Моделирование и анализ информационных систем. Т. 23, № 6 (2016), с. 741–753
Modeling and Analysis of Information Systems. Vol. 23, No 6 (2016), pp. 741–753

DOI: 10.18255/1818-1015-2016-6-741-753

УДК 517.9

О минимизации конечных автоматов-преобразователей над полугруппами

Захаров В. А.[1], **Темербекова Г.Г.**

получена 15 августа 2016

Аннотация. Автоматы-преобразователи над полугруппами можно использовать в качестве модели последовательных реагирующих программ, работающих в постоянном взаимодействии со своим окружением. Получив очередную порцию данных, реагирующая программа выполняет некоторую последовательность действий и предъявляет результат. Такие программы возникают при проектировании компьютерных драйверов, алгоритмов, работающих в оперативном режиме, сетевых коммутаторов. Во многих случаях проблема верификации программ такого рода может быть сведена к задачам минимизации и проверки эквивалентности конечных автоматов-преобразователей. Минимизация преобразователей над полугруппами проводится в три этапа. Вначале для всех состояний преобразователя вычисляются наибольшие общие левые делители. Затем все вычисленные делители "поднимаются вверх" по переходам преобразователя, и в результате образуется приведенный преобразователь. Наконец, для минимизации приведенных преобразователей применяются методы минимизации классических конечных автоматов-распознавателей.

Ключевые слова: реагирующая система, автомат-преобразователь, полугруппа, минимизация, проверка эквивалентности

Для цитирования: Захаров В. А., Темербекова Г.Г., "О минимизации конечных автоматов-преобразователей над полугруппами", *Моделирование и анализ информационных систем*, **23**:6 (2016), 741–753.

Об авторах:
Захаров Владимир Анатольевич, orcid.org/0000-0002-3794-9565, доктор физ.-мат. наук, профессор,
Московский государственный университет им. М.В. Ломоносова,
факультет ВМК, Ленинские горы, д. 1, стр. 52, ГСП-1, Москва, 119991, Россия, e-mail: zakh@cs.msu.ru

Темербекова Гульгайша Габдуловна, orcid.org/0000-0002-5856-8788, магистр,
Московский государственный университет им. М.В. Ломоносова,
факультет ВМК, Ленинские горы, д. 1, стр. 52, ГСП-1, Москва, 119991, Россия, e-mail: gulgaisha93@mail.ru

Благодарности:
[1]Работа выполнена при финансовой поддержке исследовательской программы НИУ ВШЭ в 2016 г. и гранта РФФИ №16-01-00546

Введение

Автоматы-преобразователи (трансдьюсеры) представляют собой расширения конечных автоматов, предназначенные для моделирования функций над строками и списками. Область их применения охватывает разделы от тестирования и оптимизации программ [1, 15] до компьютерной лингвистики [10]. В программировании автоматы-преобразователи служат удобной моделью для разнообразных драйверов,

проводящих манипуляции со строками, преобразование изображений, коммутацию и сортировку потоков данных и др. С помощью этой модели можно конструировать композиции драйверов, анализировать их поведение, проводить тестирование. Преобразователи нашли применение в некоторых методах верификации параметризованных моделей распределенных систем: конфигурации системы с произвольным неограниченным числом процессов представимы в виде слов некоторого конечного алфавита, и отношения переходов между конфигурациями можно задать при помощи автоматов-преобразователей, работающих над словами этого алфавита [18]. Понятно, что чем проще устроены эти преобразователи, тем эффективнее работают алгоритмы верификации регулярных моделей такого рода. В статье [14] предложено использовать в качестве модели коммуникационных протоколов автоматы-преобразователи, работающие над битовыми строками, и решать проблему верификации протоколов как задачу проверки эквивалентности двух преобразователей, один из которых моделирует протокол, а другой описывает его спецификацию. Приведенные примеры свидетельствуют о том, что алгоритмы проверки эквивалентности и уменьшения размеров автоматов-преобразователей востребованы для решения задач проектирования, верификации и оптимизации некоторых программ.

Автоматы-преобразователи могут служить простой формальной моделью последовательных реагирующих программ, работающих во взаимодействии со своим окружением. После получения очередной порции данных или запроса реагирующая программа выполняет некоторую последовательность действий. При достижении определенных контрольных точек программа выдает сложившийся к этому моменту результат вычисления в качестве отклика на запросы. Поскольку разные последовательности действий могут приводить к одному и тому же результату, нам потребуется более изощренная интерпретация действий программы, нежели простое представление о них как о словах в некотором алфавите. Базовые действия программы можно рассматривать как порождающие элементы некоторой полугруппы; тогда результатом вычисления будет элемент полугруппы, представленный композицией выполненных простейших действий.

Рассмотрим в качестве примера радиоуправляемый робот, способный совершать шаги в любом из четырех направлений N, E, S, W. При получении управляющего сигнала syg в состоянии q он должен выполнить последовательность шагов (например N, N, W, S) и перейти в следующее состояние q'. Достигнув некоторого особого состояния q_{fin}, робот сообщает о своем местоположении. Наиболее естественная модель вычислений, которую можно использовать при проектировании такого робота и исследовании его поведения, — это автомат-преобразователь, оперирующий над свободной абелевой группой ранга 2. Другим примером может служить сетевой коммутатор, на вход которого поступают потоки пакетов, перемежающиеся с командами управления. Руководствуясь таблицей коммутации пакетов, это сетевое устройство отправляет модифицированные копии каждого пакета в тот или иной выходной порт. Команды управления вносят изменения в таблицу коммутации. Модификация и коммутация пакетов — это элементарные действия коммутатора. Когда два пакета из разных потоков коммутируются в разные выходные порты, соответствующие действия могут выполняться в произвольном порядке. Поэтому такой коммутатор можно моделировать конечным автоматом-преобразователем, работающим над частично коммутативной полугруппой (множеством трасс [6]).

