



ПРИОРИТЕТЫ

- Факультативное и императивное поведение
- Императивность и θ -переход
- Параллельная композиция с θ -переходом в реализации
- Приоритеты стимулов
- Проблема θ -перехода в общем случае
- ε -переход
- Переход по предикату

216

ИСП РАН

RedVerst

Для чего нужны приоритеты?	2
Факультативное и императивное поведение.	2
Императивность и θ -переход.....	2
Параллельная композиция с θ -переходом в реализации.	3
Приоритеты стимулов.	3
ПРОБЛЕМЫ: θ -переход в общем случае.....	4
ε -переход.	4
Переход по предикату.	5
Примеры предикатов на переходах.	7
Тестовые возможности. Машина тестирования.	7
Пример: приоритет стимулов над реакциями.	8
Трассы наблюдений.....	8
Безопасные трассы.	9
Гипотеза о безопасности и конвергентности.....	9
Обобщённое соответствие <i>ioco</i>	10
Спецификация.....	10
Генерация тестов.	10
Асинхронное и композиционное тестирование.	11

Для чего нужны приоритеты?

В предыдущем рассказе мы несколько раз сталкивались с необходимостью введения приоритетов между переходами, определёнными в одном состоянии.

1. В реализации приоритет стимула над внутренней активностью для выхода из цикла внутренней дивергенции.
2. В реализации приоритет стимула над реакцией и внутренней активностью для выхода из цикла осцилляции.
3. В реализации приоритет стимула над реакцией и внутренней активностью для приёма стимула в нестационарном состоянии.
4. В среде приоритет выдачи реакции над приёмом реакции для выхода из цикла дивергенции композиции среды и реализации.

Факультативное и императивное поведение.

До сих пор мы считали, что все переходы, определённые в состоянии, равноприоритетны. Если в данный момент времени могут выполняться несколько переходов, то выбирается для выполнения один из них недетерминированным образом.

Такое поведение будем называть *факультативным*.

Императивным поведением назовём такое поведение, когда приём стимулов имеет приоритет перед выдачей реакций и внутренней активностью (τ -переходами). Если в данный момент времени могут выполняться один или несколько переходов по приёму стимулов, то выбирается для выполнения один из них, независимо от наличия готовых к выполнению переходов по выдаче реакций или (всегда готовых к выполнению) τ -переходов. Если могут приниматься несколько стимулов, то выбор стимула для приёма выполняется недетерминированным образом.

Теперь можно говорить о факультативных автоматах, которые во всех состояниях ведут себя факультативно. Именно такие автоматы мы рассматривали до сих пор.

Императивные автоматы – это автоматы, которые во всех своих состояниях ведут себя императивно.

Наконец, можно говорить о факультативных или императивных состояниях в автоматах смешанного типа. В этом случае при спецификации автомата нужно указывать, какие состояния факультативные, а какие – императивные.

Императивность и θ -переход.

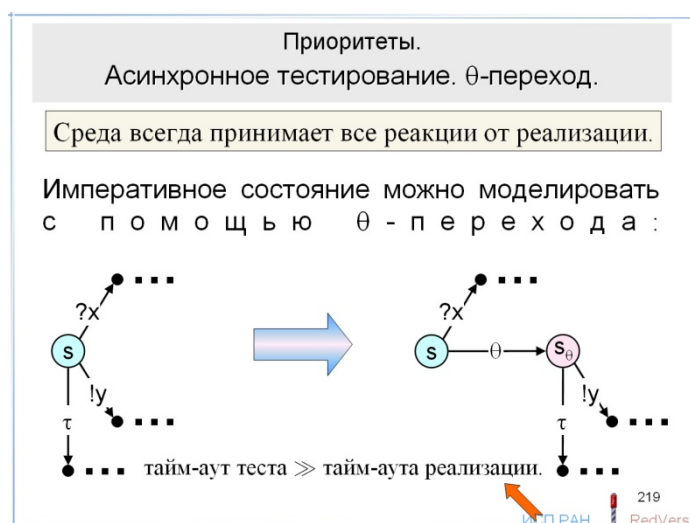
Рассмотрим асинхронное тестирование со средой, которая не тормозит реализацию по реакциям. То есть среда всегда готова принять от реализации любую реакцию.

