Метод редукции вероятностных систем переходов

А.М.Миронов, С.Л.Френкель

Содержание

Аннотация

Рассматривается задача редукции вероятностных систем переходов (ВСП) с целью понижения сложности верификации таких систем. Верификация ВСП заключается в вычислении истинностных значений формул вероятностной темпоральной логики (PCTL, Probabilistic Computational Tree Logic) в начальных состояниях ВСП. Введено понятие эквивалентности состояний ВСП, и указан алгоритм удаления эквивалентных состояний, в результате работы которого получается такая ВСП, у которой все свойства, выражаемые формулами логики PCTL, совпадают со свойствами исходной ВСП.

Ключевые слова: верификация; вероятностные системы переходов; вероятностная темпоральная логика; редукция вероятностных моделей

1 Введение

1.1 Постановка задачи, мотивация и научная новизна

В настоящей работе рассматривается задача редукции **вероятностных систем переходов (ВСП)**, целью которой является понижение сложности верификации свойств ВСП, выражаемых формулами вероятностной темпоральной логики PCTL.

ВСП представляют собой один из наиболее широко используемых классов моделей дискретных динамических систем. Понятие ВСП является обобщением понятия цепи Маркова [2], которое имеет широкие применения в естественных и гуманитарных науках. Понятие ВСП можно рассматривать также как частный случай понятия вероятностного автомата [3]. Главной отличительной особенностью понятия ВСП от понятий цепи Маркова и вероятностного автомата является наличие выразительного логического формализма, позволяющего эффективно описывать различные свойства поведения ВСП. В качестве такого формализма выступает вероятностная темпоральная логика PCTL ([4], [13]), которая

- представляет собой вероятностный аналог темпоральной логики ветвящегося времени CTL [5], использующейся для спецификации свойств параллельных и распределённых программ, и

- является эффективным инструментом для описания различных свойств дискретных вероятностных динамических систем.

Формулы логики PCTL могут отражать различные вероятностные аспекты поведения анализируемых систем, к числу которых относятся например

- частота выполнения тех или иных действий или переходов в анализируемых системах

- вероятность отказа компонентов анализируемых систем

- вероятностный характер взаимодействия анализируемой системы с её окружением, например: частота поступления входных запросов или сообщений, частота получения искажённых сообщений (для протоколов передачи сообщений в компьютерных сетях), и т.п.

В последнее время класс вероятностных моделей находит всё большее применение в различных задачах верификации программных систем. Одной из главных причин актуальности данного класса моделей в задачах верификации программных систем является невысокая (по сравнению с детерминированными моделями) сложность вероятностных моделей анализируемых систем.

Поскольку вероятностная модель является огрубленным представлением анализируемой системы, то результат её верификации может отличаться от свойств исходной системы, и одной из актуальных научных задач является оценка степени расхождения свойств исходной системы и свойств её вероятностной модели. Однако эта задача не является предметом данного исследования. В настоящей работе мы рассматриваем следующую задачу: мы предполагаем, что вероятностная модель анализируемой системы уже построена, и требуется преобразовать её в такую

модель, чтобы задача анализа свойств исходной системы, выражаемых виде формул вероятностной темпоральной логики, допускала бы более простое решение для редуцированной модели, а результаты верификации исходной и редуцированной модели были бы одинаковы.

Некоторые подходы к редукции ВСП изучались в различных работах по вероятностной верификации, однако в этих исследованиях были рассмотрены лишь частные методы редукции ВСП, такие как редукция частичных порядков ([6], [7]) и редукция основанная на понятии симметрии множества состояний ВСП ([8], [9]). Данные методы можно эффективно использовать лишь для ВСП достаточно специального вида, как правило это – вероятностные модели параллельных и распределённых программ.

В настоящей работе рассмотрен другой подход к редукции анализируемой вероятностной модели, который кратко может быть описан как удаление избыточных состояний. Данный метод является новым, и представляет собой вероятностное обобщение метода минимизации конечных автоматов.

1.2 Исторический обзор и современное состояние дел в области верификации вероятностных систем переходов

Верификация вероятностных систем переходов (англоязычное название этой области - Probabilistic Model Checking) в настоящее время является одним из наиболее широко используемых методов верификации вычислительных систем. Главным достоинством этого метода является возможность полностью автоматического анализа модели верифицируемой системы. К числу его существенных недостатков относится высокая вычислительная сложность процедуры верификации.

Первые алгоритмы вероятностной верификации были предложены в 1980-е годы в работах [10], [11], [12]. Данные алгоритмы были предназначены для верификации качественных вероятностных свойств (то есть таких, которые выполняются с вероятностью 1 или 0). Затем эти алгоритмы были обобщены на случай верификации количественных вероятностных свойств (в

спецификации таких свойств могло присутствовать любое значение вероятности). Эти алгоритмы были изложены в работах [13], [14], [15]. Программные реализации этих алгоритмов были представлены в работах [16], [17].

Первые промышленные системы вероятностной верификации были разработаны в 2000-х годах [18], [19]. Эти системы вероятностной верификации успешно применяются во многих областях, в том числе: анализ распределенных алгоритмов, телекоммуникационные протоколы, компьютерная безопасность, криптографические протоколы, моделирование биологических процессов. С использованием этих систем верификации были обнаружены уязвимости и аномальные поведения анализируемых систем, подробнее см. в [20] и [22]. При помощи систем вероятностной верификации могут быть вычислены такие характеристики программных систем как например вероятность вторжения злоумышленника в компьютерную сеть, мат. ожидание времени отклика веб-сервиса, и другие количественные и качественные характеристики.

Наиболее популярной практической системой вероятностной верификации в настоящее время является система PRISM [21], [22], разработанная на факультете компьютерных наук Оксфордского Университета (Великобритания) в группе Quantitative Analysis and Verification под руководством Марты Квиатковской [1].

1.3 Структура статьи

Во второй главе мы даём формулировку основного понятия, изучаемого в настоящей работе - вероятностной системы переходов (ВСП), и приводим два простых примера ВСП. В третьей главе мы вводим представление ВСП в терминах случайных функций и матриц. В четвертой главе мы описываем логику PCTL и определяем её семантику (т.е. правила вычисления значений формул логики PCTL в состояниях ВСП). Пятая глава является центральной - в ней мы излагаем метод редукции ВСП, для чего вводим понятие эквивалентности состояний и описываем алгоритм вычисления классов эквивалентности на множестве состояний анализируемой ВСП. Также мы излагаем алгоритм преобразования ВСП, связанный с удалением избыточных со-

стояний, и обосновываем его корректность. В шестой и седьмой главах мы рассматриваем практические иллюстрации изложенного метода редукциии ВСП. В заключении мы формулируем актуальные проблемы для дальнейших исследований в области вероятностной верификации.

2 Вероятностные системы переходов

2.1 Понятие вероятностной системы переходов

Мы предполагаем, что задано конечное множество AP, элементы которого называются **атомарными утверждениями**. Ниже запись 2^{AP} обозначает множество всех подмножеств AP.

Вероятностная система переходов (ВСП) (называемая также в англоязычной литературе **Discrete Time Markov Chain**) – это четверка D вида

$$D = (S, s^0, P, L) \tag{1}$$

компоненты которой имеют следующий смысл.

1. S – конечное множество, элементы которого называются **состояниями** ВСП D.

2. $s^0 \in S$ – выделенное состояние, называемое **начальным состоянием** ВСП D.

3. P – функция вида

   $$P : S \times S \to [0, 1]$$

   называемая **функцией перехода** ВСП D, и удовлетворяющая условию:

   $$\forall\, s \in S \quad \sum_{s' \in S} P(s, s') = 1.$$

   Для каждой пары $(s_1, s_2) \in S \times S$ число $P(s_1, s_2)$ понимается как вероятность того, что если в текущий момент времени D находится в состоянии s_1, то через один такт времени D

будет находиться в состоянии s_2. Если $P(s_1, s_2) > 0$, то мы будем называть тройку $(s_1, s_2, P(s_1, s_2))$ **переходом** из s_1 в s_2 с вероятностью $P(s_1, s_2)$. Ниже запись $s_1 \xrightarrow{a} s_2$ является другим обозначением перехода (s_1, s_2, a).

4. L – функция вида
$$L : S \to 2^{AP} \tag{2}$$
называемая **оценкой**, которая имеет следующий смысл: для каждого состояния $s \in S$ и каждого атомарного утверждения $p \in AP$ утверждение p считается

   - **истинным** в состоянии s, если $p \in L(s)$, и
   - **ложным** в состоянии s, если $p \notin L(s)$.

ВСП удобно рассматривать как помеченный граф,

- вершинами которого являются состояния, помеченные элементами множества 2^{AP}: каждая вершина $s \in S$ имеет метку $L(s)$, и
- для каждой пары $(s_1, s_2) \in S \times S$ такой, что $P(s_1, s_2) > 0$, граф содержит ребро из s_1 в s_2 с меткой $P(s_1, s_2)$.

2.2 Пути в вероятностных системах переходов

Путь в ВСП (1) – это конечная или бесконечная последовательность состояний
$$\pi \;=\; (\; s_0,\; s_1,\; \ldots\;) \tag{3}$$
такая, что для каждой пары (s_i, s_{i+1}) соседних состояний в этом пути верно неравенство $P(s_i, s_{i+1}) > 0$. Если последовательность (3) бесконечна, то путь π называется **бесконечным**, в противном случае он называется **конечным**.

При рассмотрении ВСП как графа, каждый путь (3) в ней можно отождествлять с соответствующей последовательностью рёбер (из s_0 в s_1, из s_1 в s_2, и т.д.).

Мы будем говорить, что путь π **выходит из состояния** s, если первым состоянием (т.е. состоянием с номером 0) этого пути является s.

Если π – конечный путь вида

$$\pi = (\ s_0,\ \ldots,\ s_n\ ) \tag{4}$$

то мы будем говорить, что π – **путь из** s_0 **в** s_n. Мы будем обозначать знакосочетанием $s_0 \xrightarrow{*} s_n$ тот факт, что существует путь из s_0 в s_n.

Для каждого пути π вида (3) и каждого $s \in S$ запись $s \in \pi$ означает, что $s = s_i$ для некоторого $i \geq 0$.

Отрезком пути (3) называется произвольная подпоследовательность π' последовательности (3), т.е. произвольный путь вида

$$\pi' = (\ s_i,\ s_{i+1},\ \ldots,\ s_{i+k}\ ) \tag{5}$$

где $k \geq 0$. Число k называется **длиной** отрезка (5). Отрезок (5) обозначается записью $[s_i, s_{i+k}]$. Отрезок (5) называется **начальным** отрезком пути (3), если $i = 0$.

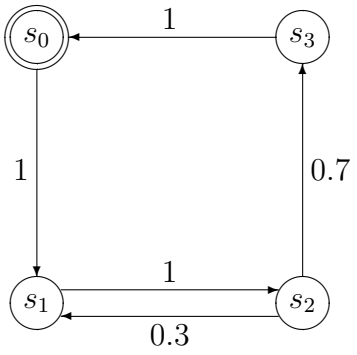
2.3 Примеры вероятностных систем переходов

Приведём два примера ВСП.

1. Упрощённая модель протокола передачи сообщений через ненадёжный канал (в котором сообщения могут пропадать). Протокол представляет собой систему, состоящую из

   - двух агентов - отправителя и получателя, а также
   - канала, в который помещаются сообщения, пересылаемые от одного агента другому.

   Граф, представляющий эту ВСП, имеет следующий вид:



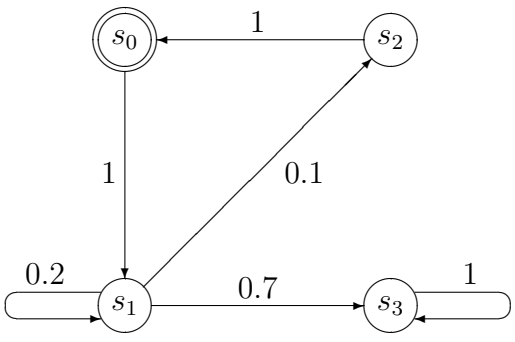
(6)

Переходы этой ВСП имеют следующий смысл.

- Переход $s_0 \xrightarrow{1} s_1$ заключается в получении отправителем от внешнего источника сообщения, которое должно быть передано через канал получателю.
- Переход $s_1 \xrightarrow{1} s_2$ заключается в помещении сообщения в канал отправителем.
- Переход $s_2 \xrightarrow{0.3} s_1$ заключается в потере сообщения в канале.
- Переход $s_2 \xrightarrow{0.7} s_3$ заключается в передаче сообщения из канала получателю.
- Переход $s_3 \xrightarrow{1} s_0$ заключается в получении сообщения получателем и посылка им подтверждения отправителю.

2. Упрощённая модель агента, работа которого представляет собой последовательность сеансов. В каждом из этих сеансов агент пытается выполнить некоторое действие. Если действие было выполнено, то он переходит к следующему сеансу, а если не было выполнено, то он завершает свою работу.

   Граф, представляющий эту ВСП, имеет следующий вид:



Состояния этой ВСП имеют следующий смысл:

- В состоянии s_0 агент начинает очередной сеанс.
- В состоянии s_1 агент предпринимает попытку выполнения действия (переход $s_1 \xrightarrow{0.2} s_1$ означает, что попытка выполнения действия пока не осуществима).
- В состоянии s_2 агент находится тогда, когда действие было выполнено успешно.
- В состоянии s_3 агент оказывается тогда, когда его попытка выполнения действия закончилась неудачей.

3 Матричное представление вероятностных систем переходов

3.1 Случайные функции

Пусть X и Y – два конечных множества.