В данной статье исследуются задачи минимизации и проверки эквивалентности конечных автоматов-преобразователей, работающих над полугруппами. Изучение этих задач в основном проводилось для классических преобразователей, работающих над словами. В [9] удалось установить, что проблема эквивалентности неразрешима для недетерминированных преобразователей. Но этот эффект возникает лишь тогда, когда входные слова могут иметь неограниченно много образов. Для ограниченно недетерминированных преобразователей проблема эквивалентности разрешима; решение достигается для детерминированных преобразователей [3] за полиномиальное время [8], а для функциональных [2] и k-значных преобразователей [5] за экспоненциальное время [17]. Более общий метод проверки эквивалентности [19] позволил получить аналогичные результаты для автоматов-преобразователей, которые работают над полугруппами, вложимыми в разрешимые группы.

Начало исследованию задачи минимизации конечных автоматов-преобразователей было положено в статье [12], но приемлемое решение было впервые получено в работе [11]. Предложенный в ней алгоритм минимизации был исправлен и улучшен в статьях [4, 13]. В работе [7] была предпринята попытка адаптировать этот алгоритм к взвешенным преобразователям, используемым в компьютерной лингвистике. Другой подход к решению задачи минимизации был предложен в статье [20]: при помощи алгоритма проверки эквивалентности, разработанного в статье [19], задачу минимизации удалось решить для преобразователей, работающих над группами. Для этого метода требуется, чтобы все элементы полугруппы были обратимы.

В данной статье показано, как обобщить метод минимизации, предложенный в статье [11], применительно к преобразователям, работающим над упорядоченными полугруппами. Минимизация преобразователя π, оперирующего над полугруппой S, проводится в три этапа. Вначале для каждого состояния q находится наибольший общий левый делитель $gcd(\pi, q)$ всех элементов полугруппы S, которые вычисляются на прогонах π из состояния q. Затем все $gcd(\pi, q)$ "поднимаются вверх" по переходам преобразователя; в результате образуется приведенный преобразователь π', у которого все $gcd(\pi, q)$ равны нейтральному элементу e полугруппы S. И, наконец, для минимизации приведенных преобразователей применяются методы минимизации классических конечных автоматов-распознавателей (см. [16]).

1. Автоматы-преобразователи над полугруппами

Пусть заданы два конечных множества $\mathcal{C}$ и $\mathcal{A}$. Элементы множества $\mathcal{C}$ будем называть *входными сигналами*; их нужно рассматривать как сообщения (команды управления, показания датчиков, пакеты данных и др.), поступающие на вход реагирующей системы от окружающей среды. Мы абстрагируемся от природы и структуры сигналов. Конечные последовательности входных сигналов (слова в алфавите $\mathcal{C}$) будем называть *потоками сигналов*. Обозначим записью $\mathcal{C}^*$ множество всевозможных потоков сигналов, а записью uv — конкатенацию потоков сигналов u и v.

Элементы множества $\mathcal{A}$ будем называть *простыми действиями*; к их числу относятся операции обработки данных, отправления сообщений и др. Конечные по-

следовательности простых действий, представляющие собой слова в алфавите $\mathcal{A}$, будем называть *составными действиями*.

Для интерпретации действий воспользуемся полугруппами. Рассмотрим полугруппу $(S, e, \circ)$, порожденную множеством простых действий $\mathcal{A}$, в которой e обозначает нейтральный элемент, а $\circ$ — полугрупповую операцию. Элементы полугруппы S выступают в роли *состояний данных*. Применив простое действие a к состоянию данных s, получаем результат $s \circ a$. Составное действие $h = a_1 a_2 \dots a_k$ представляет собой композицию $[h]_S = a_1 \circ a_2 \circ \dots \circ a_k$ простых действий системы. Индекс S может быть опущен, если из контекста понятно, о какой полугруппе идет речь.

Детерминированный *автомат-преобразователь* с конечным числом состояний над множеством сигналов $\mathcal{C}$ и множеством базовых действий $\mathcal{A}$ — это размеченная система переходов $\pi = (\mathcal{C}, \mathcal{A}, Q, q_0, F, T, h_0, E)$, состоящая из

- конечного множества *состояний управления* Q,
- *начального состояния* $q_0 \in Q$,
- множества *состояний выхода* $F \subseteq Q$,
- *функции переходов* $T : Q \times \mathcal{C} \to Q \times \mathcal{A}^*$,
- *инициализирующего действия* $h_0 \in \mathcal{A}^*$,
- *функции финализации* $E : F \to \mathcal{A}^*$.

Каждая четверка (q, c, q', h), удовлетворяющая равенству $T(q, c) = (q', h)$, называется *переходом* и традиционно обозначается записью $q \xrightarrow{c,h} q'$. *Размером* $|\pi|$ автомата-преобразователя π называется число $|Q|$ его состояний управления.

Автоматы-преобразователи можно использовать в качестве формальных моделей последовательных реагирующих систем. В начале работы реагирующей системы происходит ее инициализация, которая состоит в выполнении последовательности действий h_0. Далее реагирующая система может функционировать неограниченно долго, обрабатывая поступающие на ее вход сообщения (сигналы). Обработка сообщений сопряжена с выполнением переходов. Каждый переход $q \xrightarrow{c,h} q'$ соответствует элементарному шагу вычислений: система, пребывающая в состоянии q, приняв на входе сообщение c, выполняет последовательность действий h и переходит в новое состояние q'. Стороннему наблюдателю доступны лишь те результаты вычислений системы, которые образуются при достижении состояний выхода. В них реагирующая система выполняет финализирующие действия и формирует отклик: мобильный робот сообщает о своем местоположении, блочный шифратор отправляет очередной блок шифртекста, сетевой драйвер завершает обработку пакета данных. По откликам можно судить о функции, реализуемой реагирующей программой. Более формально это можно определить при помощи следующих понятий.

