S©MR

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ
Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru
ISSN 1813-3304

Том 20, № 2, стр. 144–145 (2023) УДК 510.64
https://doi.org/10.33048/semi.2023.20.??? MSC 03B44,
03B42, 03A05, 03B45, 03B70,03H05

ПРОЕКТИВНАЯ УНИФИКАЦИЯ И СВОЙСТВО КОНЕЧНОЙ МОДЕЛИ ДЛЯ СТУПЕНЧАТОЙ ВРЕМЕННОЙ ЛОГИКИ ЗНАНИЙ С ПРОШЛЫМ pLTK.sl

С.И. Башмаков, Ю.А. Бушакова

COMMUNICATED BY S.V. SUDOPLATOV

Abstract: This paper is devoted to the study of unification and the finite model property for the linear temporal step-like logic with past and future operators and multiagent knowledge $p\mathcal{LTK}.sl$. For this logic, a modal degree is defined in a special way that takes into account the nesting of temporal operators. Based on this definition, a p-morphism is constructed that reduces an arbitrary model to a model that is finite in time. Subsequent filtration yields the finiteness of clusters of worlds. It is established that every unifiable formula in this logic possesses a projective unifier, and consequently the logic has unitary type of unification. An explicit construction of the most general unifier is proposed.

Keywords: modal logic, temporal logic, linear time, finite model property, Kripke relational semantics, unification.

Bashmakov, S.I., Bushakova Yu.A. Проективная унификация и свойство конечной модели для ступенчатой временной логики знаний с прошлым $pLTK.sl$.

Работа поддержана Красноярским математическим центром, финансируемым Минобрнауки РФ (Соглашение № 075-02-2026-1314).

Поступила 1 января 2025 г., опубликована 31 декабря 2025 г.

1 Введение

Модальные логики представляют наиболее значимый раздел теории неклассических логик. Различные типы модальных логик исследуются как с точки зрения своей фундаментальной природы объекта, так и в качестве инструмента моделирования реальных задач. В этой части наиболее эффективным показывает себя инструментарий временных и многоагентных систем, моделируемых как многомодальные логики со всем разнообразием их семантик, [9].

Наиболее широко используемыми и хорошо зарекомендовавшими себя являются временные логики $\mathcal{LTL}$ и $\mathcal{CTL}$, моделирующие линейный и ветвящийся временной процессы соответственно, [10]. Такие логики активно используются в теории верификации программного обеспечения и алгоритмов, когда время характеризуется как рефлексивный транзитивный процесс, допускающий или не допускающий существование параллельных веток вычисления. Однако, в больших сложных системах целостность процесса передачи данных может не гарантироваться, а доступность информации может ограничиваться конкретными временными промежутками, [11].

Идея моделирования таких систем лежит в основе логик, входящих в класс линейных многоагентных логик ступенчатого времени $\mathcal{LTK}.sl$, где агенты моделируются как носители знания в каждый момент времени. В данной статье к рассмотрению предлагается логика в классе $\mathcal{LTK}.sl$, допускающая ступенчатый доступ к информации в прошлом и в будущем. Начало исследованиям логик класса ступенчатого времени $\mathcal{LTK}.sl$ положено в работе С.И. Башмакова и Т.Ю. Зверевой. Обогащение языка временным модальным оператором P (Previous) позволяет дополнить достижимость миров предшествующими временными сгустками и требует корректировки определения модальной степени, которое теперь позволит независимо оценивать положительную и отрицательную модальную глубину каждой формулы.

Теория унификации в логике представляет собой важный раздел современной математической логики в силу своей интеграции с фундаментальными задачами информатики. Значимость этого аппарата обусловлена его непосредственной применимостью в качестве базового вычислительного механизма при решении широкого круга практических проблем. Методы унификации находят эффективное применение в задачах анализа и оптимизации программного кода. В частности, в работе В.А. Захарова и Т.А. Новиковой [7], посвященной применению унификации в задачах рефакторинга, показано, как естественное обобщение классической задачи унификации выражений до задачи двусторонней унификации подстановок (нахождения решения уравнения в полугруппе подстановок) позволяет применять этот аппарат к анализу формальных моделей программ и формализовать вопрос о наличии у двух заданных объектов некоторого общего инварианта. Таким образом, развитие теории унификации обеспечивает математический фундамент для создания эффективных инструментов верификации, что подтверждает её статус важного связующего звена между абстрактными логическими исчислениями и инженерными задачами разработки программного обеспечения.

В статье рассматривается вопрос об унификации логики $p\mathcal{LTK}.sl$. Под задачей унификации в логике понимается поиск подстановки формул вместо переменных, обращающей данную формулу в теорему рассматриваемой логики. Если такая подстановка существует, формулу называем унифицируемой, а саму подстановку — унификатором. На унификаторах определен порядок «более общий», и, если каждая унифицируемая формула в логике имеет унификатор более общий, чем любой другой, логика имеет унитарный тип унификации, [5]. В ходе работы установлен унитарный тип унификации для $p\mathcal{LTK}.sl$.

Финитная аппроксимируемость (иначе — свойство конечной модели) является важным свойством логики, которое позволяет реализовывать алгоритм проверки истинности формул на конечных моделях. Говорим, что логика $\mathcal{L}$ финитно аппроксимируема если найдётся класс конечных шкал F со свойствами $F \vDash L$ и $F \nvDash \varphi$ для любой формулы $\varphi \notin L$ языка логики, [8, 4]. Нами доказана финитная аппроксимируемость логики $p\mathcal{LTK}.sl$ и получена подходящая скрученная по времени фильтрованнная по сгусткам модель.

2 Семантика логики $p\mathcal{LTK}.sl$

Алфавит языка $p\mathcal{LTK}.sl$ состоит из счётного множества пропозициональных переменных $Prop := \{p_1, \dots, p_n, \dots\}$, констант $\top$, $\bot$, временных операторов N и P, модальных операторов $\{\Box_e, \Box_1, \dots, \Box_n,\}$, скобок и стандартных булевых операций.