В этом случае императивное поведение можно моделировать с помощью θ -перехода.

Для этого делаем копию состояния и проводим θ -переход из оригинала в копию. Из состояния-оригинала переходы по реакциям и τ -переходы переносятся в состояние-копию.

Сначала в состоянии-оригинале автомат пытается принять стимулы. Если подходящих стимулов нет, срабатывает θ -переход и автомат выдаёт реакции или совершает τ -переход.

Для того, чтобы эта схема работала, нужно, чтобы тайм-аут θ -перехода в реализации был заведомо меньше тайм-аута θ -перехода в тесте. Тогда тест «не замечает» θ -переходов в реализации. Эти θ -переходы только реализуют приоритеты, но не «обманывают» тест, который по-прежнему понимает истечение тайм-аута в принимающем состоянии как стационарность, а в посылающем – как блокировку посылаемого стимула.



Параллельная композиция с θ -переходом в реализации.

В определении оператора параллельной композиции добавляется ещё одно правило для θ -перехода в реализации.

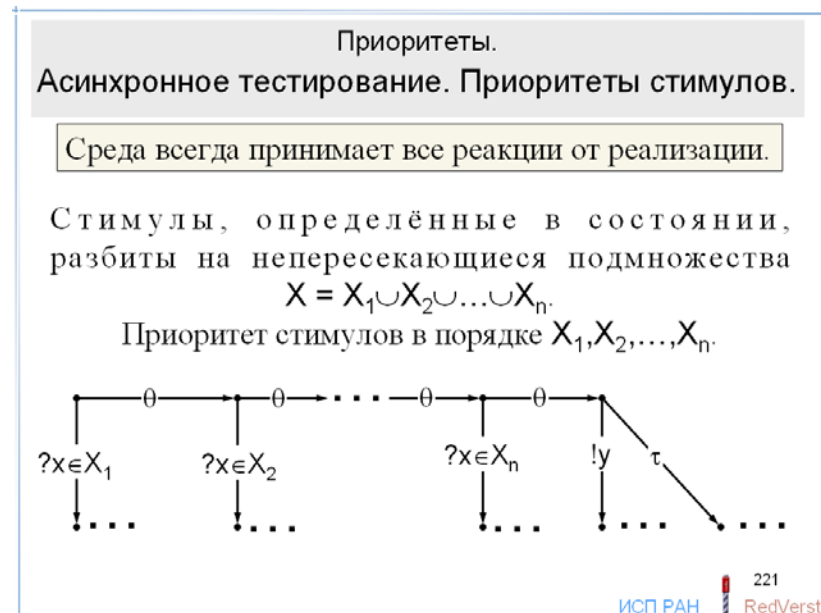
Приоритеты. Параллельная композиция.			
Реализация S в алфавите A	Тест T в алфавите B	$S \parallel T$ в алфавите $(A \setminus \underline{B}) \cup (B \setminus \underline{A})$	
$s \xrightarrow{z} s'$	$t \xrightarrow{z} t'$	$st \xrightarrow{\tau} s't'$ <i>синхронно</i>	1
$s \xrightarrow{a} s'$	$a = \tau \vee a \notin B \vee a = \gamma$	$st \xrightarrow{a} s't$	2
$b = \tau \vee b \notin A \vee b = \gamma$	$t \xrightarrow{b} t'$	$st \xrightarrow{b} st'$	3
deadlock	$t \xrightarrow{\theta} t'$	$st \xrightarrow{\theta} st'$	4
deadlock	$s \xrightarrow{\theta} s'$	$st \xrightarrow{\theta} s't$	5

deadlock = $\forall z \in A \setminus \underline{B} (s \xrightarrow{z} \nleftrightarrow \vee t \xrightarrow{z} \nleftrightarrow) \ \& \ s \xrightarrow{\tau} \nleftrightarrow \ \& \ t \xrightarrow{\tau} \nleftrightarrow$

220 ИСП РАН RedVerst

Приоритеты стимулов.

