Моделирование и анализ информационных систем. Т. 25, №5 (2018), с. 506–524
Modeling and Analysis of Information Systems. Vol. 25, No 5 (2018), pp. 506–524

Прикладные логики
Applied Logics

DOI: 10.18255/1818-1015-2018-5-506-524

УДК 517.9

О выразительных возможностях некоторых расширений линейной темпоральной логики

Гнатенко А. Р., Захаров В. А.

получена 10 сентября 2018

Аннотация. Конечные автоматы, задающие преобразования потоков входных сигналов в последовательности элементарных действий, являются простейшей моделью вычислений, пригодной для описания поведения реагирующих систем. Это поведение проявляется в соответствии между потоком входных сигналов и последовательностью элементарных действий, выполняемых системой. Мы полагаем, что для формального описания поведения реагирующих систем такого рода требуются более сложные и выразительные средства спецификации, нежели традиционная темпоральная логика линейного времени LTL. В этой работе рассматривается новый язык формальных спецификаций $\mathcal{LP}$-LTL, представляющий собой расширение темпоральной логики LTL, специально разработанное для описания свойств вычислений автоматов-преобразователей. Темпоральные операторы в формулах $\mathcal{LP}$-LTL снабжены параметрами, в качестве которых используются множества слов (языки), описывающие потоки сигналов управления, поступающих на вход реагирующей системы. Базовые предикаты в формулах $\mathcal{LP}$-LTL также определяются языками в алфавите элементарных действий; эти языки описывают ожидаемую реакцию системы в ответ на воздействия окружающей среды. В более ранних работах авторов этой статьи изучалась задача верификации конечных автоматов-преобразователей относительно спецификаций, представленных формулами логик $\mathcal{LP}$-LTL и $\mathcal{LP}$-CTL. Было показано, что для обеих логик эта задача алгоритмически разрешима. Цель данной работы — оценить выразительные возможности логики $\mathcal{LP}$-LTL и сравнить ее с широко известными логиками, применяющимися в информатике для спецификации реагирующих систем. В логике $\mathcal{LP}$-LTL были выделены два фрагмента $\mathcal{LP}$-1-LTL и $\mathcal{LP}$-n-LTL. Нам удалось установить, что язык спецификаций $\mathcal{LP}$-1-LTL более выразителен, чем LTL, в то время как фрагмент $\mathcal{LP}$-n-LTL обладает точно такими же выразительными возможностями, что и монадическая логика второго порядка S1S.

Ключевые слова: темпоральные логики, выразительность, спецификация, верификация, автоматы Бюхи, бесконечные слова

Для цитирования: Гнатенко А. Р., Захаров В. А., "О выразительных возможностях некоторых расширений линейной темпоральной логики", *Моделирование и анализ информационных систем*, **25**:5 (2018), 506–524.

Об авторах:
Гнатенко Антон Романович, orcid.org/0000-0003-1499-2090, студент,
Московский государственный университет им. М.В. Ломоносова,
Ленинские горы, 1, г. Москва, 119991 Россия, e-mail: gnatenko.cmc@gmail.com

Захаров Владимир Анатольевич, orcid.org/0000-0002-3794-9565, доктор физ.-мат. наук, профессор,
Национальный исследовательский университет «Высшая школа экономики»,
ул. Мясницкая, 20, г. Москва, 101000 Россия, e-mail: zakh@cs.msu.ru

Благодарности:
Работа выполнена при финансовой поддержке гранта РФФИ N 18-01-00854

1. Введение

Автоматы-преобразователи являются одной из самых ранних моделей вычислений; они находят применение в информатике, программировании, микроэлектронике, лингвистике. В этой статье мы будем использовать их в качестве формальной модели реагирующих информационных систем. На каждом шаге вычисления автомат-преобразователь получает на входе один из символов входного алфавита и выдает на выходе последовательность символов (слово) выходного алфавита. Буквы входного алфавита соответствуют сигналам управления, поступающим из внешней среды, а буквы выходного алфавита — элементарным действиям, выполняемым реагирующей системой. На множестве элементарных действий, выполняемых автоматом-преобразователем, можно ввести операцию композиции, сформировав, таким образом, полугруппу действий автомата. Поведение системы в этом случае характеризуется отношением преобразования потока управляющих сигналов в действия, выполняемые системой. Модель последовательных реагирующих систем такого вида была предложена и исследована в статьях [6, 21–23].

Но для проверки правильности поведения реагирующих систем, моделируемых конечными автоматами-преобразователями над полугруппами, необходим адекватный этой модели язык формальных спецификаций. Предлагаемый нами подход к построению такого языка основывается на следующих соображениях. Типичное требование правильного поведения реагирующей системы состоит в том, что для каждого входного слова (потока сигналов управления), подпадающего под некоторый заданный шаблон, вырабатывается реакция, которая также подпадает под некоторый заданный шаблон. Для выражения таких требований понадобятся не только темпоральные операторы, при помощи которых можно описывать порядок следования событий (управляющих сигналов и действий), но также и средства для описания и учета указанных шаблонов. Для задания шаблонов потоков управляющих сигналов годятся любые способы описания формальных языков — автоматы, формальные грамматики, регулярные выражения и пр. Для описания реакции автомата-преобразователя, работающего над некоторой заданной полугруппой элементарных действий, можно использовать любые предикаты, определенные в этой полугруппе. В случае свободной полугруппы такие предикаты фактически представляют собой множества слов (языки) над алфавитом элементарных действий. Для их задания также годятся любые средства задания формальных языков. В этом случае свойства поведения реагирующих систем можно описывать, например, при помощи темпоральных формул вида $\mathbf{G}_L P$, означающих, что для каждого входного слова w, принадлежащего языку L, выходное слово h, которое вырабатывает автомат-преобразователь в качестве реакции, принадлежит языку P.

Такой подход обладает двумя достоинствами. Во-первых, он позволяет явно выражать соответствия между входными и выходными словами автоматов-преобразователей и описывать тем самым типичные требования правильного поведения реагирующих систем. И, кроме того, он позволяет адаптировать методы верификации моделей программ, пригодные для традиционных темпоральных логик (см. [2]). В статье [11] этот замысел был осуществлен для языка спецификаций $\mathcal{LP}$-LTL, основанного на темпоральной логике LTL, а затем в статье [7] он был распространен на язык спецификаций $\mathcal{LP}$-CTL, расширяющий логику деревьев вычислений CTL.

Цель данной статьи — оценить выразительные возможности новой темпоральной логики $\mathcal{LP}$-LTL, сравнив ее с другими прикладными логиками, используемыми для верификации программ. Проведение такого сравнения осложняется тем, что семантика $\mathcal{LP}$-LTL и семантика наиболее известных темпоральных логик определяются на разных структурах. Чтобы преодолеть эту трудность, мы выделили в логике $\mathcal{LP}$-LTL два фрагмента $\mathcal{LP}$-1-LTL и $\mathcal{LP}$-n-LTL, семантики которых можно адаптировать к бесконечным словам (сверхсловам), и за счет этого сделать возможным сравнение этих фрагментов, темпоральной логикой линейного времени LTL и монадической логикой второго порядка S1S. В результате удалось обнаружить, что $\mathcal{LP}$-1-LTL является строго более выразительной, нежели LTL, а $\mathcal{LP}$-n-LTL и S1S обладают одинаковыми выразительными способностями.

Статья устроена следующим образом. В разделе 2 определена модель конечного последовательного автомата-преобразователя. В следующем разделе описаны синтаксис и семантика темпоральной логики $\mathcal{LP}$-LTL. В разделе 4 выделены фрагменты $\mathcal{LP}$-1-LTL и $\mathcal{LP}$-n-LTL логики $\mathcal{LP}$-LTL, семантику которых можно определить на сверхсловах. В следующих двух разделах представлены основные результаты этой статьи. В заключении проведено краткое сопоставление введенного нами расширения логики LTL с другими расширениями той же логики и намечены направления дальнейших исследований свойств темпоральной логики $\mathcal{LP}$-LTL.

2. Моделирование реагирующих систем автоматами-преобразователями

Реагирующая система производит вычисление, получая входные сигналы от окружающей среды и выполняя в ответ на них определенные элементарные действия. Рассмотрим конечное множество $\mathcal{C}$ *входных сигналов* и конечное множество $\mathcal{A}$ *элементарных действий*. Конечные слова в алфавите $\mathcal{C}$ мы будем называть *потоками сигналов*, конечные слова в алфавите $\mathcal{A}$ — *составными действиями*, которые можно рассматривать как последовательные композиции элементарных действий. Множества всевозможных потоков сигналов и всевозможных составных действий обозначим записями $\mathcal{C}^*$ и $\mathcal{A}^*$ соответственно. Мы используем стандартные обозначения, принятые в теории формальных языков: запись uv обозначает конкатенацию слов u и v, а знак ε — пустое слово.