$(n+3)$ – модальной $p\mathcal{LTK}.sl$-шкалой назовём упорядоченный набор $F := \langle Z_t, R_f, R_p, R_e, R_1, \dots, R_n \rangle$, где

- $Z_t = \bigcup_{t \in \mathbb{Z}} C_t$ – объединение непересекающихся сгустков C_t : $C_{t_1} \cap C_{t_2} = \varnothing$, где сгусток C_t — множество миров $\{x_i, y_i, z_i, \dots\}$;
- R_f – отношение «следующий момент времени»: $\forall x, y \in Z_t : xR_f y \iff \exists t \in Z_t (x \in C_t$ и $y \in C_{t+1})$;
- R_f – отношение «предыдущий момент времени»: $\forall x, y \in Z_t : xR_f y \iff \exists t \in Z_t (x \in C_t$ и $y \in C_{t-1})$;
- R_e – отношение эквивалентности на каждом отдельном сгустке: $\forall t \in \mathbb{Z}, \forall x, y \in C_t : xR_e y$;
- $R_1, \dots, R_n$ – сериальные отношения агентов на каждом отдельном сгустке: $\forall i \in [1, \dots, n]$ $R_i \subseteq R_e$.

Стандартно определяем $p\mathcal{LTK}.sl$-модель $M := \langle F, V \rangle$, где F – $p\mathcal{LTK}.sl$-шкала, а V – означивание на шкале $Prop \to 2^{Z_t}$. Выполнимость переменных и формул, содержащих стандартные логические связки и модальные операторы $\forall x \in C_t \subset Z_t, \forall t \in \mathbb{Z}$ определяется следующим образом:

- $\langle M, x\rangle \vDash p \Leftrightarrow x_i \in V(p)$;
- $\langle M, x\rangle \vDash \varphi \vee \psi \Leftrightarrow \langle M, x\rangle \vDash \varphi$ или $\langle M, x\rangle \vDash \psi$;
- $\langle M, x\rangle \vDash \varphi \wedge \psi \Leftrightarrow \langle M, x\rangle \vDash \varphi$ и $\langle M, x\rangle \vDash \psi$;
- $\langle M, x\rangle \vDash \neg\varphi \Leftrightarrow \langle M, x\rangle \nvDash \varphi$;
- $\langle M, x\rangle \vDash N\varphi \Leftrightarrow \forall y : xR_f y\langle M, y\rangle \vDash \varphi$;
- $\langle M, x\rangle \vDash P\varphi \Leftrightarrow \forall y : xR_p y\langle M, y\rangle \vDash \varphi$;
- $\langle M, x\rangle \vDash \Box_l\varphi \Leftrightarrow \forall y : xR_l y\langle M, y\rangle \vDash \varphi$, $\forall l \in \{e, 1, ..., n\}$.

Для $\Box_e, \Box_1, \ldots, \Box_n$ двойственный модальный оператор $\Diamond$ выражается стандартно: $\Diamond_l\varphi = \neg\Box_l\neg\varphi$, где $l \in \{e, 1, \ldots, n\}$.

Отметим отдельно, что отношения агентов R_i чаще всего определяются как отношения эквивалентности [1], но мы потребуем лишь их сериальность: $\forall x \exists y : xR_i y$. Действительно, в случае отсутствия сериальности, по определению оператора $\Box_i$, найдутся граничные миры, т.е. такие миры x, что $\nexists y : xR_i y$, на которых выполняется любая формула с модальным оператором $\Box_i\psi$, в том числе в случае $\psi = \bot$. Подобное поведение операторов делает невозможным доказательство теоремы, формулируемой в последнем разделе, и потому требует наличия сериальности.

Логику $p\mathcal{LTK}.sl$ определяем как множество формул, истинных на всех $p\mathcal{LTK}.sl$-моделях. Как и $\mathcal{LTK}.sl$ наша логика тоже моделирует временной процесс без ветвлений, а ступенчатый характер времени позволяет в текущем сгустке C_t проверять истинность формулы в следующем или предыдущем сгустке.

Концептуально ступенчатый процесс времени в логике позволяет моделировать процессы с любой требуемой степенью детализации, что хорошо соответствует его физической природе: согласно представлениям квантовой механики, время на фундаментальном уровне дискретно, и в качестве минимального «шага» между моментами времени (в нашем случае – сгустками) может быть принят планковский квант времени, [3]. Аналогично, в вычислительных системах таким неделимым шагом является такт работы процессора или дискретный интервал дискретизации сигнала. Доступность же предыдущего момента времени, в нашем случае, может быть интерпретирована, как обратимость временного процесса в любой момент.

3 Финитная аппроксимируемость $p\mathcal{LTK}.sl$

Чтобы установить финитную аппроксимируемость, первым шагом используем метод p-морфного отображения, затем — фильтрации. Последовательное получение в начале конечной по времени, а затем — и по мощности сгустков моделей, позволяет заключить финитную аппроксимируемость логики.

Для применения метода p-морфизма потребуется понятие модальной степени. Стандартное определение модальной степени не подойдёт для нашей логики, т.к. оно не позволяет точно определить количество требуемых для проверки формулы шагов вперед по оператору N и назад по оператору P. Прежде чем дать адекватное логике pLTK.sl определение модальной степени, введём два вспомогательных понятия.

Подформулу ψ вместе со своими модальными временными операторами ($N^{n_h}P^{m_g}\psi$) формулы φ назовём *простой подформулой*, если φ представима в одном из следующих видов:
$\varphi = \psi \wedge \chi$ ($\varphi = \chi \wedge \psi$);
$\varphi = \psi \vee \chi$ ($\varphi = \chi \vee \psi$);
$\varphi = \psi \to \chi$ ($\varphi = \chi \to \psi$);
$\varphi = \neg\chi$.

Путём $\pi(p)$ переменной p в формуле $N^nP^m\varphi$ назовём последовательность вложенных простых подформул, определённую по индукции: первым членом последовательности определим переменную вместе со своими временнными операторами $N^{n_h}P^{m_g}p$, вторым элементом – формулу (вместе со своими временнными операторами), в которую предыдущий элемент входит как простая подформула и далее. Последним элементом получаем $N^nP^m\varphi$.

Описанное выше можно определить формально. Так, путь представляет собой множество, где $\pi_i^j(p)$ – j-й элемент пути i-го индекса, а ψ_i – подформула φ, в которую ψ_{i-1} входит как простая подформула:

- $\pi_i^1(p) = N^{n_{i_1}}P^{k_{i_1}}p$,
- ...
- $\pi_i^{j-1}(p) = N^{n_{i_{j-1}}}P^{k_{i_{j-1}}}\psi_{j-1}$,
- $\pi_i^j(p) = N^{n_{i_j}}P^{k_{i_j}}\psi_j$,
- ...
- $\pi_k^l(p) = N^{n_k}P^{m_l}\varphi$,
  где $n_{i_j}, k_{i_j} \in \omega$.