Случайной функцией (СФ) из X в Y называется произвольная функция f вида

$$f : X \times Y \to [0, 1] \tag{7}$$

удовлетворяющая условию:

$$\forall\, x \in X \quad \sum_{y \in Y} f(x, y) = 1$$

Для любых $x \in X$ и $y \in Y$ значение $f(x, y)$ можно интерпретировать как вероятность того, СФ f отображает x в y.

СФ (7) называется **детерминированной**, если для каждого $x \in X$ существует единственный $y \in Y$, такой, что $f(x, y) = 1$. Если f – детерминированная СФ вида (7), и x, y – такие элементы X и Y соответственно, что $f(x, y) = 1$, то мы будем говорить, что f **отображает** x **в** y.

Если f – СФ из X в Y, то мы будем обозначать этот факт записью

$$f : X \underset{r}{\to} Y$$

Мы будем называть X **областью определения** СФ f, а Y – **областью значений** СФ f.

Для каждого конечного множества X запись id_X обозначает детерминированную СФ $X \to X$, которая отображает каждый $x \in X$ в x.

3.2 Матрицы, соответствующие случайным функциям

Если СФ f имеет вид $f : X \underset{r}{\to} Y$, и на множествах X и Y заданы упорядочения их элементов, которые имеют вид

$$(x_1, \ldots . x_m) \quad \text{и} \quad (y_1, \ldots, y_n)$$

соответственно, то СФ f можно представить в виде матрицы (обозначаемой тем же символом f)

$$f = \begin{pmatrix} f(x_1, y_1) & \ldots & f(x_1, y_n) \\ \ldots & \ldots & \ldots \\ f(x_m, y_1) & \ldots & f(x_m, y_n) \end{pmatrix} \tag{8}$$

Ниже мы будем отождествлять каждую СФ f с соответствующей ей матрицей (8).

Мы будем предполагать, что для каждого конечного множества X, являющегося областью определения или областью значений какой-либо из рассматриваемых СФ, на X задано фиксированное упорядочение его элементов. Таким образом, для каждой рассматриваемой СФ соответствующая ей матрица определена однозначно.

Для каждой СФ $f : X \underset{r}{\to} Y$ и произвольных $x \in X$, $y \in Y$ мы будем называть

- строку $(f(x, y_1), \ldots, f(x, y_n))$ матрицы f – **строкой** x, и
- столбец $\begin{pmatrix} f(x_1, y) \\ \ldots \\ f(x_m, y) \end{pmatrix}$ матрицы f – **столбцом** y.

Если f и g – СФ вида

$$f : X \underset{r}{\to} Y, \quad g : Y \underset{r}{\to} Z$$

то их **композицией** называется СФ $f \cdot g : X \underset{r}{\to} Z$, определяемая следующим образом:

$$\forall\, x \in X \quad (f \cdot g)(x) \stackrel{\text{def}}{=} \sum_{y \in Y} f(x,y) \cdot g(y,z) \tag{9}$$

Согласно определению произведения матриц, из (9) следует, что матрица $f \cdot g$ является произведением матриц f и g.

3.3 Матрицы, соответствующие вероятностным системам переходов

Пусть задана ВСП $D = (S, s^0, P, L)$, и список элементов множества S имеет вид $(s_1, \ldots, s_n)$.

Мы будем использовать следующие обозначения.

1. Символ $\mathbf{1}$ обозначает множество, состоящее из одного элемента, который мы будем обозначать символом e.

2. Для каждого состояния $s \in S$ запись I_s обозначает детерминированную СФ вида

   $$I : \mathbf{1} \underset{r}{\to} S$$

   отображающую элемент $e \in \mathbf{1}$ в состояние s ВСП D.

3. Для каждого $n \geq 0$ обозначим записью P^n СФ вида

   $$P^n : S \underset{r}{\to} S$$

   определяемую индуктивно:

   - $P^0 \stackrel{\text{def}}{=} id_S$, и
   - $\forall\, n \geq 0 \quad P^{n+1} \stackrel{\text{def}}{=} P^n \cdot P$.

   Нетрудно видеть, что матрицы, соответствующие СФ P^i, имеют следующий вид: P^0 – единичная матрица, и $\forall\, n > 0$ матрица P^n является n-й степенью матрицы P.

   Для любых $n \geq 0, \quad s_1, s_2 \in S$ число $P^n(s_1, s_2)$ можно понимать как вероятность того, что если в текущий момент времени ВСП D находится в состоянии s_1, то через n тактов времени D будет находиться в состоянии s_2.

4 Логика PCTL

4.1 Свойства вероятностных систем переходов

Логика PCTL (её название является аббревиатурой англоязычного названия **Probabilistic Computation Tree Logic**) – это темпоральная логика, предназначенная для формального описания свойств ВСП. Логика PCTL была введена Х. Ханссоном (H. Hansson) и Б. Джонссоном (B. Jonsson) в работе [13].

Приведём несколько примеров свойств ВСП, которые могут быть описаны в виде формул логики PCTL.

1. Вероятность доставки сообщения в заданном временном интервале $[t_1, t_2]$ не меньше 0.999.

2. В результате работы алгоритма избрания процесса-лидера избрание такого процесса завершится с вероятностью 1.

3. Вероятность успешной атаки на криптографический протокол не превышает 0.0001%.

4.2 Формулы логики PCTL

В определении понятия формулы логики PCTL мы будем использовать множество AP атомарных утверждений, введённое в пункте 2.1.

Формулы логики PCTL делятся на два класса:

- $StateFm$ – **формулы состояний**, и
- $PathFm$ – **формулы путей**.

Формулы из классов $StateFm$ и $PathFm$ мы будем обозначать символами φ и α соответственно (возможно, с индексами), а формулу произвольного вида – символом f (возможно, с индексом).

Классы $StateFm$ и $PathFm$ определяются следующим образом.

- $StateFm$:

    1. Каждое атомарное утверждение p из AP является формулой из $StateFm$.

2. Символы $\top$ и $\bot$ является формулами из $StateFm$.
3. Если φ_1 и φ_2 – формулы из $StateFm$, то следующие знакосочетания являются формулами из $StateFm$:

$$\neg\varphi_1, \quad \varphi_1 \wedge \varphi_2, \quad \varphi_1 \vee \varphi_2, \quad \varphi_1 \to \varphi_2, \quad \varphi_1 \leftrightarrow \varphi_2$$

4. Если
   - $\triangle$ – функциональный символ, которому соответствует функция (обозначаемая тем же символом) вида
   $$\triangle : [0,1] \times [0,1] \to \{0,1\}$$
   - a – число из $[0,1]$, и
   - α – формула из $PathFm$

   то знакосочетание $\mathcal{P}_{\triangle a}\alpha$ является формулой из $StateFm$.

- $PathFm$:

1. Если f – формула логики PCTL, то то знакосочетание $\mathbf{X}f$ является формулой из $PathFm$.
2. Если φ_1 и φ_2 – формулы из $StateFm$, то следующие знакосочетания являются формулами из $PathFm$:
   (a) $\varphi_1\mathbf{U}^{\leq n}\varphi_2$, где n – натуральное число
   (b) $\varphi_1\mathbf{U}\varphi_2$
3. Если α – формула из $PathFm$, то знакосочетание $\neg\alpha$ является формулой из $PathFm$.

В записи формул из $PathFm$ могут использоваться символы **F** и **G**, которые являются сокращением знакосочетаний $\top\mathbf{U}$ и $\neg\mathbf{F}\neg$ соответственно (т.е., например, знакосочетания $\mathbf{F}\alpha$ и $\mathbf{G}^{\leq n}\alpha$ обозначают формулы $\top\mathbf{U}\alpha$ и $\neg\mathbf{F}^{\leq n}\neg\alpha$ соответственно).

4.3 Значения формул логики PCTL в состояниях вероятностных систем переходов

Пусть $D = (S, s^0, P, L)$ – некоторая ВСП.

Для каждого состояния $s \in S$ и каждой формулы f логики PCTL определено **значение** формулы f в состоянии s, которое обозначается записью $s(f)$, и

1. если $f \in StateFm$, то $s(f) \in \{0, 1\}$,

   и

   - в случае $s(f) = 1$ формула f считается истинной в s,
   - в случае $s(f) = 0$ формула f считается ложной в s

2. если $f \in PathFm$, то значение $s(f)$ является числом из $[0, 1]$ и интерпретируется как вероятность того, что формула f истинна в состоянии s.

Для каждой формулы f логики PCTL мы будем обозначать записью $S(f)$ вектор-столбец

$$\begin{pmatrix} s_1(f) \\ \dots \\ s_n(f) \end{pmatrix}$$

Значения формул логики PCTL в состояниях ВСП определяются индукцией по структуре формул в соответствии с излагаемыми ниже правилами. В одних из этих правил мы определяем значение $s(f)$, в других – определяем вектор-столбец $S(f)$ целиком. В этих определениях мы будем использовать следующие обозначения.

- Для любых векторов $U = \begin{pmatrix} u_1 \\ \dots \\ u_n \end{pmatrix}$, $V = \begin{pmatrix} v_1 \\ \dots \\ v_n \end{pmatrix}$ из $[0, 1]^n$ записи $\max(U, V)$ и $U \circ V$ обозначают вектора

  $$\begin{pmatrix} \max(u_1, v_1) \\ \dots \\ \max(u_n, v_n) \end{pmatrix} \quad \text{и} \quad \begin{pmatrix} u_1 \cdot v_1 \\ \dots \\ u_n \cdot v_n \end{pmatrix}$$

  соответственно.

- Если A и B – матрицы порядков $n \times n$ и $n \times 1$ соответственно с компонентами из $[0, 1]$, то запись $[A^* \cdot B]$ обозначает матрицу, получаемую

  – заменой всех ненулевых компонентов A и B на 1, и

– вычислением $(\sum\limits_{i\geq 0} A^i)\cdot B$, где сложение понимается как дизъюнкция (т.е. сумма $\sum\limits_{i\geq 0} A^i$ является конечной)

Правила определения значений формул логики PCTL в состояниях ВСП имеют следующий вид.

- Для каждого $p \in AP \quad s(p) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{если } p \in L(s) \\ 0, & \text{иначе} \end{cases}$
- $s(\top) \stackrel{\text{def}}{=} 1, \;\; s(\bot) \stackrel{\text{def}}{=} 0.$
- $s(\neg f) \stackrel{\text{def}}{=} 1 - s(f), \;\; s(\varphi_1 \wedge \varphi_2) \stackrel{\text{def}}{=} s(\varphi_1)\cdot s(\varphi_2)$, и т.д. (т.е. значения формул коммутируют с булевыми операциями).
- $s(\mathcal{P}_{\triangle a}\alpha) \stackrel{\text{def}}{=} \triangle(s(\alpha), a).$
- $S(\mathbf{X}f) \stackrel{\text{def}}{=} P \cdot S(f).$
- Пусть $\alpha_n = \varphi_1 \mathbf{U}^{\leq n} \varphi_2$ (где $n \geq 0$). Тогда

$$S(\alpha_0) \stackrel{\text{def}}{=} S(\varphi_2)$$
$$\forall\, n > 0 \quad S(\alpha_n) \stackrel{\text{def}}{=} \max\left(S(\varphi_2),\; S(\varphi_1) \circ S(\mathbf{X}\alpha_{n-1})\right)$$

- Пусть $\alpha = \varphi_1 \mathbf{U} \varphi_2$. Тогда $S(\alpha)$ определяется системой линейных уравнений

$$S(\alpha) = \max\Big(S(\varphi_2),\; [P^* \cdot S(\varphi_2)] \circ S(\varphi_1) \circ (P \cdot S(\alpha))\Big)$$

4.4 Смысл формул логики PCTL

Формулы логики PCTL представляют собой утверждения, выражающие различные свойства ВСП. Эти свойства могут выражать, например, динамические аспекты поведения ВСП, т.е. описывать зависимость истинности какого-либо утверждения в некотором состоянии s рассматриваемой ВСП от истинности другого утверждения в состояниях, достижимых из s.

Смысл формул состояний логики PCTL непосредственно усматривается из определения значений этих формул в состояниях.

Смысл формул путей логики PCTL можно описать следующим образом.

1. Формулу $\mathbf{X}\varphi$ можно интерпретировать как утверждение "в следующий момент времени будет верно φ", а значение формулы $\mathbf{X}\varphi$ в состоянии s – как вероятность того, что φ будет истинна в произвольном состоянии, в которое можно перейти из s за один такт времени.

2. Значение формулы $\varphi_1 \mathbf{U}^{\leq n} \varphi_2$ в состоянии s можно интерпретировать как вероятность того, что для произвольного пути π, выходящего из s, существует состояние $s' \in \pi$, такое, что длина начального отрезка $[s, s']$ пути π не превосходит n, и

   - в каждом состоянии этого отрезка, кроме, может быть, s', верна формула φ_1
   - в состоянии s' верна формула φ_2.

3. Значение формулы $\varphi_1 \mathbf{U} \varphi_2$ в состоянии s можно интерпретировать так же, как значение предыдущей формулы, без упоминания того, что длина $[s, s']$ не превосходит n.

4.5 Пример формулы логики PCTL, выражающей свойство вероятностной системы переходов

В этом пункте мы приведём пример формулы логики PCTL, выражающей одно из свойств первого протокола из пункта 2.3 (представленного ВСП (6)). Данное свойство заключается в том, что каждое сообщение, полученное отправителем от внешнего источника, будет с вероятностью ≥ 0.9 доставлено получателю не более чем за 5 шагов .

