftr}(\boldsymbol{\Sigma},\boldsymbol{safe\ by})$.

Обозначим класс конформных реализаций для f-модели $\boldsymbol{\Omega}$:

$\boldsymbol{ConfImp}(\boldsymbol{\Omega})=\boldsymbol{ConfImp}(\boldsymbol{\Sigma},\boldsymbol{safe\ by})$, где $\boldsymbol{\Omega}=\boldsymbol{ftr}(\boldsymbol{\Sigma},\boldsymbol{safe\ by})$.

f-модель и ∇-пополнение

f-модель в общем случае может содержать безопасные трассы, которые, тем не менее, не нужны для генерации тестов, поскольку они неконформны, то есть не встречаются в конформных реализациях. В [17] определяется преобразование трассовой спецификации, целью которого является удаление из спецификации неконформных трасс. Результат такого преобразования называется ∇-пополнением (то, что мы здесь называем ∇-пополнением исходной **R**-модели $\boldsymbol{\Sigma}$ в [17] обозначается как $\boldsymbol{\Sigma}^{\mathbf{01}\nabla}$). В общем случае ∇-пополнение в исходной **R**/**Q**-семантике может не существовать. Поэтому ∇-пополнение строится не в исходной **R**/**Q**-семантике, а в расширенной $\mathbf{R}^{\#}/\mathbf{Q}^{\#}$-семантике, которая получается добавлением в каждую кнопку P фиктивного действия, которое называется не-отказом и обозначается ₽ . Тестирование в такой семантике эквивалентно тестированию в исходной семантике для реализаций в исходном алфавите **L**, то есть реализаций, в которых не-отказы не встречаются. В трассах ∇-пополнения не-отказы (и только они) непосредственно предшествуют дивергенции или разрушению, то есть не-отказы предназначены только для индикации безопасности кнопок после трасс. Кнопка P безопасна после трассы, если трасса не продолжается не-отказом ₽ , за которым следует разрушение (либо не продолжается ₽, либо после ₽ дивергенция).

Заметим, что ∇-пополнение может быть пустым, что означает отсутствие конформных реализаций. Далее будем считать, что такие реализации есть и ∇-пополнение не пусто.

∇-пополнение не является $\mathbf{R}^{\#}$-моделью, но его ***d***-замыкание является $\mathbf{R}^{\#}$-моделью, причем множество безопасных трасс этой $\mathbf{R}^{\#}$-модели совпадает с множеством всех трасс без не-отказов ∇-пополнения.

∇-пополнение строится в два этапа. На первом этапе в спецификацию добавляются все трассы, по которым можно вести безопасное тестирование всех реализаций из класса безопасно-тестируемых реализаций исходной спецификации $\boldsymbol{SafeImp}(\boldsymbol{\Sigma},\boldsymbol{safe\ by})$. Это значит, что примитивный тест, сгенерированный по такой трассе, будет безопасным для любой реализации из этого класса. Заметим, что такие трассы могут и не принадлежать исходной спецификации. На втором этапе из спецификации удаляются неконформные трассы. Поэтому, хотя все безопасные трассы без не-отказов ∇-пополнения конформны, однако некоторые из них отсутствуют в исходной спецификации, поскольку были добавлены на первом этапе пополнения и оказались конформными.

Такие добавленные трассы можно было бы удалить из ∇-пополнения, оставив только те безопасные трассы без не-отказов, которые есть в f-модели исходной спецификации. Это первая задача. Можно поставить и вторую задачу: удалить из f-модели неконформные трассы, используя ∇-пополнение, то есть удалить те безопасные трассы, которых нет в ∇-пополнении.

При решении обеих этих задач тесты для реализации в алфавите **L** генерируются по одним и тем же трассам: безопасным конформным трассам исходной спецификации (или, что то же самое, ее f-модели). Отличие только в том, как определяется безопасность кнопок после этих трасс: по ∇-пополнению или по f-модели.

Опишем решения этих задач формально.

Пусть **Ω** f-модель, а **Ξ** ∇-пополнение.

Поскольку **Ω** и **Ξ** заданы в разных семантиках, сначала определим перевод трасс из одной семантики в другую.

Преобразование из **R**/**Q**-семантики в $\mathbf{R}^{\#}/\mathbf{Q}^{\#}$-семантику. Если мы берем трассу без не-отказов $\sigma\in\mathbf{\Omega}$, то ей соответствует трасса в $\mathbf{R}^{\#}/\mathbf{Q}^{\#}$-семантике, которая строится следующим образом: в каждый **R**-отказ в трассе σ добавляется соответствующий ему не-отказ. В результате будет получена трасса, которую будем обозначать как $\sigma^{\#}$. Формально: $\forall P\in\mathbf{R}\cup\mathbf{Q}\ \forall z\in\mathbf{L}\ \forall\sigma\in(\mathbf{L}\cup\mathbf{R})^{*}$

$P^{\#}\triangleq P\cup\{\not R\}$, $z^{\#}\triangleq z$, $\sigma^{\#}\triangleq\langle\sigma(i)^{\#}|i=1..|\sigma|\rangle$.

Преобразование из $\mathbf{R}^{\#}/\mathbf{Q}^{\#}$-семантики в **R**/**Q**-семантику. Если мы берем трассу без не-отказов $\sigma\in\mathbf{\Xi}$, то ей соответствует трасса в **R**/**Q**-семантике, которая строится следующим образом: из каждого $\mathbf{R}^{\#}$-отказа в трассе удаляется не-отказ, то есть оставляются только действия из **L**. В результате будет получена трасса, которую будем обозначать как $\sigma_{\mathbf{L}}$. Формально: $\forall P\in\mathbf{R}^{\#}\ \forall z\in\mathbf{L}\ \forall\sigma\in(\mathbf{L}\cup\mathbf{R}^{\#})^{*}$

$P_{\mathbf{L}}\triangleq P\cap\mathbf{L}$, $z_{\mathbf{L}}\triangleq z$, $\sigma_{\mathbf{L}}\triangleq\langle\sigma(i)_{\mathbf{L}}|i=1..|\sigma|\rangle$.

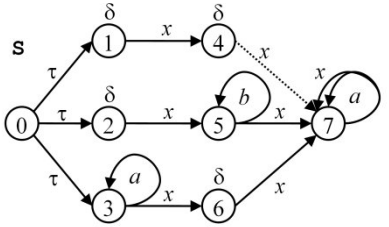
Первая задача. Для решения первой задачи мы должны, во-первых, взять все трассы из **Ξ**, которые встречаются в $\mathbf{\Omega}^{\#}$. Это будут безопасные и конформные трассы, которые есть и в **Ξ**, и в $\mathbf{\Omega}^{\#}$. Во-вторых, мы должны взять все их продолжения не-отказами и далее разрушением или дивергенцией, которые есть в **Ξ**. В результате получится следующее множество:

$\mathbf{\Xi①\Omega}\triangleq\mathbf{\Xi}\cap\mathbf{\Omega}^{\#}\cup\{\mu\cdot\langle P\rangle\cdot\lambda|\mu\in\mathbf{\Xi}\cap\mathbf{\Omega}^{\#}\ \&\ P\in\mathbf{R}\cup\mathbf{Q}\ \&\ \mu\cdot\langle\not R\rangle\cdot\lambda\in\mathbf{\Xi}\}$.

Проблема здесь в том, что тесты, сгенерированные по **Ξ①Ω** могут ловить «ложные» ошибки. Это объясняется тем, что оставшаяся в **Ξ①Ω** трасса может не продолжаться наблюдениями, которые были в **Ξ**, но удалены из **Ξ①Ω**. Если удалены все такие наблюдения из некоторой **Q**-кнопки, которая была опасной в f-модели, но безопасна в ∇-пополнении, то после нажатия такой кнопки будет ловиться «ложная» ошибка.

Пример приведен на рис. 3. Здесь все трассы LTS-спецификации **S** безопасны, поэтому все они будут в ее f-модели **Ω**. Также все эти трассы конформны, поэтому все они будут в ∇-пополнении **Ξ**. Но в **Ξ** добавляется переход, изображенный пунктиром. В этом примере имеем $\boldsymbol{Safe}_{\gamma\Delta}(\mathbf{\Omega})\subset(\boldsymbol{Safe}_{\gamma\Delta}(\mathbf{\Xi}))_{\mathbf{L}}$. Кроме того, добавляются не изображенные переходы по не-отказам, ведущие в состояние, где есть дивергенция (и нет разрушения). В состояниях, после

трасс, заканчивающихся отказом δ (эти состояния помечены символом δ), проводятся переходы только по не-отказу ~~{x}~~, а в остальных состояниях – по обоим не-отказам ~~{x}~~ и ~~δ~~. Все добавленные трассы $\delta\delta^{*}x\delta\delta^{*}x(xa)^{*}$ конформны. После удаления этих трасс в **Ξ①Ω** кнопка {x} остаётся безопасной после трассы δxδ, поскольку по-прежнему нет трассы δxδ~~{x}~~γ. В любой безопасно-тестируемой реализации после трассы δxδ должно быть действие x, однако теперь оно отсутствует в спецификации **Ξ①Ω** и, следовательно, считается ошибкой. В исходной спецификации **S** (и в ее f-модели **Ω**) кнопка {x} опасна после трассы δxδ, поэтому действие x после этой трассы не считается ошибкой. Из-за этого реализация может быть конформна исходной спецификации **S** (и ее f-модели **Ω**) и неконформна **Ξ①Ω**.



Q={{x}}, **R**={δ}={{a,b}}

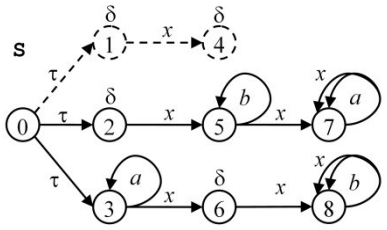
Рис. 3. **Ξ①Ω**

Вторая задача. Для решения второй задачи мы должны, во-первых, взять все трассы без не-отказов из $\mathbf{\Xi}_{\mathbf{L}}$, которые встречаются в **Ω**. Это будут безопасные конформные трассы, которые есть и в $\mathbf{\Xi}_{\mathbf{L}}$, и в **Ω**. Во-вторых, мы должны взять все их продолжения финальными суффиксами, которые есть в **Ω**. Поскольку пустая трасса тоже является финальным суффиксом, достаточно только «во-вторых». В результате получится множество:

$\mathbf{\Omega②\Xi}\triangleq\{\mu\cdot\lambda|\mu\in\mathbf{\Omega}\cap\mathbf{\Xi}_{\mathbf{L}}\ \&\ \lambda\in\boldsymbol{fs}(\mathbf{\Omega},\mu)\}$.

Множество безопасных трасс множества **Ω②Ξ** совпадает с множеством безопасных трасс без не-отказов множества **Ξ①Ω**. В обоих случаях это безопасные конформные трассы исходной спецификации. Однако теперь безопасность кнопок после этих трасс определяется по множеству **Ω②Ξ**, то есть по их финальным суффиксам в **Ω②Ξ**, которые такие же, как в f-модели **Ω**. Поэтому набор тестов, сгенерированный по всем безопасным трассам **Ω②Ξ** является полным для исходной спецификации (и ее f-модели **Ω**).

В качестве примера можно рассмотреть спецификацию на рис. 4. Здесь пунктиром отмечены переходы и состояния, удаляемые в ∇-пополнении **Ξ**: трасса δxδ неконформна. Неконформные трассы удаляются и из f-модели **Ω** при построении **Ω②Ξ**. В этом примере имеем $\boldsymbol{Safe}_{\gamma\Delta}(\mathbf{\Omega})\supset(\boldsymbol{Safe}_{\gamma\Delta}(\mathbf{\Xi}))_{\mathbf{L}}$.



Q={{x}}, **R**={δ}={{a,b}}

Рис. 4. **Ω②Ξ**

Проблема лишь в том, какие классы безопасно-тестируемых ***SafeImp***(**Ω②Ξ**) и конформных ***ConfImp***(**Ω②Ξ**) реализаций определяет само множество **Ω②Ξ**. Класс безопасно-тестируемых реализаций не сужается: ***SafeImp***(**Ω**)=***SafeImp***(**Σ**,***safe by***)⊆***SafeImp***(**Ω②Ξ**). Это объясняется тем, что, удаляя трассы из **Ω**, мы оставляем все финальные суффиксы остающихся трасс. Тем самым, мы, быть может, только ослабляем требования к реализации по безопасности. Также очевидно, что не меняется подмножество конформных реализаций множества безопасно-тестируемых реализация для исходной спецификации: ***ConfImp***(**Ω②Ξ**)∩***SafeImp***(**Ω**)=***ConfImp***(**Ω**). Остается открытым вопрос о том, не появляются в расширенном классе безопасно-тестируемых реализаций новые конформные реализации:

(***SafeImp***(**Ω②Ξ**)***SafeImp**(**Ω**))∩***ConfImp***(**Ω②Ξ**
= ***ConfImp***(**Ω②Ξ**)***ConfImp**(**Ω**) = ∅?

Здесь следует отметить, что множество **Ω②Ξ** в общем случае не является f-моделью. Это следствие того, что ∇-пополнение в исходной **R**/**Q**-семантике может не существовать. В частности, может быть нарушена **R**-полнота множества $(\mathbf{\Omega②\Xi})_{\mathbf{R}}$ в **Ω②Ξ**: могут не сохраняться финальные суффиксы при вставке **R**-отказа $\boldsymbol{i}_{\mathbf{R}}$-операцией (пример в [17] на рис.15). Это является еще одной проблемой.