Прогоном автомата-преобразователя π на потоке сигналов $w = c_1 c_2 \dots c_n$ из состояния управления q называется последовательность переходов

$$q \xrightarrow{c_1,h_1} q_1 \xrightarrow{c_2,h_2} q_2 \xrightarrow{c_3,h_3} \dots \xrightarrow{c_n,h_n} q' \ , \tag{1}$$

которую будем обозначать записью $q \xrightarrow{w,h}_* q'$, где $h = h_1 h_2 \dots h_n$. Если состояние управления q является инициальным, то прогон также называется *инициальным*, а если $q' \in F$, то прогон называется *финальным*. Прогон, являющийся одновременно инициальным и финальным, называется *полным*. Для полного прогона

$q \xrightarrow{w,h}_* q'$ автомата-преобразователя π, действия которого интерпретируются в полугруппе S, состояние данных $[h_0 hE(q')]_S$ считается *результатом* этого прогона. Таким образом, поведение реагирующей системы, которая моделируется автоматом-преобразователем π, характеризуется частичной функцией $\pi : \mathcal{C}^* \to S$; ее значения для потока сигналов w определяются соотношением

$$\pi(w) = \begin{cases} [h_0 hE(q')], \text{ если преобразователь } \pi \text{ имеет полный прогон } q \xrightarrow{w,h}_* q', \\ \text{не определено, в противном случае.} \end{cases}$$

Состояние управления q преобразователя π считается *полезным*, если через него проходит хоть один полный прогон. Бесполезные состояния не влияют на функцию, вычисляемую преобразователем, и могут быть удалены. Далее мы будем полагать, что все состояния рассматриваемых преобразователей являются полезными.

Для заданной полугруппы $(S, e, \circ)$ преобразователи π_1 и π_2 называются *S-эквивалентными*, если равенство $\pi_1(w) = \pi_2(w)$ выполняется для любого потока сигналов w. Отношение S-эквивалентности условимся обозначать записью $\pi_1 \sim_S \pi_2$. Задача проверки эквивалентности автоматов-преобразователей над полугруппой S состоит в том, чтобы для произвольной заданной пары преобразователей π_1 и π_2 выяснить, являются ли они S-эквивалентными. Автомат-преобразователь π' называется *S-минимальным*, если неравенство $|\pi'| \leq |\pi|$ выполняется для любого преобразователя π, S-эквивалентного преобразователю π'. Задача минимизации автоматов-преобразователей над полугруппой S состоит в том, чтобы для произвольного заданного преобразователя π построить S-эквивалентный ему S-минимальный преобразователь π'.

Для некоторых моделей вычислений задачи проверки эквивалентности и минимизации взаимосвязаны. Так, например, разделив при помощи алгоритма проверки эквивалентности все состояния детерминированного конечного автомата-распознавателя (автомата Рабина-Скотта) на классы эквивалентности, можно легко построить минимальный автомат, а для проверки эквивалентности двух детерминированных конечных автоматов достаточно провести их минимизацию и проверить изоморфность полученных минимальных автоматов. Аналогичный эффект имеет место для конечных автоматов-преобразователей, работающих над свободными полугруппами [11], над группами [20], и, как показано в данной статье, над полугруппами действий, которые обладают некоторыми свойствами, «естественными» как с алгебраической, так и с вычислительной точки зрения.

2. Унифицированные автоматы-преобразователи

Вначале при помощи простого приема унифицируем действия инициализации и финализации преобразователей. Пусть задан конечный автомат-преобразователь $\pi = (\mathcal{C}, \mathcal{A}, Q, q_0, F, T, h_0, E)$. Расширим множество сигналов $\mathcal{C}$, введя два новых сигнала $init$ и fin; положим $\mathcal{C}_0 = \mathcal{C} \cup \{init, fin\}$. Расширим множество состояний управления Q, введя два новых состояния *start* и *stop*; положим $Q_0 = Q \cup \{start, stop\}$. Доопределим функцию переходов T на новых элементах области определения; по-

ложим

$$T_0(q,x) \;=\; \begin{cases} T(q,x), \text{ если } q \in Q \text{ и } x \in \mathcal{A}, \\ (q_0, h_0), \text{ если } q = start \text{ и } x = init, \\ (stop, E(q)), \text{ если } q \in F \text{ и } x = fin, \\ \text{не определено, в остальных случаях.} \end{cases}$$

В итоге получим автомат-преобразователь $\pi_0 = (\mathcal{C}_0, \mathcal{A}, Q_0, start, \{stop\}, T_0, \varepsilon, E_0)$, в котором определена новая функция финализации E_0, принимающая значение $E_0(stop) = \varepsilon$. Такой преобразователь назовем *унифицированным*. Как видно из приведенного описания, для любого потока сигналов $w = c_1c_2\ldots c_n$ преобразователь π имеет полный прогон (1) тогда и только тогда, когда унифицированный преобразователь π_0 для потока сигналов $w' = init, c_1c_2\ldots c_n, fin$ имеет полный прогон

$$start \xrightarrow{init,h_0} q \xrightarrow{c_1,h_1} q_1 \xrightarrow{c_2,h_2} q_2 \xrightarrow{c_3,h_3} \cdots \xrightarrow{c_n,h_n} q' \xrightarrow{fin,E(q')} stop\ . \tag{2}$$

Отсюда следует

Утверждение 1. *Для любой пары автоматов-преобразователей π' и π'' верно соотношение $\pi' \sim_S \pi'' \iff \pi'_0 \sim_S \pi''_0$.*

