Метод понижения вычислительной сложности в задачах верификации вероятностных моделей программ

Андрей Миронов
Институт Проблем Информатики РАН
Москва, Россия
amironov66@gmail.com

Сергей Френкель
Институт Проблем Информатики РАН
Москва, Россия
fsergei@mail.ru

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 описан 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

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 [1], которое имеет широкие применения в естественных и гуманитарных науках. Понятие ВСП можно рассматривать также как частный случай понятия вероятностного автомата [2]. 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 ([3], [4]), 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 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.

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

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 они были обобщены на случай верификации произвольных вероятностных свойств (в спецификации таких свойств могло присутствовать любое значение вероятности). Эти алгоритмы были изложены в работах [4], [13], [14]. Программные реализации этих алгоритмов были представлены в работах [15], [16].

Первые промышленные системы вероятностной верификации были разработаны в 2000-х годах [17], [18]. 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 [19]. 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 [20], разработанная на факультете компьютерных наук Оксфордского Университета (Великобритания) в группе Quantitative Analysis and Verification под руководством Марты Квиатковской [21].

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

ВСП удобно рассматривать как 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 записью $s_0 \overset{*}{\to} s_n$ 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 $i = 0$.

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

$$f : X \times Y \to [0,1] \tag{6}$$

такая, что $\forall\, x \in X \quad \sum\limits_{y \in Y} f(x,y) = 1$.

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 (6) называется **детерминированной**, если для каждого $x \in X$ существует единственный $y \in Y$, такой, что $f(x,y) = 1$. Если f – детерминированная СФ вида (6), 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 $(x_1, \ldots . x_m)$ и $(y_1, \ldots, y_n)$ соответственно, то СФ f можно представить в виде матрицы (обозначаемой тем же символом f)

$$f = \begin{pmatrix} f(x_1,y_1) & \ldots & f(x_1,y_n) \\ \ldots & \ldots & \ldots \\ f(x_m,y_1) & \ldots & f(x_m,y_n) \end{pmatrix} \tag{7}$$

Ниже мы будем отождествлять СФ f сматрицей (7).

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 любых $x \in X$, $y \in Y$ 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:

$$\forall\, x \in X \quad (f \cdot g)(x) \overset{\text{def}}{=} \sum_{y \in Y} f(x,y) \cdot g(y,z) \tag{8}$$

Согласно определению произведения матриц, из (8) 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 **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 [4].

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 $StateFm$ 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 $StateFm$. Данные символы обозначают тождественно истинное и тождественно ложное утверждение соответственно.
  3. Если φ_1 и φ_2 – формулы из $StateFm$, то следующие записи являются формулами из $StateFm$:

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

  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 запись $\mathcal{P}_{\triangle a}\alpha$ – формула из $StateFm$.

- $PathFm$:

  1. Если f – формула логики PCTL, то запись $\mathbf{X}f$ является формулой из $PathFm$.
  2. Если φ_1 и φ_2 – формулы из $StateFm$, то следующие записи 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 запись $\neg\alpha$ является формулой из $PathFm$.

В записи формул из $PathFm$ могут использоваться символы $\mathbf{F}$ и $\mathbf{G}$, которые являются сокращением записей $\top\mathbf{U}$ и $\neg\mathbf{F}\neg$ соответственно (т.е., например, записи $\mathbf{F}\alpha$ 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 правилах мы определяем значение $s(f)$, в других – определяем вектор-столбец $S(f)$ целиком. В этих определениях мы будем использовать следующие обозначения.

- Для любых $U = \begin{pmatrix} u_1 \\ \dots \\ u_n \end{pmatrix}$, $V = \begin{pmatrix} v_1 \\ \dots \\ v_n \end{pmatrix}$ 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

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

5 Метод редукции вероятностных систем переходов

5.1 Задача редукции вероятностных систем переходов

Если анализируемая ВСП имеет большой размер, то анализ её свойств, выражаемых формулами логики PCTL (т.е. вычисление значений формул логики PCTL в состояниях этой ВСП), может быть связан с трудновыполнимыми требованиями к вычислительным ресурсам, с использованием которых производится этот анализ. 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 склейку эквивалентных состояний.