Для формального представления этого свойства мы будем предполагать, что среди атомарных утверждений имеются утверждения, комбинации значений которых соответствуют состояниям ВСП. Обозначим эти утверждения записями p_1 и p_2, и будем

считать, что значения p_1 и p_2 в состояниях ВСП (6) определяются следующим образом:

	p_1	p_2
s_0	0	0
s_1	0	1
s_2	1	0
s_3	1	1

в частности, состоянию s_0 соответствует формула $\neg p_1 \wedge \neg p_2$, а состоянию s_3 – формула $p_1 \wedge p_2$.

Тогда формула логики PCTL, соответствующая указанному выше свойству, имеет следующий вид:

$$\mathbf{G}((\neg p_1 \wedge \neg p_2) \rightarrow \mathcal{P}_{\geq 0.9}(\mathbf{F}^{\leq 5}(p_1 \wedge p_2))).$$

где символ $\geq$ в данной формуле обозначает функцию вида

$$\geq\colon [0,1] \times [0,1] \rightarrow \{0,1\}$$

которая сопоставляет паре $(a,b) \in [0,1] \times [0,1]$ элемент 1, если $a \geq b$, и 0 – иначе.

5 Метод редукции вероятностных систем переходов

5.1 Задача редукции вероятностных систем переходов

Вычисление значений формул логики PCTL в состояниях ВСП может иметь большую вычислительную сложность в том случае, когда анализируемая ВСП имеет большой размер. В связи с этим, представляет большую актуальность проблема редукции ВСП, т.е. удаления части состояний и переходов анализируемой ВСП, с таким расчетом, чтобы получившая ВСП была эквивалентна исходной в следующем смысле: для каждой формулы состояний f логики PCTL формула f истинна в начальном состоянии исходной ВСП тогда и только тогда, когда она истинная в начальном состоянии редуцированной ВСП.

Основная идея предлагаемого в настоящей работе метода редукции ВСП основана на понятии эквивалентности состояний ВСП: мы называем состояния эквивалентными, если значения всех формул логики PCTL в этих состояниях совпадают. Алгоритм редукции ВСП представляет собой вычисление классов эквивалентности состояний анализируемой ВСП и удалении эквивалентных состояний.

5.2 Эквивалентность вероятностных систем переходов

Пусть заданы две ВСП:

$$D_i = (S_i, s_i^0, P_i, L_i) \ \ (i = 1, 2) \tag{10}$$

Мы будем называть состояния $s_1 \in S_1$ и $s_2 \in S_2$ **эквивалентными**, если для каждой формулы f логики PCTL верно равенство

$$s_1(f) = s_2(f).$$

Если состояния s_1 и s_2 эквивалентны, то мы будем обозначать этот факт записью $s_1 \sim s_2$.

Мы будем называть ВСП D_1 и D_2 вида (10) **эквивалентными**, если $s_1^0 \sim s_2^0$. Если ВСП D_1 и D_2 эквивалентны, то мы будем обозначать этот факт записью $D_1 \sim D_2$.

Если ВСП D_1 и D_2 совпадают, и S – множество их состояний, то бинарное отношение на S, состоящее из всех пар (s_1, s_2), таких, что $s_1 \sim s_2$, является отношением эквивалентности. Мы будем обозначать это отношение символом $\sim$.

Отношение $\sim$ может быть найдено при помощи алгоритма, излагаемого в пункте (5.4). Для обоснования этого алгоритма мы введём необходимые понятия и докажем несколько вспомогательных утверждений.

5.3 Вспомогательные понятия и утверждения

5.3.1 Эквивалентности и связанные с ними разбиения множества состояний

Пусть заданы

- ВСП $D = (S, s^0, P, L)$, и
- некоторое множество Fm формул логики PCTL.

Мы будем обозначать записью $\rho(Fm)$ бинарное отношение на S, определяемое следующим образом:

$$\rho(Fm) \stackrel{\text{def}}{=} \{(s_1, s_2) \in S^2 \mid \forall\, f \in Fm \quad s_1(f) = s_2(f)\}.$$

Нетрудно видеть, что $\rho(Fm)$ – отношение эквивалентности, и

$$\rho(Fm) = \bigcap_{f \in Fm} \rho(\{f\}). \tag{11}$$

Если множество Fm конечно и имеет вид $\{f_1, \ldots, f_n\}$, то отношение $\rho(Fm)$ мы будем обозначать также записью $\rho(f_1, \ldots, f_n)$.

Лемма 1
Для каждого множества Fm формул логики PCTL существует конечное множество Fm' формул логики PCTL, такое, что

$$\rho(Fm) = \rho(Fm').$$

Доказательство.
Будем строить искомое множество Fm' следующим итеративным алгоритмом.

1. Сначала полагаем $Fm' \stackrel{\text{def}}{=} \{f\}$, где f – произвольная формула из Fm.
2. Если $\rho(Fm') = \rho(Fm)$, то алгоритм заканчивает свою работу, иначе –

   $$\exists\, f' \in Fm: \quad \rho(Fm') \neq \rho(Fm' \cup \{f'\}).$$

   Добавляем формулу f' к множеству Fm' и выполняем данный шаг ещё раз.

Отметим, что шаг 2 не может выполняться бесконечно, потому что разбиение, соответствующее отношению эквивалентности

$\rho(Fm' \cup \{f'\})$, является измельчением разбиения, соответствующего отношению эквивалентности $\rho(Fm')$, и поскольку множество S, на котором заданы эти отношения эквивалентности, является конечным, то таких измельчений не может быть более чем $|S|$. ■

Лемма 2

Для каждой формулы путей α существует конечное множество Fm формул состояний, такое, что $\rho(\alpha) = \rho(Fm)$.

Доказательство.

Обозначим записью $(a_1, \ldots, a_k)$ совокупность всех чисел вида $s(\alpha)$, где $s \in S$. Нетрудно видеть, что в качестве Fm можно взять множество формул вида

$$\varphi_i \stackrel{\text{def}}{=} \mathcal{P}_{\leq a_i}\alpha \quad (i = 1, \ldots, k-1). \quad \blacksquare$$

Лемма 3

Пусть задано множество Fm формул логики PCTL, и разбиение Σ множества S, соответствующее отношению эквивалентности $\rho(Fm)$, состоит из классов S_1, …, S_k. Тогда для каждого $j \in \{1, \ldots, k\}$ существует формула φ_j^Σ логики PCTL, такая, что

$$\begin{array}{l} \forall\, s \in S_j \quad s(\varphi_j^\Sigma) = 1 \\ \forall\, s \in S \setminus S_j \quad s(\varphi_j^\Sigma) = 0 \end{array} \tag{12}$$

Доказательство.

Из лемм 1 и 2 следует, что существует конечное множество φ_1, …, φ_n формул состояний, такое, что

$$\rho(Fm) = \rho(\varphi_1, \ldots, \varphi_n). \tag{13}$$

Обозначим записью φ_{ij} (где $i \in \{1, \ldots, n\}$ формулу, определяемую следующим образом:

$$\varphi_{ij} \stackrel{\text{def}}{=} \begin{cases} \varphi_i & \text{если } \forall\, s \in S_j \quad s(\varphi_i) = 1 \\ \neg\varphi_i & \text{иначе} \end{cases} \tag{14}$$

Искомая формула φ_j^Σ определяется следующим образом:

$$\varphi_j^\Sigma \stackrel{\text{def}}{=} \varphi_{1j} \wedge \ldots \wedge \varphi_{nj}. \tag{15}$$

Первое соотношение в (12) следует из того, что, согласно определению (14),

$$\forall\, i = 1, \ldots, n \quad \forall\, s \in S_j \quad s(\varphi_{ij}) = 1$$

и, следовательно,

$$\forall\, s \in S_j \quad s(\varphi_j^{\Sigma}) = s(\varphi_{1j} \wedge \ldots \wedge \varphi_{nj}) = 1.$$

Второе соотношение в (12) доказывается от противного: пусть для некоторого $s' \in S \setminus S_j$ верно равенство $s'(\varphi_j^{\Sigma}) = 1$, тогда из определения (15) следует, что

$$\forall\, i = 1, \ldots, n \quad s'(\varphi_{ij}) = 1. \tag{16}$$

Из (16) и (14) следует, что

$$\forall\, i = 1, \ldots, n \quad \forall\, s \in S_j \quad s'(\varphi_i) = s(\varphi_i). \tag{17}$$

Из (17) следует, что состояние s' эквивалентно состояниям из класса S_j относительно эквивалентности $\rho(\varphi_1, \ldots, \varphi_n)$, откуда, согласно (13) и определению классов $S_1, \ldots, S_k$, следует $s' \in S_j$, что противоречит предположению $s' \in S \setminus S_j$. ■

Ниже мы будем использовать следующие обозначения. Пусть задано разбиение Σ множества S, которое имеет вид $\{S_1, \ldots, S_k\}$.

- Мы будем обозначать символом π детерминированную СФ

$$\pi : S \underset{r}{\to} \Sigma \tag{18}$$

  (а также соответствующую матрицу), отображающую каждое состояние $s \in S$ в тот класс разбиения Σ, которому принадлежит s.

- Для каждого $j = 1, \ldots, k$ мы будем обозначать записью Σ_j столбец S_j вышеуказанной матрицы π

Нетрудно видеть, что в этих обозначениях соотношение (12) может быть записано более компактно в виде равенства

$$S(\varphi_j^{\Sigma}) = \Sigma_j.$$

5.3.2 Разбиения множества состояний, совместимые с функцией перехода

Пусть заданы ВСП $D = (S, s^0, P, L)$ и разбиение Σ множества S.

Мы будем говорить, что Σ **совместимо** с функцией перехода P, если существует СФ σ, делающая коммутативной диаграмму

$$\begin{array}{ccc} S & \xrightarrow{P} & S \\ {\scriptstyle\pi}\downarrow & & \downarrow{\scriptstyle\pi} \\ \Sigma & \xrightarrow{\sigma} & \Sigma \end{array} \qquad (19)$$

Лемма 4
Пусть заданы

- ВСП $D = (S, s^0, P, L)$, и
- разбиение $\Sigma = \{S_1, \ldots, S_k\}$ множества S.

Σ совместимо с функцией перехода P тогда и только тогда, когда для каждого $j = 1, \ldots, k$ вектор $P \cdot \Sigma_j$ является линейной комбинацией векторов $\Sigma_1, \ldots, \Sigma_k$.

Доказательство.
Если Σ совместимо с P, то, согласно (19) существует матрица σ, обладающая свойством

$$P \cdot \pi = \pi \cdot \sigma \qquad (20)$$

Пусть σ имеет вид $\begin{pmatrix} a_{11} \ldots a_{1k} \\ \ldots \\ a_{k1} \ldots a_{kk} \end{pmatrix}$. Нетрудно видеть, что для каждого $j = 1, \ldots, k$

- столбец S_j левой части (20) равен $P \cdot \Sigma_j$, и
- столбец S_j правой части (20) равен линейной комбинации

$$\Sigma_1 \cdot a_{1j} + \ldots + \Sigma_k \cdot a_{kj} \qquad (21)$$

Обратно, пусть для каждого $j = 1, \ldots, k$ вектор $P \cdot \Sigma_j$ представляется в виде линейной комбинации (21). Определим σ как матрицу коэффициентов этих линейных комбинаций. Нетрудно видеть, что σ удовлетворяет равенству (20).

Докажем, что σ соответствует СФ вида $\Sigma \underset{r}{\to} \Sigma$, т.е. верны утверждения

$$\forall\, i, j = 1, \ldots, k \quad a_{ij} \geq 0 \tag{22}$$

$$\forall\, i = 1, \ldots, k \quad \sum_{j=1}^{k} a_{ij} = 1. \tag{23}$$

- Докажем (22). Поскольку все компоненты матрицы P и вектора Σ_j неотрицательны, то, следовательно, все компоненты вектора $P \cdot \Sigma_j$ (который совпадает с (21)) тоже неотрицательны. Однако совокупность компонентов вектора (21) совпадает с множеством $\{a_{1j}, \ldots, a_{kj}\}$, т.к.
  - в каждой строке матрицы π один элемент равен 1, а все остальные элементы равны 0, и
  - $\forall\, j \in \{1, \ldots, k\}\ \exists\, s \in S$: элемент в строке s столбце S_j матрицы π равен 1.
- Утверждение (23) можно переформулировать в виде равенства $\sigma \cdot 1^{\downarrow} = 1^{\downarrow}$, где $1^{\downarrow}$ – вектор-столбец, все компоненты которого равны 1. Данное равенство следует из цепочки равенств

  $$\pi \cdot (\sigma \cdot 1^{\downarrow}) = (\pi \cdot \sigma) \cdot 1^{\downarrow} = (P \cdot \pi) \cdot 1^{\downarrow} = P \cdot (\pi \cdot 1^{\downarrow}) = P \cdot 1^{\downarrow} = 1^{\downarrow}$$

  и описанного в предыдущем пункте свойства матрицы π. ■

Ниже мы будем обозначать записью $PCTL$ совокупность всех формул логики PCTL.

Лемма 5

Для любой ВСП $D = (S, s^0, P, L)$ разбиение Σ множества S, соответствующее эквивалентности $\rho(PCTL)$, совместимо с функцией перехода P.

Доказательство.

Пусть Σ имеет вид $\{S_1, \ldots, S_k\}$, и Σ_1, ..., Σ_k – столбцы матрицы π, которая соответствует детерминированной СФ $\pi : S \underset{r}{\to} \Sigma$.