Определение 1. Конечным автоматом-преобразователем *над алфавитами* $\mathcal{C}$ *и* $\mathcal{A}$ *называется пятерка* $\Pi = (Q, \mathcal{C}, \mathcal{A}, q_{init}, T)$*, где 1)* Q *— это множество управляющих состояний, 2)* $q_{init} \in Q$ *— начальное состояние, 3)* $T \subseteq Q \times \mathcal{C} \times Q \times \mathcal{A}^*$ *— отношение переходов.*

Мы будем полагать, что отношение переходов T является тотальным, т. е. для каждого состояния q и сигнала управления c существуют такие состояние q' и действие h, для которых выполняется отношение переходов $(q, c, q', h) \in T$. Четвёрка $(q', c, q'', h) \in T$ называется *переходом*: пребывая в состоянии q' и получив сигнал c, автомат переходит в состояние q'' и выполняет составное действие h. Для наглядности будем обозначать переход записью $q' \xrightarrow{c,h} q''$.

Для описания поведения автомата-преобразователя нам понадобятся понятия прогона и траектории. *Прогон* автомата Π — это бесконечная последовательность переходов: $\rho = q_1 \xrightarrow{c_1, h_1} q_2 \xrightarrow{c_2, h_2} \cdots \xrightarrow{c_n, h_n} q_{n+1}, \ldots$ Прогон, начинающийся в состоянии q_{init}, называется *основным.*

Определение 2. *Пусть s_0 — некоторое составное действие, а ρ — указанный выше прогон автомата Π. Тогда* траекторией *автомата Π на прогоне ρ называется пара $tr = (s_0, \alpha)$, где последовательность $\alpha = (c_1, s_1), (c_2, s_2), \ldots, (c_i, s_i), \ldots$, удовлетворяет условию $s_i = s_{i-1}h_i$ для каждого натурального i.*

Траектория представляет собой возможную реакцию системы, ранее выполнившей действие s_0, в ответ на поток входных сигналов $w = c_1c_2 \ldots c_i \ldots$. В случае, если прогон ρ является основным прогоном автомата, а $s_0 = \varepsilon$, то соответствующая траектория также называется *основной.* Множество всех основных траекторий автомата Π будем обозначать с помощью $Tr(\Pi)$. Под записью $tr|^i$ понимается траектория, представляющая собой «хвост» траектории tr, т. е. траектория $(s_i, \alpha|^i)$, где $\alpha|^i = (c_{i+1}, s_{i+1}), (c_{i+2}, s_{i+2}), \ldots$

3. Язык спецификаций $\mathcal{LP}$-LTL

Для проверки правильности поведения реагирующих систем необходимо иметь формальный язык спецификаций, на котором записываются требования правильности. Поведение реагирующей системы представляет собой процесс, протекающий во времени и проявляющийся в соответствии между потоками сигналов управления, поступающими на вход системы, и действиями, выполняемыми ею. Для спецификации поведения такого рода обычно используются те или иные разновидности темпоральных логик. При выборе подходящей темпоральной логики для формального описания поведения автомата-преобразователя нужно иметь в виду следующие две отличительные особенности предлагаемой модели реагирующих систем:

1. результат работы автомата-преобразователя — это последовательность выполненных им действий из множества $\mathcal{A}$; поэтому атомарные формулы (базовые предикаты) должны интерпретироваться на множестве $\mathcal{A}^*$,

2. поведение автомата зависит не только от течения времени, но также и от потока входных управляющих сигналов; поэтому темпоральные операторы следует *параметризовать*, то есть снабдить описанием тех потоков управляющих сигналов, на которых проверяется поведение автомата.

Чтобы придать языку спецификаций поведения реагирующих систем указанные свойства, авторы [11] предложили новое расширение темпоральной логики линейного времени LTL. В основу этого расширения положен следующий замысел. Проверку правильности реакций автомата-преобразователя необходимо уметь проводить на потоках сигналов управления. Для стороннего наблюдателя поведение автомата на заданном потоке сигналов управления полностью определяется траекториями прогонов автомата на этом потоке. Событиями, доступными наблюдению, являются действия, выполняемые автоматом по ходу прогонов. Правильность поведения автомата проявляется в том, что эти события происходят в некотором желаемом порядке. Для описания порядка осуществления событий можно привлечь формулы LTL, в которых те или иные типы событий выступают в роли элементарных высказываний (базовых предикатов). При этом множества потоков сигналов управления, на которых проверяется требование правильности поведения автомата, указываются в качестве параметров темпоральных операторов. Таким образом, образуется параметризованное расширение LTL, в котором отношение выполнимости формул определяется на траекториях автоматов-преобразователей. Более подробное описание этого расширения таково.

Любое множество потоков сигналов управления будем рассматривать как язык в алфавите $\mathcal{C}$. Выделим некоторое отдельное семейство языков $\mathcal{L}$ в алфавите $\mathcal{C}$ (например, регулярные языки) и будем называть всякий язык из этого семейства *шаблоном поведения окружения.* Шаблоны поведения призваны описывать допустимые воздействия внешней среды на реагирующую систему.

Составные действия, выполняемые системой в ответ на сигналы управления, также являются словами в алфавите $\mathcal{A}$. Выделим некоторый класс языков $\mathcal{P}$ в алфавите $\mathcal{A}$ и будем называть всякий язык из этого класса *базовым предикатом.* Каждый базовый предикат можно рассматривать как некоторый тип наблюдаемых событий-откликов реагирующей системы в ответ на воздействие внешнего окружения. При использовании шаблонов поведения окружения и базовых предикатов в логических формулах мы предполагаем, что соответствующие языки определены каким-либо конструктивным образом (например, с помощью автоматов, грамматик, машин Тьюринга, и т. п.).

Определение 3. *Пусть задано некоторое семейство $\mathcal{L}$ шаблонов поведения окружения и класс $\mathcal{P}$ базовых предикатов, допускающих конструктивное описание. Множество формул логики $\mathcal{LP}$-LTL — это наименьшее множество выражений, которые могут быть построены по следующим правилам:*

1. *каждый базовый предикат $P \in \mathcal{P}$ является формулой;*
2. *если φ_1, φ_2 — формулы, то выражения $\neg\varphi_1$ и $\varphi_1 \wedge \varphi_2$ являются формулами;*
3. *если φ_1 и φ_2 — формулы, $c \in \mathcal{C}$, и $L \in \mathcal{L}$, то выражения $\mathbf{X}_c\varphi_1$, $\mathbf{Y}_c\varphi_1$, $\mathbf{F}_L\varphi_1$, $\mathbf{G}_L\varphi_1$ и $\varphi_1\,\mathbf{U}_L\varphi_2$ являются формулами.*

Семантика $\mathcal{LP}$-LTL определяется на основе отношения выполнимости формул на интерпретациях, в роли которых выступают траектории автоматов-преобразователей. Предположим, что имеется некоторый автомат Π и его траектория $tr = (s_0, \alpha)$, где $\alpha = (c_1, s_1), (c_2, s_2), \ldots, (c_i, s_i), \ldots$ Тогда для любой формулы ψ запись $tr \models \psi$ будет означать, что формула ψ выполняется на траектории tr автомата Π.

Определение 4. *Отношение выполнимости $\models$ в логике $\mathcal{LP}$-LTL определим индукцией по глубине формулы:*

1. *если P — базовый предикат, то $tr \models P \Longleftrightarrow s_0 \in P$;*
2. *$tr \models \neg\varphi \Longleftrightarrow$ неверно, что $tr \models \varphi$;*
3. *$tr \models \varphi_1 \wedge \varphi_2 \Longleftrightarrow tr \models \varphi_1$ и $tr \models \varphi_2$;*

4. $tr \models \mathbf{X}_c\varphi \iff c = c_1$ *и* $tr|^1 \models \varphi$;

5. $tr \models \mathbf{Y}_c\varphi \iff c \neq c_1$ *или* $tr|^1 \models \varphi$;

6. $tr \models \mathbf{F}_L\varphi \iff \exists i \geqslant 0\colon c_1c_2\dots c_i \in L$ *и* $tr|^i \models \varphi$;

7. $tr \models \mathbf{G}_L\varphi \iff \forall i \geqslant 0\colon$ *если* $c_1c_2\dots c_i \in L$, *то* $tr|^i \models \varphi$;

8. $tr \models \varphi\mathbf{U}_L\psi \iff \exists i \geqslant 0\colon c_1c_2\dots c_i \in L$, *такое, что* $tr|^i \models \psi$ *и* $\forall j,\ 0 \leqslant j < i$, *если* $c_1c_2\dots c_i \in L$, *то* $tr|^j \models \varphi$.