9. RTS-модели спецификации

Как было отмечено в п.4.1, LTS-модель является наиболее «наглядной» моделью. Кроме того, LTS-модель является способом конечного представления регулярных трассовых моделей. Этот способ, однако, обладает одним существенным недостатком: LTS-модель, вообще говоря, недетерминирована: трасса может заканчиваться не в одном, а в нескольких состояниях. Работать с такими трассами на LTS неудобно. В то же время этот недетерминизм вовсе не является неизбежным следствием недетерминизма моделируемой системы. Причина недетерминизма LTS-модели в том, что наблюдения делятся на два вида: действия и отказы, которые существенно различным образом отображаются в LTS-модели. Если трасса продолжается как отказом R, так и действием z∈R, то эти два продолжения не могут быть определены в одном и том же состоянии LTS-модели.

9.1. Порождающий граф

С другой стороны, для любого множества последовательностей (в том числе для трассовой модели) всегда существует порождающий его граф, в котором выделены множества начальных и конечных вершин. Такой граф всегда можно построить с одной начальной вершиной[6]. Поэтому его можно рассматривать как LTS, если вершины графа назвать состояниями LTS, дуги – переходами, а переходы, соответствующие непомеченным дугам, пометить символом τ. Будем считать, что порождающий граф задан парой: LTS и множество ее конечных состояний.

Множество последовательностей префикс-замкнуто тогда и только тогда, когда существует порождающий ее граф, в котором все вершины конечные. Такой граф будем задавать только LTS, порождаемое множество – это множество простых трасс этой LTS.

Любой порождающий граф можно детерминизировать, то есть преобразовать в детерминированную LTS с выделенным множеством ее конечных состояний. При этом сохраняется множество порождающих трасс. Процедуру детерминизации мы опишем ниже.

Поскольку каждая рассматриваемая нами выше трассовая модель префикс-замкнута, порождающий ее граф можно задать одной LTS, но не в исходном алфавите **L**, а (для **R**/**Q**-семантики) в алфавите **L**∪**R**∪{Δ}.

Детерминированную LTS в алфавите **L**∪**R**∪{Δ} будем называть RTS (Refusal Transition System).

Нужно отметить, что за детерминизм модели приходится чем-то «жертвовать»: не любая LTS в алфавите **L**∪**R**∪{Δ} является графом, порождающим трассовую модель в **R**/**Q**-семантике. Характеристические свойства RTS зависят от типа трассовой модели, множество трасс которой она представляет.

Для спецификации мы рассматривали три вида таких моделей: **R**-модель, p-модель и f-модель, причем для первых двух моделей отдельно задается отношение ***safe by***.

8. Отношение safe by

Отношение ***safe by*** однозначно определяется для **R**-кнопок и разрушающих **Q**-кнопок, то есть вычисляется по модели (**R**- или p-модели). Поэтому достаточно определить только те **Q**-кнопки, которые неразрушающие, но опасные по ***safe by*** после каждой неразрушающей трассы σ. Это задается множеством трасс вида σ·⟨Q⟩, где σ неразрушающая трасса спецификации, а Q∈**Q** кнопка, неразрушающая, но опасная по ***safe by*** после σ. Для такого

[6] Достаточно добавить новую начальную вершину и провести из нее непомеченные дуги во все старые начальные вершины.

множества тоже существует порождающий граф, но только в нем могут быть уже не конечные вершины. Поскольку неразрушающие трассы не содержат **Q**-отказов, все конечные вершины этого графа терминальные.

Будем говорить, что отношение ***safe by*** регулярно, если это множество трасс регулярно, то есть существует конечный порождающий граф. В [17] было введено понятие ограниченного отношения ***safe by***, когда безопасность кнопок одинакова после трасс, заканчивающихся в одном множестве состояний LTS-спецификации (эти трассы заканчиваются в одном состоянии RTS, которую мы ниже построим по LTS-спецификации). Для конечной LTS спецификации ограниченный ***safe by*** регулярный, хотя обратное, вообще говоря, не верно.

9. *Детерминизация порождающего графа*

Сначала определим формально процедуру детерминизации порождающего графа, заданного LTS $\mathbf{T}=LTS(V_\mathbf{T},\mathbf{L}_\mathbf{T},E_\mathbf{T},t_0)$ и множеством конечных состояний $F\subseteq V_\mathbf{T}$. Эта процедура ***determ***(**T**,F) строит детерминированную LTS с тем же множеством простых трасс: $LTS(P(V_\mathbf{T}),\mathbf{L}_\mathbf{T},E,t_0$ ***after*** $\epsilon)$, где множество переходов E определяется как наименьшее множество, порождаемое следующим правилом вывода: $\forall A,B\in P(V_\mathbf{T})\ \forall u\in\mathbf{L}_\mathbf{T}$

$B=\cup(A$ ***after*** $\langle u\rangle)\neq\varnothing \quad\vdash\quad A—u\rightarrow B.$

Состояние A этой LTS объявляется конечным, если $A\cap F\neq\varnothing$. Если $F=V_\mathbf{T}$, то на этом процедура детерминизации заканчивается. Иначе, нам нужно удалить из построенной LTS все состояния, из которых недостижимо хотя бы одно конечное состояние; естественно, вместе с удаляемым состоянием удаляются и все входящие в него и выходящие из него переходы.

Мы построим детерминированный порождающий граф для **R**-модели, p-модели и f-модели. Поскольку все эти модели префикс-замкнуты, будет строиться только RTS, все ее состояния считаются конечными.

10. *RTS для R-модели*

Построим RTS для **R**-модели **Σ**, заданной как множество $\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$-трасс LTS **S** в алфавите **L**. Эту RTS обозначим ***rts***(**S**). Напомним, что для определения $\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$-трасс LTS **S** строится LTS $\mathbf{S_R}=LTS(V_{\mathbf{SR}},\mathbf{L}\cup\mathbf{R}\cup\{\Delta\},E_{\mathbf{SR}},s_0)$. Для этого добавляются в каждом стабильном состоянии LTS **S** петли s—R→s, помеченные **R**-отказами, порождаемыми в этом состоянии, добавляется новое терминальное состояние ω, перенаправляются в это состояние все γ-переходы, а также проводятся в него Δ-переходы из дивергентных состояний. По определению $\mathbf{\Sigma}=\mathit{tr}(\mathbf{S_R})$. Для построения ***rts***(**S**) выполняется преобразование детерминизации LTS $\mathbf{S_R}$: ***rts***(**S**)= ***determ***($\mathbf{S_R}$).

RTS ***rts***(**S**) будет конечной тогда и только тогда, когда конечна LTS $\mathbf{S_R}$. А для этого достаточно конечности исходной LTS **S** и семейства **R** (поскольку в $\mathbf{S_R}$ добавляются петли по **R**-отказам).

11. *RTS для p-модели*

Поскольку любая RTS **U** – это детерминированная LTS, каждая ее трасса σ заканчивается только в одном состоянии s, и для всех трасс, заканчивающихся в s, безопасность кнопок после них по отношению $\mathit{safe}_{\gamma\Delta}$ зависит только от этого состояния. Определим: $\forall s\in V_\mathbf{U}\ \forall P\in\mathbf{R}\cup\mathbf{Q}\ \forall u\in\mathbf{L}\cup\mathbf{R}$

$P\ \mathit{safe}_{\gamma\Delta}\ s \triangleq s—\Delta\nrightarrow\ \&\ \forall z\in P\ s=\langle z,\gamma\rangle\not\Rightarrow,$

$u\ \mathit{safe}_{\gamma\Delta}\ s \triangleq \exists P\in$ ***but***$(u)\ P\ \mathit{safe}_{\gamma\Delta}\ s.$

Очевидно, если **U** ***after*** σ = {s}, то:

$P\ \mathit{safe}_{\gamma\Delta}$ ***tr***(**U**) ***after*** $\sigma \Leftrightarrow P\ \mathit{safe}_{\gamma\Delta}\ s,\ u\ \mathit{safe}_{\gamma\Delta}$ ***tr***(**U**) ***after*** $\sigma \Leftrightarrow u\ \mathit{safe}_{\gamma\Delta}\ s.$

Определим преобразование любой RTS, то есть детерминированной LTS $\mathbf{T}=LTS(V_\mathbf{T},\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\},E_\mathbf{T},t_0)$, которое оставляет только переходы по разрушению, по дивергенции при отсутствии разрушения, по действию с последующим разрушением, а также по неразрушающим наблюдениям: ***delete***(**T**)=$LTS(V_\mathbf{T},\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\},E,t_0)$, где множество переходов E определяется как наименьшее множество, порождаемое следующими правилами вывода: $\forall a,b\in V_\mathbf{T}\ \forall u\in\mathbf{L}\cup\mathbf{R}$

$a—\gamma\rightarrow b \quad\vdash\quad a—\gamma\rightarrow b,$

$a—\gamma\nrightarrow\ \&\ a—\Delta\rightarrow b \quad\vdash\quad a—\Delta\rightarrow b,$

$a—\gamma\nrightarrow\ \&\ a—\Delta\nrightarrow\ \&\ a—u\rightarrow b\ \&\ b—\gamma\rightarrow \quad\vdash\quad a—u\rightarrow b,$

$a—\gamma\nrightarrow\ \&\ a—\Delta\nrightarrow\ \&\ a—u\rightarrow b\ \&\ u\ \mathit{safe}_{\gamma\Delta}\ a \quad\vdash\quad a—u\rightarrow b.$

С помощью этого преобразования RTS для p-модели ***ptr***(**Σ**), где **Σ**=***tr***($\mathbf{S_R}$), строится так: ***prts***(**S**)=***delete***(***rts***(**S**))=***delete***(***determ***($\mathbf{S_R}$)).

Если ***rts***(**S**) конечна, то ***prts***(**S**) конечная. Обратное, вообще говоря, не верно, поскольку в ***rts***(**S**) может быть бесконечное число переходов, но только конечное число их лежит на маршрутах с предфинальными трассами.

12. *RTS для f-модели*

Теперь построим RTS для f-модели ***ftr***(**Σ**,***safe by***), где **Σ**=***tr***($\mathbf{S_R}$). Эту RTS обозначим ***frts***(**S**,***safe by***) и будем называть финальной RTS. По определению ***ftr***(**Σ**,***safe by***) = ***ftr***(***ptr***(**Σ**),***safe by***). Будем считать, что у нас уже построена RTS **P**=***prts***(**S**) для p-модели ***ptr***(**Σ**). Также будем считать, что отношение ***safe by*** задано парой (**G**,F), где **G** RTS, а F множество ее конечных состояний. Определим преобразование ***final***(**P**,**G**,F), которое строит новую RTS **F** в алфавите $\mathbf{L}\cup\mathbf{R}\cup\mathbf{Q}\cup\{\Delta\}$. Состояниями RTS будут все пары состояний **P** и **G**, а также два новых выделенных состояния: терминальное состояние ω и состояние γ с единственным выходящим переходом γ—γ→ω. Начальное состояние – пара начальных состояний.

Формально $\mathbf{F} = LTS((V_P \times V_G) \cup \{\omega,\gamma\}, \mathbf{L} \cup \mathbf{R} \cup \mathbf{Q} \cup \{\Delta\}, E_F, (p_0,g_0))$, где множество переходов E_F – наименьшее множество, порождаемое следующими правилами вывода: $\forall p \in V_P \ \forall g \in V_G \ \forall u \in \mathbf{L} \cup \mathbf{R} \ \forall Q \in \mathbf{Q}$

1. $p \text{—} u \rightarrow p` \ \& \ g \text{—} u \rightarrow g` \quad \vdash \quad (p,g) \text{—} u \rightarrow (p`,g`)$,
2. $p \text{—} \Delta \rightarrow p` \quad \vdash \quad (p,g) \text{—} \Delta \rightarrow \omega$,
3. $p \text{—} u \rightarrow p` \ \& \ p` \text{—} \gamma \rightarrow \quad \vdash \quad (p,g) \text{—} u \rightarrow \gamma$,
4. $\quad \vdash \quad \gamma \text{—} \gamma \rightarrow \omega$,
5. $g \text{—} Q \rightarrow g` \quad \vdash \quad (p,g) \text{—} Q \rightarrow \omega$.

Правило вывода 1 гарантирует наличие в RTS всех безопасных трасс. Правила вывода 2,3 и 4 сохраняют все финальные суффиксы безопасных трасс, которые были в p-модели. Тем самым, первые четыре правила сохраняют все трассы p-модели. Правило вывода 5 добавляет Q-финальные суффиксы трасс, задаваемые отношением ***safe by***. Заметим, что в правиле 2 $p`=\{\omega\}$, а в правиле 5 состояние g` конечное. Состояния, не достижимые по финальным трассам, будут недостижимы, то есть мы получим RTS, множество простых трасс которой является f-моделью, а это и есть финальная RTS.

Все состояния RTS **P** будут входить в левые части пар, представляющих достижимые состояния RTS **F** (кроме состояния $\{\omega\}$, вместо которого будет состояние ω, и начал γ-переходов, вместо которых будет состояние γ). Также все состояния RTS **G** будут входить в правые части пар, представляющих достижимые состояния RTS **F** (кроме конечных состояний, вместо которых будет состояние ω).

Поскольку RTS – это детерминированная LTS, каждая ее трасса σ заканчивается только в одном состоянии s, и для всех трасс, заканчивающихся в s, безопасность кнопок после них по отношению $\boldsymbol{safe}_{\gamma\Delta}$ зависит только от этого состояния. В финальной RTS отношение ***safe by*** определяется равным отношению ***safe in***, и безопасность кнопки по ***safe in*** после трассы зависит только от состояния в конце трассы. Определим: $\forall s \in V_U \ \forall P \in \mathbf{R} \cup \mathbf{Q}$: P ***safe in*** s $\triangleq$ P $\boldsymbol{safe}_{\gamma\Delta}$ s & $(P \in \mathbf{Q} \Rightarrow s \text{—} P \not\rightarrow)$.

Очевидно, если **F** ***after*** $\sigma = \{s\}$, то P ***safe in tr***(**F**) ***after*** $\sigma \Leftrightarrow$ P ***safe in*** s.