Заметим, что путь $\pi_i(p)$ соответствует i-му вхождению переменной в формулу, поэтому у переменной может быть несколько путей $\pi_i(p)$.

Пример 1. *Для формулы* $\varphi = N^{80}(((p \to Np) \wedge Pp) \to P^{98}q) \wedge N^{101}q$ *найдём все пути переменной* p:

$\pi_1(p) = \{N^0P^0p, N^0P^0(p \to Np), N^0P^0((p \to Np) \wedge Pp), N^{80}P^0(((p \to Np) \wedge Pp) \to P^{98}q), N^0P^0(N^{80}(((p \to Np) \wedge Pp) \to P^{98}q) \to N^{101}P^0q)\}$,

$\pi_2(p) = \{N^1P^0p, N^0P^0(p \to Np), N^0P^0((p \to Np) \wedge Pp), N^{80}P^0(((p \to Np) \wedge Pp) \to P^{98}q), N^0P^0(N^{80}(((p \to Np) \wedge Pp) \to P^{98}q) \to N^{101}P^0q)\}$,

$\pi_3(p) = \{N^0P^1p, N^0P^0((p \to Np) \wedge Pp), N^{80}P^0(((p \to Np) \wedge Pp) \to P^{98}q), N^0P^0(N^{80}(((p \to Np) \wedge Pp) \to P^{98}q) \to N^{101}P^0q)\}$,

и пути переменной q:

$\pi_1(q) = \{N^0P^{98}q, N^{80}P^0(((p \to Np) \wedge Pp) \to P^{98}q), N^0P^0(N^{80}(((p \to Np) \wedge Pp) \to P^{98}q) \to N^{101}P^0q)\}$,

$\pi_2(q) = \{N^{101}P^0q, N^0P^0(N^{80}(((p \to Np) \wedge Pp) \to P^{98}q) \to N^{101}P^0q)\}$.

3.1. Подсчёт модальной степени в $p\mathcal{LTK}.sl$. Теперь мы можем перейти к подсчёту модальной степени формулы. Определим *положительную и отрицательную модальную степени одного пути*:

$$md(\pi_i(p))_+ = \sum_{j=1}^{k} n_i,\ md(\pi_i(p))_- = \sum_{j=1}^{k} m_i,$$

где $\pi_i(p) = \{N^{n_1}P^{m_1}p, \dots, N^{n_k}P^{m_k}\varphi\}$ – путь i-го вхождения в формулу переменной p.

Положительную и отрицательную модальную степень переменной определим как максимумы положительных и отрицательных модальных степеней путей по всем её вхождениям в формулу соответственно:

$$md(p)_+ = \max_i(md(\pi_i(p))_+),\ md(p)_- = \max_i(md(\pi_i(p))_-).$$

Наконец, определим *положительную и отрицательную модальную степень формулы* как максимумы по положительным и отрицательным модальным степеням всех переменных:

$$md(\varphi)_+ = \max_{p\in Var(\varphi)}(md(p)_+),\ md(\varphi)_- = \max_{p\in Var(\varphi)}(md(p)_-).$$

Обозначим $b := md(\varphi)_+$, $a := -md(\varphi)_-$ — две границы диапазона, на которые достаточно отступить от сгустка C_t назад и вперёд по направлению времени, чтобы проверить выполнимость формулы в мире сгустка C_t. Действительно, этого достаточно потому что на границах $((t+a), (t+b)$ соответственно для нижней и верхней границы) при проверке выполнимости требуется установить не более чем выполнимость переменных из $Var(\varphi)$ или формул, не содержащих временных модальных операторов, проверка которых сведётся к проверке переменных на этом же сгустке.

Такая последовательность подсчёта модальных степеней интуитивно повторяет подсчёт стандартной модальной степени формулы. В нашем случае на её величину влияют только временные модальные операторы, т.к. проверка операторов знания $\Box_i$ и $\Diamond_i$ не выходит за пределы временных сгустков, количество которых нам необходимо ограничить. Отметим, что полученный диапазон избыточен, т.к. он может содержать «петли» (т.е. диапазоны временных шагов за пределами проверки истинности переменных), которые возникают из-за последовательного перебора всех подформул, входящих в путь по каждому из временных операторов отдельно.

Пример 2. *Найдём модальную степень формулы*

$$\varphi = N^{80}((p \to Np) \wedge Pp) \to P^{98}q) \wedge N^{101}q.$$

$md(p)_+ = max(80, 80, 81) = 81;$
$md(p)_- = max(0, 0, 1) = 1;$
$md(q)_+ = max(80, 101) = 101;$
$md(q)_- = max(98, 0) = 98;$

$md(\varphi)_+ = max(81, 101) = 101;$
$md(\varphi)_- = min(1, 98) = 98.$

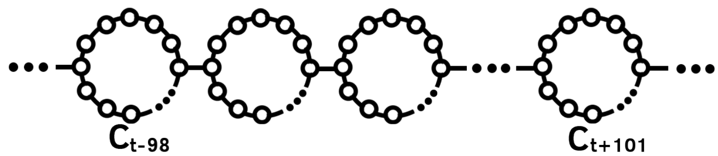

Рис. 1. $p\mathcal{LTK}.sl$-шкала, где $a := -98$, $b := 101$.

3.2. р-морфизм $p\mathcal{LTK}.sl$. Покажем, что определённого выше диапазона достаточно, используя метод p-морфного отображения. Отображение f: $M \xrightarrow{f} M'$ называется *р-морфизмом моделей*, если $\forall x, y \in Z_t, \forall l \in \{f, p, e, 1, \ldots, n\}$:

(1) $xR_ly \Rightarrow f(x)R'_lf(y)$;
(2) $f(x)R'_lf(y) \Rightarrow \exists z \in Z_t[xR_lz$ и $f(z) = f(y)]$;
(3) $\forall x \in Z_t \forall p \in Prop\ (x \in V(p) \iff f(x) \in V'(p))$.