Впервые логика $\mathcal{LP}$-LTL была введена в статье [11]. В отличие от LTL, в этой логике возникает необходимость использовать два разных оператора обращения к следующему моменту времени (NextTime) $\mathbf{X}_c$ и $\mathbf{Y}_c$ в зависимости от того, насколько обязательным является появление в траектории в текущий момент времени управляющего сигнала c, используемого в качестве параметра. Операторы $\mathbf{X}_c$ и $\mathbf{Y}_c$ двойственны друг другу. Двойственными являются также и операторы $\mathbf{F}_L$ и $\mathbf{G}_L$. С помощью отрицания $\neg$ и конъюнкции $\wedge$ можно выразить другие булевы связки, такие как $\vee, \rightarrow, \equiv$. Подобно тому, как это было сделано в приведенном выше определении для оператора $\mathbf{U}$ (Until), в логике $\mathcal{LP}$-LTL можно ввести параметризованные аналоги других темпоральных операторов LTL, например $\mathbf{R}$ (Release) или $\mathbf{W}$ (Weak Until). В статье [11] показано, что многие полезные соотношения равносильности между формулами LTL, наподобие тождеств неподвижной точки, остаются справедливыми и для параметризованных аналогов этих тождеств.

В логике $\mathcal{LP}$-LTL задача верификации автоматов-преобразователей имеет следующую постановку: для произвольного заданного автомата-преобразователя Π и формулы φ проверить, выполняется ли отношение $tr \models \varphi$ для каждой основной траектории tr из множества $Tr(\Pi)$. В статье [11] показано, что эта задача разрешима за дважды экспоненциальное время в том случае, когда семейства $\mathcal{L}$ и $\mathcal{P}$ являются классами регулярных языков и при этом шаблоны поведения окружения и базовые предикаты в формулах $\mathcal{LP}$-LTL описываются с помощью детерминированных конечных автоматов-распознавателей.

В статье [7] идея параметризации темпоральных операторов шаблонами поведения окружения была распространена на логику деревьев вычислений CTL. В этой статье были определены синтаксис и семантика логики $\mathcal{LP}$-CTL и показано, что задача верификации автоматов-преобразователей относительно спецификаций их поведения, представленных формулами $\mathcal{LP}$-CTL над регулярными семействами языков $\mathcal{L}$ и $\mathcal{P}$, разрешима за экспоненциальное время.

Можно предложить и более изощренную интерпретацию действий, выполняемых автоматами-преобразователя, нежели простое формирование выходных слов. Например, эти действия можно рассматривать как элементы некоторой полугруппы или группы с разрешимой проблемой тождеств (чтобы можно было эффективно сравнивать результаты выполнения действий). В этом случае базовые предикаты можно задавать специальными алгебраическими средствами, характерными для того или иного вида полугрупп. Задача верификации автоматов-преобразователей может тогда существенно усложниться: в статье [6] установлено, например, что задача верификации автомата, работающего над свободной коммутативной полугруппой, алгоритмически неразрешима.

4. Два фрагмента логики $\mathcal{LP}$-LTL

Существует немало логик, которые используются для описания поведения вычислительных систем. В этих логиках каждая формула является описанием множества всех тех вычислений-интерпретаций, на которых она выполняется. Таким образом, возникает возможность сравнивать выразительные способности различных логик в зависимости от того, какие множества вычислений могут быть описаны формулами этих логик. Более строго отношение сравнения выразительных возможностей логик определяется следующим образом.

Определение 5. *Пусть имеются две логики $\mathscr{L}_1$ и $\mathscr{L}_2$, у которых отношение выполнимости формул определяется в одном и том же классе интерпретаций. Будем говорить, что формула ψ_1 одной из этих логик* эквивалентна *формуле ψ_2 какой-либо из этих логик, если соотношение $I \models \psi_1 \Leftrightarrow I \models \psi_2$ верно для каждой интерпретации I. Логика $\mathscr{L}_1$ считается* не менее выразительной*, чем логика $\mathscr{L}_2$ (обозначается $\mathscr{L}_2 \preccurlyeq \mathscr{L}_1$), если для любой формулы логики $\mathscr{L}_2$ существует эквивалентная ей формула логики $\mathscr{L}_2$. Логики $\mathscr{L}_1$ и $\mathscr{L}_2$ имеют* равные выразительные возможности *(обозначается $\mathscr{L}_2 \equiv \mathscr{L}_1$), если верны соотношения $\mathscr{L}_2 \preccurlyeq \mathscr{L}_1$ и $\mathscr{L}_1 \preccurlyeq \mathscr{L}_2$. Логика $\mathscr{L}_1$ считается* более выразительной*, чем логика $\mathscr{L}_2$ (обозначается $\mathscr{L}_2 \prec \mathscr{L}_1$), если $\mathscr{L}_2 \preccurlyeq \mathscr{L}_1$, но при этом и $\mathscr{L}_2 \not\equiv \mathscr{L}_1$.*

Размеченные системы переходов широко используются в качестве моделей реагирующих систем. Наблюдаемое поведение таких систем представляется бесконечной последовательностью наблюдаемых событий. На множестве таких последовательностей можно определить семантику трех логик — теории линейного порядка $(\mathbb{N}, <)$, темпоральной логики линейного времени LTL и монадической логики второго порядка с одной функцией следования S1S. В работе [10] было установлено, что логики $(\mathbb{N}, <)$ и LTL имеют равные выразительные возможности, а в статьях [17, 19] было показано, что логика S1S является более выразительной, чем LTL.

Автоматы-преобразователи также являются моделями реагирующих систем, а их поведение можно описывать посредством нового расширения логики линейного времени $\mathcal{LP}$-LTL. Цель данной статьи — сравнить выразительные возможности трех логик, применяемых для описания поведения реагирующих систем — LTL, S1S и $\mathcal{LP}$-LTL. Однако отношение выполнимости для формул $\mathcal{LP}$-LTL определяется на интерпретациях, которые отличаются от тех, что используются в определениях семантики логик LTL и S1S. Чтобы сравнивать $\mathcal{LP}$-LTL с LTL и S1S, мы выделим два фрагмента $\mathcal{LP}$-LTL, семантику которых можно равносильным образом определить на бесконечных последовательностях событий, и сравним выразительные возможности этих фрагментов с описательными способностями логик LTL и S1S.

Формулы первого из этих фрагментов $\mathcal{LP}$-1-LTL строятся над однобуквенным алфавитом $\mathcal{C} = \{c\}$ входных сигналов и произвольным конечным алфавитом элементарных действий $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$. В качестве семейства $\mathcal{L}$ шаблонов поведения окружения в формулах этого фрагмента разрешается использовать любые регулярные языки над алфавитом $\{c\}$, а в качестве семейства $\mathcal{P}$ базовых предикатов будем использовать набор из n регулярных языков вида $P_i = \mathcal{A}^* a_i$. Будем говорить, что сверхслово $w = a_{i_1} a_{i_2} \ldots a_{i_m} \ldots$ удовлетворяет формуле φ' из фрагмента $\mathcal{LP}$-1-LTL тогда и только тогда, когда для траектории $tr' = (\varepsilon, \alpha'_w)$, где $\alpha'_w = (c, a_{i_1}), (c, a_{i_1} a_{i_2}), \ldots, (c, a_{i_1} a_{i_2} \ldots a_{i_m}), \ldots$, верно, что $tr' \models \varphi'$.

Фрагмент $\mathcal{LP}$-n-LTL определяется аналогично, с той лишь разницей, что алфавиты $\mathcal{C}$ и $\mathcal{A}$ совпадают, т. е. $\mathcal{C} = \mathcal{A} = \{a_1, a_2, \dots, a_n\}$. Сверхслово $w = a_{i_1}a_{i_2}\dots a_{i_m}\dots$ удовлетворяет формуле φ'' из фрагмента $\mathcal{LP}$-n-LTL тогда и только тогда, когда для траектории $tr'' = (\varepsilon, \alpha''_w)$, где $\alpha''_w = (a_{i_1}, a_{i_1}), (a_{i_2}, a_{i_1}a_{i_2}), \dots, (a_{i_m}, a_{i_1}a_{i_2}\dots a_{i_m}), \dots,$ верно, что $tr'' \models \varphi''$.