Исследуем вопрос о конечности финальной RTS ***frts***(**S**,***safe by***). Достаточным условием является конечность каждого шага построения:

$\mathbf{S} \text{—}_{\mathbf{R}} \rightarrow \mathbf{S}_{\mathbf{R}} \text{—} \boldsymbol{determ} \rightarrow \boldsymbol{rts}(\mathbf{S}) \text{—} \boldsymbol{delete} \rightarrow \boldsymbol{prts}(\mathbf{S}) \text{—} \boldsymbol{final} \rightarrow \boldsymbol{frts}(\mathbf{S},\boldsymbol{safe\ by})$.

Для конечности ***prts***(**S**) достаточно конечности исходной LTS **S** (конечность множества достижимых переходов) и семейства **R**. При этих условиях для конечности ***frts***(**S**) достаточно конечности регулярности отношения ***safe by***. Заметим, что мы не требует конечности алфавита **L** и конечности семейства **Q**.

10. Заключение

В работе исследованы различные модели спецификации. В качестве результата предложены финальные трассовые и RTS-модели, последние дают возможность конечного представления регулярных финальных трассовых моделей.

Финальная RTS как модель спецификации обладает целым рядом полезных для тестирования свойств, выгодно отличающих ее от LTS-модели, **R**- и p-моделей, а также от RTS для **R**- и p-моделей:

1. **Детерминизм**. RTS детерминирована, следовательно, каждая трасса, по которой нужно генерировать тесты, заканчивается в одном состоянии.
2. **Безопасные трассы**. Безопасные трассы спецификации – это все ее простые трассы, заканчивающиеся в состояниях, отличных от ω и γ.
3. **Безопасные кнопки**. Кнопка P безопасна по ***safe by*** в финальной RTS **F** после неразрушающей трассы σ тогда и только тогда, когда она безопасна по ***safe in*** в конечном состоянии трассы s: P ***safe by tr***(**F**) ***after*** σ $\Leftrightarrow$ P ***safe in*** s, где **F** ***after*** $\sigma = \{s\}$.

Также предложены алгоритмы преобразования LTS-модели с заданным отношением ***safe by*** в финальную RTS-модель. Если конечна исходная LTS **S** и семейство **R**, а отношение ***safe by*** регулярно, то финальная RTS конечна и строится за конечное время.

В то же время множество финальных трасс, хотя и достаточно, но, вообще говоря, не необходимо для генерации тестов. Это объясняется тем, что среди безопасных трасс спецификации могут встречаться неконформные трассы, то есть трассы, которых не может быть ни в одной конформной реализации. Понятно, что по таким трассам не нужно генерировать тесты. Такие трассы можно удалить из множества трасс, по которым генерируются тесты с помощью ∇-пополнения. Исследованию неконформных трасс спецификации и построению алгоритмов, удаляющих такие трассы из спецификации, посвящены наши работы [16],[17].

Доказательства утверждений

13. Доказательство Теорема 2:

1. Пусть $\langle\gamma\rangle \in \boldsymbol{\Psi}$.

Тогда $\langle\gamma\rangle \in \boldsymbol{ptr}(\boldsymbol{\Psi})$. Имеем $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi}) = \varnothing = \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Psi}))$.

Утверждение доказано.

2. Пусть $\langle\gamma\rangle \notin \boldsymbol{\Psi}$.

Будем вести доказательство индукцией по трассе σ.

2.1. База индукции.

Из $\langle\gamma\rangle\notin\boldsymbol{\Psi}$ следует $\epsilon\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

Из $\langle\gamma\rangle\notin\boldsymbol{\Psi}$ следует $\langle\gamma\rangle\notin\boldsymbol{ptr}(\boldsymbol{\Psi})$, что влечет $\epsilon\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Psi}))$.

2.2. Шаг индукции. Пусть $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})\cap\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Psi}))$ и $P\in\mathbf{R}\cup\mathbf{Q}$.

2.2.1. Докажем, что $P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}\ \boldsymbol{after}\ \sigma \Rightarrow P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{ptr}(\boldsymbol{\Psi})\ \boldsymbol{after}\ \sigma$).

Допустим обратное. Тогда либо $\sigma\cdot\langle\Delta\rangle\in\boldsymbol{ptr}(\boldsymbol{\Psi})$, либо $\sigma\cdot\langle z,\gamma\rangle\in\boldsymbol{ptr}(\boldsymbol{\Psi})$ для некоторого $z\in P$. Поскольку $\boldsymbol{ptr}(\boldsymbol{\Psi})\subseteq\boldsymbol{\Psi}$, имеем $\sigma\cdot\langle\Delta\rangle\in\boldsymbol{\Psi}$, либо $\sigma\cdot\langle z,\gamma\rangle\in\boldsymbol{\Psi}$ для некоторого $z\in P$. Но это противоречит условию $P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}\ \boldsymbol{after}\ \sigma$. Следовательно, наше допущение не верно, и импликация доказана.

2.2.2. Докажем, что $P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}\ \boldsymbol{after}\ \sigma \Leftarrow P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{ptr}(\boldsymbol{\Psi})\ \boldsymbol{after}\ \sigma$).

Допустим обратное. Тогда либо $\sigma\cdot\langle\Delta\rangle\in\boldsymbol{\Psi}$, либо $\sigma\cdot\langle z,\gamma\rangle\in\boldsymbol{\Psi}$ для некоторого $z\in P$. Поскольку $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$, имеем либо $\langle\Delta\rangle\in\boldsymbol{fs}(\boldsymbol{\Psi},\sigma)$, либо $\langle z,\gamma\rangle\in\boldsymbol{fs}(\boldsymbol{\Psi},\sigma)$ для некоторого $z\in P$. Но тогда либо $\sigma\cdot\langle\Delta\rangle\in\boldsymbol{ptr}(\boldsymbol{\Psi})$, либо $\sigma\cdot\langle z,\gamma\rangle\in\boldsymbol{ptr}(\boldsymbol{\Psi})$ для некоторого $z\in P$. Но это противоречит условию $P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{ptr}(\boldsymbol{\Psi})\ \boldsymbol{after}\ \sigma$. Следовательно, наше допущение не верно, и обратная импликация доказана.

2.2.3. Для $u\in\mathbf{L}\cup\mathbf{R}$ докажем $\sigma\cdot\langle u\rangle\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi}) \Rightarrow \sigma\cdot\langle u\rangle\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Psi}))$.

Поскольку $\sigma\cdot\langle u\rangle\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$, найдется такое $P\in\boldsymbol{but}(u)$, что $P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}\ \boldsymbol{after}\ \sigma$. По доказанному в п.2.2.1 $P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{ptr}(\boldsymbol{\Psi})\ \boldsymbol{after}\ \sigma$). Поскольку $\sigma\cdot\langle u\rangle\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$, имеем $\sigma\cdot\langle u\rangle\in\boldsymbol{ptr}(\boldsymbol{\Psi})$. Следовательно, $\sigma\cdot\langle u\rangle\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Psi}))$.

2.2.4. Для $u\in\mathbf{L}\cup\mathbf{R}$ докажем $\sigma\cdot\langle u\rangle\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi}) \Leftarrow \sigma\cdot\langle u\rangle\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Psi}))$.

Поскольку $\sigma\cdot\langle u\rangle\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Psi}))$, найдется такое $P\in\boldsymbol{but}(u)$, что $P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{ptr}(\boldsymbol{\Psi})\ \boldsymbol{after}\ \sigma$. По доказанному в п.2.2.2 $P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}\ \boldsymbol{after}\ \sigma$). Поскольку $\sigma\cdot\langle u\rangle\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Psi}))$, имеем $\sigma\cdot\langle u\rangle\in\boldsymbol{ptr}(\boldsymbol{\Psi})$. Поскольку $\boldsymbol{ptr}(\boldsymbol{\Psi})\subseteq\boldsymbol{\Psi}$, имеем $\sigma\cdot\langle u\rangle\in\boldsymbol{\Psi}$. Следовательно, $\sigma\cdot\langle u\rangle\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

14. Доказательство Теорема 3:

1. Прямая импликация. Пусть $\boldsymbol{ptr}(\boldsymbol{\Psi})=\boldsymbol{\Psi}$.

Докажем, что $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi}) = \{\sigma\in\boldsymbol{\Psi}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Psi}\}$.

1.1. Если $\langle\gamma\rangle\in\boldsymbol{\Psi}$, то $\boldsymbol{ptr}(\boldsymbol{\Psi})=\{\epsilon,\langle\gamma\rangle\}$. Тогда $\boldsymbol{\Psi}=\{\epsilon,\langle\gamma\rangle\}$ и утверждение очевидно.

1.2. Пусть $\langle\gamma\rangle\notin\boldsymbol{\Psi}$. По определению $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$ влечет $\sigma\in\boldsymbol{\Psi}\cap(\mathbf{L}\cup\mathbf{R})^*$.

Поскольку $\langle\gamma\rangle\notin\boldsymbol{fs}(\boldsymbol{\Psi},\sigma)$, а $\boldsymbol{ptr}(\boldsymbol{\Psi})=\boldsymbol{\Psi}$, имеем $\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Psi}$.

Следовательно, $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi}) \subseteq \{\sigma\in\boldsymbol{\Psi}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Psi}\}$.

Покажем обратную вложенность. Допустим противное: существует трасса $\sigma\in\boldsymbol{\Psi}\cap(\mathbf{L}\cup\mathbf{R})^*$ такая, что $\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Psi}$, но $\sigma\notin\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

Поскольку $\langle\gamma\rangle\notin\boldsymbol{\Psi}$, имеем $\epsilon\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$. Следовательно, у трассы σ есть префикс $\mu\cdot\langle u\rangle$ такой, что $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$, но $\mu\cdot\langle u\rangle\notin\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

Следовательно, либо $u=\Delta$, либо $u=\gamma$, либо $u\in\mathbf{L}\cup\mathbf{R}$ и $u\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}\ \boldsymbol{after}\ \mu$. Случаев $u=\Delta$ и $u=\gamma$ быть не может, так как $\mu\cdot\langle u\rangle\le\sigma\in(\mathbf{L}\cup\mathbf{R})^*$.

Следовательно, $u\in\mathbf{L}\cup\mathbf{R}$ и $u\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}\ \boldsymbol{after}\ \mu$.

Если $\mu\cdot\langle u,\gamma\rangle\notin\boldsymbol{\Psi}$, то $\langle u\rangle\notin\boldsymbol{fs}(\boldsymbol{\Psi},\mu)$, поэтому $\mu\cdot\langle u\rangle\notin\boldsymbol{ptr}(\boldsymbol{\Psi})=\boldsymbol{\Psi}$, что противоречит префикс-замкнутости $\boldsymbol{\Psi}$, поскольку $\mu\cdot\langle u\rangle\le\sigma\in\boldsymbol{\Psi}$.

Если $\mu\cdot\langle u,\gamma\rangle\in\boldsymbol{\Psi}$, то возможны два варианта: 1) $\mu\cdot\langle u\rangle=\sigma$ и 2) $\mu\cdot\langle u\rangle<\sigma$. Случай 1 противоречит $\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Psi}$. В случае 2 найдется такое $v\in\mathbf{L}\cup\mathbf{R}$, что $\mu\cdot\langle u,v\rangle\le\sigma$. Но $\langle u,v\rangle\notin\boldsymbol{fs}(\boldsymbol{\Psi},\mu)$, поэтому $\mu\cdot\langle u,v\rangle\notin\boldsymbol{ptr}(\boldsymbol{\Psi})=\boldsymbol{\Psi}$, что противоречит префикс-замкнутости $\boldsymbol{\Psi}$, поскольку $\mu\cdot\langle u,v\rangle\le\sigma\in\boldsymbol{\Psi}$.

Мы пришли к противоречию, следовательно, наше допущение не верно и обратная вложенность доказана.

2. Обратная импликация. Пусть выполнены условия обратной импликации.

Докажем, что $\boldsymbol{ptr}(\boldsymbol{\Psi})=\boldsymbol{\Psi}$.

2.1. Пусть $\langle\gamma\rangle\in\boldsymbol{\Psi}$.

Тогда $\boldsymbol{\Psi}=\{\epsilon,\langle\gamma\rangle\}$ и $\boldsymbol{ptr}(\boldsymbol{\Psi})=\{\epsilon,\langle\gamma\rangle\}$.

2.2. Пусть $\langle\gamma\rangle\notin\boldsymbol{\Psi}$.

Тогда $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=\{\sigma\in\boldsymbol{\Psi}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Psi}\}$, трассы из $\boldsymbol{\Psi}$ допустимы и после отказа в них нет разрушения.

Поскольку по определению $\boldsymbol{ptr}(\boldsymbol{\Psi})\subseteq\boldsymbol{\Psi}$, нам нужно доказать обратную вложенность. Допустим, это не так: существует трасса $\sigma\in\boldsymbol{\Psi}$ такая, что $\sigma\notin\boldsymbol{ptr}(\boldsymbol{\Psi})$. Поскольку по определению предфинальности $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})\subseteq\boldsymbol{ptr}(\boldsymbol{\Psi})$, имеем $\sigma\notin\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$. Поскольку $\langle\gamma\rangle\notin\boldsymbol{\Psi}$, имеем $\epsilon\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$. Следовательно, трассу σ можно представить в виде $\sigma=\mu\cdot\lambda$ таком, что $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$ максимальный безопасный префикс σ, но $\lambda\notin\boldsymbol{fs}(\boldsymbol{\Psi},\mu)$.

Поскольку все трассы из $\boldsymbol{\Psi}$ допустимы, возможны только следующие случаи: 1) $\lambda=\lambda_1\cdot\langle\Delta\rangle$, либо 2) $\lambda=\lambda_1\cdot\langle\gamma\rangle$, либо 3) $\lambda=\lambda_1$, где $\lambda_1\in(\mathbf{L}\cup\mathbf{R})^*$.