Аналогичный приём можно использовать для моделирования приоритетов между стимулами, которые определены в состоянии. Мы предполагаем, что имеется линейная последовательность приоритетов, и каждому стимулу назначен один из приоритетов. Тогда автомат сначала пытается принять стимулы с наивысшим приоритетом. Если ни один из них не может быть принят, выполняется θ -переход в состояние, где автомат пытается принять стимулы следующего приоритета. И так далее.



ПРОБЛЕМЫ: θ -переход в общем случае.

В общем случае θ -переход в реализации создаёт проблемы.

- 1) При асинхронном тестировании, когда среда может тормозить реализацию по реакциям.
- 2) Во-вторых при композиционном тестировании. Здесь возникает вопрос о соотношении тайм-аутов θ -переходов взаимодействующих компонентов.

ε -переход.

ε -переходом назовём переход по отсутствию стимулов. Его можно понимать как переход по пустому стимулу, где пустой стимул обозначается символом ε и понимается именно как отсутствие стимулов, передаваемых в автомат в данном состоянии в текущий момент времени.

Мы можем рассматривать в автомате трассы, обогащённые символом ε .

В общем, для всюду определённых по стимулам автоматов ε -переход эквивалентен θ -переходу. Однако в общем случае они имеют разную семантику и не сводятся друг к другу.

Переход по предикату.

Общий случай приоритетов – это зависимость возможности выполнения перехода по данному символу не только от наличия этого символа (в парном состоянии окружения определён переход по противоположному символу), но и от наличия или отсутствия других символов (переходов по противоположным символам в парном состоянии окружения).

Для автомата в алфавите A рассматриваем предикаты на переходах, имеющие общий вид:

$$P: \wp(A \cup \{\tau\}) \rightarrow \mathbf{Bool}.$$

Это можно понимать как булевскую функцию от булевских переменных, индекс которых пробегает множество A .

Расширение предиката надмножеством.

Пусть задано надмножество $E \supseteq A$.

Определим расширение предиката надмножеством E :

$$\begin{aligned} P[E]: \wp(E \cup \{\tau\}) &\rightarrow \mathbf{Bool}. \\ U \subseteq E \cup \{\tau\} &: P[E](U) =_{\text{def}} P(U \cap (A \cup \{\tau\})). \end{aligned}$$

Сужение предиката подмножествами.

Пусть заданы два непересекающихся подмножества $T, F \subseteq A$, $T \cap F = \emptyset$. Будем считать, что подмножество T задаёт индексы переменных, значение которых фиксируется как **true**, а подмножество F задаёт индексы переменных, значение которых фиксируется как **false**.

Определим сужение предиката подмножествами T и F :

$$\begin{aligned} P\langle T, F \rangle: \wp((A \cup \{\tau\}) \setminus (T \cup F)) &\rightarrow \mathbf{Bool}. \\ U \subseteq (A \cup \{\tau\}) \setminus (T \cup F) &: P\langle T, F \rangle(U) =_{\text{def}} P(U \cup T). \end{aligned}$$

Задание предикатов на переходах.

Будем считать, что предикат P , заданный на переходе из состояния S , сужен подмножествами $P\langle T, F \rangle = P$, где

$$T = \{\tau \mid S \xrightarrow{\tau} \},$$

$$F = \{z \in A \mid S \xrightarrow{z} \nrightarrow \}.$$

Сужение-расширение предиката при композиции.