Согласно лемме 3, для каждого $j \in \{1, \ldots, k\}$ существует формула φ_j^{Σ} логики PCTL, обладающая свойством (12), т.е. такая, что

$$\forall\, j = 1, \ldots, k \quad S(\varphi_j^{\Sigma}) = \Sigma_j$$

Поскольку $\rho(PCTL) \subseteq \rho(\mathbf{X}\varphi_j^{\Sigma})$ то, следовательно, для каждого $j = 1, \ldots, k$ вектор $S(\mathbf{X}\varphi_j^{\Sigma}) = P \cdot S(\varphi_j^{\Sigma}) = P \cdot \Sigma_j$ имеет одинаковые компоненты, соответствующие состояниям из одного из того же класса разбиения Σ, т.е. вектор $P \cdot \Sigma_j$ является линейной комбинацией векторов Σ_1, ..., Σ_k. Согласно лемме 4, отсюда следует, что Σ совместимо с функцией перехода P. ■

Теорема 1

Пусть заданы

- ВСП $D = (S, s^0, P, L)$, и
- множество Fm формул логики PCTL, содержащее все атомарные утверждения из AP.

Тогда следующие утверждения эквивалентны:

(1) разбиение, соответствующее $\rho(Fm)$, совместимо с P,

(2) $\rho(Fm) = \rho(PCTL)$.

Доказательство.

Импликация (2) $\Rightarrow$ (1) следует из леммы 5.

Докажем импликацию (1) $\Rightarrow$ (2).

Поскольку $Fm \subseteq PCTL$, то $\rho(PCTL) \subseteq \rho(Fm)$.

Согласно (11), для доказательства обратного включения достаточно доказать, что для каждой формулы f логики PCTL верно включение

$$\rho(Fm) \subseteq \rho(f). \tag{24}$$

Докажем это включение индукцией по структуре формулы f.

Если $f \in AP$, то (24) следует из предположения $AP \subseteq Fm$.

Предположим, что для каждой подформулы f' формулы f верно включение (24), в котором f заменено на f'. Докажем, что в этом случае будет верно включение (24).

- Пусть f является булевой комбинацией. Рассмотрим, например, случай $f = \neg f'$. Если $(s_1, s_2) \in \rho(Fm)$, то

$$s_1(f) = s_1(\neg f') = 1 - s_1(f') = 1 - s_2(f') = s_2(\neg f') = s_2(f)$$

  Другие случаи возможного вида f, когда f является булевой комбинацией, рассматриваются аналогично.

- Пусть $f = \mathcal{P}_{\triangle a}\alpha$. Если $(s_1, s_2) \in \rho(Fm)$, то

$$s_1(f) = \triangle(s_1(\alpha), a) = \triangle(s_2(\alpha), a) = s_2(f).$$

Для рассмотрения остальных возможных вариантов вида f мы введём дополнительные обозначения. Обозначим символом Σ разбиение, соответствующее отношению эквивалентности $\rho(Fm)$. Пусть Σ состоит из классов $S_1, \ldots, S_k$. Заметим, что включение (24) равносильно следующему утверждению:

$$\forall\, j = 1, \ldots, k, \;\; \forall\, s_1, s_2 \in S_j \quad s_1(f) = s_2(f). \tag{25}$$

Нетрудно видеть, что (25) можно переформулировать следующим образом:

$$\exists\, \tilde{S}(f) : \; S(f) = \pi \cdot \tilde{S}(f) \tag{26}$$

где π – матрица, соответствующая СФ (18).

Докажем (26) для случая, когда f имеет вид $\mathbf{X}f'$, $\varphi_1 \mathbf{U}^{\leq n} \varphi_2$ и $\varphi_1 \mathbf{U} \varphi_2$.

- Пусть $f = \mathbf{X}f'$. По индуктивному предположению,

$$\exists\, \tilde{S}(f') : \; S(f') = \pi \cdot \tilde{S}(f')$$

  Следовательно,

$$S(\mathbf{X}f') = P \cdot S(f') = P \cdot \pi \cdot \tilde{S}(f') \tag{27}$$

По предположению, Σ совместимо с P, поэтому существует матрица σ, удовлетворяющая равенству (20). Поэтому цепочку равенств (27) можно продолжить следующим образом:

$$P \cdot \pi \cdot \tilde{S}(f') = \pi \cdot \sigma \cdot \tilde{S}(f') \tag{28}$$

Из (27) и (28) следует, что в качестве искомого вектор-столбца $\tilde{S}(f)$ в (26) можно взять вектор $\sigma \cdot \tilde{S}(f')$.

- Пусть $f = \varphi_1 \mathbf{U}^{\leq n} \varphi_2$. Обозначим эту формулу символом α_n.

  Докажем индукцией по n, что будет верно утверждение (25), в котором f заменено на α_n.

  Если $n = 0$, то, по определению, $S(\alpha_n) = S(\varphi_2)$, и доказываемое утверждение верно потому, что φ_2 – подформула формулы f.

  Если $n > 0$, то, по определению,

  $$S(\alpha_n) = \max\left(S(\varphi_2),\ S(\varphi_1) \circ S(\mathbf{X}\alpha_{n-1})\right)$$

  и истинность доказываемого утверждения следует из его истинности для φ_1, φ_2, α_{n-1} и $\mathbf{X}\alpha_{n-1}$, а также из определения операций max и $\circ$ на вектор-столбцах.

- Пусть $f = \varphi_1 \mathbf{U} \varphi_2$. Согласно определению, вектор-столбец $S(f)$ удовлетворяет соотношению

  $$\begin{array}{l} S(f) = \\ \max\Big(S(\varphi_2),\ [P^* \cdot S(\varphi_2)] \circ S(\varphi_1) \circ (P \cdot S(f))\Big) \end{array} \tag{29}$$

  Докажем утверждение (26) для данного случая.

  По индуктивному предположению, будет верно утверждение (26), в котором f заменено на φ_1 или на φ_2, т.е.

  $$\exists\, \tilde{S}(\varphi_i) :\ S(\varphi_i) = \pi \cdot \tilde{S}(\varphi_i) \quad (i = 1, 2)$$

  Определим вектор-столбец $\tilde{S}(\varphi)$ как решение системы линейных уравнений

  $$\begin{array}{l} \tilde{S}(f) = \\ \max\Big(\tilde{S}(\varphi_2),\ [\sigma^* \cdot \tilde{S}(\varphi_2)] \circ \tilde{S}(\varphi_1) \circ (\sigma \cdot \tilde{S}(f))\Big) \end{array} \tag{30}$$

где σ – матрица, соответствующая той СФ $\sigma : \Sigma \to \Sigma$, которая делает диаграмму (19) коммутативной.

Утверждение (26) следует из равенств (30), (20), и из следующих утверждений.

– Для любых вектор-столбцов V_1, V_2 длины k верны равенства

$$\pi \cdot \max(V_1, V_2) = \max(\pi \cdot V_1, \pi \cdot V_2)$$
$$\pi \cdot (V_1 \circ V_2) = (\pi \cdot V_1) \circ (\pi \cdot V_2)$$

Данные равенства непосредственно следуют из определения операций max и $\circ$ на вектор-столбцах.

– Для любого вектор-столбца V длины k с неотрицательными компонентами верно равенство

$$\pi \cdot [\sigma^* \cdot V] = [P^* \cdot \pi \cdot V]. \tag{31}$$

Данное равенство обосновывается следующим образом.

Из определения матрицы вида $[A^* \cdot B]$ в пункте 4.3 следует, что для каждого $s \in S$ следующие утверждения равносильны:

* элемент s столбца $[P^* \cdot \pi \cdot V]$ (являющегося правой частью равенства (31)) равен 1
* $\exists\, n \geq 0$: элемент s столбца $P^n \cdot \pi \cdot V$ больше 0.

Второе из этих утверждений равносильно следующему:

$$\exists\, n \geq 0 \;\; I_s \cdot P^n \cdot \pi \cdot V > 0 \tag{32}$$

где I_s – матрица, определённая в пункте 3.3.

Из (20) следует, что (32) можно переписать следующим образом:

$$\exists\, n \geq 0 \;\; I_s \cdot \pi \cdot \sigma^n \cdot V > 0 \tag{33}$$

С другой стороны, элемент s вектор-столбца $\pi \cdot [\sigma^* \cdot V]$ (являющегося левой частью равенства (31)) равен 1 тогда и только тогда, когда

$$I_s \cdot \pi \cdot [\sigma^* \cdot V] = 1. \tag{34}$$

Равенство (34) эквивалентно следующему утверждению: элемент S_j вектор-столбца $[\sigma^* \cdot V]$ равен 1, где S_j – тот класс разбиения Σ, которому принадлежит s.

Утверждение в предыдущем абзаце равносильно следующему: $\exists\, n \geq 0$: элемент S_j столбца $\sigma^n \cdot V$ больше 0, или:

$$\exists\, n \geq 0 : \quad I_{S_j} \cdot \sigma^n \cdot V > 0 \tag{35}$$

Поскольку $I_{S_j} = I_s \cdot \pi$, то (35) равносильно (33), что и требовалось доказать.

Таким образом, теорема 1 полностью доказана. ■

5.4 Редукция вероятностных систем переходов

5.4.1 Задача редукции ВСП

Пусть задана ВСП $D = (S, s^0, P, L)$.

Задача редукции ВСП D заключается в построении ВСП D', которая эквивалентна D, и число состояний которой меньше, чем число состояний ВСП D.

Излагаемый в настоящем параграфе алгоритм редукции ВСП является вероятностным обобщением алгоритма редукции детерминированных автоматов. Идея данного алгоритма основана на отождествлении неразличимых состояний ВСП:

- алгоритм вычисляет классы эквивалентности $S_1, \ldots, S_k$ множества S, соответствующие разбиению $\rho(PCTL)$, и
- ВСП D преобразуется путем удаления состояний в классах эквивалентности S_1, ..., S_k (и соответствующего переопределения функции перехода), до тех пор, пока не останется по одному состоянию в каждом из этих классов.

В результате этих удалений получается искомая ВСП D'.

5.4.2 Построение разбиения множества состояний редуцируемой ВСП

Разбиение множества S состояний ВСП $D = (S, s^0, P, L)$, соответствующее отношению эквивалентности $\rho(PCTL)$, вычисляется следующим образом:

1. Вычисляется разбиение Σ^0, соответствующее отношению эквивалентности $\rho(AP)$. Нетрудно видеть, что

$$\rho(AP) = \{(s_1, s_2) \in S \times S \mid L(s_1) = L(s_2)\}.$$

2. Затем работает цикл, состоящий из следующих шагов.

   Пусть для некоторого $i \geq 0$ определены

   - отношение эквивалентности ρ^i, которое имеет вид $\rho(Fm)$ для некоторого множества Fm формул логики PCTL, причем $AP \subseteq Fm$, и
   - соответствующее ему разбиение Σ^i, которое состоит из классов
   $$S_1^i, \ldots, S_k^i$$

   Обозначим записями

   - $\Sigma_1^i, \ldots, \Sigma_k^i$ – строки матрицы π^i, соответствующей детерминированной СФ $\pi^i : S \to \Sigma^i$, и
   - $\varphi_1^{\Sigma^i}, \ldots, \varphi_k^{\Sigma^i}$ – список формул, таких, что
   $$\forall\, j = 1, \ldots, k \quad S(\varphi_j^{\Sigma^i}) = \Sigma_j^i$$
   (существование таких формул следует из леммы 3)

   Определим отношение эквивалентности ρ^{i+1} на S:

$$\rho^{i+1} \stackrel{\text{def}}{=} \rho^i \cap \rho(\mathbf{X}\varphi_1^{\Sigma^i}, \ldots, \mathbf{X}\varphi_k^{\Sigma^i}). \tag{36}$$

   Нетрудно видеть, что если $\rho^i = \rho(Fm)$ для некоторого множества Fm формул логики PCTL, то

$$\rho^{i+1} = \rho(Fm \cup \{\mathbf{X}\varphi_1^{\Sigma^i}, \ldots, \mathbf{X}\varphi_k^{\Sigma^i}\}).$$

   Разбиение Σ^{i+1}, соответствующее отношению ρ^{i+1}, можно построить следующим образом:

- вычисляются вектор-столбцы

$$S(\mathbf{X}\varphi_j^{\Sigma^i}) = P \cdot \Sigma_j^i \tag{37}$$

(каждый из которых, как нетрудно видеть, является суммой некоторых столбцов матрицы P: для каждого $j = 1, \ldots, k$ вектор-столбец (37) является суммой таких столбцов s матрицы P, для которых $s \in S_j$)

- классы разбиения Σ^{i+1} получаются путём измельчения классов разбиения Σ^i: в один и тот же класс разбиения Σ^{i+1} попадают такие состояния, для которых соответствующие им компоненты векторов (37) совпадают для каждого $j = 1, \ldots, k$.

Возможны два случая.

(a) $\Sigma^{i+1} = \Sigma^i$.

В этом случае искомое разбиение $\rho(PCTL)$ найдено: оно совпадает с Σ^i.

Действительно, из равенства $\rho^{i+1} = \rho^i$ и из определения (36) следует включение

$$\rho^i \subseteq \rho(\mathbf{X}\varphi_1^{\Sigma^i}, \ldots, \mathbf{X}\varphi_k^{\Sigma^i}) \tag{38}$$

Лемма 6

Определённое выше разбиение Σ^i совместимо с функцией перехода P.

Доказательство.

Согласно лемме 4, для доказательства леммы достаточно доказать, что $\forall\, j = 1, \ldots, k$ вектор $P \cdot \Sigma_j^i$ является линейной комбинацией векторов $\Sigma_1^i, \ldots, \Sigma_k^i$.