Это утверждение позволяет ограничиться исследованием задач проверки эквивалентности и минимизации только для унифицированных автоматов-преобразователей, которые не выполняют действий инициализации и финализации. Каждому такому преобразователю $\pi_0 = (\mathcal{C}_0, \mathcal{A}, Q_0, start, \{stop\}, T_0, \varepsilon, E_0)$, работающему над полугруппой $(S, e, \circ)$, можно сопоставить детерминированный автомат-распознаватель $A_{\pi_0} = (\mathcal{C}_0 \times S, Q_0, start, \{stop\}, \varphi)$ над алфавитом (в общем случае бесконечным) пар $\mathcal{C}_0 \times S$. Функция переходов $\varphi : Q_0 \times (\mathcal{C}_0 \times S) \to Q_0$ этого автомата определяется следующим соотношением: $\varphi(q,(c,s)) = q' \iff T_0(q,c) = (q',h) \wedge s = [h]$. На вход автомата, пребывающего в начальном состоянии $start$, поступает конечная последовательность пар $\alpha = (c_1,s_1),(c_2,s_2),\ldots,(c_n,s_n)$. Она допускается автоматом, если после ее прочтения автомат переходит в состояние выхода $stop$. Нетрудно заметить, что автомат A_{π_0} допускает последовательность пар α в том и только том случае, когда преобразователь π_0 имеет такой полный прогон (2), для которого равенство $[h_i] = s_i$ выполняется для каждого $i, 1 \le i \le n$. Унифицированные преобразователи π'_0 и π''_0 назовем *строго эквивалентными* на полугруппе S (и обозначим это отношение записью $\pi'_0 \approx_S \pi''_0$), если автоматы-распознаватели $A_{\pi'_0}$ и $A_{\pi''_0}$ допускают одно и то же множество слов. Из определения строгой эквивалентности следует

Утверждение 2. *Для любой пары унифицированных автоматов-преобразователей π'_0 и π''_0 верно соотношение $\pi'_0 \approx_S \pi''_0 \Rightarrow \pi'_0 \sim_S \pi''_0$.*

Обратное соотношение в общем случае неверно. Ключевая идея решения задачи минимизации для автоматов-преобразователей состоит в том, чтобы для некоторого класса полугрупп S суметь выделить семейство приведенных преобразователей, удовлетворяющих следующим двум требованиям:

1) для каждого преобразователя π можно эффективно построить S-эквивалентный приведенный преобразователь π' того же самого размера, т.е. $|\pi| = |\pi'|$,

2) для каждой пары приведенных преобразователей π' и π'' верно соотношение $\pi' \sim_S \pi'' \iff \pi' \approx_S \pi''$.

Тогда для минимизации преобразователя π нужно построить эквивалентный приведенный преобразователь π', а затем, применив любой из известных методов минимизации автоматов-распознавателй (см., например, [16]), минимизировать детерминированный конечный автомат-распознаватель $A_{\pi'}$. Тот преобразователь π'', который соответствует построенному минимальному автомату-распознавателю, и будет являться результатом минимизации исходного преобразователя π. А для проверки выполнимости отношения $\pi_1 \sim_S \pi_2$ достаточно построить S-эквивалентные приведенные преобразователи π_1' и π_2', а затем проверить эквивалентность детерминированных конечных автоматов-распознавателей $A_{\pi_1'}$ и $A_{\pi_2'}$.

3. Упорядоченные левосократимые полугруппы

В этом разделе перечислены требования, которым должна удовлетворять полугруппа $(S, e, \circ)$, чтобы для нее можно было осуществить описанную выше стратегию минимизации детерминированных автоматов-преобразователей.

На множестве S элементов полугруппы определим бинарное отношение $\preceq_S$ следующим образом: для любой пары элементов s_1, s_2 отношение $s_1 \preceq_S s_2$ выполняется тогда и только тогда, когда для некоторого элемента s имеет место равенство $s_1 \circ s = s_2$. Полугруппа называется *упорядоченной*, если $\preceq_S$ является отношением частичного порядка на множестве S. Далее мы будем опускать индекс в обозначении отношения $\preceq_S$, если из контекста ясно, о какой полугруппе идет речь.

Первое требование, предъявляемое к рассматриваемой полугруппе, таково.

Req1: Частично упорядоченное множество $(S, \preceq)$ является фундированной решеткой, в которой для каждой пары элементов $[h_1]$ и $[h_2]$ эффективно вычислима их точная нижняя грань.

Будем использовать записи $s_1 \vee s_2$ и $s_1 \wedge s_2$ для обозначения соответственно точной нижней грани и точной верхней грани элементов s_1 и s_2. По сути дела, $s_1 \vee s_2$ — это наибольший общий левый делитель элементов s_1 и s_2, а $s_1 \wedge s_2$ — это наименьшее общее кратное элементов s_1 и s_2. Из определения отношения $\preceq$ следует, что справедлив закон левой дистрибутивности операции композиции действий относительно операции взятия точной нижней грани: $s \circ s_1 \vee s \circ s_2 = s \circ (s_1 \vee s_2)$. Нейтральный элемент полугруппы e является наименьшим элементом решетки $(S, \preceq)$, но у решетки может не оказаться наибольшего элемента. Мы добавим к множеству S еще один мнимый элемент τ, для которого будем полагать справедливыми равенства $s \circ \tau = \tau \circ s = \tau$ для всякого элемента s из S. Ясно, что $s \preceq \tau$ для любого s из S. Положим $S_\tau = S \cup \{\tau\}$. Таким образом, если рассматриваемая полугруппа удовлетворяет требованию **Req1**, то частично упорядоченное множество $(S_\tau, \preceq)$ является полной фундированной решеткой. Для всякого множества S' элементов этой решетки условимся обозначать записью $\bigvee S'$ точную нижнюю грань множества S'.

Второе требование касается разрешимости линейных уравнений в рассматриваемой полугруппе $(S, e, \circ)$.

