Московский Государственный Университет
имени М.В. Ломоносова
Российская Академия Наук
Международная Академия Технологических Наук
Российская Академия Естественных Наук

Интеллектуальные Системы.

Теория и приложения

ТОМ **23** ВЫПУСК 1 * **2019**

МОСКВА

ISSN 2411–4448

УДК 519.95; 007:159.955
ББК 32.81

Издается с 1996 г.*

Главный редактор: д.ф.-м.н., профессор В. Б. Кудрявцев

Редакционная коллегия:

д.ф.-м.н., проф. А. Е. Андреев **(зам. главного редактора)**
д.ф.-м.н., проф. Э. Э. Гасанов **(зам. главного редактора)**
к.ф.-м.н., доц. А. С. Строгалов **(зам. главного редактора)**
к.ф.-м.н., м.н.с. В. В. Осокин **(ответственный секретарь)**
д.ф.-м.н., проф. В. В. Александров, д.ф.-м.н., проф. С. В. Алешин, д.ф.-м.н., проф. Д. Н. Бабин, академик РАН, д.ф.-м.н., проф. Ю. Л. Ершов, академик РАН, д.ф.-м.н., проф. Ю. И. Журавлев, д.ф.-м.н., проф. В. Н. Козлов, чл.-корр. РАН, д.ф.-м.н., проф. А. В. Михалев, к.ф.-м.н., проф. В. А. Носов, д.ф.-м.н., проф. А. С. Подколзин, д.т.н., проф. Д. А. Поспелов, д.ф.-м.н., проф. Ю. П. Пытьев, академик РАН, д.т.н., проф. А. С. Сигов, д.ф.-м.н., проф. А. В. Чечкин

Международный научный совет журнала:

С. Н. Васильев (Россия), К. Вашик (Германия), В. В. Величенко (Россия), А. И. Галушкин (Россия), И. В. Голубятников (Россия), Я. Деметрович (Венгрия), Л. Заде (США), Г. Килибарда (Сербия), Ж. Кнап (Словения), П. С. Краснощеков (Россия), А. Нозаки (Япония), В. Н. Редько (Украина), И. Розенберг (Канада), А. П. Рыжов (Россия) — ученый секретарь совета, А. Саломаа (Финляндия), С. Саксида (Словения), Б. Тальхайм (Германия), Ш. Ушчумлич (Сербия), Фан Дин Зиеу (Вьетнам), А. Шайеб (Сирия), Р. Шчепанович (США), Г. Циммерман (Германия)

Секретари редакции: И. О. Бергер, М. А. Ильгова, А. А. Коровин

В журнале «Интеллектуальные системы. Теория и приложения» публикуются научные достижения в области теории и приложений интеллектуальных систем, новых информационных технологий и компьютерных наук.

Издание журнала осуществляется под эгидой МГУ им. М. В. Ломоносова, Научного Совета по комплексной проблеме «Кибернетика» РАН, Отделения «Математическое моделирование технологических процессов» АТН РФ, Секции «Информатики и кибернетики» РАЕН.

Учредитель журнала: ООО «Интеллектуальные системы».

Журнал входит в список изданий, включенных ВАК РФ в реестр публикаций материалов по кандидатским и докторским диссертациям по математике и механике.

Спонсором издания является:

ООО «Два Облака»
Разработка корпоративных информационных систем
http://www.dvaoblaka.ru

Индекс подписки на журнал: 64559 в каталоге НТИ «Роспечать».

Адрес редакции: 119899, Россия, Москва, Воробьевы Горы, МГУ, ГЗ, механико-математический факультет, комн. 12-01.
Адрес издателя: 115230, Россия, Москва, Хлебозаводский проезд, д. 7, стр. 9, офис 9. Тел. +7 (495) 939-46-37, e-mail: mail@intsysjournal.org

*) Прежнее название журнала: «Интеллектуальные системы».

© ООО «Интеллектуальные системы», 2019.

ОГЛАВЛЕНИЕ

Часть 1.
Общие проблемы теории интеллектуальных систем

Алгоритм обучения систем с дискретным управлением

Голиков К.А.

Разработан алгоритм обучения для задачи позиционирования систем с дискретным управлением, основанный на методе обобщения проб и ошибок, сохраняющихся в базе данных, с помощью глобальной интерполяции и градиентного спуска. Оптимизация алгоритма производится по критерию сокращения времени обучения (числа попыток). Алгоритм был протестирован на симуляторе для моделей систем, действующих на плоскости, двух разных типов: для мобильного робота с двумя ведущими гусеницами и для открытой кинематической цепи с вращательными и призматическими сочленениями.

Ключевые слова: позиционирование, алгоритм обучения, робот, интерполяция, аппроксимация.

Система работает в дискретном времени $\mathbf{t}$. В каждый момент времени система находится в одном из своих состояний:

$$s(t) = (x(t), \dot{x}(t))$$

Состояние системы можно разбить на две составляющие: *статическое состояние* $x(t)$, известно с достаточной точностью, и *динамическое состояние* $\dot{x}(t)$ – неизвестно.