Поскольку $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=\{\sigma\in\boldsymbol{\Psi}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Psi}\}$, в этих случаях имеет место: 1) $\lambda_1=\epsilon$ и $\lambda=\langle\Delta\rangle$, 2) $\lambda_1=\epsilon$ и $\lambda=\langle\gamma\rangle$, 3) $\lambda=\langle u\rangle$, где $u\in\mathbf{L}\cup\mathbf{R}$ и $\mu\cdot\langle u,\gamma\rangle\in\boldsymbol{\Psi}$.

Случай 1 противоречит $\lambda=\langle\Delta\rangle\notin\boldsymbol{fs}(\boldsymbol{\Psi},\mu)$.

В случае 2 трасса μ не может быть пустой, так как $\langle\gamma\rangle \notin \boldsymbol{\Psi}$, и не может заканчиваться отказом, так как в трассах из $\boldsymbol{\Psi}$ после отказа нет разрушения. Следовательно, трасса μ заканчивается действием, за которым в $\boldsymbol{\Psi}$ следует разрушение, что противоречит $\mu \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

Рассмотрим случай 3. Поскольку в трассах из $\boldsymbol{\Psi}$ после отказа нет разрушения, должно быть $u \in \mathbf{L}$. Но тогда $\lambda=\langle u\rangle \in \boldsymbol{fs}(\boldsymbol{\Psi},\mu)$, что неверно.

Мы пришли к противоречию, следовательно, наше допущение не верно и обратная вложенность доказана.

3. Необходимость условий теоремы для обратной импликации показывается следующими примерами. В каждом из этих примеров нарушено только одно условие обратной импликации.

3.1. Выполнены все условия, кроме условия на множество неразрушающих трасс при $\langle\gamma\rangle \in \boldsymbol{\Psi}$.

Пример $\boldsymbol{\Psi}=\{\epsilon,\langle P\rangle,\langle z\rangle,\langle z,\gamma\rangle\}$, где $P \in \mathbf{R}$ и $z \in P$.

$\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=\{\epsilon\} \neq \{\epsilon,\langle P\rangle\}=\{\sigma \in \boldsymbol{\Psi} \cap (\mathbf{L}\cup\mathbf{R})^{*} \mid \sigma\cdot\langle\gamma\rangle \notin \boldsymbol{\Psi}\}$,

$\boldsymbol{ptr}(\boldsymbol{\Psi})=\{\epsilon,\langle z\rangle,\langle z,\gamma\rangle\} \neq \boldsymbol{\Psi}$.

3.2. Выполнены все условия, кроме условия $\boldsymbol{\Psi}=\{\epsilon,\langle\gamma\rangle\}$ при $\langle\gamma\rangle \in \boldsymbol{\Psi}$.

Пример $\boldsymbol{\Psi}=\{\epsilon,\langle\gamma\rangle,\langle u\rangle\}$, где $u \in \mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$. $\boldsymbol{ptr}(\boldsymbol{\Psi})=\{\epsilon,\langle\gamma\rangle\} \neq \boldsymbol{\Psi}$.

3.3. Выполнены все условия, кроме допустимости трасс.

Пример 1 $\boldsymbol{\Psi}=\{\epsilon,\langle\Delta\rangle,\langle\Delta,u\rangle\}$, где $u \in \mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$.

$\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=\{\sigma \in \boldsymbol{\Psi} \cap (\mathbf{L}\cup\mathbf{R})^{*} \mid \sigma\cdot\langle\gamma\rangle \notin \boldsymbol{\Psi}\}=\{\epsilon\}$, $\boldsymbol{ptr}(\boldsymbol{\Psi})=\{\epsilon,\langle\Delta\rangle\}$.

Пример 2 $\boldsymbol{\Psi}=\{\epsilon,\langle z\rangle,\langle z,\gamma\rangle,\langle z,\gamma,u\rangle\}$, где $z \in \mathbf{L}$ и $u \in \mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\}$.

$\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=\{\sigma \in \boldsymbol{\Psi} \cap (\mathbf{L}\cup\mathbf{R})^{*} \mid \sigma\cdot\langle\gamma\rangle \notin \boldsymbol{\Psi}\}=\{\epsilon\}$,

$\boldsymbol{ptr}(\boldsymbol{\Psi})=\{\epsilon,\langle z\rangle,\langle z,\gamma\rangle\} \neq \boldsymbol{\Psi}$.

3.4. Выполнены все условия, кроме отсутствия разрушения после отказа.

Пример $\boldsymbol{\Psi}=\{\epsilon,\langle P\rangle,\langle P,\gamma\rangle,\langle z\rangle,\langle z,\gamma\rangle\}$, где $P \in \mathbf{R}$ и $z \in P$.

$\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=\{\sigma \in \boldsymbol{\Psi} \cap (\mathbf{L}\cup\mathbf{R})^{*} \mid \sigma\cdot\langle\gamma\rangle \notin \boldsymbol{\Psi}\}=\{\epsilon\}$,

$\boldsymbol{ptr}(\boldsymbol{\Psi})=\{\epsilon,\langle z\rangle,\langle z,\gamma\rangle\}$.

15. Доказательство Теорема 4:

1. Непустота множества. Поскольку ***d***-замыкание только добавляет трассы, ***d***-замыкание непустого множества не пусто.
2. Префикс-замкнутость. Поскольку при ***d***-замыкании отказ P удаляется после трассы μ перед каждой продолжающей ее трассой λ, ***d***-замыкание префикс-замкнутого множества префикс-замкнуто.

3. Допустимость. Поскольку при ***d***-замыкании только удаляются отказы из трасс, ***d***-замыкание множества допустимых трасс добавляет только допустимые трассы.
4. Согласованность. Поскольку при ***d***-замыкании только удаляются отказы из трасс, ***d***-замыкание множества согласованных трасс добавляет только согласованные трассы.
5. **R**-конвергентность. Поскольку при ***d***-замыкании отказ P удаляется после трассы μ перед каждой продолжающей ее трассой λ, ***d***-замыкание **R**-конвергентного множества **R**-конвергентно.
6. Замкнутость. По определению ***d***-замыкание любого множества трасс ***d***-замкнуто.
7. **R**-полнота. Поскольку при ***d***-замыкании отказ P удаляется после трассы μ перед каждой продолжающей ее трассой λ, ***d***-замыкание полного множества полно.

16. Доказательство Теорема 5:

Если $\langle\gamma\rangle \in \boldsymbol{\Sigma}$, то $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})=\varnothing$, поэтому $\boldsymbol{ptr}(\boldsymbol{\Sigma})=\{\epsilon,\langle\gamma\rangle\}$ и выполнение всех свойств p-модели очевидно. Далее будем считать, что $\langle\gamma\rangle \notin \boldsymbol{\Sigma}$.

1. Непустота множества. Поскольку трассовая модель не пуста и префикс-замкнута $\epsilon \in \boldsymbol{\Sigma}$. Поэтому по определению предфинальных трасс $\epsilon \in \boldsymbol{ptr}(\boldsymbol{\Sigma})$, следовательно, множество $\boldsymbol{ptr}(\boldsymbol{\Sigma})$ не пусто.
2. Префикс-замкнутость. Множество $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$ префикс-замкнуто по определению. По определению префикс финального продолжения неразрушающей трассы является либо префиксом этой трассы, либо ее финальным продолжением. Тем самым множество $\boldsymbol{ptr}(\boldsymbol{\Sigma})$ префикс-замкнуто.
3. Допустимость. Все трассы трассовой модели допустимы, в том числе и ее предфинальные трассы.
4. Согласованность. Все трассы трассовой модели согласованы, в том числе и ее предфинальные трассы.
5. **R**-конвергентность. Пусть σ предфинальная трасса. Рассмотрим два случая.

5.1. $\sigma \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$.

5.1.1. Если **R**-кнопка P разрушающая в $\boldsymbol{\Sigma}$ после σ, то $\sigma\cdot\langle\Delta\rangle \in \boldsymbol{\Sigma}$ или $\sigma\cdot\langle z,\gamma\rangle \in \boldsymbol{\Sigma}$ для некоторого $z \in P$. А тогда $\sigma\cdot\langle\Delta\rangle \in \boldsymbol{ptr}(\boldsymbol{\Sigma})$ или $\sigma\cdot\langle z,\gamma\rangle \in \boldsymbol{ptr}(\boldsymbol{\Sigma})$ для некоторого $z \in P$.

5.1.2. Если **R**-кнопка P неразрушающая в $\boldsymbol{\Sigma}$ после σ, то по **R**-конвергентности трассовой модели в $\boldsymbol{\Sigma}$ имеется продолжение трассы σ неразрушающим наблюдением $u \in \boldsymbol{obs}(P)$, и трасса $\sigma\cdot\langle u\rangle$ неразрушающая и, следовательно, предфинальная.

В обоих случаях трасса σ **R**-конвергентна в $\boldsymbol{ptr}(\boldsymbol{\Sigma})$.

5.2. $\sigma \notin \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$.

Тогда по определению финального продолжения трасса σ либо заканчивается дивергенцией или разрушением, либо продолжается разрушением и, следовательно, **R**-конвергентна в $\boldsymbol{ptr}(\boldsymbol{\Sigma})$.

6. **R**-полнота. Пусть трасса $\mu\cdot\lambda \in \boldsymbol{ptr}(\boldsymbol{\Sigma})$, трасса μ заканчивается некоторым отказом и не продолжается в $\boldsymbol{ptr}(\boldsymbol{\Sigma})$ действиями из отказа P. Нужно доказать, что $\mu\cdot\langle P\rangle\cdot\lambda \in \boldsymbol{ptr}(\boldsymbol{\Sigma})$.

6.1. Покажем, что трасса $\mu \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$.

Трасса μ как префикс предфинальной трассы $\mu\cdot\lambda$ по доказанной префикс-замкнутости также предфинальна. А тогда, поскольку трасса μ заканчивается на отказ, по определению предфинальных трасс $\mu \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$.

6.2. Покажем, что P $\boldsymbol{safe}_{\gamma\Delta}$ $\boldsymbol{\Sigma}$ ***after*** μ. Действительно, в противном случае в $\boldsymbol{\Sigma}$ имелась бы либо 1) предфинальная трасса $\mu\cdot\langle\Delta\rangle$, либо 2) предфинальная трасса $\mu\cdot\langle z,\gamma\rangle$ для некоторого $z \in P$. Случая 1 не может быть, так как трасса μ заканчивается отказом и по согласованности **R**-модели не может продолжаться в $\boldsymbol{\Sigma}$ дивергенцией. В случае 2 $\mu\cdot\langle z,\gamma\rangle \in \boldsymbol{ptr}(\boldsymbol{\Sigma})$, чего быть не может, так как трасса μ не продолжается в $\boldsymbol{ptr}(\boldsymbol{\Sigma})$ действиями из отказа P.

6.3. Покажем, что трасса μ не продолжается в $\boldsymbol{\Sigma}$ действиями из отказа P.

Действительно, поскольку P $\boldsymbol{safe}_{\gamma\Delta}$ $\boldsymbol{\Sigma}$ ***after*** μ, такие продолжения были бы неразрушающими в $\boldsymbol{\Sigma}$ и, следовательно, финальными, что противоречит тому, что трасса μ не продолжается в $\boldsymbol{ptr}(\boldsymbol{\Sigma})$ действиями из отказа P.

6.4. Покажем, что $\mu\cdot\langle P\rangle \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$. Поскольку $\boldsymbol{\Sigma}$ **R**-конвергентна и трасса μ не продолжается в $\boldsymbol{\Sigma}$ действиями из отказа P, должно быть $\mu\cdot\langle P\rangle \in \boldsymbol{\Sigma}$. Поскольку P $\boldsymbol{safe}_{\gamma\Delta}$ $\boldsymbol{\Sigma}$ ***after*** μ, $\mu\cdot\langle P\rangle \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$.

Поскольку $\mu \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$, у предфинальной трассы $\mu\cdot\lambda$ имеется максимальный неразрушающий префикс $\mu\cdot\lambda_1$.

6.5. Покажем, что $\mu\cdot\langle P\rangle\cdot\lambda_1 \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$.

Поскольку $\mu\cdot\langle P\rangle\cdot\lambda \in \boldsymbol{\Sigma}$, по префикс-замкнутости **R**-модели $\mu\cdot\langle P\rangle\cdot\lambda_1 \in \boldsymbol{\Sigma}$. Допустим $\mu\cdot\langle P\rangle\cdot\lambda_1 \notin \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$. Тогда, поскольку $\mu\cdot\langle P\rangle \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$, у трассы λ_1 имеется такой префикс $\lambda_2\cdot\langle u\rangle$, что $\mu\cdot\langle P\rangle\cdot\lambda_2 \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$, а $\mu\cdot\langle P\rangle\cdot\lambda_2\cdot\langle u\rangle \notin \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$. Отсюда, поскольку $\mu\cdot\langle P\rangle\cdot\lambda \in \boldsymbol{\Sigma}$ и, следовательно, $\mu\cdot\langle P\rangle\cdot\lambda_2\cdot\langle u\rangle \in \boldsymbol{\Sigma}$, имеем, либо $u=\Delta$, либо $u \in \mathbf{L}\cup\mathbf{R}$ и u $\boldsymbol{safe}_{\gamma\Delta}$ $\boldsymbol{\Sigma}$ ***after*** $\mu\cdot\langle P\rangle\cdot\lambda_2$. Поэтому в $\boldsymbol{\Sigma}$ имеется трасса $\mu\cdot\langle P\rangle\cdot\lambda_2\cdot\langle\Delta\rangle$ или для каждой кнопки $R \in \boldsymbol{but}(u)$ имеется трасса $\mu\cdot\langle P\rangle\cdot\lambda_2\cdot\langle z,\gamma\rangle$ для некоторого $z \in R$. Но тогда по замкнутости **R**-модели в $\boldsymbol{\Sigma}$ имеется трасса $\mu\cdot\lambda_2\cdot\langle\Delta\rangle$ или для каждой кнопки $R \in \boldsymbol{but}(u)$ имеется трасса $\mu\cdot\lambda_2\cdot\langle z,\gamma\rangle$ для

некоторого $z \in R$, что, поскольку $\mu\cdot\lambda_2\cdot\langle u\rangle \le \mu\cdot\lambda_1$, противоречит тому, что $\mu\cdot\lambda_1 \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$.