Req2: Существует алгоритм решения уравнений вида $[g] \circ X = [h]$ для любой заданной пары действий $g, h \in \mathcal{A}^*$.

Заметим, что если рассматриваемая полугруппа удовлетворяет требованиям **Req1** и **Req2**, то проблема тождества $[g] \stackrel{?}{=} [h]$ в этой полугруппе разрешима.

Последнее требование касается свойства сократимости. Полугруппа называется *левосократимой*, если всякое уравнение вида $s \circ X = \widehat{s}$ имеет не более одного решения, т.е. соотношение $s \circ s' = s \circ s'' \;\Rightarrow\; s' = s''$ верно для любой тройки s, s', s''.

Req3: $(S, e, \circ)$ — левосократимая полугруппа.

Многие полугруппы, используемые в тех или иных моделях в теории вычислений, удовлетворяют перечисленным выше требованиям. Например, требованиям **Req1**–**Req3** удовлетворяют частично коммутативные моноиды (трассы) [6], при помощи которых можно описывать поведение систем взаимодействующих процессов.

4. Наибольшие общие делители состояний

Минимизация автоматов-преобразователей проводится в три этапа. Вначале для каждого состояния управления q униформного преобразователя π_0 нужно найти наибольшие общие делители всех состояний данных, которые вычисляются на финальных прогонах из q.

Предположим, что униформный преобразователь π_0 имеет множество состояний управления $Q_0 = \{q_1, q_2, \ldots, q_n\}$. Для каждого состояния управления q_i преобразователя π_0 сформируем множество $S(\pi_0, q_i) = \{[h] : q_i \xrightarrow{w,h}_* stop\}$ результатов финальных прогонов, начинающихся из q_i. Точную нижнюю грань $gcd(\pi_0, q_i) = \bigvee S(\pi_0, q_i)$ будем называть *наибольшим общим делителем* состояния управления q_i; обозначим записью $GCD(\pi_0)$ набор $(gcd(\pi_0, q_1), gcd(\pi_0, q_2), \ldots, gcd(\pi_0, q_n))$ наибольших общих делителей всех состояний преобразователя. Чтобы вычислить наибольшие общие делители всех состояний управления преобразователя π_0, введем оператор $\Psi_{\pi_0} : S_\tau^n \to S_\tau^n$, значения которого определяются так: для каждого набора элементов $(s_1, s_2, \ldots, s_n)$ из S_τ^n будем полагать, что $\Psi_{\pi_0}(s_1, s_2, \ldots, s_n) = (s'_1, s'_2, \ldots, s'_n)$, где

$$s'_i = \begin{cases} e, \text{ если } q_i = stop, \\ \bigvee\{[h] \circ s_j : T_0(q_i, c) = (q_j, h), c \in \mathcal{C}_0\}, \text{ в противном случае,} \end{cases}$$

для каждого $i, 1 \le i \le n$. Частичный порядок $\preceq$ можно естественным образом распространить на множество наборов S_τ^n: $(s_1, s_2, \ldots, s_n) \preceq (s'_1, s'_2, \ldots, s'_n) \iff \forall i : s_i \preceq s'_i$. Максимальным набором здесь является набор $\top = (\tau, \tau, \ldots, \tau)$.

Утверждение 3. *Если полугруппа $(S, e, \circ)$ удовлетворяет требованию **Req1**, то оператор Ψ_{π_0} является монотонным.*

Справедливость утверждения непосредственно следует из определения оператора Ψ_{π_0}. Поскольку решетка $(S_\tau, \preceq)$ полна, по теореме Кнастера–Тарского оператор Ψ_{π_0} имеет наибольшую неподвижную точку $gfp(\Psi_{\pi_0})$. По теореме Клини она является пределом убывающей последовательности наборов $\top \succeq_S \Psi_{\pi_0}(\top) \succeq_S \Psi_{\pi_0}(\Psi_{\pi_0}(\top)) \succeq_S \ldots$. Так как согласно требованию **Req1** решетка $(S, \preceq)$ является фундированной, существует такое k, для которого $\Psi_{\pi_0}^k(\top) = \Psi_{\pi_0}^{k+1}(\top) = gfp(\Psi_{\pi_0})$. Значит, $gfp(\Psi_{\pi_0})$ вычисляется эффективно.

Утверждение 4. *Если полугруппа $(S, e, \circ)$ удовлетворяет требованию **Req1**, то $gfp(\Psi_{\pi_0}) = GCD(\pi_0)$.*

Доказательство. 1). Если $q_i = stop$, то $S(\pi_0, q_i) = \{e\}$, и поэтому $gcd(\pi_0, q_i) = e$. В противном случае $S(\pi_0, q_i) = \bigcup\limits_{c \in \mathcal{C}_0} \{[h] \circ [g] : T_0(q_i, c) = (q_j, h), g \in S(\pi_0, q_j)\}$, и по закону левой дистрибутивности $\circ$ относительно $\vee$ имеем

$$gcd(\pi_0, q_i) = \bigvee \{[h] \circ gcd(\pi_0, q_j) : T_0(q_i, c) = (q_j, h), c \in \mathcal{C}_0\}.$$

Следовательно, $GCD(\pi_0)$ — это неподвижная точка оператора Ψ_{π_0}.

2). Допустим, что $gfp(\Psi_{\pi_0}) = (s'_1, s'_2, \ldots, s'_n)$. Рассмотрим какой-либо финальный прогон $q_i \xrightarrow{w,h}_* stop$ преобразователя π_0. Индукцией по длине этого прогона можно показать, что $s'_i \preceq [h]$. Если $q_i = stop$, то по определению оператора Ψ_{π_0} неравенство $s'_i = e \preceq [h]$ справедливо для любого действия h. Предположим, что прогон имеет вид $q_i \xrightarrow{c,g} q_k \xrightarrow{w',h'}_* stop$. Тогда согласно индуктивной гипотезе и определению оператора Ψ_{π_0} справедливы неравенства $s'_i \preceq [g] \circ s'_k \preceq [g] \circ [h'] = [gh'] = [h]$.