5.2 Эквивалентность ВСП

Пусть заданы две ВСП:

$$D_i = (S_i, s_i^0, P_i, L_i) \;\; (i = 1, 2) \tag{9}$$

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 это записью $s_1 \sim s_2$.

Мы будем называть ВСП D_1 и D_2 вида (9) **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 (5.3).

5.3 Редукция вероятностных систем переходов

5.3.1 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'.

5.3.2 Построение разбиения множества состояний редуцируемой ВСП

Разбиение множества S состояний ВСП $D = (S, s^0, P, L)$, соответствующее 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 $\forall\, j = 1, \ldots, k \quad S(\varphi_j^{\Sigma^i}) = \Sigma_j^i$.

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

   $$\rho^{i+1} \stackrel{\text{def}}{=} \rho^i \cap \rho(\mathbf{X}\varphi_1^{\Sigma^i}, \ldots, \mathbf{X}\varphi_k^{\Sigma^i}). \qquad (10)$$

   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

     $$S(\mathbf{X}\varphi_j^{\Sigma^i}) = P \cdot \Sigma_j^i \qquad (11)$$

     (каждый из которых, как нетрудно видеть, является суммой некоторых столбцов матрицы P: для каждого $j = 1, \ldots, k$ вектор-столбец (11) 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 (11) 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 (10) следует включение

   $$\rho^i \subseteq \rho(\mathbf{X}\varphi_1^{\Sigma^i}, \ldots, \mathbf{X}\varphi_k^{\Sigma^i}) \qquad (12)$$

   Поскольку $\rho^i = \rho(Fm)$ для некоторого множества Fm формул логики PCTL, причем $AP \subseteq Fm$, то 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).

5.3.3 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

$$D_1 \stackrel{\text{def}}{=} (S_1, s^0, P_1, L_1) \tag{13}$$

где $S_1 \stackrel{\text{def}}{=} S \setminus \{s_1\}$, $\forall\, s, s' \in S_1$

$$P_1(s, s') \stackrel{\text{def}}{=} \begin{cases} P(s, s') + P(s, s_1) & \text{если } s' = s_2 \\ P(s, s') & \text{если } s' \neq s_2 \end{cases}$$

$\forall\, s \in S_1 \quad L_1(s) \stackrel{\text{def}}{=} L(s)$.

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, и матрица L_1 получается из матрицы L удалением строки s_1.

Ниже мы будем использовать следующие обозначения. Пусть A – матрица, соответствующая СФ вида $S \underset{r}{\to} S$. 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:

$$\begin{array}{l} P_1 = (id_S \backslash \vec{s}_1) \cdot P \cdot id_S(s_1, s_2, 1) \cdot (id_S \backslash s_1^{\downarrow}) \\ L_1 = (id_S \backslash \vec{s}_1) \cdot L \end{array} \tag{14}$$

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 (13) получается из ВСП D путем **удаления состояния s_1, эквивалентного состоянию s_2**. Согласно определению ВСП (13), 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 1. Пусть ВСП (13) получается из ВСП D путем удаления состояния s_1, эквивалентного состоянию s_2. Тогда $\forall\, s \in S_1 \quad s_{D_1}(f) = s_D(f)$.

5.3.4 Описание алгоритма редукции ВСП

Теорема 1 является обоснованием излагаемого ниже алгоритма редукции ВСП $D = (S, s^0, P, L)$. Этот алгоритм имеет следующий вид.

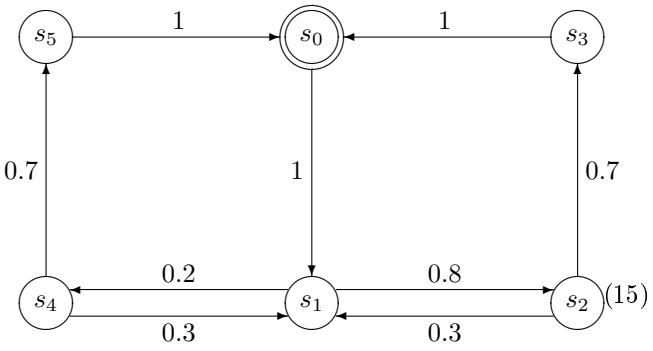
1. Вычисляется разбиение множества состояний ВСП D, соответствующее отношению эквивалентности $R \stackrel{\text{def}}{=} \rho(PCTL)$ (для этого выполняются действия, изложенные в пункте 5.3.2).
2. Искомая ВСП D' строится путём удаления состояний из ВСП D и переопределения функции перехода и отношения 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, и переходим на шаг 2a.
   (b) Если каждая пара, входящая в R, имеет вид (s, s), то завершаем работу.