Из (38) следует, что для каждого $j = 1, \ldots, k$ верно включение

$$\rho^i \subseteq \rho(\mathbf{X}\varphi_j^{\Sigma^i})$$

которое можно интерпретировать следующим образом: классы $S_1^i, \ldots, S_k^i$ разбиения Σ^i содержатся в классах разбиения, соответствующего вектору

$$S(\mathbf{X}\varphi_j^{\Sigma^i}) = P \cdot S(\varphi_j^{\Sigma^i}) = P \cdot \Sigma_j^i$$

т.е. компоненты вектора $P \cdot \Sigma^i_j$, соответствующие элементам из одного и того класса разбиения Σ^i, одинаковы. Это равносильно тому, что вектор $P \cdot \Sigma^i_j$ является линейной комбинацией векторов $\Sigma^i_1, \ldots, \Sigma^i_k$. ■

Поскольку $\rho^i = \rho(Fm)$ для некоторого множества Fm формул логики PCTL, причем $AP \subseteq Fm$, то на основании леммы 6 и теоремы 1 отсюда следует желаемое равенство

$$\rho^i = \rho(PCTL).$$

(b) $\Sigma^i \neq \Sigma^{i+1}$.

В этом случае мы увеличиваем i на 1 и возвращаемся в начало цикла (т.е. выполняем шаг 2 с увеличенным значением i).

Нетрудно видеть, что таких возвращений может быть не больше количества элементов множества S (т.к. разбиение Σ^{i+1} является измельчением разбиения Σ^i).

5.4.3 Удаление эквивалентных состояний из вероятностных систем переходов

Пусть ВСП $D = (S, s^0, P, L)$ содержит пару эквивалентных состояний s_1, s_2, где $s_1 \neq s^0$. Определим ВСП

$$D_1 \stackrel{\text{def}}{=} (S_1, s^0, P_1, L_1) \tag{39}$$

где

- $S_1 \stackrel{\text{def}}{=} S \setminus \{s_1\}$
- $\forall\, s, s' \in S_1 \quad P_1(s, s') \stackrel{\text{def}}{=} \begin{cases} P(s, s') + P(s, s_1) & \text{если } s' = s_2 \\ P(s, s') & \text{если } s' \neq s_2 \end{cases}$
- $\forall\, s \in S_1 \quad L_1(s) \stackrel{\text{def}}{=} L(s)$.

Таким образом,

- матрица P_1 получается из матрицы P

– прибавлением к столбцу s_2 столбца s_1, и

– удалением строки s_1 и столбца s_1

- матрица L_1 получается из матрицы L удалением строки s_1.

Ниже мы будем использовать следующие обозначения. Пусть A – матрица, соответствующая СФ вида $S \underset{r}{\rightarrow} S$. Для каждого $s \in S$ мы будем обозначать

- записью $A\backslash\vec{s}$ матрицу, получаемую из A удалением строки s, и
- записью $A\backslash s^{\downarrow}$ – матрицу, получаемую из A удалением столбца s.

Нетрудно видеть, что матрицы P_1 и L_1 связаны с матрицами P и L следующим образом:

$$\begin{array}{l} P_1 = (id_S\backslash\vec{s}_1) \cdot P \cdot id_S(s_1, s_2, 1) \cdot (id_S\backslash s_1^{\downarrow}) \\ L_1 = (id_S\backslash\vec{s}_1) \cdot L \end{array} \tag{40}$$

где $id_S(s_1, s_2, 1)$ – матрица, получаемая из матрицы id_S заменой в ней элемента в строке s_1 столбце s_2 на 1.

Мы будем говорить что ВСП (39) получается из ВСП D путем **удаления состояния s_1, эквивалентного состоянию s_2**. Согласно определению ВСП (39), каждое её состояние является также и состоянием ВСП D.

Для каждого $s \in S_1$ и каждой формулы f логики PCTL мы будем обозначать

- записями $s_D(f)$ и $s_{D_1}(f)$ значения формулы f в состоянии s в ВСП D и D_1 соответственно, и
- записями $S_D(f)$ и $S_{D_1}(f)$ – вектор-столбцы значений формулы f в состояниях ВСП D и D_1 соответственно.

Теорема 2

Пусть ВСП (39) получается из ВСП D путем удаления состояния s_1, эквивалентного состоянию s_2. Тогда

$$\forall\, s \in S_1 \quad s_{D_1}(f) = s_D(f). \tag{41}$$

Доказательство.

Отметим, что утверждение (41) эквивалентно равенству

$$S_{D_1}(f) = S_D(f) \setminus \vec{s}_1 \tag{42}$$

Доказательство того, что для произвольной формулы f логики PCTL верно (41) или (42), мы будем вести индукцией по структуре формулы f.

Пусть $f = p \in AP$. Поскольку $\forall\, s \in S_1 \quad L_1(s) = L(s)$, то

$$s_{D_1}(p) = 1 \quad \Leftrightarrow \quad p \in L_1(s) \quad \Leftrightarrow \quad p \in L_(s) \quad \Leftrightarrow \quad s_D(p) = 1$$

т.е. в данном случае (41) верно.

Предположим, что для каждой подформулы f' формулы f верно равенство (41) или (42), в котором f заменено на f'. Докажем, что в этом случае будет верно (41) или (42).

1. Пусть f является булевой комбинацией. Рассмотрим, например, случай $f = \neg f'$. Для каждого $s \in S_1$

   $$s_{D_1}(f) = s_{D_1}(\neg f') = 1 - s_{D_1}(f') = 1 - s_D(f') = s_D(\neg f') = s_D(f)$$

   Другие случаи возможного вида f, когда f является булевой комбинацией, рассматриваются аналогично.

2. Пусть $f = \mathcal{P}_{\triangle a}\alpha$. Тогда

   $$s_{D_1}(f) = \triangle(s_{D_1}(\alpha), a) = \triangle(s_D(\alpha), a) = s_D(f).$$

3. Пусть $f = \mathbf{X}f'$.

   Докажем, что верно равенство (42). Согласно определению матрицы $id_S \backslash \vec{s}_1$, данное равенство равносильно равенству

   $$S_{D_1}(f) = (id_S \backslash \vec{s}_1) \cdot S_D(f) \tag{43}$$

   Мы предполагаем, что (43) будет верно, если f заменить на f', т.е.

   $$S_{D_1}(f') = (id_S \backslash \vec{s}_1) \cdot S_D(f') \tag{44}$$

Согласно определению значений формулы $\mathbf{X}f'$, верны равенства

$$S_{D_1}(f) = S_{D_1}(\mathbf{X}f') = P_1 \cdot S_{D_1}(f')$$
$$S_D(f) = S_D(\mathbf{X}f') = P \cdot S_D(f')$$

Учитывая (40) и индуктивное предположение (44), можно переписать доказываемое равенство (43) следующим образом:

$$(id_S \backslash \vec{s}_1) \cdot P \cdot id_S(s_1, s_2, 1) \cdot (id_S \backslash s_1^{\downarrow}) \cdot (id_S \backslash \vec{s}_1) \cdot S_D(f') = \\ = (id_S \backslash \vec{s}_1) \cdot P \cdot S_D(f') \tag{45}$$

Нетрудно проверить, что

- произведение матриц $(id_S \backslash s_1^{\downarrow}) \cdot (id_S \backslash \vec{s}_1)$ совпадает с матрицей $id_S(s_1, s_1, 0)$, получаемой из id_S заменой элемента в строке s_1 столбце s_1 на 0
- произведение $id_S(s_1, s_2, 1) \cdot id_S(s_1, s_1, 0)$ совпадает с матрицей $id_S \begin{pmatrix} s_1, s_2, 1 \\ s_1, s_1, 0 \end{pmatrix}$, получаемой из id_S заменой
  - элемента в строке s_1 столбце s_2 на 1, и
  - элемента в строке s_1 столбце s_1 на 0
- произведение

$$id_S \begin{pmatrix} s_1, s_2, 1 \\ s_1, s_1, 0 \end{pmatrix} \cdot S_D(f') \tag{46}$$

  совпадает с вектор-столбцом, получаемым из $S_D(f')$ заменой элемента в строке s_1 на элемент $s_2(f')$.

Поскольку состояния s_1 и s_2 по предположению эквивалентны, то, по определению понятия эквивалентности состояний, верно равенство $s_1(f') = s_2(f')$. Следовательно, (46) совпадает с $S_D(f')$.

Из вышеприведённых утверждений следует, что левая часть доказываемого равенства (45) совпадает с его правой частью, что и требовалось доказать.

4. Пусть $f = \varphi_1 \mathbf{U}^{\leq n} \varphi_2$. Обозначим эту формулу символом α_n. Докажем индукцией по n, что $S_{D_1}(\alpha_n) = S_D(\alpha_n) \setminus \vec{s}_1$.

   Если $n = 0$, то, по определению, $S(\alpha_n) = S(\varphi_2)$, и доказываемое утверждение верно потому, что φ_2 – подформула формулы f.

   Если $n > 0$, то, по определению,

$$S(\alpha_n) = \max\left(S(\varphi_2),\ S(\varphi_1) \circ S(\mathbf{X}\alpha_{n-1})\right)$$

   и истинность доказываемого утверждения следует из его истинности для φ_1, φ_2, α_{n-1} и $\mathbf{X}\alpha_{n-1}$, а также из определения операций max и $\circ$ на вектор-столбцах.

5. Пусть $f = \varphi_1 \mathbf{U} \varphi_2$. Согласно определению, вектор-столбцы $S_D(f)$ и $S_{D_1}(f)$ удовлетворяют соотношениям

$$\begin{array}{l} S_D(f) = \\ \max\Big(S_D(\varphi_2),\ [P^* \cdot S_D(\varphi_2)] \circ S_D(\varphi_1) \circ (P \cdot S_D(f))\Big) \end{array} \tag{47}$$

$$\begin{array}{l} S_{D_1}(f) = \\ \max\Big(S_{D_1}(\varphi_2),\ [(P_1)^* \cdot S_{D_1}(\varphi_2)] \circ S_{D_1}(\varphi_1) \circ (P_1 \cdot S_{D_1}(f))\Big) \end{array} \tag{48}$$

   Для доказательства равенства (42) мы докажем следующее утверждение: замена в соотношении (48) компонентов вектора $S_{D_1}(f)$ на соответствующие компоненты вектора $S_D(f) \setminus \vec{s}_1$ не нарушает истинности этого соотношения.

   Утверждение в предыдущем абзаце следует из (47), определения операций max и $\circ$ на вектор-столбцах, и из следующих соотношений:

   - равенств

$$\begin{array}{c} S_{D_1}(\varphi_1) = S_D(\varphi_1) \setminus \vec{s}_1 \\ S_{D_1}(\varphi_2) = S_D(\varphi_2) \setminus \vec{s}_1 \end{array} \tag{49}$$

     которые верны по индуктивному предположению

   - равенств

$$[P^* \cdot S_D(\varphi_2)] \setminus \vec{s}_1 = [(P_1)^* \cdot S_{D_1}(\varphi_2)] \tag{50}$$

$$(P \cdot S_D(f)) \setminus \vec{s}_1 = P_1 \cdot (S_D(f) \setminus \vec{s}_1) \tag{51}$$

Из второго равенства в (49) следует, что (50) можно заменить на равенство

$$[P^* \cdot S_D(\varphi_2)] \setminus \vec{s}_1 = [(P_1)^* \cdot (S_D(\varphi_2) \setminus \vec{s}_1)] \qquad (52)$$

Истинность равенства (51) доказывается рассуждениями, аналогичными тем, которые были приведены в пункте 3 настоящего доказательства.

Докажем (52). Для каждого $n \geq 0$ верно равенство

$$(P^n \cdot S_D(\varphi_2)) \setminus \vec{s}_1 = P_1^n \cdot (S_D(\varphi_2) \setminus \vec{s}_1) \qquad (53)$$

которое нетрудно доказать индукцией по n.

Для каждого $s \in S_1$ элемент s столбца $[P^* \cdot S_D(\varphi_2)] \setminus \vec{s}_1$ равен 1 тогда и только тогда, когда $\exists\, n \geq 0$: элемент s столбца $(P^n \cdot S_D(\varphi_2)) \setminus \vec{s}_1$ отличен от 0. Согласно (53), это эквивалентно тому, что $\exists\, n \geq 0$ элемент s столбца $P_1^n \cdot (S_D(\varphi_2) \setminus \vec{s}_1)$ отличен от 0, а это эквивалентно тому, что элемент s столбца $[P_1^* \cdot (S_D(\varphi_2) \setminus \vec{s}_1)]$ равен 1. Это доказывает равенство (52).

Таким образом, теорема 2 полностью доказана. ■

5.4.4 Описание алгоритма редукции ВСП

Теорема 2 является обоснованием излагаемого ниже алгоритма редукции ВСП $D = (S, s^0, P, L)$. Этот алгоритм имеет следующий вид.

1. Вычисляется разбиение множества состояний ВСП D, соответствующее отношению эквивалентности $R \stackrel{\text{def}}{=} \rho(PCTL)$ (для этого выполняются действия, изложенные в пункте 5.4.2).

2. Искомая ВСП D' строится путём удаления состояний из ВСП D и переопределения функции перехода и отношения R следующим образом.

   (a) Если отношение R содержит пару (s_1, s_2), такую, что $s_1 \neq s_2$, и $s_2 \neq s^0$, то выберем произвольную такую

пару (s_1, s_2), и преобразуем компоненты ВСП D описываемым ниже образом. Мы будем излагать данное преобразование в терминах графа, соответствующего ВСП D (данный граф мы будем обозначать тем же символом D).

i. Если граф D содержит ребро с началом в некоторой вершине s и с концом s_2, то данное ребро удаляется, а к метке ребра с началом в s и с концом в s_1 прибавляется число, равное метке удалённого ребра. Данная операция выполняется до тех пор, пока имеются рёбра с концом в s_2.

ii. Вершина s_2 удаляется, и кроме того удаляются все рёбра, выходящие из этой вершины.

iii. Из R удаляются все пары, содержащие s_2.

iv. После этого мы переходим на шаг 2a.