Так как неравенство $s'_i \preceq [h]$ соблюдается для каждого состояния данных $[h]$ из множества $S(\pi_0, q_i)$, верно неравенство $s'_i \preceq gcd(\pi_0, q_i)$. Таким образом, $gfp(\Psi_{\pi_0}) \preceq GCD(\pi_0)$, и, значит, $gfp(\Psi_{\pi_0}) = GCD(\pi_0)$. □

5. Редукция преобразователей

На следующем этапе минимизации проводится редукция автомата-преобразователя. Униформный преобразователь π_0, который работает над полугруппой, удовлетворяющей требованию **Req1**, назовем *приведенным*, если $gcd(\pi_0, q) = e$ для любого состояния q, отличного от начального состояния $start$.

Теорема 1. *Если полугруппа $(S, e, \circ)$ удовлетворяет требованиям **Req1**–**Req3**, то для каждого унифицированного преобразователя π_0 можно эффективно построить S-эквивалентный приведенный преобразователь π'_0 такого же размера.*

Доказательство. Для каждого перехода $q_i \xrightarrow{c,h} q_j$ в преобразователе π_0 рассмотрим уравнение $gcd(\pi_0, q_i) \circ X = [h] \circ gcd(q_j)$. Так как $S(\pi_0, q_i) \supseteq \{[h] \circ s : s \in S(\pi_0, q_j)\}$, из определения наибольшего общего делителя следует неравенство $gcd(\pi_0, q_i) \preceq [h] \circ gcd(\pi_0, q_j)$. Значит, указанное уравнение обязательно имеет решение $X = g_{c,i}$, которое согласно требованию **Req2** можно вычислить эффективно. Заметим, что для перехода $start \xrightarrow{init,h_0} q_1$ соответствующее уравнение имеет решение $X = e$. Образуем преобразователь π'_0 из преобразователя π_0 заменой каждого перехода $q_i \xrightarrow{c,h} q_j$ переходом $q_i \xrightarrow{c,g_{c,i}} q_j$ в случае $q_i \neq start$ или переходом $q_i \xrightarrow{init,g_0} q_j$, где $g_0 = gcd(\pi_0, start)$, в случае $q_i = start$.

Равенство $gcd(\pi_0, q_i) \circ [g_{c,i}] = [h] \circ gcd(\pi_0, q_j)$, обеспечивающее взаимосвязь между переходами $q_i \xrightarrow{c,h} q_j$ и $q_i \xrightarrow{c,g_{c,i}} q_j$ преобразователей π_0 и π'_0, можно распространить на прогоны этих преобразователей. Рассмотрим пару соответствующих прогонов преобразователей π_0 и π'_0 на одном и том же потоке сигналов $w = c_1 c_2 \ldots c_{m-1} c_m$:

$$q_1 \xrightarrow{c_1,h_1} q_2 \xrightarrow{c_2,h_2} \cdots q_{m-1} \xrightarrow{c_{m-1},h_{m-1}} q_m \xrightarrow{c_m,h_m} q_{m+1},$$
$$q_1 \xrightarrow{c_1,g_1} q_2 \xrightarrow{c_2,g_2} \cdots q_{m-1} \xrightarrow{c_{m-1},g_{m-1}} q_m \xrightarrow{c_m,g_m} q_{m+1}.$$

Учитывая устройство преобразователя π'_0, можно получить цепочку равенств

$$[h_1 \dots h_{m-1}h_m] \circ gcd(\pi_0, q_m) = [h_1h_2 \dots h_{m-1}] \circ [h_m] \circ [gcd(\pi_0, q_{m+1})] =$$
$$= [h_1h_2 \dots h_{m-1}] \circ [gcd(\pi_0, q_m)] \circ [g_m] = [h_1h_2 \dots h_{m-2}] \circ gcd(\pi_0, q_{m-1}) \circ [g_{m-1}g_m] = \cdots$$
$$\cdots = [h_1] \circ [gcd(\pi_0, q_2)] \circ [g_2 \dots g_{m-1}g_m] = gcd(\pi_0, q_1) \circ [g_1 \dots g_{m-1}g_m] \ .$$

Чтобы убедиться в том, что $\pi_0 \sim_S \pi'_0$, прежде всего заметим, что обе функции $\pi_0(\cdot)$ и $\pi'_0(\cdot)$ определены на одном и том же множестве потоков сигналов. Рассмотрим произвольный поток сигналов w, на котором определено значение $\pi_0(w)$, и полные прогоны $start \xrightarrow{init,h_0} q_1 \xrightarrow{w,h}_* stop$ и $start \xrightarrow{init,g_0} q_1 \xrightarrow{w,g}_* stop$ преобразователей π_0 и π'_0 на w. Так как $gcd(\pi_0, stop) = e$, имеет место следующая цепочка равенств:

$$\pi_0(w) = [h_0h] = [h_0] \circ [h] \circ [gcd(\pi_0, stop)] = [h_0] \circ [gcd(\pi_0, q_1)] \circ [g] =$$
$$= [gcd(\pi_0, start)] \circ [g] = [g_0] \circ [g] = \pi'_0(w) \ .$$

Следовательно, $\pi_0(w) = \pi'_0(w)$ для каждого потока сигналов w.