Пусть $M = \langle Z_t, R_f, R_p, R_e, R_1, \ldots, R_n, V\rangle$ — $p\mathcal{LTK}.sl$-модель. Для неё определим конечную по времени модель $M' = \langle Z'_t, R'_f, R'_p, R'_e, R'_1, \ldots, R'_n, V'\rangle$ следующим образом:

- $Z'_t = \bigcup\limits_{i=t+a-1}^{t+b+1} C'_i$, где:
  $C'_i = C_i, \forall i \in [t+a; t+b]$ — объединение сгустков из определенного диапазона,
  $C'_{t+a-1} = \bigcup\limits_{j=t+a-1}^{-\infty} C_j$ — вырожденный сгусток, содержащий все миры модели M, находящиеся до выделенного диапазона,
  $C'_{t+b+1} = \bigcup\limits_{j=t+b+1}^{+\infty} C_j$ — вырожденный сгусток, содержащий все миры модели M, находящиеся после выделенного диапазона;
- R'_f : если $x \in C_i, y \in C_{i+1}$ и:
  $i \in [t+a-1; t+b]$, тогда $f(x)R'_ff(y)$;
  $i \in (-\infty; t+a-2]$, тогда $f(x)R'_ff(y)$ и $f(x), f(y) \in C'_{t+a-1}$;
  $i \in [t+b+1; +\infty)$, тогда $f(x)R'_ff(y)$ и $f(x), f(y) \in C'_{t+b+1}$;
- R'_p : если $x \in C_i, y \in C_{i-1}$ и:
  $i \in [t+a; t+b+1]$, тогда $f(x)R'_pf(y)$;
  $i \in (-\infty; t+a-1]$, тогда $f(x)R'_pf(y)$ и $f(x), f(y) \in C'_{t+a-1}$;
  $i \in [t+b+2; +\infty)$, тогда $f(x)R'_pf(y)$ и $f(x), f(y) \in C'_{t+b+1}$;
- $R'_l \in R_e, R_1, \ldots, R_n$: если $f(x)R_lf(y)$ и:
  $i \in [t+a; t+b]$, тогда $f(x)R'_lf(y)$;
  $i \in (-\infty; t+a-1]$, тогда $f(x)R'_lf(y)$ и $f(x), f(y) \in C'_{t+a-1}$;
  $i \in [t+b+1; +\infty)$, тогда $f(x)R'_lf(y)$ и $f(x), f(y) \in C'_{t+b+1}$;
- $\forall p \in Prop\ V'(p) := V(p)$.

Таким образом, получили два вырожденных сгустка на концах выделенного диапазона. Отдельное склеивание миров в сгусток в начале и в конце диапазона необходимо для дальнейшего сохранения корректности отношений R_f и R_p.

Отметим, что временной процесс не нарушен т.к. по построению модели не происходит перемешивание сгустков по времени внутри значимого диапазона сгустков. Полученную таким образом модель называем *скрученной.*

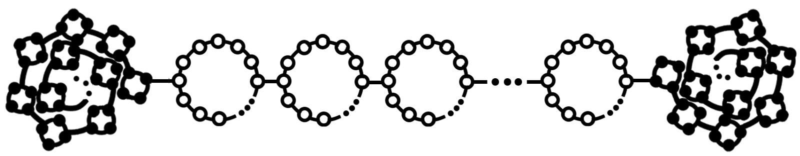

Рис. 2. Скрученная конечная по времени модель M'.

Теорема 1. *Модель M' является р-морфным образом модели M.*

Доказательство. Действительно, проверим выполнение определения р-морфного отображения f. Для этого укажем соответствие между элементами шкал F и F'

(1) $\forall x \in \bigcup\limits_{j=t+a}^{t+b} C_j, f(x) = x;$

(2) $\forall x \in \bigcup\limits_{j=t+b+1}^{+\infty} C_j, f(x) \in C'_{t+b+1};$

(3) $\forall x \in \bigcup\limits_{j=t+a-1}^{-\infty} C_j, f(x) \in C'_{t+a-1}.$

Покажем, что такое отображение f сохраняет условие р-морфизма $\forall x, y \in Z_t$:

1)$xR_l y \Rightarrow f(x)R'_l f(y), \forall l \in \{f, p, e, 1, \ldots, n\}$

- Пусть $xR_f y \Rightarrow x \in C_j, y \in C_{j+1}$.
  Если $i \in \{t+a-1, \ldots, t+b\}$, то $f(x) = x, f(y) = y \Rightarrow f(x)R'_f f(y)$.
  Если $i \in \{-\infty, \ldots, t+a-2\}$, то $f(x), f(y) \in C'_{t+a-1} \Rightarrow f(x)R'_f f(y)$.
  Если $i \in \{t+b+1, \ldots, +\infty\}$, то $f(x), f(y) \in C'_{t+b+1} \Rightarrow f(x)R'_f f(y)$.
- Для случая $xR_p y$ аналогично, с учетом сдвига диапазонов.
- Пусть $xR_i y \Rightarrow x, y \in C_j$, $\forall i \in \{1, \ldots, n\}$. Тогда $\forall j \in \mathbb{Z}$ по определению выполняется $f(x)R'_i f(y)$.

2)$f(x)R'_l f(y) \Rightarrow \exists z \in F[xR_l z$ и $f(z) = f(y)]$.

- Пусть $f(x)R'_f f(y)$
  Если $f(x), f(y) \in \{C'_{t+a}, \ldots, C'_{t+b}\}$, отношение R_f сохраняется по определению, $z = y$, $xR_f y$, поэтому для $i \in (-\infty, t+a-2]$ $x \in C_i, y \in C_{i+1}$. Следовательно, $\exists z \in F[xR_f z$ и $f(z) = f(y)]$.

Если $f(x), f(y) \in C'_{t+a-1}$, отношение R_f сохраняется по определению, $z = y$, $xR_f y$. Следовательно, для $i \in (-\infty, t+a-2]$ $x \in C_i, y \in C_{i+1}$.

Если $f(x), f(y) \in C'_{t+b+1}$, отношение R_f сохраняется по определению, $z = y$, $xR_f y$. Следовательно, для $i \in [t+b+1, +\infty)$ $x \in C_i, y \in C_{i+1}$.

- Пусть $f(x)R'_p f(y)$. В этом случае рассуждения аналогичны предыдущему пункту.
- $\forall x \in Z_t\ \forall p \in Prop\ (x \in V(p) \Leftrightarrow f(x) \in V'(p))$ верно в силу сохранения оценки на р-морфном образе модели. Остаётся отметить, что $f(x) = x$ по определению отображения f.

$\square$

Доказанная теорема выше позволяет строить из бесконечных моделей конечные по времени р-морфные образы, сохраняя выполнимость формул, (см. [4] стр 62).