Рассмотрим композицию автомата A в алфавите A и автомата B в алфавите B . Пусть ab – достижимое композиционное состояние. Для каждого перехода из состояния a его предикат P сузим подмножествами T и F и расширим надмножеством E :

$$T(a, b) = \{ z \in A \cap B \mid a \xrightarrow{z} \ \& \ b \xrightarrow{z} \},$$

$$F(a, b) = \{ z \in A \cap B \mid a \xrightarrow{z} \ \& \ b \xrightarrow{z} \nrightarrow \},$$

$$E(a, b) = A \setminus B \cup B \setminus A,$$

$$P(a, b) = P\langle T(a, b), F(a, b) \rangle[E(a, b)].$$

Вычисление предикатов композиционных переходов.

Предикат R асинхронного перехода композиционного автомата, порождённого переходом автомата A с предикатом P , вычисляется так:

$$R = P(a, b).$$

Предикат R асинхронного перехода композиционного автомата, порождённого переходом автомата B с предикатом Q , вычисляется так:

$$R = Q(b, a).$$

Предикат R синхронного перехода композиционного автомата, порождённого переходом автомата A с предикатом P и переходом автомата B с предикатом Q , вычисляется так:

$$R = P(a, b) \ \& \ Q(b, a).$$

Оператор композиции.

$$\begin{array}{llll} z \in A_{\gamma\tau} \setminus \underline{B} & \& a \xrightarrow{\gamma} z, P \rightarrow a' & \vdash ab \xrightarrow{\gamma} z, P(a, b) \rightarrow a' b'; \\ z \in B_{\gamma\tau} \setminus \underline{A} & \& b \xrightarrow{\gamma} z, Q \rightarrow b' & \vdash ab \xrightarrow{\gamma} z, Q(b, a) \rightarrow ab'; \\ z \in A \cap B & \& a \xrightarrow{\gamma} z, P \rightarrow a' \quad \& \quad b \xrightarrow{\gamma} z, Q \rightarrow b' & \vdash ab \xrightarrow{\gamma} z, P(a, b) \& Q(b, a) \rightarrow a' b'. \end{array}$$

Приоритеты.			
Предикаты на переходах			
<u>Вычисление предикатов переходов композиции в состоянии ab.</u>			
$P(a,b)$	- асинхронный композиционный переход, порождённый переходом из состояния a		
$Q(b,a)$	- асинхронный композиционный переход, порождённый переходом из состояния b		
$P(a,b) \ \& \ Q(b,a)$	- синхронный композиционный переход		
A в алфавите A	B в алфавите B	$A \upharpoonright B$ в алфавите $(A \setminus B) \cup (B \setminus A)$	
$s \xrightarrow{\gamma} z, P \rightarrow s'$	$t \xrightarrow{\gamma} z, Q \rightarrow t'$	$st \xrightarrow{\gamma} z, P(a,b) \& Q(b,a) \rightarrow s't'$	1
$s \xrightarrow{\gamma} a, P \rightarrow s'$	$a = \tau \vee a \notin B \vee a = \gamma$	$st \xrightarrow{\gamma} a, P(a,b) \rightarrow s't$	2
$b = \tau \vee b \notin A \vee b = \gamma$	$t \xrightarrow{\gamma} b, Q \rightarrow t'$	$st \xrightarrow{\gamma} b, Q(b,a) \rightarrow st'$	3

Выполнение автомата.

Для выполнения автомата A нужно скомпоновать его с автоматом B , в котором нет переходов, а алфавит $B = \underline{A}$.

В такой композиции будут только τ -переходы, а все предикаты – константы (**true** или **false**). Удаляем переходы с ложными предикатами.

На самом деле, константные предикаты могут возникать и раньше. Мы можем сразу удалять переходы с константными ложными предикатами.

Примеры предикатов на переходах.

Приоритет стимулов над реакциями и τ .

Предикат на переходе по стимулу:

$$P(U) = \text{true}.$$

Предикат на переходе по реакции или τ :

$$P(U) = (?U=\emptyset),$$

где $?U = \{?x \in U\}$.

Приоритеты символов $1 \dots n$.

$A = A_1 \cup \dots \cup A_n$ – разбиение на непересекающиеся подмножества.