Чтобы убедиться в том, что π'_0 — приведенный преобразователь, рассмотрим произвольное состояние управления q_i, отличное от начального, и элемент $gcd(\pi_0, q_i) = \bigvee\{[h] : q_i \xrightarrow{w,h}_* stop\}$. Опираясь на установленную взаимосвязь между соответствующими прогонами преобразователей π_0 и π'_0, а также принимая во внимание очевидное равенство $gcd(\pi_0, stop) = e$, можно заметить, что

$$gcd(\pi_0, q_i) \circ gcd(\pi'_0, q_i) = \bigvee\{gcd(\pi_0, q_i) \circ [g] : q_i \xrightarrow{w,g}_* stop\} =$$
$$= \bigvee\{[h] \circ gcd(\pi_0, stop) : q_i \xrightarrow{w,h}_* stop\} = gcd(\pi_0, q_i) \ .$$

Поскольку S — левосократимая полугруппа, равенство $gcd(\pi_0, q_i) \circ gcd(\pi'_0, q_i) = gcd(\pi_0, q)$ влечет $gcd(\pi'_0, q_i) = e$. □

6. Минимизация приведенных преобразователей

На последнем этапе для минимизации приведенных преобразователей применяются методы минимизации детерминированных конечных автоматов-распознавателей на основании взаимосвязи между приведенными преобразователями и конечными автоматами, которая устанавливается в следующем утверждении.

Утверждение 5. *Пусть π'_0 и π''_0 — пара приведенных S-эквивалентных преобразователей, которые работают над полугруппой $(S, e, \circ)$, удовлетворяющей требованиям **Req1, Req3**, и пусть $start \xrightarrow{w,h'}_* q'_1 \xrightarrow{c,g'} q'_2$ и $start \xrightarrow{w,h''}_* q''_1 \xrightarrow{c,g''} q''_2$ — пара начальных прогонов этих преобразователей на некотором потоке сигналов wc, где $w \in \mathcal{C}_0^*, c \in \mathcal{C}_0$. Тогда $[g'] = [g'']$.*

Доказательство. Применим индукцию по длине потока сигналов wc. Обоснование базиса индукции и индуктивного перехода проводится по одной и той же схеме.

Так как все состояния преобразователей π'_0 и π''_0 полезные, преобразователь π'_0 имеет финальный прогон $q'_2 \xrightarrow{u,f'}_* stop$. Значит, преобразователь π'_0 имеет полный прогон $start \xrightarrow{w,h'}_* q'_1 \xrightarrow{c,g'} q'_2 \xrightarrow{u,f'}_* stop$. Поскольку $\pi'_0 \sim_S \pi''_0$, преобразователь π''_0 также имеет полный прогон $start \xrightarrow{w,h''}_* q''_1 \xrightarrow{c,g''} q''_2 \xrightarrow{u,f''} stop$, и при этом

$[h'g'f'] = [h''g''f'']$. Последнее равенство означает, что $[h'g'] \wedge [h''g''] \neq \tau$. Следовательно, существует тройка элементов s, s', s'', для которых выполняются равенства $[h'g'] \wedge [h''g''] = [h'g'] \circ s' = [h''g''] \circ s''$ и $[h'g'f'] = [h''g''f''] = ([h'g'] \wedge [h''g'']) \circ s$. Таким образом, $[h'g'] \circ s' \circ s = [h'g'f'] = [h''g''f''] = [h''g''] \circ s'' \circ s$.

Если проводится обоснование базиса индукции, то $h' = h'' = \varepsilon$. Если же проводится обоснование индуктивного перехода, то по индуктивному предположению $[h'] = [h'']$. И в том, и в другом случае закон левого сокращения приводит к равенству $[g'] \circ s' = [g''] \circ s''$ и неравенствам $s' \preceq [f']$ и $s'' \preceq [f'']$.

Заметим, что элементы s' и s'' не зависят от f' и f'', и поэтому указанные неравенства верны для любых элементов f' из $S(\pi'_0, q'_2)$ и f'' из $S(\pi''_0, q''_2)$. Отсюда следуют неравенства $s' \preceq gcd(\pi'_0, q'_2)$ и $s'' \preceq gcd(\pi''_0, q''_2)$. Состояния управления q'_2 и q''_2 не являются начальными, и поэтому согласно определению приведенных преобразователей верны равенства $gcd(\pi'_0, q'_2) = gcd(\pi''_0, q''_2) = e$. Таким образом, $s' = s'' = e$, и, следовательно, $[g'] = [g'']$. □

Из утверждения 5 следует

Теорема 2. *Если полугруппа $(S, e, \circ)$ удовлетворяет требованиям **Req1, Req3**, то для любой пары приведенных преобразователей π' и π'' справедливо соотношение*

$$\pi' \sim_S \pi'' \iff \pi' \approx_S \pi'' .$$

Теоремы 1 и 2 дают способ решения задач минимизации и проверки эквивалентности для детерминированных автоматов-преобразователей, работающих над полугруппами, которые удовлетворяют требованиям **Req1**–**Req3**. Для этого достаточно редуцировать анализируемые преобразователи, воспользовавшись теоремой 1, а затем обратиться к соответствующим конечным автоматам-распознавателям и алгоритмам решения упомянутых задач для этих автоматов.

7. Заключение

В данной статье предложена общая схема решения задач минимизации конечных автоматов-преобразователей, работающих над полугруппами специального вида. Так как преобразователи над полугруппами служат формальными моделями последовательных реагирующих программ, эта схема предоставляет один из подходов к решению задач оптимизации и верификации таких программ. В этом состоит теоретическая значимость полученных в статье результатов.

Вместе с тем, остаются вопросы, требующие дальнейшего исследования. Мы не оценивали сложность алгоритмов минимизации, построенных на основе предложенной схемы. Это объясняется тем, что сложность этих алгоритмов зависит не только от сложности решения вычислительных задач, упомянутых в требованиях **Req1**–**Req3** (вычисление наибольших общих делителей, решение линейных уравнений), но также от выбора подходящих структур данных для представления элементов рассматриваемых полугрупп. Эти вопросы относятся, прежде всего, к области вычислительной алгебры.