6 Пример редукции вероятностной системы переходов

В этом пункте мы рассматриваем пример редукции вероятностной модели протокола передачи сообщений через ненадёжный канал связи, в котором пересылаемые сообщения могут пропадать или искажаться. Протокол представляет собой систему, состоящую из двух агентов - отправителя и получателя, а также канала, в который помещаются сообщения, пересылаемые от одного агента другому агенту. Мы предполагаем, что факт искажения получаемых сообщений может быть установлен, и если исходное сообщение не может быть восстановлено из искажённого, то отправитель получает сигнал о необходимости повторной посылки этого сообщения. Как только сообщение успешно доходит до получателя, отправителю посылается сигнал подтверждения успешного получения, и он переходит к отправке следующего

сообщения. Мы предполагаем, что сигналы и подтверждения отправителю не пропадают и не искажаются в канале.

Графовая модель этого протокола имеет следующий вид:



Переходы в этом графе имеют следующий смысл.

1. Переход $s_0 \overset{1}{\to} s_1$ заключается в получении отправителем от внешнего источника сообщения, которое должно быть передано через канал получателю.

2. Переход $s_1 \overset{0.8}{\to} s_2$ заключается в помещении сообщения в канал отправителем, причём сообщение в канале не искажается.

3. Переход $s_1 \overset{0.2}{\to} s_4$ заключается в помещении сообщения в канал отправителем, причём сообщение в канале искажается.

4. Переход $s_2 \overset{0.3}{\to} s_1$ заключается в потере неискажённого сообщения в канале и посылке отправителю сигнала о необходимости повторной передачи.

5. Переход $s_4 \overset{0.3}{\to} s_1$ заключается в посылке отправителю сигнала о том, что исходное сообщение не может быть восстановлено из искажённого сообщения и должно быть передано повторно.

6. Переход $s_2 \overset{0.7}{\to} s_3$ заключается в передаче неискажённого сообщения из канала получателю.

7. Переход $s_4 \overset{0.7}{\to} s_5$ заключается в восстановлении исходного сообщения из искажённого и передаче восстановленного сообщения получателю.

8. Переходы $s_3 \overset{1}{\to} s_0$ и $s_5 \overset{1}{\to} s_0$ заключаются в получении сообщения получателем и посылке им отправителю уведомления о том, что получение сообщения было выполнено успешно.

Одно из свойств протокола, представленного моделью (15). заключается в том, что каждое сообщение, полученное отправителем от внешнего источника, будет с вероятностью ≥ 0.9 доставлено получателю не более чем через 5 единиц времени. Для формального представления этого свойства мы будем полагать множество AP атомарных утверждений состоящим из одной переменной p, и эта переменная принимает в состоянии s_0 (15) значение 1, а в остальных состояниях (15) – значение 0. Таким образом, множество 2^{AP} состоит из двух элементов: $\emptyset$ и $\{p\}$. Мы будем обозначать эти элементы символами 0 и 1 соответственно.

Формула логики PCTL, соответствующая указанному выше свойству, имеет следующий вид:

$$\mathbf{G}((\neg p) \to \mathcal{P}_{\geq 0.9}(\mathbf{F}^{\leq 5} p)) \tag{16}$$

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

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

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

Анализируемая ВСП получается из графа (15) приписыванием к каждой его вершине s метки $L(s)$, которая равна 1, если $s = s_0$, и 0, иначе. Для вычисления значения формулы (16) в состояниях этой ВСП можно использовать описанный выше метод редукции.