В качестве наглядного изображения системы с дискретным управлением рассматриваем модели роботов разных конструкций, оперирующих на плоскости (x, y).

Статическое состояние системы может быть описано позициями n точек на плоскости $x(t)=(x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$ и их массами $m=(m_1, \ldots, m_n), m_k=const, k=\overline{1,n}$. Кроме того, в системе имеются *приводы*, которые действуют на точки системы и обеспечивают изменение состояния системы во времени – движения системы. В позициях $(n-1)$ точек системы закреплены по два привода, назовём эти точки *сочленениями*, а соответствующие сочленению два привода – *парой приводов-антагонистов сочленения.* Оставшаяся одна точка будет той, которой

нужно научиться управлять, она не имеет приводов, назовём её *особой точкой* робота (например, представляет собой центр захвата манипулятора или центр мобильного робота). Таким образом, в системе имеется $m{=}2(n{-}1)$ приводов, ими можно управлять, $(n-1)$ сочленение с координатами $((x_1, y_1), (x_2, y_2), \ldots, (x_{n-1}, y_{n-1}))$ и особая точка (x_n, y_n).

На каждый привод независимо можно подавать *дискретное управление* $u_k(t){\in}\{0,1\}, k{=}\overline{1,m}$, которое означает, действует ли привод или выключен в текущий момент. Существует специальный вход $reset{\in}\{0,1\}$, который мгновенно возвращает систему в известное *начальное положение* (абсолютно точно) из любого её состояния. Тогда общее управление системы записывается в виде:

$$u(t) = (u_1(t), u_2(t), \ldots, u_m(t), reset) \in \{0,1\}^{m+1}$$

Динамическое состояние системы – это скорости **n** точек системы $\dot{x}(t)$, которые слагаются из нескольких составляющих: усилий приводов и остальных сил, действующих на робота, таких как инерция, кориолисовы/центростремительные силы, гравитация и другие вне-модельные силы.

Пробой, попыткой, полным действием или *эпизодом обучения* будет фиксирование наблюдаемых состояний системы $s(t)$ в заданный промежуток времени $t_0 < t < t_1$ от известного начального состояния s_0 в некоторое конечное $s(t_1) = (x(t_1), 0)$, при чём конечное состояние должно быть с нулевой динамической частью, т.е. установившимся (после конца эпизода предполагается использовать кнопку $reset$).

Данные эпизода формируют одну запись в *базе данных* — $t_0 < t < t_1$
1. управление $u(t)$
2. траектория движения особой точки системы $x(t)$
3. смещения сочленений $q(t)$
4. приближение динамики $\dot{q}(t)$, $\dot{x}(t)$

ПОСТАНОВКА ЗАДАЧИ

Итак, есть система

$$s(t) = (x(t), \dot{x(t)}), \qquad (s(t), u(t)) \to s(t{+}1)$$

в каждый момент t на её входы подаются *управляющие сигналы* $u(t)$, которые задаются бинарной кусочно-постоянной вектор-функцией, включают приводы, которые изменяют состояние системы $s(t)$. Частично

определить характеристики состояния системы в каждый момент t можно при помощи статической части – данных *обратной связи* $x(t)$ – вещественнозначной вектор-функции, представляющей собой координаты точек в глобальной неподвижной системе отсчёта. Умышленно отказываемся от возможности аналитически построить точную или приближённую физическую модель системы, полагаемся только на опыты и обратную связь – это принципиальное ограничение нашего исследования.

ЗАДАЧИ:
1. За минимальное время произвести ценные действия с помощью приводов системы, отражающие возможности и принципы функционирования этой системы. Получить управления, хорошо подходящие для обобщения и предсказания поведения системы для решения поставленных задач.
В рамках задачи позиционирования: определить равномерную плотную решётку целевых точек, получить за минимальное время множество движений робота, покрывающих конечными установившимися положениями особой точки системы все целевые точки из начального положения с нужной точностью, отразить накопленный опыт в базе данных движений наиболее полным образом.

2. Выработать подход на основе интерполяции и градиентного спуска, позволяющий системе с накопленной базой действий производить новые действия, помещающие конечное положение особой точки системы в *недостигнутые ранее цели* с заданной точностью в области рабочего пространства, плотно исследованной на прошлом этапе. *Получить аппроксимацию функции*, отображающую принцип функционирования системы с учётом неизвестной части состояния системы по сохранённым в базе данных действиям.
В рамках задачи позиционирования: аппроксимацию функции обратной кинематики с учётом неизвестной динамики.

3. Обеспечить *в условиях изменяющегося динамического функционирования* системы (подмены действующих сил) *повторяемость* с заданной точностью ранее достигнутых целевых точек. В реальном времени обеспечить *коррекцию действия*, минимизирующую конечное отклонение от достигнутой ранее цели в плотно изученной области действия робота. Разработать быстрый метод обновления аппроксимации при получении новых данных, учитывающий уменьшение актуальности старого опыта.