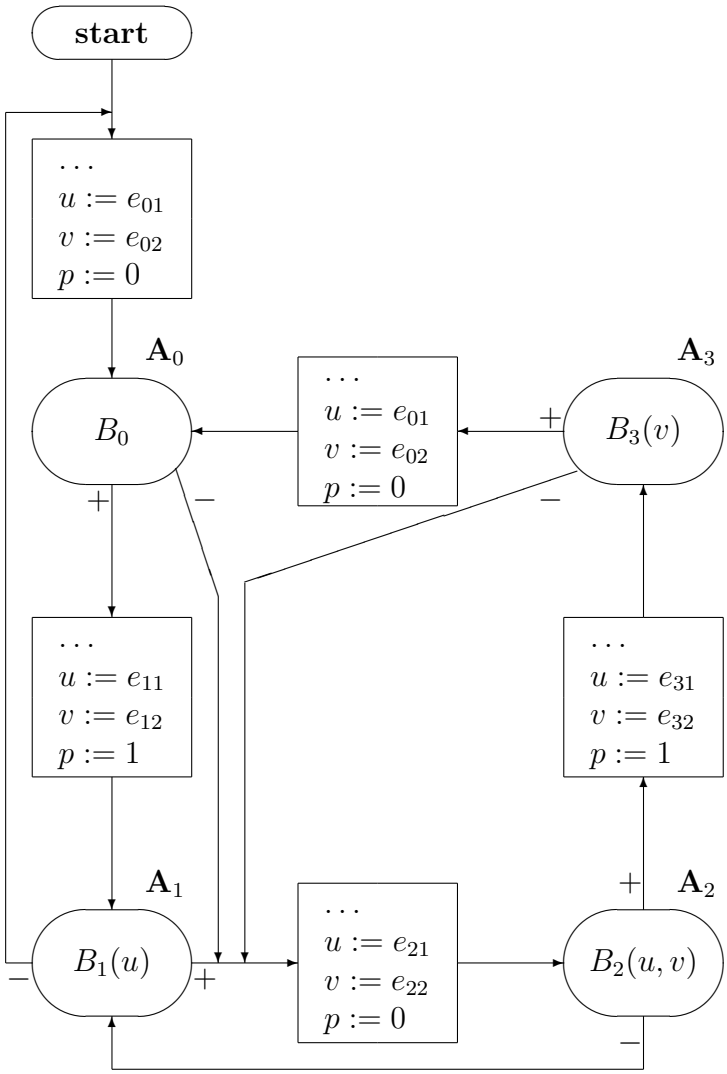
(b) Если каждая пара, входящая в R, имеет вид (s, s), то алгоритм завершает работу.

6 Пример редукции вероятностной системы переходов

В этом параграфе мы рассмотрим пример использования предложенного в настоящей работе алгоритма редукции ВСП.

6.1 Описание редуцируемой ВСП

ВСП, рассматриваемая в этом примере, является вероятностной абстракцией последовательной незавершающейся программы, представляемой следующей блок-схемой:



(54)

где символы $\mathbf{A}_0$, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$, нарисованные рядом с условными операторами, можно рассматривать как имена данных операторов.

Программа содержит

- булевы переменные p, u, v (которые мы будем рассматривать как существенные переменные данной блок-схемы), и

- некоторые другие переменные, не указанные на данной блок-схеме (которые мы будем рассматривать как несущественные переменные данной блок-схемы).

В условных операторах данной блок-схемы используются булевозначные выражения B_0 и т.д., в каждом из которых в скобках указаны лишь существенные переменные, от которых зависит данное булевозначное выражение.

Значения переменных p, u и v изменяются операторами присваивания, в правой части которых стоит терм, зависящий от u, v и других переменных блок-схемы. Операторы присваивания, которые изменяют значения переменных, отличных от p, u и v, не принимаются во внимание и изображены многоточием.

Предположим, что анализируется какое-либо частное свойство данной блок-схемы, связанное со статистическими закономерностями изменения значения переменной p. Для анализа свойства данного типа мы построим ВСП, являющуюйся вероятностной абстракцией данной блок-схемы.

Процесс функционирования данной блок-схемы можно интерпретировать как процесс перехода от одного условного оператора $\mathbf{A}_i$ к другому оператору $\mathbf{A}_j$, с выполнением операторов присваивания, которые встречаются на пути от $\mathbf{A}_i$ к $\mathbf{A}_j$, т.е.

- при переходе от $\mathbf{A}_0$ к $\mathbf{A}_1$ выполняются операторы присваивания $\ldots$, $u := e_{11}$, $v := e_{12}$, $p := 1$,
- при переходе от $\mathbf{A}_0$ к $\mathbf{A}_2$ выполняются операторы присваивания $\ldots$, $u := e_{21}$, $v := e_{22}$, $p := 0$,
- и т.д.

Напомним (см. [5]), что полная диаграмма состояний, соответствующая блок-схеме, представляет собой граф,

- вершины которого соответствуют состояниям блок-схемы, где под состоянием понимается набор значений, сопоставленных
  - переменным блок-схемы, а также

– управляющей переменной, которая не входит в число переменных блок-схемы, и значениями которой являются операторы блок-схемы.

Построим упрощённую диаграмму состояний данной блок-схемы, соответствующую такому уровню абстракции, при котором

- из переменных блок-схемы мы будем учитывать лишь переменные p, u и v, и
- множество значений управляющей переменной будем считать равным $\{\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3\}$ (принимая во внимание высказанное выше замечание о том, что каждый шаг функционирования блок-схемы можно интерпретировать как переход от оператора вида $\mathbf{A}_i$ к оператору вида $\mathbf{A}_j$, с выполнением присваиваний, находящихся на пути от $\mathbf{A}_i$ к $\mathbf{A}_j$).

Таким образом, при выбранном уровне абстракции диаграмма состояний должна содержать $2 \cdot 2 \cdot 2 \cdot 4$ состояний. Однако, из визуального анализа блок-схемы мы можем заключить, что значение переменной p однозначно определяется значением управляющей переменной (в операторах $\mathbf{A}_0$ и $\mathbf{A}_2$ значение p равно 0, а в операторах $\mathbf{A}_1$ и $\mathbf{A}_3$ значение p равно 1), и, следовательно, количество состояний может быть сокращено до $2 \cdot 2 \cdot 4$ состояний. Мы будем обозначать эти состояния записями вида $\mathbf{A}_i(j, k)$, где j и k – значения переменных u и v в состоянии $\mathbf{A}_i(j, k)$.

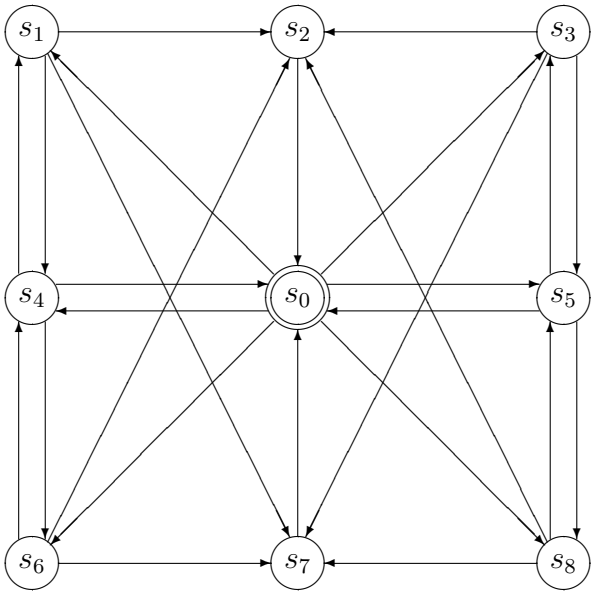
Из визуального анализа блок-схемы видно, что

- функционирование блок-схемы в тот момент, когда значение её управляющей переменной равно $\mathbf{A}_0$, не зависит от значений переменных u и v, поэтому все состояния вида $\mathbf{A}_0(j, k)$ можно рассматривать как одинаковые, мы будем обозначать все такие состояния одним символом $\mathbf{A}_0$,
- функционирование блок-схемы в тот момент, когда значение её управляющей переменной равно $\mathbf{A}_1$, зависит только от значения переменной u, поэтому состояния вида $\mathbf{A}_1(j, k)$ с одинаковыми первыми компонентами в скобках можно рассматривать как одинаковые, мы будем обозначать эти состояния записями $\mathbf{A}_1(0)$ и $\mathbf{A}_1(1)$ соответственно, и

- функционирование блок-схемы в тот момент, когда значение её управляющей переменной равно $\mathbf{A}_3$, зависит только от значения переменной , поэтому состояния вида $\mathbf{A}_3(j,k)$ с одинаковыми вторыми компонентами в скобках можно рассматривать как одинаковые, мы будем обозначать эти состояния записями $\mathbf{A}_3(0)$ и $\mathbf{A}_3(1)$ соответственно.

Таким образом, после всех вышеперечисленных отождествлений осталось 9 состояний: $\mathbf{A}_0$, $\mathbf{A}_1(0)$, $\mathbf{A}_1(1)$, $\mathbf{A}_2(0,0)$, $\mathbf{A}_2(0,1)$, $\mathbf{A}_2(1,0)$, $\mathbf{A}_2(1,1)$, $\mathbf{A}_3(0)$, $\mathbf{A}_3(1)$. Мы будем обозначать их более коротко символами s_0, s_4, s_5, s_1, s_6, s_3, s_8, s_7, s_2.

Путем дополнительного анализа данной блок-схемы можно установить, что граф переходов, соответствующий данному множеству состояний, имеет следующий вид:

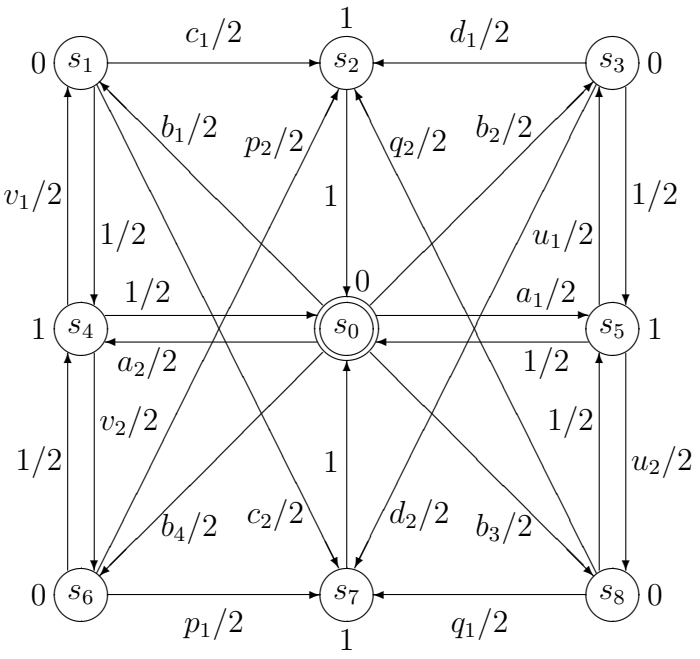


(55)

Далее, путём проведения тестовых испытаний с блок-схемой мы определяем статистические закономерности: для каждого перехода $s_i \to s_j$ в данной диаграмме мы определяем вероятность того, что если если в текущий момент времени блок-схема находится в состоянии s_i, то в следующий момент времени она будет находиться в состоянии s_j.

Искомая ВСП представляет собой граф (55), с каждым ребром которого связана соответствующая вероятность (полученная экспериментальным образом).

Предположим, что в результате проведения таких испытаний была получена следующая ВСП:



(56)

где в обозначениях вероятностей переходов мы используем буквенные обозначения, которые имеют следующий смысл: символы a_1, a_2 и т.д. обозначают неотрицательные действительные числа, удовлетворяющие следующим ограничениям:

$$\begin{aligned} &a_1 + a_2 = 1 \\ &b_1 + b_2 + b_3 + b_4 = 1 \\ &c_1 + c_2 = 1 \\ &d_1 + d_2 = 1 \\ &p_1 + p_2 = 1 \\ &q_1 + q_2 = 1 \\ &u_1 + u_2 = 1 \\ &v_1 + v_2 = 1 \end{aligned} \tag{57}$$

В этой ВСП множество AP атомарных утверждений состоит из одного утверждения p, т.е. множество 2^{AP} состоит из двух эле-

ментов: $\emptyset$ и $\{p\}$. Мы будем обозначать эти элементы символами 0 и 1 соответственно.

Начальным состоянием данной ВСП является состояние s_0, оно обозначено двойным кружком. Рядом с каждым состоянием s приписана метка, которая равна значению $L(s)$.

Матрицу P, соответствующую данной ВСП мы представим в виде следующей таблицы:

	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
s_0	0	$b_1/2$	0	$b_2/2$	$a_2/2$	$a_1/2$	$b_4/2$	0	$b_3/2$
s_1	0	0	$c_1/2$	0	$1/2$	0	0	$c_2/2$	0
s_2	1	0	0	0	0	0	0	0	0
s_3	0	0	$d_1/2$	0	0	$1/2$	0	$d_2/2$	0
s_4	$1/2$	$v_1/2$	0	0	0	0	$v_2/2$	0	0
s_5	$1/2$	0	0	$u_1/2$	0	0	0	0	$u_2/2$
s_6	0	0	$p_2/2$	0	$1/2$	0	0	$p_1/2$	0
s_7	1	0	0	0	0	0	0	0	0
s_8	0	0	$q_2/2$	0	0	$1/2$	0	$q_1/2$	0

Левый столбец и верхняя строка в этой таблице не являются элементами матрицы P, они предназначены для интерпретации коэффициентов в этой таблице: для каждой пары состояний $(s_i, s_j) \in S \times S$ значение $P(s_i, s_j)$ располагается в пересечении строки, содержащей символ s_i, и столбца, содержащего символ s_j.