3.3. Фильтрация $p\mathcal{LTK}.sl$-моделей. Полученная р-морфная модель конечна по времени, однако мощность сгустков по-прежнему не ограничена. Используем метод фильтрации для построения конечной модели M''.

Пусть $M' = \langle Z'_t, R'_f, R'_p, R'_e, R'_1, \dots, R'_n, V'\rangle$ – р-морфный образ $p\mathcal{LTK}.sl$-модели, $\Phi \subseteq For(L^{p\mathcal{LTK}.sl})$ – набор формул, замкнутых относительно взятия подформул. Определим стандартным образом отношение эквивалентности $\equiv_\Phi$ на сгустках из Z'_t, $\forall t \in \mathbb{Z}, \forall x, y \in C'_t$:

$$x \equiv_\Phi y \Longleftrightarrow [\forall \alpha \in \Phi\ (\langle M', x\rangle \vDash \alpha \Leftrightarrow \langle M', y\rangle \vDash \alpha)].$$

Опишем полученную шкалу в результате фильтрации модели M' относительно такого отношения эквивалентности:

- $[x]_{\equiv_\Phi} := \{y \mid y \in \mathbb{Z}'{}_t, x \equiv_\Phi y\}$;
- $C''_i := \{[x]_{\equiv_\Phi} | x \in C'_i\}$;
- $\mathbb{Z}''_t := \bigcup\limits_{i=t+a-1}^{t+b+1} C''_i$.

Теперь определим означивание и отношения на полученном множестве классов эквивалентностей:

(1) $\forall p \in Var(\Phi)[V''(p) = \{[x]_{\equiv_\Phi} | \langle M', x\rangle \vDash p\}]$;
(2) $\forall x, y \in Z'_t, \forall l \in \{e, 1, \dots, n\}(xR'_l y \Rightarrow [x]_{\equiv_\Phi} R''_l [y]_{\equiv_\Phi})$;
(3) $\forall x, y \in Z'_t :$
  (a) $\forall l \in \{e, 1, \dots, n\}\ ([x]_{\equiv_\Phi} R''_l [y]_{\equiv_\Phi} \Longrightarrow [\forall \Box_l \alpha \in \Phi$ $\langle M', x\rangle \vDash \Box_l \alpha \Rightarrow \langle M', y\rangle \vDash \alpha])$;
  (b) $([x]_{\equiv_\Phi} R''_f [y]_{\equiv_\Phi} \Longrightarrow [\forall N\alpha \in \Phi \langle M', x\rangle \vDash N\alpha \Rightarrow \langle M', y\rangle \vDash \alpha])$;
  (c) $([x]_{\equiv_\Phi} R''_p [y]_{\equiv_\Phi} \Longrightarrow [\forall P\alpha \in \Phi \langle M', x\rangle \vDash P\alpha \Rightarrow \langle M', y\rangle \vDash \alpha])$.

Полученную в результате такой процедуры модель обозначим $M'' = \langle \mathbb{Z}''_t, R''_f, R''_p, R''_e, R''_1, \dots, R''_n, V''\rangle$ и будем называть *фильтрованной* моделью для модели M.

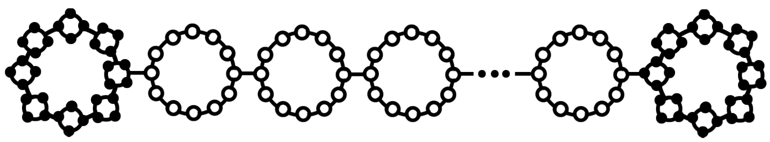

Рис. 3. Конечная по времени и сгусткам модель M''.

Лемма 1. *Пусть* $M' = \langle Z'_t, R'_f, R'_p, R'_e, R'_1, \dots, R'_n, V' \rangle$ *— p-морфный образ* $p\mathcal{LTK}.sl$*-модели,* $M'' = \langle \mathbb{Z}''_t, R''_f, R''_p, R''_e, R''_1, \dots, R''_n, V'' \rangle$ *— фильтрованная модель,* Φ *— множество всех подформул формулы* φ*. Тогда* $\forall \varphi \in For(L^{p\mathcal{LTK}.sl}), \forall z \in Z'_t$:

$$\langle M', x \rangle \vDash \alpha \Leftrightarrow \langle M'', [x]_{\equiv_\Phi} \rangle \vDash \alpha,$$

для любой подформулы $\alpha \in \Phi$.

Доказательство. Покажем индукцией по сложности формулы α.

Пусть $\langle M', x \rangle \vDash p \Rightarrow \exists [x]_{\equiv_\Phi} : V''(p) \ni [x]_{\equiv_\Phi}$. Пусть $\langle M'', x \rangle \vDash p \Rightarrow \forall x \in [x]_{\equiv_\Phi} : V(p) \ni [x]_{\equiv_\Phi}$. Получили $\langle M', x \rangle \vDash p \Leftrightarrow \langle M'', [x]_{\equiv_\Phi} \rangle \vDash p$.

Допустим, верно $\langle M', x \rangle \vDash \chi \Leftrightarrow \langle M'', [x]_{\equiv_\Phi} \rangle \vDash \chi$. Рассмотрим $\chi \wedge \theta$. Пусть $\langle M', x \rangle \vDash \chi \wedge \theta$, тогда по определению выполнимости $\langle M', x \rangle \vDash \chi$ и $\langle M', x \rangle \vDash \theta$, затем по индукционному предположению $\langle M'', [x]_{\equiv_\Phi} \rangle \vDash \chi$ и $\langle M'', [x]_{\equiv_\Phi} \rangle \vDash \theta$, снова по определению выполнимости получаем $\langle M'', [x]_{\equiv_\Phi} \rangle \vDash \theta$. Данная цепочка рассуждений верна и в обратную сторону в силу определений выполнимости.

Для случая $\chi \vee \theta$ доказательство аналогично рассуждениям выше.

Для $\neg\chi$ имеем: $\langle M', x \rangle \vDash \neg\chi$, по определению выполнимости $x \notin V'(p)$, значит по индуктивному предположению $\langle M'', [x]_{\equiv_\Phi} \rangle \nvDash \chi$, значит $\langle M'', [x]_{\equiv_\Phi} \rangle \vDash \neg\chi$.