Чтобы сравнивать выразительные возможности логик LTL, S1S и двух выделенных фрагментов логики $\mathcal{LP}$-LTL, приведем альтернативное определение семантики формул из рассмотренных фрагментов на сверхсловах. Пусть $\Sigma = \{a_1, a_2, \dots, a_n\}$ — некоторый конечный алфавит. *Сверхсловом* будем называть всякое отображение $w\colon \mathbb{N} \to \Sigma$, где $\mathbb{N}$ обозначает множество неотрицательных целых чисел. Множество всех сверхслов обозначим записью Σ^ω. Запись $w|^k$ будет обозначать суффикс слова w, начинающийся с буквы с номером k, т. е. бесконечного слова v, для которого $v(i) = w(i+k)$ при любом $i,\ i \geqslant 0$. Запись $w[0\dots k]$ будет обозначать префикс w, т. е. конечное слово $w(0)w(1)\dots w(k)$. В случае, когда L является регулярным языком над однобуквенным алфавитом $\{c\}$, будем вместо $c^i \in L$ писать кратко $i \in L$.

Переформулируем определение семантики формул фрагмента $\mathcal{LP}$-1-LTL как множества формул, построенных из букв алфавита Σ (используемых вместо базовых предикатов) с помощью булевых связок $\neg, \wedge$ и темпоральных операторов $\mathbf{X}$, $\mathbf{F}_L$, $\mathbf{G}_L$ и $\mathbf{U}_L$, параметризованных регулярными языками L над однобуквенным алфавитом $\mathcal{C} = \{c\}$.

Определение 6. *Отношение выполнимости $\models$ формул $\mathcal{LP}$-1-LTL на множестве сверхслов определяется следующими правилами:*

1. *для атомарной формулы $a \in \Sigma$: $w \models a \iff w(0) = a$;*
2. *$w \models \neg\varphi \iff$ неверно, что $w \models \varphi$;*
3. *$w \models \varphi \wedge \psi \iff w \models \varphi$ и $w \models \psi$;*
4. *$w \models \mathbf{X}\varphi \iff w|^1 \models \varphi$;*
5. *$w \models \mathbf{F}_L\varphi \iff \exists\, i \geqslant 0$, такое, что $i \in L$ и $w|^i \models \varphi$;*
6. *$w \models \mathbf{G}_L\varphi \iff \forall\, i \geqslant 0$, если $i \in L$, то $w|^i \models \varphi$;*
7. *$w \models \varphi\,\mathbf{U}_L\psi \iff \exists\, i \geqslant 0$, такое, что $i \in L$ и $w|^i \models \psi$, и $\forall\, j,\ 0 \leqslant j < i$, если $j \in L$, то $w|^j \ \models\ \varphi$.*

Примером формулы фрагмента $\mathcal{LP}$-1-LTL может служить выражение $\mathbf{G}_{(cc)^*}a$. Как можно понять из определения, сверхслово w удовлетворяет данной формуле тогда и только тогда, когда для любого $i,\ i \geqslant 0$, верно $w(2i) = a$.

Синтаксис $\mathcal{LP}$-n-LTL отличается от случая $\mathcal{LP}$-1-LTL в трех аспектах:

1. помимо оператора $\mathbf{X}$ ("NeXttime", в следующий момент времени) появляется двойственный ему оператор $\mathbf{Y}$;
2. операторы $\mathbf{X}$ и $\mathbf{Y}$ параметризованы буквами алфавита Σ: $\mathbf{X}_a,\ \mathbf{Y}_b$;
3. операторы $\mathbf{F}_L$, $\mathbf{G}_L$ и $\mathbf{U}_L$ параметризованы регулярными языками над алфавитом Σ.

Определение 7. *Отношение выполнимости* $\models$ *формул* $\mathcal{LP}$-n-LTL *на сверхсловах определяется следующим образом:*

1. *выполнимость атомарных формул, отрицания* $\neg$ *и конъюнкции* $\wedge$ *определяются точно так же, как в случае* $\mathcal{LP}$-1-LTL*;*
2. $w \models \mathbf{X}_c\varphi \Longleftrightarrow w(0) = c$ *и* $w|^1 \models \varphi$*;*
3. $w \models \mathbf{Y}_c\varphi \Longleftrightarrow w(0) \neq c$ *или* $w|^1 \models \varphi$*;*
4. $w \models \mathbf{F}_L\varphi \Longleftrightarrow \exists\, i \geqslant 0$, *такое, что* $w[0\dots i] \in L$ *и* $w|^i \models \varphi$*;*
5. $w \models \mathbf{G}_L\varphi \Longleftrightarrow \forall\, i \geqslant 0$, *если* $w[0\dots i] \in L$, *то* $w|^i \models \varphi$*;*
6. $w \models \varphi\,\mathbf{U}_L\psi \Longleftrightarrow \exists\, i \geqslant 0$, *такое, что* $w[0\dots i] \in L$ *и* $w|^i \models \psi$, *и* $\forall\, j,\ 0 \leqslant j < i$, *если* $w[0\dots j] \in L$, *то* $w|^j \models \varphi$.

5. $\mathcal{LP}$-1-LTL vs LTL

Легко заметить, что темпоральная логика LTL является подмножеством $\mathcal{LP}$-1-LTL, в котором в качестве параметра всех темпоральных операторов используется регулярный язык $\mathcal{C}^*$. Отсюда следует, что $LTL \preccurlyeq \mathcal{LP}$-1-$LTL$. Более того, справедливо следующее более сильное утверждение.

Теорема 1. $LTL \prec \mathcal{LP}$-1-LTL

В статье [19] показано, что формулами логики LTL невозможно описать множество бесконечных слов, имеющих заданную букву во всех позициях, номер которых кратен некоторому числу $k, k \geq 2$. Мы приводим здесь другое доказательство этого результата с помощью *игры Эренфойхта–Фреше* (см. [3]).

Всюду далее в формулах LTL вместо оператора $\mathbf{G}$ будем использовать эквивалентную комбинацию $\neg\,\mathbf{F}\,\neg$. Техника *игр Эренфойхта–Фреше* для темпоральных логик была введена авторами статьи [3] для построения иерархии формул LTL. Для описания этой техники нам понадобится понятие *глубины вложенности операторов* формулы.

Определение 8. *Определим* глубину вложенности операторов $depth(\varphi)$ *формулы* $\varphi \in LTL$ *индукцией по ее строению:*

1. *для атомарной формулы* $a \in \Sigma$ *считается, что* $depth(a) = 0$*;*
2. $depth(\neg\varphi) = depth(\varphi)$*;*
3. $depth(\varphi \wedge \psi) = \max(depth(\varphi), depth(\psi))$*;*
4. $depth(\mathbf{X}\varphi) = depth(\varphi) + 1$*;*
5. $depth(\mathbf{F}\varphi) = depth(\varphi) + 1$*;*
6. $depth(\varphi\mathbf{U}\psi) = \max(depth(\varphi), depth(\psi)) + 1$.

В игре Эренфойхта–Фреше принимают участие два игрока: Игрок 1 и Игрок 2; один из них в результате выигрывает, а другой проигрывает. Игра ведется на паре сверхслов (w_0, w_1), которая называется *конфигурацией* игры. Игрок 1 стремится показать, что сверхслова w_0 и w_1 *различимы* некоторой формулой LTL, в то время как Игрок 2 пытается доказать обратное.

Определение 9. *Игра Эренфойхта–Фреше, состоящая из* k раундов *(*k*-игра) с начальной конфигурацией* (w_0, w_1)*, определяется индукцией по* k*:*

- *Игрок 1 выигрывает в игру из 0 раундов, если* $w_0(0) \neq w_1(0)$*. В противном случае выигрывает Игрок 2.*
- *Игрок 2 выигрывает в игру из* $(k+1)$ *раундов, если* $w_0(0) \neq w_1(0)$*. В противном случае разыгрывается раунд, что приводит либо к победе Игрока 1, либо к новой конфигурации* (w'_0, w'_1)*, на которой разыгрывается игра из* k *раундов.*

В каждом *раунде* проводится изменение конфигураций. Игрок 1 выбирает один из трех возможных *ходов*:

- **X-ход.** В качестве новой конфигурации устанавливает пару $(w_0|^1, w_1|^1)$;
- **F-ход.**
  1. Игрок 1 выбирает слово w_s, $s \in \{0, 1\}$, и позицию $i_s \geqslant 0$;
  2. Игрок 2 отвечает, выбирая позицию $i_{1-s} \geqslant 0$ в слове w_{1-s};
  3. В качестве новой конфигурации устанавливается пара $(w_s|^{i_s}, w_{1-s}|^{i_{1-s}})$.
- **U-ход.**
  1. Игрок 1 выбирает w_s, $s \in \{0, 1\}$, и позицию $i \geqslant 0$;
  2. Игрок 2 отвечает, выбирая такую позицию $i_{1-s} \geqslant 0$ в слове w_{1-s}, что если $i_s = 0$, то $i_{1-s} = 0$;
  3. Игрок 1 выбирает один из двух вариантов развития событий:
     (a) В качестве новой конфигурации устанавливает пару $(w_s|^{i_s}, w_{1-s}|^{i_{1-s}})$;
     (b) Игрок 1 выбирает позицию i'_{1-s} в слове w_{1-s}, где $0 \leqslant i'_{1-s} < i_{1-s}$, и Игрок 2 отвечает, выбирая позицию $i'_s \in w_s$, $0 \leqslant i'_s < i_s$. В качестве новой конфигурации устанавливается пара $(w_s|^{i'_s}, w_{1-s}|^{i'_{1-s}})$.