6.6. Поскольку $\mu\cdot\lambda_1$ максимальный неразрушающий префикс предфинальной трассы $\mu\cdot\lambda$, трасса $\mu\cdot\lambda$ является финальным продолжением трассы $\mu\cdot\lambda_1$: 1) $\mu\cdot\lambda=\mu\cdot\lambda_1$, или 2) $\mu\cdot\lambda=\mu\cdot\lambda_1\cdot\langle\Delta\rangle$, или 3) $\mu\cdot\lambda=\mu\cdot\lambda_1\cdot\langle z,\gamma\rangle$, или 4) $\mu\cdot\lambda=\mu\cdot\lambda_1\cdot\langle z\rangle$ и $\mu\cdot\lambda_1\cdot\langle z,\gamma\rangle \in \boldsymbol{\Sigma}$, или 5) $\mu\cdot\lambda=\mu\cdot\lambda_1\cdot\langle Q\rangle$, где $Q \in \mathbf{Q}$. Случая 5 быть не может, так как **R**-модель $\boldsymbol{\Sigma}$ не содержит **Q**-отказов.

Поскольку $\mu\cdot\lambda \in \boldsymbol{\Sigma}$ и трасса μ заканчивается отказом и не продолжается в $\boldsymbol{\Sigma}$ действиями из отказа P, по **R**-полноте трассовой модели: 1) $\mu\cdot\langle P\rangle\cdot\lambda=\mu\cdot\langle P\rangle\cdot\lambda_1 \in \boldsymbol{\Sigma}$, или 2) $\mu\cdot\langle P\rangle\cdot\lambda=\mu\cdot\langle P\rangle\cdot\lambda_1\cdot\langle\Delta\rangle \in \boldsymbol{\Sigma}$, или 3) $\mu\cdot\langle P\rangle\cdot\lambda=\mu\cdot\langle P\rangle\cdot\lambda_1\cdot\langle z,\gamma\rangle \in \boldsymbol{\Sigma}$, или 4) $\mu\cdot\langle P\rangle\cdot\lambda=\mu\cdot\langle P\rangle\cdot\lambda_1\cdot\langle z\rangle \in \boldsymbol{\Sigma}$ и $\mu\cdot\langle P\rangle\cdot\lambda_1\cdot\langle z,\gamma\rangle \in \boldsymbol{\Sigma}$. Поэтому, поскольку $\mu\cdot\langle P\rangle\cdot\lambda_1 \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$, во всех случаях трасса $\mu\cdot\langle P\rangle\cdot\lambda$ является финальным продолжением трассы $\mu\cdot\langle P\rangle\cdot\lambda_1$. Следовательно, $\mu\cdot\langle P\rangle\cdot\lambda \in \boldsymbol{ptr}(\boldsymbol{\Sigma})$, что и требовалось доказать.

7. предфинальность: $\boldsymbol{ptr}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))=\boldsymbol{ptr}(\boldsymbol{\Sigma})$.

По определению предфинальных трасс $\boldsymbol{ptr}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))\subseteq\boldsymbol{ptr}(\boldsymbol{\Sigma})$. Нам достаточно показать, что $\boldsymbol{ptr}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))\supseteq\boldsymbol{ptr}(\boldsymbol{\Sigma})$. Рассмотрим трассу $\sigma \in \boldsymbol{ptr}(\boldsymbol{\Sigma})$. По определению предфинальных трасс $\sigma \in \boldsymbol{\Sigma}$ и возможны два варианта.

7.1. $\sigma \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$. Покажем, что $\sigma \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$, из чего будет следовать, что $\sigma \in \boldsymbol{ptr}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$.

Допустим противное. Тогда, поскольку $\langle\gamma\rangle \notin \boldsymbol{\Sigma}$, у трассы σ имеется такой префикс $\mu\cdot\langle u\rangle$, что $\mu \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$, а $\mu\cdot\langle u\rangle \notin \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$. Поскольку $\sigma \in \boldsymbol{ptr}(\boldsymbol{\Sigma})$, $\mu\cdot\langle u\rangle \in \boldsymbol{ptr}(\boldsymbol{\Sigma})$. Следовательно, $u=\Delta$ или $u \in \mathbf{L}\cup\mathbf{R}$ и u $\boldsymbol{safe}_{\gamma\Delta}$ $\boldsymbol{ptr}(\boldsymbol{\Sigma})$ ***after*** μ. Тогда в $\boldsymbol{ptr}(\boldsymbol{\Sigma})$ либо имеется трасса $\mu\cdot\langle\Delta\rangle$, либо для каждой кнопки $R \in \boldsymbol{but}(u)$ имеется трасса $\mu\cdot\langle z,\gamma\rangle$ для некоторого $z \in R$. Но тогда такие трассы есть и в $\boldsymbol{\Sigma}$, что противоречит тому, что $\sigma \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$. Следовательно, $\sigma \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$.

7.2. $\sigma \notin \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$.

Тогда, поскольку $\langle\gamma\rangle \notin \boldsymbol{\Sigma}$, σ является финальным продолжением некоторой неразрушающей трассы μ. По доказанному $\mu \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$, что влечет $\sigma \in \boldsymbol{ptr}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$.

8. финально-замкнутость: Пусть $\mu\cdot\langle P\rangle\cdot\lambda \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$, где $P \in \mathbf{R}$. Нужно доказать, что $\mu\cdot\lambda \in \boldsymbol{UnSafe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$ или $\mu\cdot\lambda \in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$ и $\boldsymbol{fs}(\boldsymbol{ptr}(\boldsymbol{\Sigma}),\mu\cdot\langle P\rangle\cdot\lambda)\subseteq\boldsymbol{fs}(\boldsymbol{ptr}(\boldsymbol{\Sigma}),\mu\cdot\lambda)$.

Поскольку $\mu\cdot\langle P\rangle\cdot\lambda\in \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$, имеем $\mu\cdot\langle P\rangle\cdot\lambda\in\boldsymbol{ptr}(\boldsymbol{\Sigma})$ и $\mu\cdot\langle P\rangle\cdot\lambda\in\boldsymbol{\Sigma}$. По замкнутости **R**-модели $\mu\cdot\lambda\in\boldsymbol{\Sigma}$. Поскольку $\mu\cdot\langle P\rangle\cdot\lambda\in\boldsymbol{ptr}(\boldsymbol{\Sigma})$, по доказанной префикс-замкнутости $\mu\cdot\langle P\rangle\in\boldsymbol{ptr}(\boldsymbol{\Sigma})$. А тогда, поскольку $P\in\mathbf{R}$, должно быть $\mu\cdot\langle P\rangle\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$, поскольку трасса $\mu\cdot\langle P\rangle$ заканчивается **R**-отказом. По префикс-замкнутости неразрушающих трасс имеем $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$. Далее рассмотрим два случая.

8.1. $\mu\cdot\lambda\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$.

Тогда по доказанному в п.7.1 $\mu\cdot\lambda\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$. Пусть $\kappa\in\boldsymbol{fs}(\boldsymbol{ptr}(\boldsymbol{\Sigma}),\mu\cdot\langle P\rangle\cdot\lambda)$. Тогда 1) $\kappa=\epsilon$, или 2) $\kappa=\langle\Delta\rangle$, или 3) $\kappa=\langle z,\gamma\rangle$, 4) $\kappa=\langle z\rangle$ и $\mu\cdot\langle P\rangle\cdot\lambda\cdot\langle z,\gamma\rangle\in\boldsymbol{ptr}(\boldsymbol{\Sigma})$, или 5) $\mu\cdot\lambda=\mu\cdot\lambda_1\cdot\langle Q\rangle$, где $Q\in\mathbf{Q}$. Случая 5 быть не может, так как **R**-модель $\boldsymbol{\Sigma}$ не содержит **Q**-отказов. Поскольку $\boldsymbol{ptr}(\boldsymbol{\Sigma})\subseteq\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}$ ***d***-замкнуто, имеем $\kappa\in\boldsymbol{fs}(\boldsymbol{\Sigma},\mu\cdot\lambda)$. Поскольку $\mu\cdot\lambda\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$, имеем $\kappa\in\boldsymbol{fs}(\boldsymbol{ptr}(\boldsymbol{\Sigma}),\mu\cdot\lambda)$. Следовательно, $\boldsymbol{fs}(\boldsymbol{ptr}(\boldsymbol{\Sigma}),\mu\cdot\langle P\rangle\cdot\lambda)\subseteq\boldsymbol{fs}(\boldsymbol{ptr}(\boldsymbol{\Sigma}),\mu\cdot\lambda)$.

8.2. $\mu\cdot\lambda\notin\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Sigma})$.

Тогда, поскольку $\mu\cdot\langle P\rangle\cdot\lambda\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$, имеем $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$. Поэтому у трассы λ найдется такой префикс $\lambda_1\cdot\langle u\rangle$, что $\mu\cdot\lambda_1\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$, но $\mu\cdot\lambda_1\cdot\langle u\rangle\notin\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$.

Следовательно, $u=\Delta$ или $u\in\mathbf{L}\cup\mathbf{R}$ и $u\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Sigma}\ \boldsymbol{after}\ \mu\cdot\lambda_1$. Поскольку, $\mu\cdot\langle P\rangle\cdot\lambda_1\cdot\langle u\rangle\le\mu\cdot\langle P\rangle\cdot\lambda\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{ftr}(\boldsymbol{\Sigma}))$, трасса $\mu\cdot\lambda_1\cdot\langle u\rangle$ является $\mathbf{L}\cup\mathbf{R}$-трассой, следовательно, случая $u=\Delta$ быть не может. Поэтому $u\in\mathbf{L}\cup\mathbf{R}$ и $u\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Sigma}\ \boldsymbol{after}\ \mu\cdot\lambda_1$. Тогда либо $\mu\cdot\lambda_1\cdot\langle\Delta\rangle\in\boldsymbol{\Sigma}$, либо для каждой кнопки $R\in\boldsymbol{but}(u)$ имеется трасса $\mu\cdot\lambda_1\cdot\langle z,\gamma\rangle\in\boldsymbol{\Sigma}$ для некоторого $z\in R$. А тогда по определению предфинальных трасс либо $\mu\cdot\lambda_1\cdot\langle\Delta\rangle\in\boldsymbol{ptr}(\boldsymbol{\Sigma})$, либо для каждой кнопки $R\in\boldsymbol{but}(u)$ имеется трасса $\mu\cdot\lambda_1\cdot\langle z,\gamma\rangle\in\boldsymbol{ptr}(\boldsymbol{\Sigma})$ для некоторого $z\in R$. Следовательно, в обоих случаях трасса $\mu\cdot\lambda\in\boldsymbol{UnSafe}_{\gamma\Delta}(\boldsymbol{ptr}(\boldsymbol{\Sigma}))$.

17. Доказательство Теорема 6:

1. Сначала докажем, что $\boldsymbol{D}(\boldsymbol{\Psi})$ является **R**-моделью.

По определению p-модели $\boldsymbol{\Psi}$ она является незамкнутой **R**-моделью. Поэтому по теореме 4 $\boldsymbol{D}(\boldsymbol{\Psi})$ является **R**-моделью.

2. Теперь покажем, что $\boldsymbol{ptr}(\boldsymbol{D}(\boldsymbol{\Psi}))=\boldsymbol{\Psi}$.

2.1. Пусть $\langle\gamma\rangle\in\boldsymbol{\Psi}$. Тогда по определению предфинальных трасс $\boldsymbol{ptr}(\boldsymbol{\Psi})=\{\epsilon,\langle\gamma\rangle\}$. По определению p-модели, $\boldsymbol{\Psi}=\boldsymbol{ptr}(\boldsymbol{\Psi})=\{\epsilon,\langle\gamma\rangle\}$. По

определению ***d***-замыкания $\boldsymbol{D}(\boldsymbol{\Psi})=\boldsymbol{\Psi}=\{\epsilon,\langle\gamma\rangle\}$. По определению предфинальных трасс $\boldsymbol{ptr}(\boldsymbol{D}(\boldsymbol{\Psi}))=\{\epsilon,\langle\gamma\rangle\}$. Следовательно, $\boldsymbol{ptr}(\boldsymbol{D}(\boldsymbol{\Psi}))=\{\epsilon,\langle\gamma\rangle\}=\boldsymbol{\Psi}$.

2.2. Пусть $\langle\gamma\rangle\notin\boldsymbol{\Psi}$. Тогда по согласованности p-модели $\langle\gamma\rangle\notin\boldsymbol{D}(\boldsymbol{\Psi})$. А тогда $\epsilon\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$ и $\epsilon\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$.

2.2.1. Сначала покажем, что $\boldsymbol{ptr}(\boldsymbol{D}(\boldsymbol{\Psi}))\subseteq\boldsymbol{\Psi}$.

2.2.1.1. Покажем, что $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))\subseteq\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

Пусть трасса $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$.

Допустим противное: $\sigma\notin\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$. Поскольку $\epsilon\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$, у трассы σ найдется префикс $\mu\cdot\langle u\rangle$ такой, что $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$, а $\mu\cdot\langle u\rangle\notin\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$. Поскольку $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$, а $\mu\cdot\langle u\rangle\le\sigma$, имеем $u\ne\Delta$ и $u\ne\gamma$, то есть $u\in\mathbf{L}\cup\mathbf{R}$ и $u\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Psi}\ \boldsymbol{after}\ \mu$.