6.2 Вычисление эквивалентности $\rho(PCTL)$ для редуцируемой ВСП

Вычисление эквивалентности $\rho(PCTL)$ для ВСП (55) происходит следующим образом.

1. Вычисляется отношение эквивалентности ρ^0, которое состоит из всех пар $(s_1, s_2) \in S \times S$, удовлетворяющих равенству $L(s_1) = L(s_2)$.

   Нетрудно видеть, что Σ^0 состоит из двух классов

   $$\{s_0, s_1, s_3, s_6, s_8\}, \quad \{s_2, s_4, s_5, s_7\} \tag{58}$$

2. Матрица π^0, соответствующая детерминированной СФ

$$\pi^0 : S \to \Sigma^0$$

имеет вид

s_0	1	0
s_1	1	0
s_2	0	1
s_3	1	0
s_4	0	1
s_5	0	1
s_6	1	0
s_7	0	1
s_8	1	0

Затем вычисляется матрица $P \cdot \pi^0$. Данная матрица будет иметь следующий вид:

0	$b_1/2 + b_2/2 + b_4/2 + b_3/2$	$a_2/2 + a_1/2$
1	0	$c_1/2 + 1/2 + c_2/2$
2	1	0
3	0	$d_/2 + 1/2 + d_2/2$
4	$1/2 + v_1/2 + v_2/2$	0
5	$1/2 + u_1/2 + u_2/2$	0
6	0	$p_2/2 + 1/2 + p_1/2$
7	1	0
8	0	$q_2/2 + 1/2 + q_1/2$

Принимая во внимание ограничения (57), данную матрицу

можно упростить до следующего вида:

0	1/2	1/2
1	0	1
2	1	0
3	0	1
4	1	0
5	1	0
6	0	1
7	1	0
8	0	1

(59)

По матрице (59) нетрудно вычислить отношение ρ^1 и соответствующее ему разбиение Σ^1. Из определения отношения ρ^1 непосредственно следует, что состояния s_1 и s_2 находятся в одном и том же классе разбиения Σ^1 тогда и только тогда, когда они оба находятся в одном и том же классе из списка (58), и кроме того строки матрицы (59), соответствующие состояниям s_1 и s_2, совпадают.

Разбиение Σ^1 будет состоять из трех классов (измельчится первый класс в (58), а второй класс останется тем же), эти классы имеют следующий вид:

$$\{s_0\}, \quad \{s_1, s_3, s_6, s_8\}, \quad \{s_2, s_4, s_5, s_7\} \tag{60}$$

3. Затем вычисляется матрица π^1, соответствующая детерминированной СФ

$$\pi^1 : S \underset{r}{\to} \Sigma^1$$

Данная матрица будет иметь следующий вид:

0	1	0	0
1	0	1	0
2	0	0	1
3	0	1	0
4	0	0	1
5	0	0	1
6	0	1	0
7	0	0	1
8	0	1	0

Произведение $P \cdot \pi_1$ имеет следующий вид:

0	0	$b_1/2 + b_2/2 + b_4/2 + b_3/2$	$a_2/2 + a_1/2$
1	0	0	$c_1/2 + 1/2 + c_2/2$
2	1	0	0
3	0	0	$d_1/2 + 1/2 + d_2/2$
4	$1/2$	$v_1/2 + v_2/2$	0
5	$1/2$	$u_1/2 + u_2/2$	0
6	0	0	$p_2/2 + 1/2 + p_1/2$
7	1	0	0
8	0	0	$q_2/2 + 1/2 + q_1/2$

Принимая во внимание ограничения (57), данную матрицу

можно упростить до следующего вида:

0	0	1/2	1/2
1	0	0	1
2	1	0	0
3	0	0	1
4	1/2	1/2	0
5	1/2	1/2	0
6	0	0	1
7	1	0	0
8	0	0	1

После этого, действуя аналогичным образом, как и в предыдущем пункте, вычисляем классы разбиения Σ^2, соответствующего эквивалентности ρ^2. Таких классов будет четыре (измельчится третий класс в (60), а первый и второй классы останутся теми же), эти классы имеют следующий вид:

$$\{s_0\}, \quad \{s_1, s_3, s_6, s_8\}, \quad \{s_2, s_7\}, \quad \{s_4, s_5\} \tag{61}$$

4. Затем вычисляется матрица π^2, соответствующая детерминированной СФ

$$\pi^2 : S \underset{r}{\rightarrow} \Sigma^2$$

Данная матрица будет иметь следующий вид:

0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	1	0	0
4	0	0	0	1
5	0	0	0	1
6	0	1	0	0
7	0	0	1	0
8	0	1	0	0

Произведение $P \cdot \pi_2$ имеет следующий вид:

0	0	$b_1/2 + b_2/2 + b_4/2 + b_3/2$	0	$a_2/2 + a_1/2$
1	0	0	$c_1/2 + c_2/2$	$1/2$
2	1	0	0	0
3	0	0	$d_1/2 + d_2/2$	$1/2$
4	$1/2$	$v_1/2 + v_2/2$	0	0
5	$1/2$	$u_1/2 + u_2/2$	0	0
6	0	0	$p_2/2 + p_1/2$	$1/2$
7	1	0	0	0
8	0	0	$q_2/2 + q_1/2$	$1/2$

Принимая во внимание ограничения (57), данную матрицу можно упростить до следующего вида:

0	0	$1/2$	0	$1/2$
1	0	0	$1/2$	$1/2$
2	1	0	0	0
3	0	0	$1/2$	$1/2$
4	$1/2$	$1/2$	0	0
5	$1/2$	$1/2$	0	0
6	0	0	$1/2$	$1/2$
7	1	0	0	0
8	0	0	$1/2$	$1/2$

(62)

После этого, действуя аналогичным образом, как и в предыдущем пункте, вычисляем классы разбиения Σ^3, соответствующего эквивалентности ρ^3. Нетрудно проверить, что классы разбиения Σ^3 будут иметь точно такой же вид, что и классы эквивалентности разбиения Σ^2. Это означает, что искомое разбиение множества S на классы эквивалентных состояний построено, оно имеет вид (61).

6.3 Удаление избыточных состояний

Теперь можно приступить к удалению избыточных состояний (так, чтобы среди оставшихся состояний было ровно по одному

состоянию из каждого класса эквивалентности (61)). Нетрудно видеть, что можно удалить следующие состояния: s_3, s_6, s_8, s_7, s_5.

1. Удаление состояния s_3.

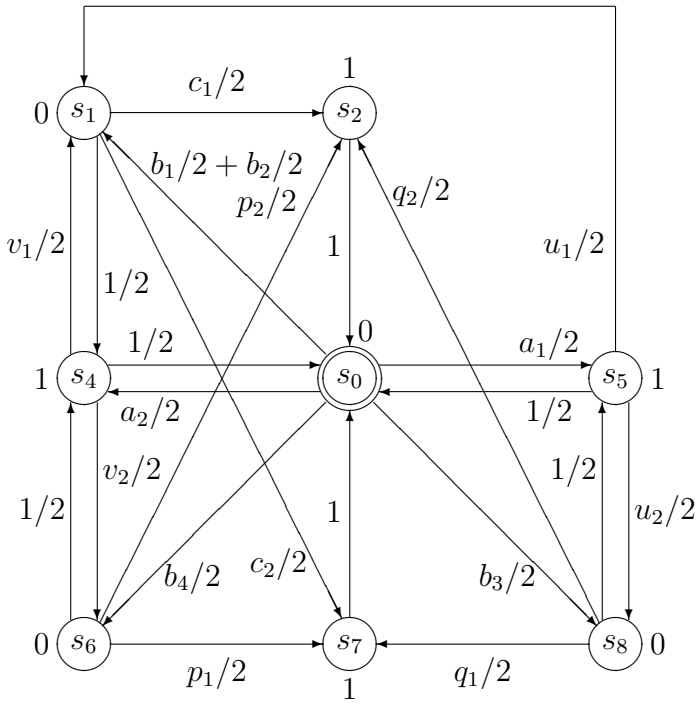
   Совокупность всех рёбер ВСП (56) с концом s_3 и ненулевыми метками имеет вид:

   - ребро из s_0 в s_3 с меткой $b_2/2$, и
   - ребро из s_5 в s_3 с меткой $u_1/2$.

   В соответствии с алгоритмом,

   - к меткам рёбер из s_0 в s_1 и из s_5 в s_1 мы прибавляем $b_2/2$ и $u_1/2$ соответственно, и
   - удаляем состояние s_3 и все связанные с ним рёбра.

   После удаления состояния s_3 и связанных с ним рёбер ВСП (56) примет следующий вид:



(63)

2. Удаление состояния s_6.

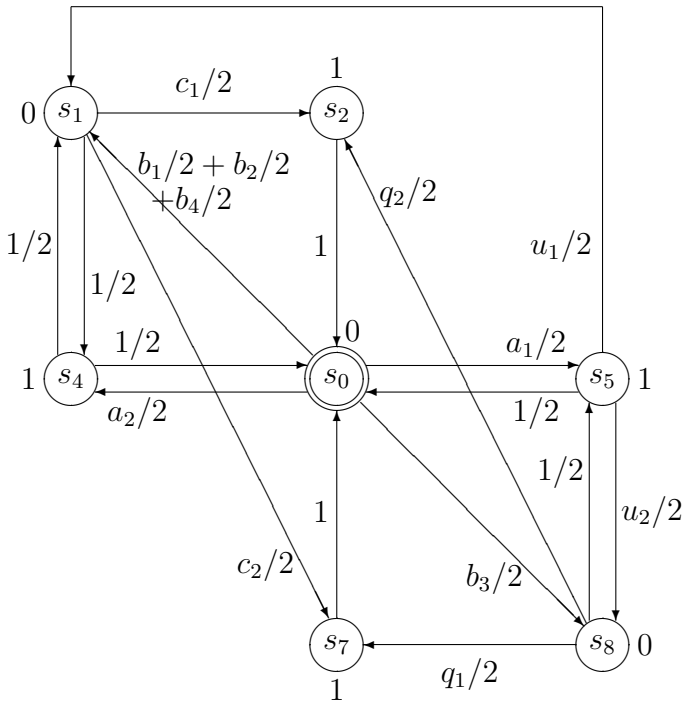
   Совокупность всех рёбер ВСП (63) с концом s_6 и ненулевыми метками имеет вид:

   - ребро из s_0 в s_6 с меткой $b_4/2$, и
   - ребро из s_4 в s_6 с меткой $v_2/2$.

   В соответствии с алгоритмом,

   - к меткам рёбер из s_0 в s_1 и из s_4 в s_1 мы прибавляем $b_4/2$ и $v_2/2$ соответственно, и
   - удаляем состояние s_6 и все связанные с ним рёбра.

   Принимая во внимание соотношение $v_1 + v_2 = 1$, получаем, что после удаления состояния s_6 и связанных с ним рёбер ВСП (63) примет следующий вид:



(64)

3. Удаление состояния s_8

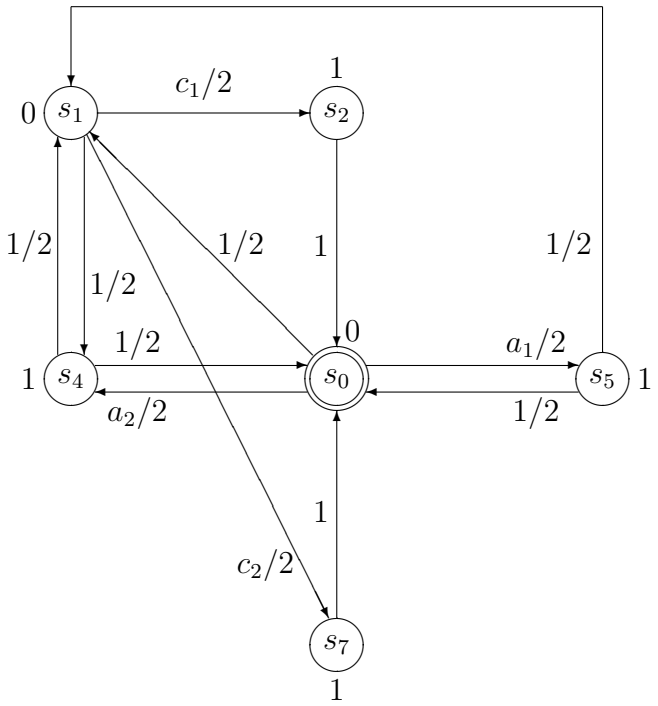
   Совокупность всех рёбер ВСП (64) с концом s_8 и ненулевыми метками имеет вид:

   - ребро из s_0 в s_8 с меткой $b_3/2$, и
   - ребро из s_5 в s_8 с меткой $u_2/2$.

   В соответствии с алгоритмом,

   - к меткам рёбер из s_0 в s_1 и из s_5 в s_1 мы прибавляем $b_3/2$ и $u_2/2$ соответственно, и
   - удаляем состояние s_8 и все связанные с ним рёбра.

   Принимая во внимание соотношения $u_1 + u_2 = 1$ и $b_1 + b_2 + b_3 + b_4 = 1$, получаем, что после удаления состояния s_8 и связанных с ним рёбер ВСП (64) примет следующий вид:



(65)

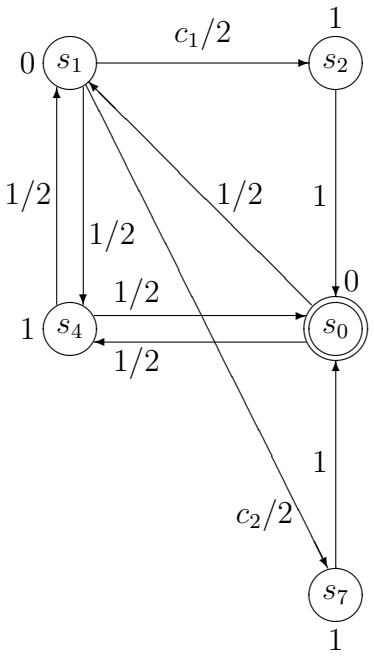
4. Удаление состояния s_5.