Особого внимания заслуживает вопрос о необходимых условиях, которым должна удовлетворять полугруппа, для того чтобы задача минимизации автоматов-преобразователей имела единственное решение. Как видно из результатов работы [20],

свойство упорядоченности полугруппы, фигурирующее в **Req1**, к числу таковых условий не относится.

Список литературы / References

[1] Alur R., Cerny P., "Streaming transducers for algorithmic verification of single-pass list-processing programs", *Proc. of 38-th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, 2011, 599–610.

[2] Blattner M, Head T., "Single-valued a-transducers", *J. of Comput. and Syst. Sci.*, **15**:3 (1977), 310–327.

[3] Blattner M, Head T., "The decidability of equivalence for deterministic finite transducers", *J. of Comput. and Syst. Sci.*, **19**:1 (1979), 45–49.

[4] Beal M.-P., Carton O., "Computing the prefix of an automaton", *Theoretical Informatics and Applications*, **34**:6 (2000), 503–514.

[5] Culik K., Karhumaki J., "The equivalence of finite-valued transducers (on HDTOL languages) is decidable", *Theor. Comput. Sci.*, **47** (1986), 71–84.

[6] Diekert V., Metivier Y., "Partial commutation and traces", *Handbook of formal languages*, **3**, 1997, 457–533.

[7] Eisner J., "Simpler and more general minimization for weighted finite-state automata", *Proc. of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, **1**, 2003, 64–71.

[8] Friedman E.P., Greibach S.A., "A polynomial time algorithm for deciding the equivalence problem for 2-tape deterministic finite state acceptors", *SIAM J. Comput.*, **11**:1 (1982), 166–183.

[9] Griffiths T., "The unsolvability of the equivalence problem for ε-free nondeterministic generalized machines", *J. of the ACM*, **15** (1968), 409–413.

[10] Mohri M., "Finite-state transducers in language and speech processing", *Comput. Ling*, **23**:2 (1997), 269–311.

[11] Mohri M., "Minimization algorithms for sequential transducers", *Theor. Comput. Sci.*, **234** (2000), 177–201.

[12] Reutenauer C., Schuzenberger M. P., "Minimization of rational word functions", *SIAM J. of Comput.*, **20**:4 (1991), 669—685.

[13] Shofrutt C., "Minimizing subsequential transducers: a survey", *Theor. Comput. Sci.*, **292**:1 (2003), 131—143.

[14] Thakkar J., Kanade A., Alur R., "A transducer-based algorithmic verification of retransmission protocols over noisy channels", *Proc. of IFIP Joint International Conference on Formal Techniques for Distributed Systems, LNCS*, **7892** (2013), 209–224.

[15] Veanes M., Hooimeijer P., Livshits B., et al., "Symbolic finite state transducers: algorithms and applications", *Proc. of the 39th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages. ACM SIGPLAN Notices*, **147** (2012), 137–150.

[16] Watson B. W., "A taxonomy of finite automata minimization algorithm", *Computing Science Report. Eindhoven University of Technology*, **93/44** (2005), 1–32.

[17] Weber A., "Decomposing finite-valued transducers and deciding their equivalence", *SIAM Journal on Computing*, **22**:1 (1993), 175–202.

[18] Wolper P., Boigelot B., "Verifying systems with infinite but regular state spaces", *Proc. 10th Int. Conf. on Computer Aided Verification (CAV-1998), LNCS*, **1427** (1998), 88—97.

[19] Zakharov V. A., "Equivalence checking problem for finite state transducers over semigroups", *Proc. of the 6-th International Conference on Algebraic Informatics (CAI-2015). LNCS*, **9270** (2015), 208-221.

[20] Захаров В. А., Подымов В. В., “Применение алгоритмов проверки эквивалентности для оптимизации программ”, *Труды Института системного программирования*, **27**:3 (2015), 145–174; [Zakharov V. A., Podymov V. V., “Primenenije algoritmov proverki ekvivalentnisti dlya optimizacii program”, *Trudy insituta sistemnogo programmirovaniya*, **27**:3 (2015), 145–174, (in Russian).]

Zakharov V. A., Temerbekova G. G., "On the Minimization of Finite State Transducers over Semigroups", *Modeling and Analysis of Information Systems*, **23**:6 (2016), 741–753.

DOI: 10.18255/1818-1015-2016-6-741-753

Abstract. Finite state transducers over semigroups are regarded as a formal model of sequential reactive programs that operate in the interaction with the environment. At receiving a piece of data a program performs a sequence of actions and displays the current result. Such programs usually arise at implementation of computer drivers, on-line algorithms, control procedures. In many cases verification of such programs can be reduced to minimization and equivalence checking problems for finite state transducers. Minimization of a transducer over a semigroup is performed in three stages. At first the greatest common left-divisors are computed for all states of the transducer, next the transducer is brought to a reduced form by pulling all such divisors ”upstream”, and finally a minimization algorithm for finite state automata is applied to the reduced transducer.

Keywords: reactive system, transducer, semigroup, minimization, equivalence checking

About the authors:
Vladimir A. Zakharov, orcid.org/0000-0002-3794-9565, PhD, professor
Lomonosov Moscow State University,
Faculty of Computational Mathematics and Cybernetics, GSP-1, 1-52 Leninskiye Gory, Moscow 119991, Russia, e-mail: zakh@cs.msu.ru

Gulgaysha G. Temerbekova, orcid.org/0009-0203-2514-7755, graduate student,
Lomonosov Moscow State University,
Faculty of Computational Mathematics and Cybernetics, GSP-1, 1-52 Leninskiye Gory, Moscow 119991, Russia, e-mail: gulgaisha93@mail.ru

Acknowledgments:
This work is supported by the Basic Research Program at the National Research University Higher School of Economics in 2016 and by RFBR grants №16-01-00546.