А тогда либо трасса $\mu\cdot\langle\Delta\rangle\in\boldsymbol{\Psi}$, либо для каждой кнопки $R\in\boldsymbol{but}(u)$ имеется трасса $\mu\cdot\langle z,\gamma\rangle\in\boldsymbol{\Psi}$ для некоторого $z\in R$. А тогда $\mu\cdot\langle\Delta\rangle\in\boldsymbol{D}(\boldsymbol{\Psi})$, либо для каждой кнопки $R\in\boldsymbol{but}(u)$ имеется трасса $\mu\cdot\langle z,\gamma\rangle\in\boldsymbol{D}(\boldsymbol{\Psi})$ для некоторого $z\in R$. Следовательно, $u\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{D}(\boldsymbol{\Psi})\ \boldsymbol{after}\ \mu$. Но это противоречит тому, что $\mu\cdot\langle u\rangle\le\sigma$ и $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$. Мы пришли к противоречию, следовательно, $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$, что и требовалось доказать.

2.2.1.2. Покажем, что $\boldsymbol{ptr}(\boldsymbol{D}(\boldsymbol{\Psi}))\setminus\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))\subseteq\boldsymbol{\Psi}\setminus\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

Пусть трасса $\sigma\in\boldsymbol{ptr}(\boldsymbol{D}(\boldsymbol{\Psi}))\setminus\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$. Тогда трасса σ является финальным продолжением в $\boldsymbol{D}(\boldsymbol{\Psi})$ некоторой трассы $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$ и $\sigma\ne\mu$. Тогда по доказанному $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$. Нам надо показать, что σ является финальным продолжением μ в $\boldsymbol{\Psi}$ и $\sigma\notin\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

Поскольку $\sigma\ne\mu$, по определению финального продолжения и отсутствию в p-модели **Q**-отказов возможны три варианта: 1) $\sigma=\mu\cdot\langle\Delta\rangle\in\boldsymbol{D}(\boldsymbol{\Psi})$, 2) $\sigma=\mu\cdot\langle z,\gamma\rangle\in\boldsymbol{D}(\boldsymbol{\Psi})$, 3) $\sigma=\mu\cdot\langle z\rangle\in\boldsymbol{D}(\boldsymbol{\Psi})$ и $\mu\cdot\langle z,\gamma\rangle\in\boldsymbol{D}(\boldsymbol{\Psi})$.

По определению ***d***-замыкания найдется такая трасса $\mu`\in\boldsymbol{\Psi}$, что $\mu\in\boldsymbol{d}(\mu`)$ и 1) $\mu`\cdot\langle\Delta\rangle\in\boldsymbol{\Psi}$, 2) $\mu`\cdot\langle z,\gamma\rangle\in\boldsymbol{\Psi}$, 3) $\mu`\cdot\langle z\rangle\in\boldsymbol{\Psi}$ и $\mu`\cdot\langle z,\gamma\rangle\in\boldsymbol{\Psi}$. Отсюда по свойству предфинальности $\boldsymbol{\Psi}$ трасса $\mu`\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

А тогда по финально-замкнутости $\boldsymbol{\Psi}$ либо 1) $\mu\in\boldsymbol{UnSafe}_{\gamma\Delta}(\boldsymbol{\Psi})$, либо 2) $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$ и $\boldsymbol{fs}(\boldsymbol{\Psi},\mu`)\subseteq\boldsymbol{fs}(\boldsymbol{\Psi},\mu)$.

Случай 1 противоречит $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$. Следовательно, 1) $\sigma=\mu\cdot\langle\Delta\rangle\in\boldsymbol{\Psi}$, 2) $\sigma=\mu\cdot\langle z,\gamma\rangle\in\boldsymbol{\Psi}$, 3) $\sigma=\mu\cdot\langle z\rangle\in\boldsymbol{\Psi}$ и $\mu\cdot\langle z,\gamma\rangle\in\boldsymbol{\Psi}$.

Следовательно, σ является финальным продолжением μ в $\boldsymbol{\Psi}$.

Во всех этих случаях также $\sigma\notin \boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

2.2.2. Теперь покажем, что $\boldsymbol{ptr}(\boldsymbol{D}(\boldsymbol{\Psi}))\supseteq\boldsymbol{\Psi}$.

2.2.2.1. Покажем, что $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))\supseteq\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

Пусть трасса $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$. Тогда $\sigma\in\boldsymbol{\Psi}$ и, следовательно, $\sigma\in\boldsymbol{D}(\boldsymbol{\Psi})$. Допустим, утверждение не верно: $\sigma\in\boldsymbol{D}(\boldsymbol{\Psi})\setminus\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$.

Поскольку $\epsilon\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$, у трассы σ найдется префикс $\mu\cdot\langle u\rangle$ такой, что $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$, а $\mu\cdot\langle u\rangle\notin\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$.

Поскольку $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$, а $\mu\cdot\langle u\rangle\le\sigma$, имеем $u\ne\Delta$ и $u\ne\gamma$, то есть $u\in\mathbf{L}\cup\mathbf{R}$ и тогда u $\boldsymbol{safe}_{\gamma\Delta}$ $\boldsymbol{D}(\boldsymbol{\Psi})$ $\boldsymbol{after}$ μ.

Следовательно, либо 1) $\mu\cdot\langle\Delta\rangle\in\boldsymbol{D}(\boldsymbol{\Psi})$, либо 2) для каждой кнопки $R\in\boldsymbol{but}(u)$ имеется трасса $\mu\cdot\langle z_R,\gamma\rangle\in\boldsymbol{D}(\boldsymbol{\Psi})$ для некоторого $z_R\in R$.

Тогда по определению ***d***-замыкания либо 1) найдется такая трасса $\mu`\in\boldsymbol{\Psi}$, что $\mu\in\boldsymbol{d}(\mu`)$ и $\mu`\cdot\langle\Delta\rangle\in\boldsymbol{\Psi}$, либо 2) для каждой кнопки $R\in\boldsymbol{but}(u)$ найдется такая трасса $\mu_R`\in\boldsymbol{\Psi}$, что $\mu\in\boldsymbol{d}(\mu_R`)$ и $\mu_R`\cdot\langle z_R,\gamma\rangle\in\boldsymbol{\Psi}$.

По свойству предфинальности $\boldsymbol{\Psi}$ 1) трасса $\mu`\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$ или 2) все трассы $\mu_R`\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

Поскольку $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$, по п.2.2.2.1.1 $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

А тогда по финально-замкнутости $\boldsymbol{\Psi}$ имеем 1) $\langle\Delta\rangle\in\boldsymbol{fs}(\boldsymbol{\Psi},\mu`)\subseteq\boldsymbol{fs}(\boldsymbol{\Psi},\mu)$ либо 2) $\langle z_R,\gamma\rangle\in\boldsymbol{fs}(\boldsymbol{\Psi},\mu_R`)\subseteq\boldsymbol{fs}(\boldsymbol{\Psi},\mu)$ для каждой кнопки R.

Следовательно, 1) трасса $\mu\cdot\langle\Delta\rangle\in\boldsymbol{\Psi}$ либо 2) для каждой кнопки $R\in\boldsymbol{but}(u)$ трасса $\mu\cdot\langle z_R,\gamma\rangle\in\boldsymbol{\Psi}$.

Тем самым, u $\boldsymbol{safe}_{\gamma\Delta}$ $\boldsymbol{\Psi}$ $\boldsymbol{after}$ μ, что, поскольку $\mu\cdot\langle u\rangle\le\sigma$, противоречит $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

Итак, мы пришли к противоречию, следовательно, наше допущение не верно и $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$, что и требовалось доказать.

2.2.2.2. Покажем, что $\boldsymbol{ptr}(\boldsymbol{D}(\boldsymbol{\Psi}))\setminus\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))\supseteq\boldsymbol{\Psi}\setminus\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$.

Пусть трасса $\sigma\in\boldsymbol{\Psi}\setminus\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$. Тогда трасса σ является финальным продолжением в $\boldsymbol{\Psi}$ некоторой трассы $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$ и $\sigma\ne\mu$. Тогда по п.2.2.2.1 $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$. Нам надо показать, что σ является финальным продолжением μ в $\boldsymbol{D}(\boldsymbol{\Psi})$ и $\sigma\notin\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$.

Поскольку $\sigma\ne\mu$, по определению финального продолжения и отсутствию в p-модели $\mathbf{Q}$-отказов возможны три варианта: 1) $\sigma=\mu\cdot\langle\Delta\rangle\in\boldsymbol{\Psi}$, 2) $\sigma=\mu\cdot\langle z,\gamma\rangle\in\boldsymbol{\Psi}$, 3) $\sigma=\mu\cdot\langle z\rangle\in\boldsymbol{\Psi}$ и $\mu\cdot\langle z,\gamma\rangle\in\boldsymbol{\Psi}$.

Тогда по определению ***d***-замыкания 1) $\sigma=\mu\cdot\langle\Delta\rangle\in\boldsymbol{D}(\boldsymbol{\Psi})$, 2) $\sigma=\mu\cdot\langle z,\gamma\rangle\in\boldsymbol{D}(\boldsymbol{\Psi})$, 3) $\sigma=\mu\cdot\langle z\rangle\in\boldsymbol{D}(\boldsymbol{\Psi})$ и $\mu\cdot\langle z,\gamma\rangle\in\boldsymbol{D}(\boldsymbol{\Psi})$.

Тогда, поскольку $\mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$, трасса σ является финальным продолжением μ в $\boldsymbol{D}(\boldsymbol{\Psi})$, во всех этих случаях также $\sigma\notin\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{D}(\boldsymbol{\Psi}))$, что и требовалось доказать.

18. Доказательство Теорема 7:

Пусть $\boldsymbol{\Omega}\subseteq(\mathbf{L}\cup\mathbf{R}\cup\mathbf{Q}\cup\{\Delta,\gamma\})^*$ и $\boldsymbol{\Psi}=\boldsymbol{\Omega}_{\mathbf{R}}$ p-модель.

Обозначим $A=\{\sigma\in\boldsymbol{\Omega}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Omega}\}$.

Поскольку $\boldsymbol{\Psi}=\boldsymbol{\Omega}_{\mathbf{R}}$, имеем $A=\{\sigma\in\boldsymbol{\Psi}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\boldsymbol{\Psi}\}$.

Условия 4b и 4c совпадают, если $\boldsymbol{SafeIn}(\boldsymbol{\Omega})=\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})$. Поскольку $\boldsymbol{\Psi}=\boldsymbol{\Omega}_{\mathbf{R}}$, имеем $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})=\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$. Поскольку $\boldsymbol{\Psi}$ p-модель, она предфинальна, поэтому по теореме 3 $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=A$. Следовательно, для эквивалентности условий 4b и 4c достаточно $\boldsymbol{SafeIn}(\boldsymbol{\Omega})=A$. Очевидно, $\boldsymbol{SafeIn}(\boldsymbol{\Omega})\subseteq A$, поэтому достаточно условия $\boldsymbol{SafeIn}(\boldsymbol{\Omega})\supseteq A$.

1. Пусть выполнены условия: $\forall z\in\mathbf{L}\ \forall Q\in\mathbf{Q}\ \forall\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})$

2b) $\sigma\cdot\langle z\rangle\in\boldsymbol{\Omega}\ \&\ \sigma\cdot\langle z,\gamma\rangle\notin\boldsymbol{\Omega}$

$\Rightarrow\exists P\in\boldsymbol{but}(z)\ P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Omega}\ \boldsymbol{after}\ \sigma\ \&\ (P\in\mathbf{Q}\Rightarrow\sigma\cdot\langle P\rangle\notin\boldsymbol{\Omega})$,

3b) $Q\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Omega}\ \boldsymbol{after}\ \sigma\ \&\ \sigma\cdot\langle Q\rangle\notin\boldsymbol{\Omega}\Rightarrow\exists v\in Q\ \sigma\cdot\langle v\rangle\in\boldsymbol{\Omega}$.

4b) $\forall P\in\mathbf{Q}\ \forall\mu\cdot\langle P\rangle\cdot\lambda\in\boldsymbol{\Omega}\ \mu\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})\ \&\ P\ \boldsymbol{safe}_{\gamma\Delta}\ \boldsymbol{\Omega}\ \boldsymbol{after}\ \mu\ \&\ \lambda=\epsilon$.

Докажем, что $\boldsymbol{SafeIn}(\boldsymbol{\Omega})\supseteq A$ и выполнены условия 2c и 3c.

1.1. Пусть $\langle\gamma\rangle\in\boldsymbol{\Psi}$.

Тогда по определению p-модели $\boldsymbol{\Psi}=\{\epsilon,\langle\gamma\rangle\}$ и $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=\varnothing$. Поскольку $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})=\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})$, по правилу 4b ни одна трасса не содержит $\mathbf{Q}$-отказов. Следовательно, $\boldsymbol{\Omega}=\boldsymbol{\Omega}_{\mathbf{R}}=\boldsymbol{\Psi}=\{\epsilon,\langle\gamma\rangle\}$, поэтому выполнены условия 2c и 3c. Также $A=\varnothing$, следовательно, $\boldsymbol{SafeIn}(\boldsymbol{\Omega})\supseteq A$.

1.2. Пусть $\langle\gamma\rangle\notin\boldsymbol{\Psi}$.