Предикат на переходе по символу из A_i : $P(U) = (U \cap (A_1 \cup \dots \cup A_{i-1}) = \emptyset)$.

Правила в состоянии S следующие:

$$s \xrightarrow{?a, P_1} s_1 \quad : \quad P_1(U) = ?b \in U$$

$$s \xrightarrow{?a, P_2} s_2 \quad : \quad P_2(U) = ?b \notin U$$

$$s \xrightarrow{?b, P_3} s_3 \quad : \quad P_3(U) = ?a \notin U$$

$$s \xrightarrow{!y, P_4} s_4 \quad : \quad P_4(U) = ?a \notin U \ \& \ ?b \notin U$$

Здесь два перехода по стимулу $?a$, имеющие разные предикаты. Один выполняется, если предлагается стимул $?b$, а другой в противном случае.

Символы Север, Восток, Юг, Запад. Нет τ -переходов. Выбираем любое направление из возможных, но только не среднее, если возможны три направления. Предикат перехода определяется его символом:

$$P_{\text{Север}}(U) = (U \neq \{ \text{Запад, Север, Восток} \})$$

$$P_{\text{Восток}}(U) = (U \neq \{ \text{Север, Восток, Юг} \})$$

$$P_{\text{Юг}}(U) = (U \neq \{ \text{Восток, Юг, Запад} \})$$

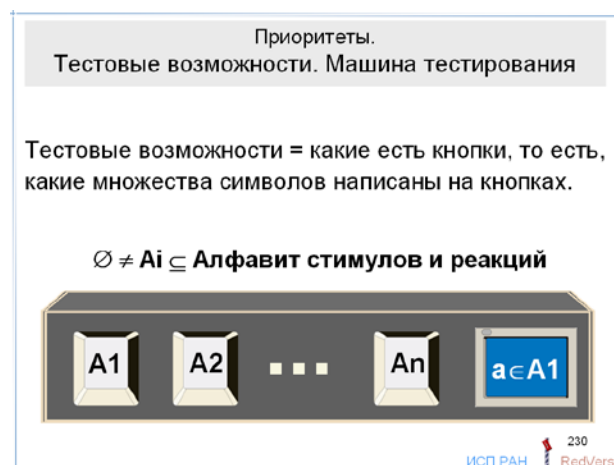
$$P_{\text{Запад}}(U) = (U \neq \{ \text{Юг, Запад, Север} \})$$

Тестовые возможности. Машина тестирования.

В связи с предикатами переходов мы должны вернуться к общему случаю машины тестирования, когда на кнопке написано множество символов наблюдаемых действий, и нажатие кнопки означает разрешение машине выполнить любое из этих действий.

Тайм-аут при нажатой кнопке означает (при отсутствии дивергенции), что ни одно из действий, разрешённых этой кнопкой, машина выполнить не может.

Тестовые возможности определяются набором таких кнопок, то есть семейством подмножеств символов, задаваемым кнопками.



Пример: приоритет стимулов над реакциями.

В качестве примера можно рассмотреть машину тестирования, которая позволяла бы тестировать приоритетное выполнение стимулов над реакциями.

В такой машине, кроме набора кнопок “!x” для посылки одиночных стимулов и кнопки “?” приёма всех реакций, должны быть комбинированные кнопки “?+!x” – «послать данный стимул и принять все реакции».

Когда может считать правильным приём в тесте реакции ?y при нажатой кнопке “?+!x” после наблюдения некоторой трассы σ ?

Если бы не было приоритета стимулов над реакциями, такая реакция может поступить, если в спецификации после трассы σ есть состояние, где выдаётся реакция !y.

$\leftarrow \sigma \rightarrow$
!x₁→!x₁ , !x₁→θ , ?→?y , ?→θ , ?+!x₁→!x₁ , ?+!x₁→?y , ?+!x₁→θ .

Без приоритетов:

в некотором состоянии спецификации после трассы σ есть выдача реакции y

Однако, если приём стимулов имеет приоритет над выдачей реакций, то такая реакция может поступить, если в спецификации после трассы σ есть состояние, где выдаётся реакция !y и не принимается стимул ?x.