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

	s_0	s_1	s_2	s_3	s_4	s_5
s_0	0	1	0	0	0	0
s_1	0	0	0.8	0	0.2	0
s_2	0	0.3	0	0.7	0	0
s_3	1	0	0	0	0	0
s_4	0	0.3	0	0.7	0	0
s_5	1	0	0	0	0	0

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

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

   По предположению, значение p в s_0 равно 1, и в каждом $s \in S$ (где S – множество состояний анализируемой ВСП), таком, что $s \neq s_0$, значение p равно 0, то $L(s_0) = 1$ и $\forall s \in S \setminus \{s_0\} \quad L(s) = 0$. Следовательно, Σ^0 состоит из двух классов

   $$\{s_0\}, \quad \{s_1, s_2, s_3, s_4, s_5\} \tag{17}$$

2. Матрица π^0, соответствующая детерминированной СФ

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

имеет вид

s_0	1	0
s_1	0	1
s_2	0	1
s_3	0	1
s_4	0	1
s_5	0	1

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

s_0	0	1
s_1	0	1
s_2	0	1
s_3	1	0
s_4	0	1
s_5	1	0

(18)

По матрице (18) нетрудно вычислить отношение ρ^1 и соответствующее ему разбиение Σ^1. Из определения отношения ρ^1 непосредственно следует, что состояния s и s' 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 (17), и кроме того строки матрицы (18), соответствующие состояниям s и s', совпадают.

Разбиение Σ^1 будет состоять из трех классов (измельчится второй класс в (17), а первый класс останется тем же), эти классы имеют следующий вид:

$$\{s_0\}, \quad \{s_1, s_2, s_4\}, \quad \{s_3, s_5\} \qquad (19)$$

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

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

Данная матрица будет иметь следующий вид:

s_0	1	0	0
s_1	0	1	0
s_2	0	1	0
s_3	0	0	1
s_4	0	1	0
s_5	0	0	1

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

s_0	0	1	0
s_1	0	1	0
s_2	0	0.3	0.7
s_3	1	0	0
s_4	0	0.3	0.7
s_5	1	0	0

После этого, действуя аналогичным образом, как и в предыдущем пункте, вычисляем классы разбиения Σ^2, соответствующего эквивалентности ρ^2. Таких классов будет четыре (измельчится второй класс в (19), а первый и третий классы останутся теми же), эти классы имеют следующий вид:

$$\{s_0\}, \quad \{s_1\}, \quad \{s_2, s_4\}, \quad \{s_3, s_5\} \qquad (20)$$

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

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

Данная матрица будет иметь следующий вид:

s_0	1	0	0	0
s_1	0	1	0	0
s_2	0	0	1	0
s_3	0	0	0	1
s_4	0	0	1	0
s_5	0	0	0	1

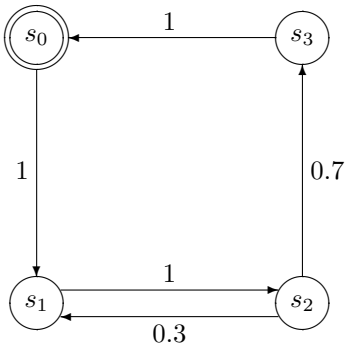
Произведение $P \cdot \pi_2$ имеет следующий вид:

s_0	0	1	0	0
s_1	0	0	1	0
s_2	0	0.3	0	0.7
s_3	1	0	0	0
s_4	0	0.3	0	0.7
s_5	1	0	0	0

(21)

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 (20).

Теперь можно приступить к удалению избыточных состояний (так, чтобы среди оставшихся состояний было ровно по одному состоянию из каждого класса эквивалентности (20)). Нетрудно видеть, что можно удалить состояния s_4 и s_5. После удаления данных состояний граф (15) примет следующий вид:



(22)

Таким образом, задача вычисления значений формулы (16) в состояниях ВСП (15) сводится к задаче вычисления значений формулы (16) в состояниях ВСП (22), что требует выполнения меньшего числа операций, чем задача вычисления значений формулы (16) в состояниях исходной ВСП.

7 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.

Список литературы

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

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

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

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

[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] **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.*

[14] **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.*

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

[16] **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.*

[17] **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.*

[18] **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.*

[19] **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.*

[20] **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.*

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