Будем вести доказательство по индукции. Из $\langle\gamma\rangle\notin\boldsymbol{\Psi}$ следует, что $\langle\gamma\rangle\notin\boldsymbol{\Omega}$, поэтому $\epsilon\in\boldsymbol{SafeIn}(\boldsymbol{\Omega})$. Также $\epsilon\in A$. Пусть для $\sigma\in\boldsymbol{SafeIn}(\boldsymbol{\Omega})\cap A$ утверждение верно и пусть $\sigma\cdot\langle u\rangle\in A$. Тогда $\sigma\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})$. Поскольку $\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})=\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Psi})=A$, имеем $\sigma\cdot\langle u\rangle\in\boldsymbol{Safe}_{\gamma\Delta}(\boldsymbol{\Omega})$. Поэтому, если $u\in\mathbf{R}$, то $\sigma\cdot\langle u\rangle\in\boldsymbol{SafeIn}(\boldsymbol{\Omega})$. Если $u\in\mathbf{L}$, то $\sigma\cdot\langle u,\gamma\rangle\notin\boldsymbol{\Omega}$. Поэтому по правилу 2b u $\boldsymbol{safe\ in}$ $\boldsymbol{\Omega}$ $\boldsymbol{after}$ σ. Следовательно, $\sigma\cdot\langle u\rangle\in\boldsymbol{SafeIn}(\boldsymbol{\Omega})$. Итак мы доказали, что $\boldsymbol{SafeIn}(\boldsymbol{\Omega})\supseteq A$. И, поскольку $\boldsymbol{SafeIn}(\boldsymbol{\Omega})\subseteq A$, имеем $\boldsymbol{SafeIn}(\boldsymbol{\Omega})=A$, то есть условие 2c доказано. Также по

правилу 3b, если трасса σ не продолжается **Q**-отказом, то либо этот **Q**-отказ разрушающий после трассы, либо она продолжается действием из этого **Q**-отказа. Если **Q**-отказ разрушающий после трассы, то трасса продолжается либо дивергенцией, либо действием из **Q**-отказа (и далее разрушением). Во всех случаях выполнено свойство 3c **Q**-конвергентности трасс из $\mathbf{\Omega_R}$ в **Ω**.

2. Пусть выполнены условия:

2c) $\langle\gamma\rangle\notin\mathbf{\Omega}$ & $\mathit{SafeIn}(\mathbf{\Omega})=\{\sigma\in\mathbf{\Omega}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\mathbf{\Omega}\}\ \vee\ \mathbf{\Omega}=\{\epsilon,\langle\gamma\rangle\}$,

3c) трассы из $\mathbf{\Omega_R}$ **Q**-конвергентны в **Ω**,

4c) $\forall P\in\mathbf{Q}\ \forall\mu\cdot\langle P\rangle\cdot\lambda\in\mathbf{\Omega}\ \mu\in\mathit{SafeIn}(\mathbf{\Omega})$ & $P\ \mathit{safe}_{\gamma\Delta}\ \mathbf{\Omega}\ \mathit{after}\ \mu$ & $\lambda=\epsilon$.

Докажем, что $\mathit{SafeIn}(\mathbf{\Omega})\supseteq A$ и выполнены условия 2b и 3b.

2.1. Пусть $\langle\gamma\rangle\in\mathbf{\Psi}$.

Тогда $\mathbf{\Omega}=\{\epsilon,\langle\gamma\rangle\}$, что влечет $\mathit{Safe}_{\gamma\Delta}(\mathbf{\Omega})=\varnothing$ и, следовательно, выполнены правила 2b и 3b. Также будет $A=\varnothing$, следовательно, $\mathit{SafeIn}(\mathbf{\Omega})\supseteq A$.

2.2. Пусть $\langle\gamma\rangle\notin\mathbf{\Psi}$.

Тогда $\mathit{SafeIn}(\mathbf{\Omega})=A$, следовательно, $\mathit{SafeIn}(\mathbf{\Omega})\supseteq A$. Также $\mathit{Safe}_{\gamma\Delta}(\mathbf{\Omega})=\mathit{Safe}_{\gamma\Delta}(\mathbf{\Psi})=A$. Пусть выполнено условие левой части импликации в правиле 2b: $z\in\mathbf{L}$, $\sigma\in\mathit{Safe}_{\gamma\Delta}(\mathbf{\Omega})$, $\sigma\cdot\langle z\rangle\in\mathbf{\Omega}$, $\sigma\cdot\langle z,\gamma\rangle\notin\mathbf{\Omega}$. Тогда $\sigma\cdot\langle z\rangle\in A=\mathit{SafeIn}(\mathbf{\Omega})$. Следовательно, выполнено условие в правой части импликации правила 2b, то есть правило 2b доказано. Пусть выполнено условие левой части импликации в правиле 3b: $\sigma\in\mathit{Safe}_{\gamma\Delta}(\mathbf{\Omega})$, $Q\ \mathit{safe}_{\gamma\Delta}\ \mathbf{\Omega}\ \mathit{after}\ \sigma$, $\sigma\cdot\langle Q\rangle\notin\mathbf{\Omega}$. Тогда по **Q**-конвергентности выполнено условие в правой части импликации правила 3b, то есть правило 3b доказано.

19. Доказательство Теорема 8:

1. Пусть $\langle\gamma\rangle\in\mathbf{\Psi}$. Тогда $\mathbf{\Omega}=\mathbf{\Psi}=\{\epsilon,\langle\gamma\rangle\}$. Доказательство утверждения в этом случае тривиально.
2. Пусть $\langle\gamma\rangle\notin\mathbf{\Psi}$.

2.1. Сначала докажем часть утверждения, связанную с множествами безопасных (по разным отношениям) трасс.

По определению отношения ***safe by*** имеет место: $\mathit{Safe}_{\gamma\Delta}(\mathbf{\Omega})=\mathit{SafeBy}(\mathbf{\Omega})$ и $\mathit{SafeBy}(\mathbf{\Psi})=\mathit{Safe}_{\gamma\Delta}(\mathbf{\Psi})$. Поскольку трассы из р-модели **Ψ** не содержат **Q**-отказов, имеем $\mathit{Safe}_{\gamma\Delta}(\mathbf{\Psi})=\mathit{SafeIn}(\mathbf{\Psi})$.

По теореме 3 $\mathit{Safe}_{\gamma\Delta}(\mathbf{\Psi})=\{\sigma\in\mathbf{\Psi}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\mathbf{\Psi}\}$, а по правилу 2c) $\mathit{SafeIn}(\mathbf{\Omega})=\{\sigma\in\mathbf{\Omega}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\mathbf{\Omega}\}$. Следовательно, $\mathit{SafeIn}(\mathbf{\Omega})=\mathit{Safe}_{\gamma\Delta}(\mathbf{\Omega})$.

Поскольку $\mathbf{\Psi}=\mathbf{\Omega_R}$, имеем $\mathit{Safe}_{\gamma\Delta}(\mathbf{\Omega})=\mathit{Safe}_{\gamma\Delta}(\mathbf{\Psi})$. Из этих равенств следует доказываемое утверждение:

$\mathit{SafeIn}(\mathbf{\Omega})=\mathit{Safe}_{\gamma\Delta}(\mathbf{\Omega})=\mathit{SafeBy}(\mathbf{\Omega})=\mathit{SafeBy}(\mathbf{\Psi})=\mathit{Safe}_{\gamma\Delta}(\mathbf{\Psi})=\mathit{SafeIn}(\mathbf{\Psi})$.

2.2. Теперь докажем выполнение свойств f-модели для $\mathbf{\Omega}=\mathit{ftr}(\mathbf{\Psi},\mathit{safe\ by})$.

2.2.1. Непустота. Поскольку **Ψ** р-модель, $\mathbf{\Psi}\neq\varnothing$. Следовательно, поскольку $\mathbf{\Psi}\subseteq\mathbf{\Omega}$, $\mathbf{\Omega}\neq\varnothing$.

2.2.2. Префикс-замкнутость. Поскольку **Ψ** р-модель, она префикс-замкнута. По теореме 7 и правилу 4c) трассы из $\mathbf{\Omega}\backslash\mathbf{\Psi}$ – это **Q**-финальные продолжения трасс из **Ψ**. Поэтому префикс-замкнутость сохраняется.

2.2.3. Допустимость. Поскольку **Ψ** р-модель, все ее трассы допустимы. По теореме 7 и правилу 4c) трассы из $\mathbf{\Omega}\backslash\mathbf{\Psi}$ – это **Q**-финальные продолжения трасс из **Ψ**, то есть трассы, заканчивающиеся **Q**-отказами. Поэтому допустимость сохраняется.

2.2.4. **Q**-допустимость. Это следует из теоремы 7 и свойства 4c).

2.2.5. Согласованность. Поскольку **Ψ** р-модель, все ее трассы согласованы. По теореме 7 и правилу 4c) трассы из $\mathbf{\Omega}\backslash\mathbf{\Psi}$ – **Q**-финальные продолжения трасс из **Ψ**. Поэтому согласованность сохраняется.

2.2.6. $\mathbf{R}\cup\mathbf{Q}$-конвергентность трасс из $\mathbf{\Omega_R}$ в **Ω**. Поскольку **Ψ** р-модель, все трассы из $\mathbf{\Omega_R}$ **R**-конвергентны $\mathbf{\Omega_R}$, следовательно, они **R**-конвергентны в $\mathbf{\Omega}\supseteq\mathbf{\Omega_R}$. По теореме 7 и правилу 3c) трассы из $\mathbf{\Omega_R}$ **Q**-конвергентны в **Ω**.

2.2.7. финально-замкнутость $\mathbf{\Omega_R}$. Поскольку **Ψ** р-модель, она финально-замкнута. Поскольку $\mathbf{\Psi}=\mathbf{\Omega_R}$, имеем финально-замкнутость $\mathbf{\Omega_R}$.

2.2.8. **R**-полнота $\mathbf{\Omega_R}$. Поскольку **Ψ** р-модель, она **R**-полная. Поскольку $\mathbf{\Psi}=\mathbf{\Omega_R}$, имеем **R**-полноту $\mathbf{\Omega_R}$.

2.2.9. финальность. Это следует из теоремы 7 и свойства 2c).

20. Доказательство Теорема 9:

1. Пусть $\langle\gamma\rangle\in\mathbf{\Omega}$. Тогда $\mathbf{\Omega}=\mathbf{\Omega_R}=\{\epsilon,\langle\gamma\rangle\}$. Доказательство утверждения в этом случае тривиально.
2. Пусть $\langle\gamma\rangle\notin\mathbf{\Omega}$.

2.1. Сначала покажем, что $\mathbf{\Omega_R}$ р-модель.

2.1.1. Алфавит. По определению $\mathbf{\Omega_R}\subseteq(\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\})^*$.

2.1.2. Непустота. Так как **Ω** не пусто и префикс-замкнуто, $\epsilon\in\mathbf{\Omega}$. Следовательно, поскольку $\langle\gamma\rangle\notin\mathbf{\Omega}$, $\epsilon\in\mathbf{\Omega_R}$, то есть $\mathbf{\Omega_R}$ не пусто.

2.1.3. Префикс-замкнутость. Поскольку **Ω** f-модель, она префикс-замкнута. Также префикс-замкнуто множество $(\mathbf{L}\cup\mathbf{R}\cup\{\Delta,\gamma\})^*$. Отсюда $\mathbf{\Omega_R}$ префикс-замкнуто как пересечение префикс-замкнутых множеств.

2.1.4. Допустимость. Поскольку **Ω** f-модель, все ее трассы допустимы. Поскольку $\mathbf{\Omega_R}\subseteq\mathbf{\Omega}$, допустимость сохраняется.

2.1.5. Согласованность. Поскольку **Ω** f-модель, все ее трассы согласованы. Поскольку $\mathbf{\Omega_R}\subseteq\mathbf{\Omega}$, согласованность сохраняется.

2.1.6. **R**-конвергентность $\mathbf{\Omega_R}$. Поскольку **Ω** f-модель, все трассы из $\mathbf{\Omega_R}$ **R**-конвергентны **Ω**, следовательно, они **R**-конвергентны в $\mathbf{\Omega_R}$.

2.1.7. финально-замкнутость $\mathbf{\Omega_R}$. Поскольку $\mathbf{\Omega}$ f-модель, $\mathbf{\Omega_R}$ финально-замкнута.

2.1.8. **R**-полнота $\mathbf{\Omega_R}$. Поскольку $\mathbf{\Omega}$ f-модель, $\mathbf{\Omega_R}$ **R**-полная.

2.1.9. предфинальность. По определению $\boldsymbol{SafeIn}(\mathbf{\Omega_R})\subseteq\boldsymbol{Safe}_{\gamma\Delta}(\mathbf{\Omega_R})$. Также очевидно $\boldsymbol{SafeIn}(\mathbf{\Omega_R})=\boldsymbol{SafeIn}(\mathbf{\Omega})$. Поскольку $\mathbf{\Omega}$ f-модель, $\mathbf{\Omega}$ финально, что означает выполнение свойства 2c для случая $\langle\gamma\rangle\notin\mathbf{\Omega}$:

$\boldsymbol{SafeIn}(\mathbf{\Omega})=\{\sigma\in\mathbf{\Omega}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\mathbf{\Omega}\}$. По определению

$\boldsymbol{Safe}_{\gamma\Delta}(\mathbf{\Omega_R})\subseteq\{\sigma\in\mathbf{\Omega_R}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\mathbf{\Omega_R}\}$

$=\{\sigma\in\mathbf{\Omega}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\mathbf{\Omega}\}$.

Следовательно, $\boldsymbol{Safe}_{\gamma\Delta}(\mathbf{\Omega_R})=\{\sigma\in\mathbf{\Omega_R}\cap(\mathbf{L}\cup\mathbf{R})^*|\sigma\cdot\langle\gamma\rangle\notin\mathbf{\Omega_R}\}$. Поэтому, учитывая, что по доказанному все трассы из $\mathbf{\Omega_R}$ допустимы и согласованы, а $\langle\gamma\rangle\notin\mathbf{\Omega}$, по теореме 3 $\mathbf{\Omega_R}$ предфинально.