Для импликации $\chi \to \theta$ остаётся только отметить тот факт, что в $p\mathcal{LTK}.sl$ $\chi \to \theta \sim \neg\chi \vee \theta$ и повторить рассуждения двух пунктов выше.

Теперь покажем для временных модальных операторов. Рассмотрим $N\chi$. Пусть $\langle M', x \rangle \vDash N\chi$, тогда $\forall y \in Z'_t : xR_f y$ $\langle M', y \rangle \vDash \chi$, по индукционному предположению $\langle M', [y]_{\equiv_\Phi} \rangle \vDash \chi$ и по выбору y имеем $\langle M'', x \rangle \vDash N\chi$. Аналогично для оператора P.

Для случая $\Box_i\chi$: если $\langle M', x \rangle \vDash \Box_i\chi$, тогда $\forall y \in Z'_t : xR_i y$, значит $\langle M', y \rangle \vDash \chi$ и так же по определению выполнимости и выбору y получим $\langle M'', x \rangle \vDash \Box_i\chi$.

$\Box$

4 Унификация в $p\mathcal{LTK}.sl$

Цель раздела — доказать, что логика $p\mathcal{LTK}.sl$ обладает унитарным типом унификации. Для этого мы получим вспомогательные результаты: установим слабый критерий унифицируемости формулы и покажем, что любая унифицируемая формула проективна, откуда непосредственно следует унитарность логики, [6]. Итак, введём основные определения.

Формула $\varphi(p_1, \ldots, p_s)$ называется *унифицируемой* в логике $\mathcal{L}$, если $\exists \sigma\colon p_i \mapsto \sigma_i$ — подстановка для каждой переменной $p_i \in Var(\varphi)$ (далее мы будем называть её *унификатором* формулы φ) такая, что $\sigma(\varphi) = \varphi(\sigma_1, \ldots, \sigma_s) \in \mathcal{L}$.

Корневым gu называется унификатор, получаемый подстановкой констант вместо переменных формулы (т.е. $gu : p_i \mapsto \{\top, \bot\}, \forall p_i \in Var(\varphi)$).

На множестве унификаторов определено отношение предпорядка: унификатор σ формулы $\varphi(p_1, \ldots, p_s)$ называется *более общим*, чем унификатор σ_1 в логике $\mathcal{L}$, если существует подстановка γ такая, что для любой переменной p_i: $\sigma_1(p_i) \equiv_{\mathcal{L}} \gamma(\sigma(p_i))$ (заданный таким образом предпорядок будем обозначать как $\sigma_1 \preceq \sigma$). По отношению $\preceq$ можно определить эквивалентность унификаторов для любой унифицируемой формулы φ: унификатор σ_1 называем *эквивалентным* унификатору σ_2 ($\sigma_1 \equiv \sigma_2$), если $\exists \gamma_1 : \sigma_1(p) \equiv_{\mathcal{L}} \gamma_1(\sigma_2(p))$ и $\exists \gamma_2 : \sigma_2(p) \equiv_{\mathcal{L}} \gamma_2(\sigma_1(p))$ для всех $p \in Var(\varphi)$.

Унификатор σ формулы $\varphi(p_1, \ldots, p_s)$ называется *максимальным*, если для любого другого унификатора σ_i выполняется $\sigma_i \preceq \sigma$ или же они несравнимы, т.е. ($\sigma_i \npreceq \sigma$) и ($\sigma \npreceq \sigma_i$). Если σ более общий, чем любой другой, он называется *наиболее общим* (сокращённо *н.о.у.*).

Проективной в $\mathcal{L}$ назовём формулу $\varphi(p_1, \ldots, p_s)$ для которой найдётся унификатор τ, удовлетворяющий условию: $\varphi \vdash_{\mathcal{L}} [p_i \equiv \tau(p_i)]$ для всех переменных $p_i \in Var(\varphi)$. Такой унификатор τ будем называть *проективным*.

Заметим, что выполнимость φ на всех мирах модели эквивалентна выполнимости $\zeta_j(\varphi)$, где

$$\zeta_j(\varphi) = \bigwedge_{i=0}^{t_j} P^i \varphi \wedge \bigwedge_{i=0}^{t_j} N^i \varphi,\ t_j \in \mathbb{N}.$$

Принимая, что t_j пробегают всевозможные значения, мы потребуем выполнимость формулы φ на всей модели. В силу показанной далее эквивалентности таких подстановок, интервал t_j для проверки формулы может быть любым фиксированным числом. Диапазон для проверки, получаемый выбором какого-то значения t_j будем обозначать Δ_j, и он будет включать в себя все сгустки $C_t \in [C_{t-t_j}; C_{t+t_j}]$. Такая конструкция будет использоваться в рассуждениях доказательств нижеследующих теорем.

Говорим, что логика $\mathcal{L}$ обладает *унитарным* типом унификации, если каждая унифицируемая формула в $\mathcal{L}$ имеет н.о.у.

Сформулируем и докажем следующую лемму, гарантирующую константную оценку истинности формул в логике, необходимую при дальнейших рассуждениях о существовании корневых унификаторов.

Лемма 2. *Для любой формулы $\varphi(p_1, \ldots, p_s)$ в языке $L^{p\mathcal{LTK}.sl}$ и любого набора $c_1, \ldots, c_s \in \{\top, \bot\}$ существует константа $c \in \{\top, \bot\}$ такая,* *что на любой $p\mathcal{LTK}.sl$-модели M*

$$\forall x \in Z_t \ \langle M, x\rangle \models \varphi(c_1, \dots, c_s) \equiv c.$$

Доказательство. Покажем индукцией по сложности формулы φ. При $\varphi = p$ получаем $\varphi(c) = c$, для любой модели M утверждение выполняется. Предположим, что утверждение верно для формул $\chi = \chi(p_1, \dots, p_s)$ и $\psi = \psi(p_1, \dots, p_s)$. Рассмотрим более сложные формулы. Если $\varphi(p_1, \dots, p_s) = \chi(p_1, \dots, p_s) \wedge \psi(p_1, \dots, p_s)$, то при подстановке любого набора констант $(c_1, \dots, c_s)$ верно $\varphi = c' \wedge c'' = c$ по определению выполнимости $\wedge$. Для остальных немодальных связок ход доказательства аналогичен.