Говорят, что у Игрока 2 есть *выигрышная стратегия* в игре, состоящей из k раундов, с начальной конфигурацией (w_0, w_1), если он может выиграть в этой игре независимо от ходов Игрока 1. Этот факт обозначается записью $w_0 \sim_k w_1$. В противном случае говорят, что выигрышную стратегию имеет Игрок 1.

В [3] доказано важное свойство игр Эренфойхта–Фреше:

Утверждение 1. *Для любого целого неотрицательного* k *и любых двух сверхслов* w_0 *и* w_1 *отношение* $w_0 \sim_k w_1$ *верно в том и только в том случае, если для любой формулы* $\varphi \in LTL$ *глубины* $depth(\varphi) \leqslant k$ *верно, что* $w_0 \models \varphi \iff w_1 \models \varphi$.

Теперь можно приступить к доказательству теоремы 1. Покажем, что свойство $W_a^{2n} = \{w \in \Sigma^\omega \mid w(2i) = a,\ i \in \mathbb{N}_0\}$ сверхслов в алфавите $\Sigma = \{a, b\}$, описываемое $\mathcal{LP}$-1-LTL формулой $\mathbf{G}_{(cc)^*}a$, невыразимо формулами логики LTL.

Доказательство. Рассмотрим пару слов $u_n = a^{2n+1}ba^\omega \in W_a^{2n}$ и $v_n = a^{2n}ba^\omega \notin W_a^{2n}$. Убедимся, используя технику игр Эренфойхта–Фреше, что при $2n > k$ формулы LTL не различают u_n и v_n. Для удобства префиксы слов u_n и v_n до буквы b будем называть их *началами*, а после буквы b — *хвостами.* В основе предлагаемого строгого доказательства лежит идея, заключающаяся в том, что Игроку 1 не удастся различить блоки a^{2n+1} и a^{2n} за k раундов, и поэтому он не сможет установить разницу между u_n и v_n.

Для строгого обоснования опишем выигрышную стратегию Игрока 2:

1. Если Игрок 1 делает **X**-ход, то Игрок 2 должен играть в соответствии с правилами (происходит "сдвиг" конфигурации на одну букву, в то время как число раундов уменьшается на единицу);

2. Если Игрок 1 делает **F**-ход

   (a) в *хвост*, то Игрок 2 должен тоже сделать ход в *хвост* на ту же самую позицию (считая от первой буквы *хвоста*);

   (b) в *начало*, то Игрок 2 должен сделать ход в *начало*, так, чтобы расстояние от новой позиции до последней буквы *начала* было таким же, как в слове Игрока 1.

   (c) в *позицию буквы* b, то Игрок 2 также должен ходить в *позицию буквы* b;

3. Если Игрок 1 делает **U**-ход, то Игроку 2 следует играть аналогично случаю **F**-хода, за исключением ситуации, когда Игрок 1 делает ход в первую букву *начала* (т. е. остается в предыдущей позиции); тогда Игрок 2 должен играть согласно правилам и тоже сделать ход в первую букву *начала*.

   Аналогично, если затем Игрок 1 сделает "ход назад" (применив третий пункт описания **U**-хода), то Игроку 2 следует отвечать на его действия, как описано выше, за исключением случая, когда Игрок 1 делает ход в первую букву *начала* (тогда Игроку 2 следует также делать ход в первую букву *начала*).

Легко заметить, что справедливо следующее

Утверждение 2. *При использовании Игроком 2 описанной выше стратегии конфигурация (u, v) на всем протяжении игры удовлетворяет следующим свойствам:*

1. $u[0] = v[0]$*;*

2. *если $u \neq v$, то длины кратчайших блоков букв a в словах u и v не меньше, чем оставшееся количество раундов;*

3. *между второй и третьей частями **U**-хода для любой позиции j в слове w_2 Игрока 2, которая меньше текущей позиции, существует соответствующая ей позиция i в слове w_1 Игрока 1, такая, что конфигурация $(w_1|^i, w_2|^j)$ удовлетворяет свойствам 1 и 2.*

Таким образом, теорема 1 доказана. □

6. $\mathcal{LP}$-n-LTL vs S1S

Перейдем теперь к исследованию выразительных возможностей фрагмента $\mathcal{LP}$-n-LTL. Очевидно, что фрагмент $\mathcal{LP}$-n-LTL является не менее выразительным, чем фрагмент $\mathcal{LP}$-1-LTL. Далее мы покажем, что он столь же выразителен, как монадическая логика второго порядка S1S. Для доказательства этого факта мы не будем рассматривать синтаксис и семантику логики S1S, а воспользуемся вместо этого взаимосвязью этой логики с конечными автоматами, предназначенными для распознавания сверхслов. Хорошо известно (см., например, [15]), что следующие три утверждения о языках сверхслов эквивалентны:

- множество L состоит из всех тех сверхслов, на которых выполняется некоторая формула Φ логики S1S, т. е. $L = \{w \mid w \models \Phi\}$,
- множество сверхслов L является *ω-регулярным языком*,
- множество сверхслов LL распознаётся некоторым *автоматом Бюхи (Рабина, Маллера, Стрита)*.

Поэтому мы будем сравнивать выразительные возможности фрагмента $\mathcal{LP}$-n-LTL с вычислительными способностями конечных ω-автоматов.

Определение 10. Конечным автоматом над сверхсловами *(*ω-автоматом*) называется пятёрка* $A = (\Sigma, Q, Q_0, \Delta, \mathcal{F})$, *где*

- Σ *— это конечный алфавит,*
- Q *— это конечное множество* состояний*,*
- $\Delta \subseteq Q \times \Sigma \times Q$ — отношение переходов*,*
- $Q_0 \in Q$ *— множество* начальных состояний*,*
- $\mathcal{F}$ — допускающее правило*.*

Детерминированным называется такой ω-автомат, у которого $Q_0 = \{q_0\}$ и отношение переходов является функцией, т. е. для любого состояния q и любой буквы a существует единственное состояние $q' \in Q$, для которого $(q, a, q') \in \Delta$. *Прогоном* ω-автомата A на сверхслове w называется всякое такое отображение $\rho\colon \mathbb{N}_0 \to Q$, у которого $\rho(0) \in Q_0$, и для каждого i выполняется соотношение $(\rho(i), w(i), \rho(i+1)) \in \Delta$.

Записью $\inf(\rho)$ обозначим множество состояний, которые встречаются в прогоне ρ бесконечно часто. Разнообразные виды ω-автоматов отличаются друг от друга допускающими правилами. Здесь мы рассматриваем автоматы Бюхи и автоматы Маллера. Допускающее правило обобщенного автомата Бюхи B представляет собой семейство $\mathcal{F} = \{F_1, F_2, \dots, F_m\}$ подмножеств F_i, $F_i \subseteq Q$, $1 \leqslant i \leqslant m$, множества состояний автомата B.

Определение 11. *Автомат Бюхи B* допускает *сверхслово w тогда и только тогда, когда существует такой прогон ρ этого автомата на сверхслове w, что* $\inf(\rho) \cap F_i \neq \varnothing$ *для каждого i, $1 \leqslant i \leqslant m$.*

Допускающее правило автомата Маллера M также задается семейством $\mathcal{F} = \{E_1, E_2, \dots, E_m\}$ подмножеств E_i, $E_i \subseteq Q$, $1 \leqslant i \leqslant m$, множества состояний ω-автомата M.

Определение 12. *Автомат Маллера M* допускает *сверхслово w тогда и только тогда, когда существует такой прогон ρ этого автомата на сверхслове w, что $\inf(\rho) \in \mathcal{F}$.*

Множество сверхслов, допускаемых автоматом A, обозначим записью $L(A)$. Говорят, что автомат A *распознает* язык $L(A)$.