2.2. Докажем, что отношение ***safe in*** на $\mathbf{\Omega}$, взятое для трасс из $\mathbf{\Omega_R}$, удовлетворяет правилам отношения ***safe by***. Поскольку $\mathbf{\Omega}$ f-модель, она **Q**-допустима и финальна, а все трассы из $\mathbf{\Omega_R}$ **Q**-конвергентны в $\mathbf{\Omega}$. Поэтому по теореме 7 выполнены условия 2b), 3b), 4b) . Условия 2b), 3b) эквивалентны условиям на ***safe by***, если **Q**-кнопка безопасна по ***safe by*** после неразрушающей трассы тогда и только тогда, когда она неразрушающая и нет **Q**-отказа после трассы. Такое отношение ***safe by*** совпадает с отношением ***safe in*** и $\mathbf{\Omega}=\boldsymbol{ftr}(\mathbf{\Omega_R},\boldsymbol{safe\ in})$.

Список литературы

[1] Bourdonov I., Kossatchev A., Kuliamin V. Formal Conformance Testing of Systems with Refused Inputs and Forbidden Actions. Proc. of MBT 2006, Vienna, Austria, March 2006.

[2] Бурдонов И.Б., Косачев А.С., Кулямин В.В. Формализация тестового эксперимента. «Программирование», 2007, No. 5.

[3] Бурдонов И.Б., Косачев А.С., Кулямин В.В. Безопасность, верификация и теория конформности. Материалы Второй международной научной конференции по проблемам безопасности и противодействия терроризму, Москва, МНЦМО, 2007.

[4] Бурдонов И.Б., Косачев А.С., Кулямин В.В. Теория соответствия для систем с блокировками и разрушением. «Наука», 2008.

[5] Игорь Бурдонов. Теория конформности (функциональное тестирование программных систем на основе формальных моделей). LAP LAMBERT Academic Publishing, Saarbrucken, Germany, 2011, ISBN 978-3-8454-1747-9, 428 стр. (содержание книги доступно по адресу: http://www.ispras.ru/~RedVerst/RedVerst/Publications/TR-01-2007.pdf)

[6] Бурдонов И.Б., Косачев А.С. Системы с приоритетами: конформность, тестирование, композиция. Труды Института системного программирования РАН, N 14.1, 2008.

[7] Бурдонов И.Б., Косачев А.С. Эквивалентные семантики взаимодействия. Труды Института системного программирования РАН, N 14.1, 2008.

[8] Бурдонов И.Б., Косачев А.С. Системы с приоритетами: конформность, тестирование, композиция. «Программирование», 2009, No. 4.

[9] Бурдонов И.Б., Косачев А.С. Аналитическая верификация конформности. Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской суперкомпьютерной конференции (21-26 сентября 2009 г., г. Новороссийск). - М.: Изд-во МГУ, 2009.

[10] Бурдонов И.Б., Косачев А.С. Тестирование конформности на основе соответствия состояний. Труды Института системного программирования РАН, N 18, 2010.

[11] Бурдонов И.Б., Косачев А.С. Симуляция систем с отказами и разрушением. 5-ый Международный симпозиум по компьютерным наукам в России. Семинар «Семантика, спецификация и верификация программ: теория и приложения».Казань 2010.

[12] Бурдонов И.Б., Косачев А.С. Тестирование безопасной симуляции. 5-ый Международный симпозиум по компьютерным наукам в России. Семинар «Семантика, спецификация и верификация программ: теория и приложения».Казань 2010.

[13] Бурдонов И.Б., Косачев А.С. Семантики взаимодействия с отказами, дивергенцией и разрушением. Часть 1. Гипотеза о безопасности и безопасная конформность. «Вестник Томского государственного университета. Управление, вычислительная техника и информатика», №4, 2010.

[14] I.Burdonov, A.Kosachev. Formal conformance verification. Short Papers of the 22nd IFIP ICTSS, Alexandre Petrenko, Adenilso Simao, Jose Carlos Maldonado (eds.), Nov. 08-10, 2010, Natal, Brazil.

[15] Бурдонов И.Б., Косачев А.С. Семантики взаимодействия с отказами, дивергенцией и разрушением. «Программирование», 2010, No. 5.

[16] Бурдонов И.Б., Косачев А.С. Пополнение спецификации для ***ioco***. «Программирование», 2011, No. 1.

[17] Бурдонов И.Б., Косачев А.С. Удаление из спецификации неконформных трасс. Препринт № 23, ИСП РАН, 2011.

[18] Bernot G. Testing against formal specifications: A theoretical view. In S. Abramsky and T.S.E. Maibaum, editors, TAPSOFT'91, Volume 2, pp. 99-119. Lecture Notes in Computer Science 494, Springer-Verlag, 1991.

[19] Edmonds J. Johnson E.L. Matching. Euler Tours, and the Chinese Postman. Math. Programming 5, 88-124 (1973).

[20] van Glabbeek R.J. The linear time – branching time spectrum. In J.C.M. Baeten and J.W. Klop, editors, CONCUR'90, Lecture Notes in Computer Science 458, Springer-Verlag, 1990, pp 278–297.

[21] 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.

[22] Lee D., Yannakakis M. Principles and Methods of Testing Finite State Machines – A Survey. Proceedings of the IEEE 84, No. 8, 1090–1123, 1996.

[23] Milner R. Modal characterization of observable machine behaviour. In G. Astesiano & C. Bohm, editors: Proceedings CAAP 81, LNCS 112, Springer, pp. 25-34.

[24] Tretmans J. Conformance testing with labelled transition systems: implementation relations and test generation. Computer Networks and ISDN Systems, v.29 n.1, p.49-79, Dec. 1996.

[25] Tretmans J. Test Generation with Inputs, Outputs and Repetitive Quiescence. In: Software-Concepts and Tools, Vol. 17, Issue 3, 1996.

The final models of specification

Igor Bourdonov <igor@ispras.ru>, Alexander Kossatchev kos@ispras.ru
ISP RAS, Moscow, Russia

Abstract. The paper describes the research in formal methods of conformance testing of the target system against requirements given in specifications. Such testing is based on interaction semantics defining test stimuli and observations of actions and refusals (absence of actions). Unobservable actions and refusals are also possible. Destruction is introduced as a forbidden action that should be avoided during interaction. A notion of safe testing is also introduced, when no unobservable refusals and destruction occur and no test stimuli applied in divergence. On this basis, the implementation hypothesis of safety and safe conformance are defined, as well as the generation of complete test suite from specification.

The most common model of specification is LTS (Labelled Transition System). However, for the described interaction semantics, only traces (sequences of observations) are important, not the LTS states. Therefore, the most natural is the trace model defined as a set of LTS traces.

The goal of this paper is to define the subset of specification traces sufficient for generation of the complete test suite. We called such subset the final trace model of specification. On the other hand, LTS model is convenient as a way of finite representation of regular trace sets. To represent the final trace model, the paper proposes a variation of LTS called final RTS (Refusal Transition System). The transitions on observable refusals are defined explicitly. Such model is very convenient for test generation: 1) it is deterministic, 2) trace of observations is safe iff it ends in non-terminal state with no destruction, 3) test stimulus is safe after the trace iff it is safe in the final state of the trace.

The paper proposes algorithms for transformation of LTS model into final RTS model. Sufficient conditions for creation of the final RTS in finite time are also defined.

Keywords: interaction semantics, refusals, destruction, divergence, conformance, safe testing, traces, LTS, test generation.

References

[1]. Bourdonov I., Kossatchev A., Kuliamin V. Formal Conformance Testing of Systems with Refused Inputs and Forbidden Actions. Proc. of MBT 2006, Vienna, Austria, March 2006.

[2]. Bourdonov I., Kossatchev A., Kuliamin V. Formalization of Test Experiments. Programming and Computer Software, Vol. 33, No. 5, 2007, pp. 239-260

[3]. Bourdonov I., Kossatchev A., Kuliamin V. Bezopasnost', verifikatsiya i teoriya konformnosti [Safety, Verification and Conformance Theory]. Materialy Vtoroj mezhdunarodnoj nauchnoj konferentsii po problemam bezopasnosti i protivodejstviya terrorizmu [The proceeding of the Second international conference on the problems of safety and counteraction against terrorizm], Moscow, MNCMO, 2007. pp. 135-158. (in Russian).

[4]. Bourdonov I., Kossatchev A., Kuliamin V. Teoriya sootvetstviya dlya system s blokirovkami I razrusheniem [Conformance theory of the systems with Refused Inputs and Forbidden Actions]. Moscow, «Nauka», 2008, 412 p. (in Russian)

[5]. Bourdonov I. Teoriya konformnosti (funkcional'noe testirovanie prorammny'kh system na osnove formal'ny'kh modelej [Conformance theory (functional testing on formal model base)]. LAP LAMBERT Academic Publishing, Saarbrucken, Germany, 2011, ISBN 978-3-8454-1747-9, 428 pp. http://www.ispras.ru/~RedVerst/RedVerst/Publications/TR-01-2007.pdf) (in Russian)

[6]. Bourdonov I., Kossatchev A. Sistemy s prioritetami: konformnost', testirovanie, kompozitsiya [Systems with priority: conformance, testing, composition]. Trudy ISP RAN [The proceeding of ISP RAS], Vol. 14.1, 2008, pp.23-54. (in Russian)

[7]. Bourdonov I., Kossatchev A. Ehkvivalentnye semantiki vzaimodejstviya [Equivalent interaction semantics]. Trudy ISP RAN [The proceeding of ISP RAS], v. 14.1, 2008, pp.55-72. (in Russian)

[8]. Bourdonov I., Kossatchev A. Systems with Priorities: Conformance, Testing, and Composition. Programming and Computer Software, Vol. 35, No. 4, 2009, pp.198-211.

[9]. Bourdonov I., Kossatchev A. Analiticheskaya verifikatsiya konformnosti [Analitical conformance verification]. Nauchnyj servis v seti Internet: masshtabiruemost', parallel'nost', ehffektivnost': Trudy Vserossijskoj superkomp'yuternoj konferentsii (21-26 sentyabrya 2009 g., g. Novorossijsk) [Scientific service of Internet: The proceeding of Russian Supercomputer conference (21-26 sept. 2009, Novorossijsk)] – Moscow, MSU publ., 2009. (in Russian)

[10]. Bourdonov I., Kossatchev A. Testirovanie konformnosti na osnove sootvetstviya sostoyanij [Conformance testing based on a state relation]. Trudy ISP RAN [The proceeding of ISP RAS, Vol. 18, 2010, pp. 183-320. (in Russian)

[11]. Bourdonov I., Kossatchev A. Simulyatsiya sistem s otkazami i razrusheniem [Simulation of the systems with Refused Inputs and Forbidden Actions]. 5-yj Mezhdunarodnyj simpozium po komp'yuternym naukam v Rossii. Seminar «Semantika, spetsifikatsiya i verifikatsiya programm: teoriya i prilozheniya» [5-th International symposium on computer science in Russia. Workshop “Semanticsand program specification and verification: Theory and applications”]. Kazan' 2010. (in Russian)

[12]. Bourdonov I., Kossatchev A. Testirovanie bezopasnoj simulyatsii [Safe simulation testing]. 5-yj Mezhdunarodnyj simpozium po komp'yuternym naukam v Rossii. Seminar «Semantika, spetsifikatsiya i verifikatsiya programm: teoriya i prilozheniya» [5-th International symposium on computer science in Russia. Workshop “Semanticsand program specification and verification: Theory and applications”]. Kazan' 2010. (in Russian)

[13]. Bourdonov I., Kossatchev A. Semantiki vzaimodejstviya s otkazami, divergentsiej i razrusheniem. CHast' 1. Gipoteza o bezopasnosti i bezopasnaya konformnost'. [Semantics of Interaction with Refused Inputs, Divergence and Forbidden Actions. Part 1. Hypothesis of safety and safe conformance]. «Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika» [Tomsk State University. Journal of Control and Computer Science]., №4, 2010. (in Russian)

[14]. I.Burdonov, A.Kosachev. Formal conformance verification. Short Papers of the 22nd IFIP ICTSS, Alexandre Petrenko, Adenilso Simao, Jose Carlos Maldonado (eds.), Nov. 08-10, 2010, Natal, Brazil.

[15]. Bourdonov I., Kossatchev A. Interaction Semantics with Refusals, Divergence, and Destruction. Programming and Computer Software, Vol. 36, No. 5, 2010, pp. 247-263.

[16]. Bourdonov I., Kossatchev A. Specification Completion for IOCO. Programming and Computer Software, Vol. 37, No. 1, 2011, pp. 1-14.

[17]. Bourdonov I., Kossatchev A. Udalenie iz spetsifikatsii nekonformnykh trass [Nonconforming traces elimination from specification]. Preprint № 23, ISP RAN [Preprints of the Institute for System Programming of RAS, Preprint 23], 2011. (in Russian)

[18]. Bernot G. Testing against formal specifications: A theoretical view. In S. Abramsky and T.S.E. Maibaum, editors, TAPSOFT'91, Volume 2, pp. 99-119. Lecture Notes in Computer Science 494, Springer-Verlag, 1991.
[19]. Edmonds J. Johnson E.L. Matching. Euler Tours, and the Chinese Postman. Math. Programming 5, 88-124 (1973).
[20]. van Glabbeek R.J. The linear time – branching time spectrum. In J.C.M. Baeten and J.W. Klop, editors, CONCUR'90, Lecture Notes in Computer Science 458, Springer-Verlag, 1990, pp 278–297.
[21]. 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.
[22]. Lee D., Yannakakis M. Principles and Methods of Testing Finite State Machines – A Survey. Proceedings of the IEEE 84, No. 8, 1090–1123, 1996.
[23]. Milner R. Modal characterization of observable machine behaviour. In G. Astesiano & C. Bohm, editors: Proceedings CAAP 81, LNCS 112, Springer, pp. 25-34.
[24]. Tretmans J. Conformance testing with labelled transition systems: implementation relations and test generation. Computer Networks and ISDN Systems, v.29 n.1, p.49-79, Dec. 1996.
[25]. Tretmans J. Test Generation with Inputs, Outputs and Repetitive Quiescence. In: Software-Concepts and Tools, Vol. 17, Issue 3, 1996.