Пусть теперь $\varphi = N\psi$. По определению $p\mathcal{LTK}.sl$ модели $\exists y : xR_f y$. Рассмотрим все возможные случаи относительно выполнимости N. Пусть набор констант $(c_1, \dots, c_s)$ такой, что $\forall y : xR_f y$ выполняется $\langle M, y\rangle \models \psi(c_1, \dots, c_s)$. Тогда, по определению выполнимости N, $\langle M, x\rangle \models N\psi(c_1, \dots, c_s)$ и $\langle M, x\rangle \models (N\psi \equiv \top)$. Пусть теперь набор $(c_1, \dots, c_s)$ такой, что $\exists y \models \neg\psi(c_1, \dots, c_s)$, тогда $\langle M, x\rangle \models \neg N\psi(c_1, \dots, c_s)$ и $\langle M, x\rangle \models (N\psi \equiv \bot)$. Для случая $\varphi = P\psi$ доказательство аналогично в силу того, что $\exists y : xR_p y$.

Для $\varphi = \Box_i\psi$ утверждение леммы очевидно выполняется, в силу сериальности отношений R_i модель не содержит граничных миров, где выполняется $\Box_i\psi$, при $\psi \equiv \bot$, и опровергается $\Diamond_i\psi$, при $\psi \equiv \top$. $\Box$

Теорема 2. *Формула φ унифицируема в $p\mathcal{LTK}.sl$ тогда и только тогда, когда φ имеет корневой унификатор.*

Доказательство. Пусть φ унифицируема, тогда найдётся $\sigma : p_i \to \sigma_i$, $\varphi(\sigma_1, \dots, \sigma_r) \in L$. Покажем, что все σ_i могут являться константами. Так как $\varphi(\sigma_1, \dots, \sigma_r) \in L$, $\sigma(\varphi)$ истинна на всех моделях логики, следовательно $\forall x\langle M, x\rangle \models \sigma(\varphi)$ получаем $\forall x\langle M, x\rangle \models \sigma(\varphi) \equiv \top$, значит в силу предыдущей леммы можем зафиксировать набор констант $c_1, \dots, c_r$. Получили, что среди всех унификаторов $\exists\sigma : p_i \to c_i$, этот унификатор σ и есть искомый корневой унификатор gu.

Пусть φ имеет корневой унификатор gu, тогда она унифицируема по определению. $\Box$

Критерий, доказанный выше, можно считать слабым, т.к., с одной стороны, он позволяет устанавливать унифицируемость формул в логике, с другой — с очевидностью указывает на унифицируемость формулы, если для неё уже найден корневой унификатор.

Теорема 3. *Любая унифицируемая формула в $p\mathcal{LTK}.sl$ проективна.*

Доказательство. Пусть φ унифицируема в логике $p\mathcal{LTK}.sl$. Рассмотрим подстановку вида:

$$\sigma(p_i) = (\zeta_j(\varphi) \wedge p_i) \vee (\neg\zeta_j(\varphi) \wedge gu(p_i)),$$

где ζ_j фиксирует диапазон проверки формулы φ на любой модели M. Корневой унификатор gu существует в силу унифицируемости φ и предыдущей теоремы. Докажем, что эта подстановка σ является унификатором для φ. Сначала отметим, что если σ – унификатор для φ, то $\forall x \langle M, x\rangle \vDash \sigma(\varphi)$.

1. Пусть $\forall x \in \Delta_j : \langle M, x\rangle \vDash \varphi$. Тогда $\langle M, x\rangle \vDash \zeta_j(\varphi)$, поэтому второй дизъюнкт подстановки опровергается на x. Рассмотрим первый дизъюнкт:

- если $\langle M, x\rangle \vDash p_i$, тогда $\langle M, x\rangle \vDash \zeta_j(\varphi) \wedge p_i$, значит $\langle M, x\rangle \vDash \sigma(p_i)$.
- если $\langle M, x\rangle \nvDash p_i$, тогда $\langle M, x\rangle \nvDash \zeta_j(\varphi) \wedge p_i$, значит $\langle M, x\rangle \nvDash \sigma(p_i)$.

Следовательно, $\sigma(p_i) \equiv p_i$. Так как $\varphi(p_1, \ldots, p_r)$ истинна на мире x, то $\varphi(\sigma(p_1), \ldots, \sigma(p_r))$ истинна на x. В этом случае $\langle M, x\rangle \vDash \sigma(\varphi)$.

2. Пусть $\exists x \in \Delta : \langle M, x\rangle \nvDash \varphi$. Тогда $\langle M, x\rangle \nvDash \zeta_j(\varphi)$, поэтому первый дизъюнкт подстановки опровергается на x. Рассмотрим второй дизъюнкт: $\langle M, x\rangle \vDash \neg\zeta_j(\varphi)$, тогда второй дизъюнкт имеет вид $\top \wedge gu(p_i)$ и справедливо $\sigma(p_i) = \bot \vee (\top \wedge gu(p_i)) \Rightarrow \sigma(p_i) = gu(p_i)$. В силу того, что $\langle M, x\rangle \vDash gu(\varphi)$, то $\langle M, x\rangle \vDash \sigma(\varphi)$.

Получаем, что σ – унификатор φ. Теперь покажем проективность унификатора σ. По определению проективности, должно выполняться

$$[\zeta_j(\varphi) \to (p_i \equiv \sigma(p_i))].$$

Если $\zeta_j(\varphi)$ истинна на x, то выполняется $p_i \equiv \sigma(p_i)$, в условиях пункта 1 выше. Если $\zeta_j(\varphi)$ ложна на x, то импликация $\zeta_j(\varphi) \to (p_i \equiv \sigma(p_i))$ выполняется. Следовательно, σ – проективный унификатор для φ.

□

Отметим, что в силу произвольности выбора индекса j в $\zeta_j(\varphi)$, проективным является целый бесконечный набор подстановок вида σ. Покажем, что такие унификаторы образуют класс эквивалентных между собой унификаторов.