Покажем, что класс языков, распознаваемых ω-автоматами, совпадает с классом языков, которые можно описать формулами логики $\mathcal{LP}$-n-LTL. Для этого, во-первых, каждой формуле $\psi \in \mathcal{LP}$-n-LTL поставим в соответствие такой недетерминированный автомат Бюхи B_ψ, что $L(B_\psi) = \{w \mid w \models \psi\}$. А во-вторых, для каждого детерминированного автомата Маллера M построим такую формулу $\psi_M \in \mathcal{LP}$-n-LTL, что $L(M) = \{w \mid w \models \psi_M\}$.

В первом случае нам потребуется несколько модифицировать хорошо известную схему построения автомата Бюхи по формуле логики LTL (см., например, [1]). Пусть $\varphi \in \mathcal{LP}$-n-LTL. Перепишем формулу φ, используя тождества $\mathbf{G}_L\psi = \neg\mathbf{F}_L\neg\psi$ и $\mathbf{F}_L\psi = \mathbf{true}\,\mathbf{U}_L\psi$ так, чтобы исключить операторы $\mathbf{F}$ и $\mathbf{G}$, оставив только $\mathbf{U}$. Далее без ограничения общности полагаем, что φ построена с помощью $\neg$, $\wedge$, $\mathbf{X}_c$, $\mathbf{Y}_c$, $\mathbf{U}_L$.

Определение 13. Замыканием $cl(\varphi)$ *формулы φ называется множество всех подформул этой формулы, т. е. наименьшее такое множество, что*

- $\varphi \in cl(\varphi)$*;*
- *если* $\neg\psi \in cl(\varphi)$*, то* $\psi \in cl(\varphi)$*;*
- *если* $\psi \wedge \chi \in cl(\varphi)$*, то* $\psi, \chi \in cl(\varphi)$*;*
- *если* $\mathbf{X}\psi \in cl(\varphi)$*, то* $\psi \in cl(\varphi)$*;*
- *если* $\psi\,\mathbf{U}_L\chi \in cl(\varphi)$*, то* $\psi, \chi \in cl(\varphi)$*.*

Сверхслову $w \in \Sigma^\omega$ поставим в соответствие такое сверхслово ρ_w в алфавите $2^{cl(\varphi)}$, что для каждого i включение $\psi \in \rho_w(i)$ имеет место тогда и только тогда, когда $w|^i \models \psi$. Далее мы построим автомат Бюхи $\mathcal{B}_\varphi$, у которого множеством состояний является семейство всех подмножеств замыкания $cl(\varphi)$ формулы φ, а последовательность ρ_w — это прогон B на слове w. С помощью состояний, переходов и допускающего правила мы опишем семантику формулы φ, причем для описания семантики параметров операторов мы воспользуемся конечными автоматами.

Определение 14. *(Неинициализированным)* конечным автоматом *назовем тройку $A = (\Sigma, Q, \delta)$, где Σ — это алфавит, Q — множество состояний, а $\delta\colon Q \times \Sigma \to Q$ — функция переходов.*

Функцию переходов обычным образом распространим на множестве Σ^* всех конечных слов в алфавите Σ: $\delta(q, \varepsilon) = q$ для пустого слова ε, и $\delta(q, \alpha a) = \delta(\delta(q, \alpha), a)$ для любых $a \in \Sigma$ и $\alpha \in \Sigma^*$. Автомат A называется *инициализированным*, если задано *начальное состояние* q_0 и множество допускающих состояний F. Инициализированный таким образом автомат обозначается $A(q_0, F)$; он *допускает* конечное слово α тогда и только тогда, когда $\delta(q_0, \alpha) \in F$. Записью $L(A(q_0, F))$ мы обозначаем множество всех слов, допускаемых автоматом $A(q_0, F)$, т. е. *язык* этого автомата. Далее, упоминая регулярный язык L, мы полагаем, что он распознается автоматом $A^L({q_0}^L, F^L)$, где $A^L = (\Sigma, Q^L, \delta^L)$.

Утверждение 3. *Для каждой формулы φ логики $\mathcal{LP}$-n-LTL существует обобщенный автомат Бюхи $\mathcal{B}_\varphi$, обладающий следующим свойством: для любого сверхслова w отношение $w \models \varphi$ выполняется тогда и только тогда, когда w допускается автоматом Бюхи $\mathcal{B}_\varphi$.*

Доказательство. Пусть φ — произвольная формула логики $\mathcal{LP}$-n-LTL. Опишем устройство соответствующего обобщенного автомата Бюхи $\mathcal{B}_\varphi$. Для этого построим множество $\widehat{Q} = 2^{cl(\varphi)} \times Q^{L_1} \times Q^{L_2} \times \ldots \times Q^{L_m}$, где $L_1, L_2, \ldots, L_m$ — параметры всех темпоральных операторов, используемых в φ. Важно отметить, что, если некоторые операторы наделены одним и тем же параметром L, этот параметр всё равно включается в список $L_1, L_2, \ldots, L_m$ отдельно для каждого оператора.

Назовём элемент $q = (S, q^{L_1}, \ldots, q^{L_m}) \in \widehat{Q}$ *нормальным*, если

- множество формул S непротиворечиво относительно классической пропозициональной логики, т. е.
  - если $\psi \wedge \chi \in S$, то $\psi \in S$ и $\chi \in S$;
  - если $\psi \in S$, то $\neg\psi \notin S$;
  - если $\mathbf{true} \in cl(\varphi)$, то $\mathbf{true} \in S$;
  - если $a \in S \cap \Sigma$ и $b \in S \cap \Sigma$, то $a = b$;
- q локально непротиворечив относительно оператора $\mathbf{U}$, т. е. для любой формулы вида $\psi\, \mathbf{U_L} \chi \in cl(\varphi)$
  - если $\chi \in S$ и $q^L \in F^L$, то $\psi\, \mathbf{U_L} \chi \in S$;
  - если $\psi\, \mathbf{U_L} \chi \in S$, и $\chi \notin S$, и $q^L \in F^L$, то $\psi \in S$;
- множество формул S максимально, т. е.
  - $S \cap \Sigma \neq \varnothing$;
  - для любой формулы $\psi \in cl(\varphi)$ если $\psi \notin S$, то $\neg\psi \in S$.

Записью $Norm(\varphi)$ обозначим множество всех нормальных элементов $\widehat{Q}$.

Приступим теперь непосредственно к построению автомата Бюхи $\mathcal{B}_\varphi$, распознающего множество сверхслов, на которых выполняется заданная формула φ логики $\mathcal{LP}$-n-LTL. В отличие от аналогичного метода, описанного в монографии [1] для построения автомата Бюхи, соответствующего формуле LTL, нам приходится учитывать конечные автоматы, которые используются для параметризации темпоральных операторов.

Рассмотрим автомат Бюхи $\mathcal{B}_\varphi = (\Sigma, Q, Q_0, \Delta, \mathcal{F})$, где

- $Q = Norm(\varphi)$;
- $Q_0 = \{q \in Q \mid q = (S, q_0^{L_1}, \ldots, q_0^{L_m}),\ \varphi \in S\}$;
- $\mathcal{F} = \{F_{\psi\,\mathbf{U}_L\chi} \mid \psi\,\mathbf{U}_L\chi \in cl(\varphi)\}$, где
$$F_{\psi\,\mathbf{U}_L\chi} = \{q \in Q \mid \psi\,\mathbf{U}_L\chi \notin S \text{ или совместно } \chi \in S \text{ и } q^L \in F^L\}$$
- Отношение переходов $\Delta\colon Q \times \Sigma \to 2^Q$ для состояния $q = (S, q^{L_1}, \ldots, q^{L_m}) \in Q$ и буквы $a \in \Sigma$ определяется следующим образом:
  - если $a \notin S \cap \Sigma$ или существует такая подформула $\mathbf{X}_c\psi \in S$, что $a \neq c$, то $\Delta(q, a) = \varnothing$;
  - если $S \cap \Sigma = \{a\}$ и для каждой подформулы $\mathbf{X}_c\psi \in S$ верно, что $a = c$, то $\Delta(q, a)$ равно множеству всех состояний таких $\hat{q} = (\hat{S}, \hat{q}^{L_1}, \ldots, \hat{q}^{L_m}) \in Q$, для которых выполнены следующие условия:
    1. для каждой подформулы $\mathbf{X}_c\psi \in cl(\varphi)$: $\mathbf{X}_c\psi \in S \iff \psi \in \hat{S}$;
    2. для каждой подформулы $\mathbf{Y}_c\psi \in cl(\varphi)$: если $a = c$, то $\mathbf{Y}_c\psi \in S \iff \psi \in \hat{S}$;
    3. для каждой подформулы $\psi\,\mathbf{U}_L\chi \in cl(\varphi)$: $\psi\,\mathbf{U}_L\chi \in S \iff$ ($q^L \in F^L$ и $\chi \in S$) или одновременно $\psi\,\mathbf{U}_L\chi \in \hat{S}$ и, если $q^L \in F^L$, то $\psi \in S$).
    4. для каждого параметра L_i, если помеченная им подформула $\psi\,\mathbf{U}_{L_i}\chi$ принадлежит S, то $\hat{q}^{L_i} = \delta^{L_i}(q^{L_i}, a)$, а иначе $\hat{q}^{L_i} = q^{L_i}$.