$\leftarrow \sigma \rightarrow$
!x₁→!x₁ , !x₁→θ , ?→?y , ?→θ , ?+!x₁→!x₁ , ?+!x₁→?y , ?+!x₁→θ .

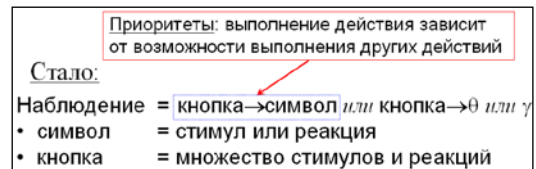
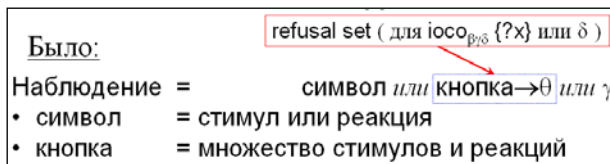
Приоритет стимулов над реакциями:

в некотором состоянии спецификации после трассы σ есть выдача реакции y, но нет приёма стимула x₁

Трассы наблюдений.

Под трассой в общем случае понимается последовательность наблюдений.

Если нет приоритетов, то есть взаимовлияния символов, разрешаемых кнопкой, то для наблюдения важно множество A символов, написанных на кнопке, только в одном случае: когда это наблюдение истечения тайм-аута. Символ θ означает, что машина не может выполнить ни одно из действий множества A, то есть это refusal set A. Мы рассматривали два вида таких refusal set: стационарность «A = все реакции» и блокировка стимула «A = один стимул». В то же время наблюдения внешнего действия z означает выполнение машиной действия $z \in A$, что не зависит от того, какие ещё символы разрешены кнопкой, то есть принадлежат A.



Если приоритеты есть, то наблюдение внешнего действия z имеет разный смысл при разных множествах A , содержащих z . Иными словами, зависит от того, какие ещё символы разрешены кнопкой.

Безопасные трассы.

Трассу будем считать безопасной, если от неё нет ответвлений по разрушению.

Это определение отличается от использованного в предыдущем тексте, поскольку раньше нельзя было ответвляться даже по последовательности реакций и далее по разрушению.

Это объясняется тем, что раньше у нас была только одна кнопка для приёма (всех) реакций. Мы могли говорить о пассивном поведении окружения как поведении, которое только принимает реакции и принимает все реакции. Теперь, в связи с приоритетами, появляется много таких кнопок, и вариантов «пассивного поведения» становится много.

Вообще говоря, всю изложенную выше теорию в связи с безопасностью можно (а, пожалуй, что и нужно) переделать под новое определение безопасности.

Тогда соответствие $ioco_{\beta, \gamma, \delta}$ для синхронного тестирования усилится, поскольку станет больше безопасных трасс. Мы будем посылать стимул, если сразу после его приёма нет разрушения, но иногда не будем принимать реакции, поскольку приём реакций может привести к тому, что будет выдана реакция, сразу после которой идёт разрушение.

Для асинхронного тестирования, когда среда не тормозит реализацию по реакциям (например, неограниченная выходная очередь) всё останется по-прежнему, но только мы, учитывая безопасность, не должны посылать стимул, если после него через цепочку реакций достижимо разрушение. Эту цепочку реакций нетормозящая среда всегда может выбрать.

Можно сказать, что теперь при асинхронном тестировании безопасных трасс становится меньше по сравнению с синхронным тестированием, что приводит к большему ослаблению соответствия в асинхронном случае или, другими словами, к большей «вседозволенности» при асинхронном тестировании.

Гипотеза о безопасности и конвергентности.

Трасса безопасно-тестируемая, если она безопасна и проходит только через конвергентные состояния.

Гипотеза о безопасности и конвергентности теперь выглядит так:

Пусть трасса μ безопасна в спецификации S и эта трасса есть в реализации I .
Тогда:

Безопасность: μ безопасна в I .