Теорема 4. *Для любых t_1 и $t_2 \in \omega$ справедливо $\tau_{t_1} \equiv \tau_{t_2}$, где $\tau_{t_j} = (\zeta_j(\varphi) \wedge p_i) \vee (\neg\zeta_j(\varphi) \wedge gu(p_i))$ и $\zeta_j(\varphi) = \bigwedge\limits_{i=0}^{t_j} P^i\varphi \wedge \bigwedge\limits_{i=0}^{t_j} N^i\varphi$.*

Доказательство. Чтобы показать, что унификаторы τ_{t_1} и τ_{t_2} эквивалентны, по определению, достаточно найти подстановки γ_1 и γ_2 соответственно, взаимно выражающие унификаторы друг из друга. В качестве таких подстановок рассмотрим сами проективные унификаторы. Рассмотрим только первый случай: применим в качестве γ_1 унификатор τ_{t_1} к τ_{t_2}:

$$\tau_{t_1}(\tau_{t_2}(p_j)) = (\tau_{t_1}(\zeta_2(\varphi)) \wedge \tau_{t_1}(p_j)) \vee (\neg\,(\tau_{t_1}(\zeta_2(\varphi))) \wedge \tau_{t_1}(gu(p_j))) = (*)$$

где $\tau_{k_1}(\zeta_2(\varphi)) = \bigwedge\limits_{i=0}^{t_j} \tau_{k_1}(P^i\varphi) \wedge \bigwedge\limits_{i=0}^{t_j} \tau_{k_1}(N^i\varphi)$,
$\tau_{k_1}(N^i\varphi) = N^i\varphi(\tau_{k_1}(p_1), \dots, \tau_{k_1}(p_s))$, $\tau_{k_1}(P^i\varphi) = P^i\varphi(\tau_{k_1}(p_1), \dots, \tau_{k_1}(p_s))$.

Поскольку τ_{k_1} — унификатор φ, получаем:

$$(*) = (\top \wedge \tau_{k_1}(p_j)) \vee (\neg(\top) \wedge \tau_{k_1}(gu(p_j))) \equiv \tau_{k_1}(p_j).$$

Следовательно, $\forall p_j \in Var(\varphi(p_1, \dots, p_s))\ \tau_{t_1}(\tau_{t_2}(p_j)) \equiv \tau_{t_1}(p_j)$, а значит $\tau_{t_1} \preceq \tau_{t_2}$. В силу произвольности выбора значений t_1 и t_2, аналогично $\tau_{t_2} \preceq \tau_{t_1}$, следовательно, $\tau_{t_2} \equiv \tau_{t_1}$. □

Гиларди предложен подход, использующий понятие проективности для доказательства унитарного типа унификации, им доказаны следующие два утверждения.

Лемма 3. [6] *Если φ — проективно унифицируемая формула в L и τ — её проективный унификатор, то τ — н.о.у. формулы φ.*

Следствие 1. [6] *Проективность унификации в L наследует унитарный тип унификации.*

В силу доказанного выше можем утверждать унитарный тип унификации в логике $p\mathcal{LTK}.sl$, причём проективная подстановка σ предлагает схему построения н.о.у. для любой унифицируемой в логике формулы.

5 Заключение

Представленное в статье исследование логики $p\mathcal{LTK}.sl$ демонстрирует, что класс линейных логик знания ступенчатого времени с операторами прошлого и будущего образует содержательную и математически корректную альтернативу широко используемой линейной временной логике $\mathcal{LTL}$. Принципиальное отличие ступенчатого подхода заключается в нетранзитивной и нерефлексивной природе временного потока, что позволяет моделировать вычислительные процессы с любой требуемой степенью детализации – от уровня планковского кванта времени в физических системах, [3], до такта работы процессора в задачах верификации программного обеспечения. В отличие от классической $\mathcal{LTL}$, где время представляет собой транзитивный и рефлексивный линейный порядок, ступенчатая модель точнее отражает дискретную природу реальных систем, в которых целостность передачи данных не гарантирована, а доступность информации может быть жёстко ограничена конкретными временными промежутками.

Вместе с тем, чтобы рассматривать логики класса $\mathcal{LTK}.sl$ как полноценную альтернативу $\mathcal{LTL}$, необходимо продолжение систематического исследования их фундаментальных свойств. Проведённый в данной статье анализ финитной аппроксимируемости и унитарного типа унификации для $\mathcal{LTK}.sl$ закладывает теоретический фундамент, и остаются актуальными вопросы разрешимости логики на основе полученных результатов и построения аксиоматизации. Отдельный интерес представляет исследование вариантов логики с различными ограничениями на отношения агентов (например, их взаимнозависимость), что потенциально расширяет выразительные возможности языка для спецификации свойств программных и аппаратных систем.

References

[1] S. I. Bashmakov, T. Yu. Zvereva, *Linear step-like logic of knowledge LTK.sl*, Siberian Electronic Mathematical Reports, 20:2 (2023), 1361–1373.

[2] S. I. Bashmakov, K. A. Smelykh, *Relational version of multi-agent computation tree logic CTLK*, Bulletin of Irkutsk State University. Series Mathematics, 47 (2024), 78–92.

[3] S. R. Beane, Z. Davoudi, M. J. Savage, *Constraints on the universe as a numerical simulation*, The European Physical Journal A, 50, 148 (2014).

[4] P. Blackburn, M. de Rijke, Y. Venema, *Modal Logic*, Cambridge University Press, 2001, 554.

[5] W. Dzik, P. Wojtylak, *Projective unification in modal logic*, Logic Journal of the IGPL, 20:1 (2012), 121–153.

[6] S. Ghilardi, *Unification through projectivity*, Journal of Logic and Computation, 7:6 (1997), 733–752.

[7] T. A. Novikova, V. A. Zakharov, *Two-sided program unification and its application to refactoring problems*, Proceedings of the Institute for System Programming of the RAS, 26:2 (2014), 245–268.

[8] S. P. Odintsov, S. O. Speranski, S. A. Drobyshevich, *Introduction to non-classical logics*, Novosibirsk, RITS NGU, 2014, 132.

[9] C. Stirling, *Modal and temporal logics*, LFCS, Department of Computer Science, University of Edinburgh, 1991, 477–563.

[10] M. Y. Vardi, *An automata-theoretic approach to linear temporal logic, Logics for concurrency: structure versus automata*, Logics for concurrency: structure versus automata, Berlin, Heidelberg, Springer Berlin Heidelberg, 1996, 238–266.

[11] V. F. Yun, *Polymodal logic of the class of inductive linear time frames*, Siberian Electronic Mathematical Reports, 12 (2015), 421–431.

Stepan Igorevich Bashmakov
Siberian Federal University,
pr. Svobodny, 79,
600041, Krasnoyarsk, Russia
Email address: krauder@mail.ru

Yulia Alexandrovna Bushakova
Siberian Federal University,
pr. Svobodny, 79,
600041, Krasnoyarsk, Russia
Email address: u_bushakova@vk.com