По сути дела, ограничения, налагаемые на отношение переходов Δ, полностью описывают семантику темпоральных операторов $\mathbf{X}$, $\mathbf{Y}$ и $\mathbf{U}$. Последнее правило в описании отношения переходов Δ, в частности, гарантирует, что каждый автомат, отвечающий за параметр темпорального оператора, начинает свою работу, как только начинается обработка подформулы, озаглавленной этим оператором.

Чтобы вычисления автомата Бюхи $\mathcal{B}_\varphi$ корректно соответствовали семантике операторов Until $\mathbf{U}_L$, встречающихся в формуле φ, для каждой подформулы $\zeta = \psi\,\mathbf{U}_L\chi$ формулы φ в допускающем правиле для этого ω-автомата имеется множество состояний F_ζ. Присутствие такого множества в допускающем правиле гарантирует, что в каждом прогоне ρ_w, для которого $\zeta \in \rho_w(0)$, будет верно включение $\chi \in S_j$ для некоторого $j \geqslant 0$, $\rho_w(j) = (S_j, \ldots, q_j^L, \ldots)$, $q_j^L \in F^L$ (т. е., $w[0 \ldots j] \in L$); и $\psi \in S_i$ для всех $i < j$, таких, что $q_i^L \in F^L$ (то есть, $w[0 \ldots i] \in L$). В конструкции F_ζ учтено также и влияние на выполнимость оператора Until его параметра: сверхслово w удовлетворяет формуле $\psi\,\mathbf{U}_L\chi$ только в том случае, когда χ в итоге становится верной в позиции, "согласованной" с параметром L.

Далее нетрудно показать, воспользовавшись той же самой схемой рассуждений, которая приведена в монографии [1] для обоснования аналогичного рассуждения, что построенный таким образом автомат Бюхи $\mathcal{B}_\varphi$ обладает требуемым свойством: для любого сверхслова w отношение $w \models \varphi$ выполняется тогда и только тогда, когда w допускается автоматом Бюхи $\mathcal{B}_\varphi$. $\square$

Если принять во внимание сделанное ранее замечание о взаимосвязи автоматов Бюхи и логики S1S, то на основании утверждения 3 можно прийти к заключению о том, что $\mathcal{LP}$-n-$LTL \preccurlyeq$ S1S.

Покажем теперь, как по автомату Маллера построить эквивалентную ему формулу логики $\mathcal{LP}$-n-LTL.

Утверждение 4. *Для каждого детерминированного автомата Маллера существует формула φ логики $\mathcal{LP}$-n-LTL, которая удовлетворяет следующему требованию: автомат M допускает сверхслово w тогда и только тогда, когда $w \models \varphi_M$.*

Доказательство. Пусть задан произвольный детерминированный автомат Маллера $M = (\Sigma, Q, q_0, \Delta, \mathcal{F})$ с единственным начальным состоянием q_0 и допускающим правилом $\mathcal{F} = \{E_1, E_2, \dots, E_m\}$. Напомним, что *допускающий прогон* автомата M — это такой его прогон ρ, что $\inf(\rho) = E_i$ для некоторого $E_i \in \mathcal{F}$. Заметим, что единственный прогон ρ_w детерминированного автомата Маллера на слове w однозначно определяется этим словом, и состояние, в котором находится автомат в некоторый момент вычисления, зависит только от префикса слова w, прочитанного к этому моменту. Эту зависимость мы и опишем формулами $\mathcal{LP}$-n-LTL.

Рассмотрим лежащий в основе автомата Маллера M неинициализированный конечный автомат $A = (\Sigma, Q, \Delta)$ и его инициализации $M(q, Q') = A(q, Q')$ для всевозможных состояний $q \in Q$ и подмножеств $Q' \subseteq Q$. Для упрощения обозначений мы будем использовать запись $M(q, Q')$ вместо $L(M(q, Q'))$.

Для каждого множества состояний $E_i \in \mathcal{F}$ построим следующие три формулы:

1. $\psi_1^i = \bigwedge_{q_e \in E_i} \mathbf{F}_{M(q_0, q_e)}\, \mathbf{true}$. Формула ψ_1^i выполняется на сверхслове w в том и только том случае, когда прогон автомата Маллера M на w проходит через все состояния множества E_i.

2. $\psi_2^i = \bigwedge_{q \in Q} \mathbf{G}_{M(q_0, q)} \left(\bigwedge_{q_e \in E_i} \mathbf{F}_{M(q, q_e) \setminus \{\varepsilon\}}\, \mathbf{true} \right)$. Данная формула выполняется на сверхслове w в том и только том случае, когда прогон автомата Маллера M на w проходит бесконечно часто через каждое состояние из множества E_i.

3. $\psi_3^i = \bigvee_{q \in Q} \mathbf{F}_{M(q_0, q)} \left(\mathbf{G}_{M(q, Q \setminus E_i)}\, \mathbf{false} \right)$. Эта формула выполняется на сверхслове w в том и только том случае, когда прогон автомата Маллера M на w проходит бесконечно часто только через состояния из множества E_i.

Из приведенных выше пояснений содержательного смысла формул ψ_1^i, ψ_2^i и ψ_3^i следует, что $\varphi_M = \bigvee_{i=1}^{m} (\psi_1^i \wedge \psi_2^i \wedge \psi_3^i)$ является искомой формулой. $\square$

Таким образом, верна

Теорема 2. $\mathcal{LP}$-n-$LTL \equiv$ S1S.

7. Заключение

После опубликования статьи [5], в которой было проведено подробное изучение темпоральной логики LTL в свете ее применения в качестве средства описания поведения вычислительных систем, было предпринято несколько попыток расширить выразительные возможности этой логики. В статье [14] в синтаксис LTL были добавлены средства квантификации по базовым предикатам. Оказалось, что выразительность квантифицированного расширения $QLTL$ существенно превосходит описательные возможности темпоральной логики LTL. Например, с использованием кванторов свойство $W_1^{2n} = \{w \in \Sigma^\omega \mid w(2i) = 1,\ i \in \mathbb{N}_0\}$ сверхслов в алфавите $\Sigma = \{0, 1\}$, невыразимое в логике LTL, описывается формулой ее расширения $QLTL$

$$\exists q(q \wedge \mathbf{G}(q \to \mathbf{X}\neg q) \wedge \mathbf{G}(\neg q \to q) \wedge \mathbf{G}(q \to p)) \ .$$

Автор статьи [19] предложил способ введения новых темпоральных операторов при помощи праволинейных грамматик. Слова, порождаемые этими грамматиками, задают шаблоны, на которых проверяется выполнимость формул в области действия темпорального оператора. Например, темпоральный оператор $\mathbf{E}p$, проверяющий указанное выше свойство W_1^{2n}, задается грамматикой из двух правил $V_0 \to pV_1$ и $V_1 \to \mathbf{true}\ V_0$. Подобным же образом шаблоны для описания семантики новых темпоральных операторов могут быть описаны при помощи конечных автоматов, как это показано в статьях [12, 16]. Не были обойдены вниманием и операторы неподвижной точки: логика линейного времени μ-LTL, обогащенная этими операторами, была введена и изучена в статье [18].

Для всех перечисленных расширений LTL (за исключением μ-LTL) было показано, что они имеют точно такие же выразительные способности, что и логика S1S, но при этом проблема выполнимости для всех этих логик остается PSPACE-полной, как и для LTL. Вводя в рассмотрение темпоральную логику $\mathcal{LP}$-LTL, мы не стремились всего лишь расширить выразительные возможности LTL как таковые. Нашей целью было создание адекватного языка для спецификации поведения реагирующих систем, моделируемых автоматами-преобразователями. Тем не менее, теорема 2 показывает, что выразительные возможности предложенного нами логического языка спецификаций совершили такой же скачок, как и другие расширения LTL, исследованные в статьях [12, 14, 16, 18, 19].