   Совокупность всех рёбер ВСП (65) с концом s_5 и ненулевыми метками состоит из одного ребра из s_0 в s_5 с меткой $a_1/2$.

   В соответствии с алгоритмом,

   - к метке ребра из s_0 в s_4 мы прибавляем $a_1/2$, и
   - удаляем состояние s_5 и все связанные с ним рёбра.

   Принимая во внимание соотношение $a_1 + a_2 = 1$, получаем, что после удаления состояния s_5 и связанных с ним рёбер ВСП (65) примет следующий вид:



(66)

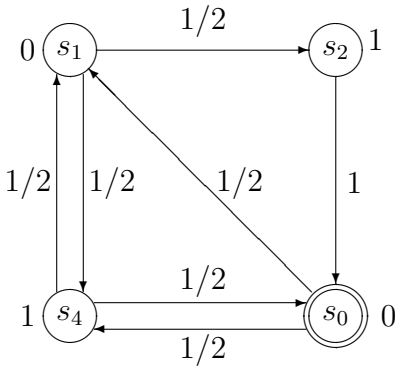
5. Удаление состояния s_7.

   Совокупность всех рёбер ВСП (66) с концом s_7 и ненулевыми метками состоит из одного ребра из s_1 в s_7 с меткой $c_2/2$.

   В соответствии с алгоритмом,

- к метке ребра из s_1 в s_2 мы прибавляем $c_2/2$, и
- удаляем состояние s_7 и все связанные с ним рёбра.

Принимая во внимание соотношение $c_1 + c_2 = 1$, получаем, что после удаления состояния s_7 и связанных с ним рёбер ВСП (66) примет следующий вид:



(67)

ВСП (67) далее нередуцируема: нетрудно видеть, что все её состояния неэквивалентны.

Таким образом, в результате редукции исходная ВСП (56) с девятью состояниями упростится до эквивалентной ей ВСП (67) с четырьмя состояниями.

7 Пример вычисления значений формулы логики PCTL в состояниях вероятностных систем переходов

Рассмотрим в качестве примера следующую формулу логики PCTL:

$$\mathcal{P}_{=1}\mathbf{F}^{\leq 2}p \tag{68}$$

Мы будем интерпретировать эту формулу следующим образом. Значение переменной p в каком-либо состоянии мы будем понимать как наличие некоторой неисправности в этом состоянии:

- если $s(p) = 1$, то мы будем считать, что в состоянии s моделируемой системы имеется неисправность, и
- если $s(p) = 0$, то мы считаем, что в состоянии s неисправность отсутствует.

При такой интерпретации формулу (68) можно интерпретировать как следующее утверждение: с вероятностью 1 из любого состояния будет достижимо за ≤ 2 шага состояние, в котором будет присутствовать неисправность.

Вычислим значения этой формулы в состояниях ВСП (56). Для этого мы должны вычислить значения всех подформул формулы (68) в состояниях ВСП (56).

Нетрудно видеть, что список подформул формулы (68) имеет следующий вид:

$$\begin{array}{l} p \\ \mathbf{F}^{\leq 2} p \\ \mathcal{P}_{=1} \mathbf{F}^{\leq 2} p \end{array} \tag{69}$$

Значения подформулы p в состояниях ВСП (56) имеют следующий вид:

	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
p	0	0	1	0	1	1	0	1	0

Для вычисления значений подформулы $\mathbf{F}^{\leq 2} p$ в состояниях ВСП (56) мы будем использовать следующий алгоритм вычисления значений формул вида $\mathbf{F}^{\leq n} \varphi$ (который непосредственно вытекает из определения значений формул вида $\varphi_1 \mathbf{U}^{\leq n} \varphi_2$): значения $a_{sn} \stackrel{\text{def}}{=} s(\mathbf{F}^{\leq n} \varphi)$ вычисляем следующим образом:

- $\forall\, i = 0, \ldots, n,\ \forall\, s \in S \quad a_{si} := s(\varphi)$
- $\forall\, i = 1, \ldots, n,\ \forall\, s \in S$ если $a_{si} = 0$, то

$$a_{si} := \sum_{s' \in S} P(s, s') \cdot a_{s', i-1}$$

Согласно этому алгоритму, значения a_{si}, соответствующие формуле $\mathbf{F}^{\leq 2}p$ и состояниям ВСП (56), имеют следующий вид:

	$i=0$	$i=1$	$i=2$
s_0	0	$b_1/2\cdot 0 + b_2/2\cdot 0 + +b_3/2\cdot 0 + b_4/2\cdot 0 + +a_1/2\cdot 1 + a_2/2\cdot 1 = = 1/2$	$b_1/2\cdot 1 + b_2/2\cdot 1 + +b_3/2\cdot 1 + b_4/2\cdot 1 + +a_1/2\cdot 1 + a_2/2\cdot 1 = 1$
s_1	0	$c_1/2\cdot 1 + c_2/2\cdot 1 + +1/2\cdot 1 = 1$	$c_1/2\cdot 1 + c_2/2\cdot 1 + +1/2\cdot 1 = 1$
s_2	1	1	1
s_3	0	$d_1/2\cdot 1 + d_2/2\cdot 1 + +1/2\cdot 1 = 1$	$d_1/2\cdot 1 + d_2/2\cdot 1 + +1/2\cdot 1 = 1$
s_4	1	1	1
s_5	1	1	1
s_6	0	$p_1/2\cdot 1 + p_2/2\cdot 1 + +1/2\cdot 1 = 1$	$p_1/2\cdot 1 + p_2/2\cdot 1 + +1/2\cdot 1 = 1$
s_7	1	1	1
s_8	0	$q_1/2\cdot 1 + q_2/2\cdot 1 + +1/2\cdot 1 = 1$	$q_1/2\cdot 1 + q_2/2\cdot 1 + +1/2\cdot 1 = 1$

Таким образом, значение подформулы $\mathbf{F}^{\leq 2}p$ во всех состояниях ВСП (56) равно 1, откуда следует, что значение формулы (68) в во всех состояниях ВСП (56) равно 1.

Теперь рассмотрим задачу вычисления значений той же самой формулы (68) в состояниях редуцированной ВСП (67). Так же, как и в предыдущем случае, мы вычисляем значения всех подформул формулы (68) в состояниях ВСП (67).

Значения подформулы p в состояниях ВСП (67) имеют следующий вид:

	s_0	s_1	s_2	s_4
p	0	0	1	1

Значения a_{si}, соответствующие формуле $\mathbf{F}^{\leq 2}p$ и состояниям ВСП (67), вычисляются по описанному выше алгоритму, и имеют

следующий вид:

	$i = 0$	$i = 1$	$i = 2$
s_0	0	$1/2 \cdot 0 + 1/2 \cdot 1 = 1/2$	$1/2 \cdot 1 + 1/2 \cdot 1 = 1$
s_1	0	$1/2 \cdot 1 + 1/2 \cdot 1 = 1$	$1/2 \cdot 1 + 1/2 \cdot 1 = 1$
s_2	1	1	1
s_4	1	1	1

Таким образом, как и в предыдущем случае, значение подформулы $\mathbf{F}^{\leq 2} p$ во всех состояниях ВСП (67) равно 1, откуда следует, что значение формулы (68) во всех состояниях ВСП (67) равно 1.

Отметим, что процедура вычисления значений формулы (68) в состояниях редуцированной ВСП (67) имеет существенно меньшую сложность, чем процедура вычисления значений этой формулы в состояниях исходной ВСП (56). Это свидетельствует о пользе предложенного в настоящей работе подхода редукции анализируемой ВСП перед вычислением значений формул логики PCTL в её состояниях.

8 Заключение

В настоящей работе изложен алгоритм редукции вероятностных систем переходов, идея которого заключается в удалении избыточных состояний. Отметим, что в результате такой редукции может получиться ВСП, которая хотя и не содержит различных эквивалентных состояний, но тем не менее может не являться минимальной по числу состояний среди всех ВСП, эквивалентных исходной ВСП. В связи с этим встает вопрос об алгоритме нахождения минимальной по числу состояний ВСП, эквивалентной заданной ВСП, и исследовании единственности такой минимальной ВСП (с точностью до подходящим образом сформулированного понятия изоморфизма).

Также представляет интерес исследование проблем минимизации других классов моделей, связанных с вероятностной верификацией, в частности, минимизации марковских решающих процессов.

Кроме этих задач, для вероятностной верификации актуальны все задачи, которые связаны с классической теорией Model Checking, в частности

- разработка методов верификации ВСП "на лету"(on-the-fly, см. [5]), т.е. таких методов, которые связаны с построением тех фрагментов ВСП, которых достаточно для проверки анализируемого свойства
- построение символьных методов вероятностной верификации (в том числе нахождение аналогов понятия бинарных разрешающих диаграмм [5] для ВСП).

Список литературы

[1] `http://qav.comlab.ox.ac.uk/`

[2] **Кемени Дж., Снелл Дж.** Конечные цепи Маркова. *Москва, Наука, 1970.*

[3] **Бухараев Р.Г.** Основы теории вероятностных автоматов. *Москва, Наука, 1985.*

[4] **Marta Kwiatkowska, David Parker.** Advances in Probabilistic Model Checking.
`http://qav.comlab.ox.ac.uk/papers/marktoberdorf11.pdf`

[5] **Э. М. Кларк, О. Грамберг, Д. Пелед.** Верификация моделей программ. Model Checking. *МЦНМО, 2002, 416 с.*

[6] **C. Baier, M. Groesser, and F. Ciesinski.** Partial order reduction for probabilistic systems. *In Proc. 1st International Conference on Quantitative Evaluation of Systems (QEST'04), pages 230-239. IEEE CS Press, 2004.*

[7] **P. D'Argenio and P. Niebert.** Partial order reduction on concurrent probabilistic programs. *In Proc. 1st International Conference on Quantitative Evaluation of Systems (QEST'04). IEEE CS Press, 2004.*

[8] **A. Donaldson and A. Miller.** Symmetry reduction for probabilistic model checking using generic representatives. *In S. Graf and W. Zhang, editors, Proc. 4th Int. Symp. Automated Technology for Verification and Analysis (ATVA'06), volume 4218 of Lecture Notes in Computer Science, pages 9-23. Springer, 2006.*

[9] **M. Kwiatkowska, G. Norman, and D. Parker.** Symmetry reduction for probabilistic model checking. *In T. Ball and R. Jones, editors, Proc. 18th International Conference on Computer Aided Verification (CAV'06), volume 4114 of LNCS, pages 234- 248. Springer, 2006.*

[10] **S. Hart, M. Sharir, A. Pnueli.** Termination of probabilistic concurrent programs. *ACM Transactions on Programming Languages and Systems, 5(3):356-380, 1983.*

[11] **M. Vardi.** Automatic verification of probabilistic concurrent finite state programs. *In Proc. 26th Annual Symposium on Foundations of Computer Science (FOCS'85), pages 327-338. IEEE Computer Society Press, 1985.*

[12] **C. Courcoubetis and M. Yannakakis.** Verifying temporal properties of finite state probabilistic programs. *In Proc. 29th Annual Symposium on Foundations of Computer Science (FOCS'88), pages 338-345. IEEE Computer Society Press, 1988.*

[13] **H. Hansson and B. Jonsson.** A logic for reasoning about time and reliability. *Formal Aspects of Computing, 6(5):512-535, 1994.*

[14] **A. Bianco and L. de Alfaro.** Model checking of probabilistic and nondeterministic systems. *In P. Thiagarajan, editor, Proc. 15th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'95), volume 1026 of LNCS, pages 499-513. Springer, 1995.*

[15] **C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen.** Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering, 29(6):524-541, 2003.*

[16] **H. Hansson.** Time and Probability in Formal Design of Distributed Systems. *Elsevier, 1994.*

[17] **C. Baier, E. Clarke, V. Hartonas-Garmhausen, M. Kwiatkowska, and M. Ryan.** Symbolic model checking for probabilistic processes. *In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, Proc. 24th International Colloquium on Automata, Languages and Programming (ICALP'97), volume 1256 of LNCS, pages 430-440. Springer, 1997.*

[18] **L. de Alfaro, M. Kwiatkowska, G. Norman, D. Parker, and R. Segala.** Symbolic model checking of probabilistic processes using MTBDDs and the Kronecker representation. *In S. Graf and M. Schwartzbach, editors, Proc. 6th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'00), volume 1785 of LNCS, pages 395-410. Springer, 2000.*

[19] **H. Hermanns, J.-P. Katoen, J. Meyer-Kayser, and M. Siegle.** A Markov chain model checker. *In S. Graf and M. Schwartzbach, editors, Proc. 6th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'00), volume 1785 of LNCS, pages 347-362. Springer, 2000.*

[20] **M. Kwiatkowska, G. Norman, and D. Parker.** Probabilistic model checking in practice: Case studies with PRISM. *ACM SIGMETRICS Performance Evaluation Review, 32(4):16-21, 2005.*

[21] **M. Kwiatkowska, G. Norman, and D. Parker.** PRISM 4.0: Verification of probabilistic real-time systems. *In G. Gopalakrishnan and S. Qadeer, editors, Proc. 23rd International Conference on Computer Aided Verification (CAV'11), volume 6806 of LNCS, pages 585-591. Springer, 2011.*

[22] `http://www.prismmodelchecker.org/`