Конвергентность: если S после μ конвергентна, то I после μ конвергентна.

Обобщённое соответствие *ioco*.

Соответствие обобщается следующим образом:

Если трасса μ безопасна в спецификации S и эта трасса есть в реализации I , то в I после μ может быть данное наблюдение только в том случае, когда это наблюдение может быть в S после μ .

Спецификация.

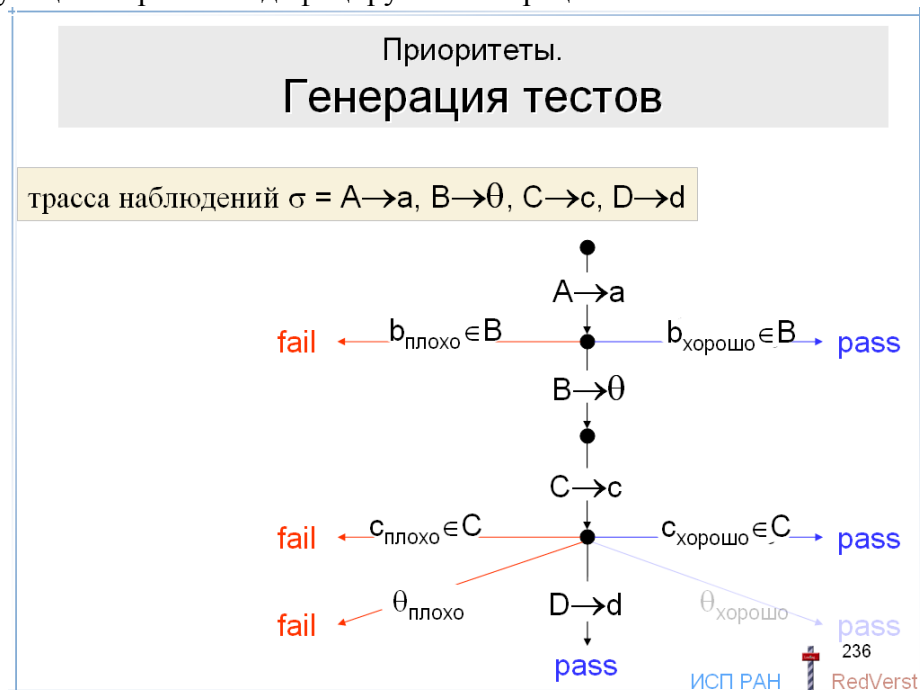
Чтобы тестирование по безопасным трассам спецификации было безопасным, к спецификации предъявляются требования:

Безопасность: Спецификация не разрушается «сама по себе», то есть из начального состояния по τ -переходам недостижимо разрушение γ .

Безопасно-конвергентность: После безопасной трассы спецификация конвергентна.

Генерация тестов.

Соответствующим образом модифицируется генерация тестов.



Асинхронное и композиционное тестирование.

Асинхронное тестирование. Во многих случаях мы можем считать, что среда не имеет предикатов (все предикаты тождественно истинны). Однако, если реализация имеет приоритеты, то композиция реализации со средой может иметь неконстантные предикаты. Поскольку асинхронное тестирование – это синхронное тестирование композиции реализации со средой, в таком тестировании должны учитываться предикаты.

Композиционное тестирование. Композиционное тестирование – это синхронное тестирование композиции компонентов. В композиции могут быть неконстантные предикаты, если компоненты имеют неконстантные предикаты.

ЧТО ОСТАЛОСЬ ЗА СКОБКАМИ?

В нашем обзоре тестирования соответствия мы обошли стороной следующие вопросы:

1. Метрики, покрытия и связанная с ними структура спецификаций.
2. Другие спецификации (например, алгебраические).
3. Другие модели (например, геометрические автоматы).
4. Время (временные автоматы).
5. Вероятности. Вероятностные алгоритмы и вероятностные автоматы.