Идея параметризации темпоральных операторов не нова: почти такая же параметризация темпоральных операторов, что и в данной работе, была введена для динамического расширения логики LTL в статье [9]. По сути дела, фрагмент $\mathcal{LP}$-n-LTL имеет очень большое сходство с логикой $DLTL$, которая была введена в этой статье. Однако наше расширение LTL отличается, главным образом, тем, что базовые предикаты в нем также параметризованы. В дальнейшем мы собираемся провести сравнение выразительных возможностей логики $\mathcal{LP}$-n-LTL и динамических логик, включая $DLTL$ [9] и PDL [4].

Доказанная нами теорема 1 открывает еще одно направление исследований. Каждое из расширений логики LTL, предложенных в статьях [9, 14, 16, 18, 19], оказывается столь же выразительным, как одна из двух логик LTL и S1S. Однако, как показно в теореме 1, рассмотренный в нашей работе фрагмент $\mathcal{LP}$-1-LTL оказался более выразительным, нежели LTL, а как показано в теореме 2, другой фрагмент $\mathcal{LP}$-n-LTL оказался столь же выразительным, как и логика S1S. Мы высказываем предположение о том, что два рассмотренных в нашей работе фрагмента логики $\mathcal{LP}$-LTL имеют разные выразительные возможности. Если это предположение окажется справедливым, то фрагмент $\mathcal{LP}$-1-LTL будет являться естественным новым примером темпоральной логики, занимающей по выразительным возможностям промежуточное положение между хорошо известными логиками $(\mathbb{N}, <)$ и S1S. Мы полагаем, что эту гипотезу удастся доказать при помощи игр Эренфойхта–Фреше, которые были использованы в данной статье.

Список литературы / References

[1] Baier C., Katoen J., *Principles of Model Checking*, MIT Press, 2008.

[2] Clarke E. M., Gramberg O., Peled D. A., *Model Checking*, MIT Press, 1999.

[3] Etessami K., Wilke T., "An Until Hierarchy and Other Applications of an Ehrenfeucht-Fraisse Game for Temporal Logic", *Information and Computation*, **160**:1 (2000), 88–108.

[4] Fisher J. M., Ladner R. E., "Propositional dynamic logic of regular programs", *Journal of Computer and System Sciences*, **18** (1979), 194–211.

[5] Gabbay D., Pnueli A., Shelach S., Stavi J., "The temporal analysis of fairness", *Proceedings of 7-th ACM Symposium on Principles of Programming Languages*, 1980, 163-173.

[6] Гнатенко А. Р., Захаров В. А., "О сложности верификации конечных автоматов-преобразователей над коммутативными полугруппами", *Труды 18-й Международной конференции "Проблемы теоретической кибернетики"*, 2017, 68–70; [Gnatenko A. R., Zakharov V. A., "O slozhnosti verifikatsii avtomatov-preobrazovateley nad kommutativnymi polugruppami", *Trudy 18 Mezhdunarodnoj konferentsii "Problemy teoteticheskoy kibernetiki"*, 2017, 68–70, (in Russian).]

[7] Гнатенко А. Р., Захаров В. А., "О верификации конечных автоматов-преобразователей над полугруппами", *Труды Института системного программирования РАН*, **30**:3 (2018), 303–324; [Gnatenko A. R., Zakharov V. A., "On the model checking of finite state transducers over semigroups", *Proceedings of ISP RAS*, **30**:3, 303–324].

[8] Harel D., Kozen D., Parikh P., "Process logic: Expressiveness, decidability, completeness", *Journal of Computer and System Science*, **25**:2 (1982), 144–170.

[9] Henriksen J. J., Thiagarajan P. S., "Dynamic linear time temporal logic", *Annals of Pure and Applied Logic*, **96**:1–3 (1999), 187–207.

[10] Kamp J. A. W., *Tense Logic and the Theory of Linear Order, PhD thesis*, University of California, Los Angeles, 1968.

[11] Kozlova D. G., Zakharov V. A., "On the model checking of sequential reactive systems", *Proceedings of the 25th International Workshop on Concurrency, Specification and Programming (CS&P 2016)*, CEUR Workshop Proceedings, **1698**, 2016, 233–244.

[12] Kupferman O., Piterman N., Vardi M. Y., "Extended Temporal Logic Revisited", *Proceedings of 12-th International Conference on Concurrency Theory*, 2001, 519–535.

[13] Leucker M., Sanchez C., "Regular Linear Temporal Logic", *Proceedings of the 4-th International Colloquium on Theoretical Aspects of Computing*, 2007, 291–305.

[14] Manna Z., Wolper P., "Synthesis of Communicating Processes from Temporal Logic Specifications", *Lecture Notes in Computer Science*, **131**, 1981, 253–281.

[15] Thomas W., "Automata on infinite objects", *Handbook of Theoretical Computer Science, volume B: Formal Models and Semantics*, 1990, 133–192.

[16] Vardi M. Y., Wolper P., "Yet Another Process Logic", *Lecture Notes in Computer Science*, **14**, 1983, 501–512.

[17] Vardi M. Y., Wolper P., "An Automata-Theoretic Approach to Automatic Program Verification", *Proceedings of the First Symposium on Logic in Computer Science*, 1986, 322–331.

[18] Vardi M. Y., "A Temporal Fixpoint Calculus", *Proceedings of the 15-th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 1989, 250–259.

[19] Wolper P., "Temporal Logic Can Be More Expressive", *Information and Control*, **56**:1–2 (1983), 72–99.

[20] Wolper P., Boigelot B., "Verifying systems with infinite but regular state spaces", *Lecture Notes in Computer Science*, **1427**, 1998, 88–97.

[21] Zakharov V. A., "Equivalence checking problem for finite state transducers over semigroups", *Lecture Notes in Computer Science*, **9270**, 2015, 208–221.

[22] Захаров В. А., Темербекова Г. Г., “О минимизации конечных автоматов-преобразователей над полугруппами”, *Моделирование и анализ информационных систем*, **23**:6 (2016), 741–753; English transl.: Zakharov V. A., Temerbekova G. G., “On the minimization of finite state trans-ducers over semigroups”, *Automatic Control and Computer Sciences*, **51**:7 (2017), 523–530.

[23] Захаров В. А., Жайлауова Ш. Р., “О задаче минимизации последовательных программ”, *Моделирование и анализ информационных систем*, **24**:4 (2017), 415–430; English transl.: Zakharov V. A., Jaylauova S. R., “On the minimization problem for sequential programs”, *Automatic Control and Computer Sciences*, **51**:7 (2017), 689–700.

Gnatenko A. R., Zakharov V.A., "On the Expressive Power of Some Extensions of Linear Temporal Logic", *Modeling and Analysis of Information Systems*, **25**:5 (2018), 506–524.

DOI: 10.18255/1818-1015-2018-5-506-524

Abstract. One of the most simple models of computation which is suitable for representation of reactive systems behaviour is a finite state transducer which operates over an input alphabet of control signals and an output alphabet of basic actions. The behaviour of such a reactive system displays itself in the correspondence between flows of control signals and compositions of basic actions performed by the system. We believe that the behaviour of this kind requires more suitable and expressive means for formal specifications than the conventional LTL. In this paper, we define some new (as far as we know) extension $\mathcal{LP}$-LTL of Linear Temporal Logic specifically intended for describing the properties of transducers computations. In this extension the temporal operators are parameterized by sets of words (languages) which represent distinguished flows of control signals that impact on a reactive system. Basic predicates in our variant of the temporal logic are also languages in the alphabet of basic actions of a transducer; they represent the expected response of the transducer to the specified environmental influences. In our earlier papers, we considered a model checking problem for $\mathcal{LP}$-LTL and $\mathcal{LP}$-CTL and showed that this problem has effective solutions. The aim of this paper is to estimate the expressive power of $\mathcal{LP}$-LTL by comparing it with some well known logics widely used in the computer science for specification of reactive systems behaviour. We discovered that a restricted variant $\mathcal{LP}$-1-LTL of our logic is more expressive than LTL and another restricted variant $\mathcal{LP}$-n-LTL has the same expressive power as monadic second order logic S1S.

Keywords: temporal logics, expressive power, specification, verification, Buchi automata, infinite words

On the authors:
Anton R. Gnatenko, orcid.org/0000-0003-1499-2090, student,
Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia, e-mail: gnatenko.cmc@gmail.com

Vladimir A. Zakharov, orcid.org/0000-0002-3794-9565, Dr. of Science, professor,
National Research University Higher School of Economics (HSE),
20 Myasnitskaya ulitsa, Moscow, 101000 Russia, e-mail: zakh@cs.msu.ru

Acknowledgments:
This work was supported by the Russian Foundation for Basic Research, Grant N 18-01-00854