Обучение системы отрабатывается в виртуальной среде, для задачи позиционирования роботов двух разных конструкций со своими особенностями. Эксперименты проводились с моделями реальных роботов с количеством приводов m от 4 до 10. Число переключений активных приводов k в управлениях эпизодов обучения находилось в пределах 4-8, а максимальная длительность эпизода N=10000 тактов. Попытки делаем не очень длительные. Предполагаем, что если в $u(t)$ общее число последовательностей единиц не велико, не превышает натурального k ($k << N$), перемещение особой точки получится не очень сложным[1].

Рассмотрим свойства первой задачи исследования: самым дорогим ресурсом при обучении робота является время реализации попытки, попыток нужно делать минимальное число, но при этом они должны быть максимально полезными для обучения. Это сделать сложно, потому что в неизученной обасти результат попытки и вид получаемого действия неизвестен.

Цели позиционирования при обучении на первом этапе выбираются произвольным равномерным образом в интересующей области, в которой будут ставиться задачи позиционирования при эксплуатации механизма. Целевая точка является центром небольшой окрестности, в соответствии с заданной наперёд точностью позиционирования, если конечное установившееся положение особой точки системы находится в этой окрестности, говорим, что робот *попал в цель*. Пробуя попадать в конкретные цели, система в итоге учится попадать в любые точки.

Решением второй задачи данного исследования разработки алгоритма обучения будут два метода аппроксимации: *глобальная интерполяция случайными функциями 2-функционала* динамической части состояния системы и *приближение попытками к целевой точке от ближайшей известной в направлении градиента*. С результатами экспериментов можно ознакомиться в приведённых таблицах в конце данной работы.

Говорим, что система *повторила* действие, если конечное установившееся положение особой точки системы оказалось в допустимой окрестности целевой точки, вне зависимости от траектории, которую эта точка реально описала.

Для третьей задачи исследования сформулируем математическое представление изучаемых изменений и шумовых погрешностей в динамическом состоянии системы при эксплуатации робота для обновления алгоритма в будущем.

[1]Энерго-эффективные траектории – когда от положения покоя отклоняются только те сочленения, которые нужны, чтобы попасть в цель

МОДЕЛЬ РОБОТА МАНИПУЛЯТОРА

1. *Особая точка* – центр захвата манипулятора. Критерий достижения цели – захват достиг цели (без учёта ориентации).
2. *Привод* – смыкающее либо размыкающие действие вращательного сочленения или выдвигающее либо сдвигающее действие призматического сочленения.
3. $n = 3$–5 точек системы с массами m_1–m_n, 2–4 сочленения, $m = 4$–8 приводов.

Модель движения идеализирована:
1. Без проскальзываний в сочленениях
2. Без упругих связок и звеньев
3. Без изгибания осей суставов
и т.д.

Будем рассматривать плоские манипуляторы, т.е. захват может двигаться только на плоскости (x, y).

Рис. 1. Модель манипулятора

ОСОБЕННОСТИ КИНЕМАТИКИ МАНИПУЛЯТОРОВ

Кинематика – геометрия движения, ветвь классической механики, описывает движение точек, тел и систем тел без рассмотрения причин их движения. В области кинематики роботов изучают взаимосвязь между конфигурациями сочленений q и точками декартового *оперативного (рабочего) пространства* x, пространства захвата, его возможные положения и ориентации. *Пространство сочленений* $q = (q_1, \ldots, q_n)$ – набор смещений в сочленениях различной структуры.

$$q \in S_q,$$

где S_q – разрешённые конфигурации, в реальных манипуляторах допускаются не все положения суставов, в механизме твёрдые элементы не могут проходить сквозь друг друга.

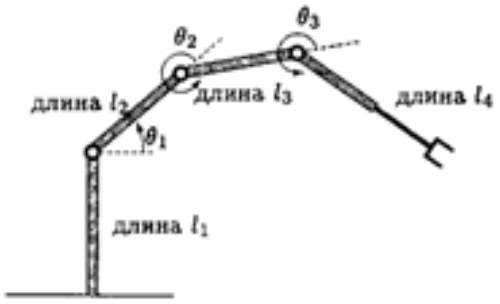


Рис. 2. Определение внутренних координат сочленений

Рассмотрим рисунок 2, начало координат рабочего пространства робота расположено в первом подвижном сочленении у основания кинетической цепи, углы вычисляются против часовой стрелки, начиная от условного продолжения предыдущего звена, а самый первый угол – начиная от оси x глобальных координат.

Задача прямой кинематики: при заданных положениях сочленений q определить положение центра захвата x в рабочем пространстве.

$$x = T\ q \tag{1}$$

Общая матрица прямого преобразования T из q в x имеет вид:

$$T = \left[\begin{array}{c|c} {}^{n}_{1}R & {}^{n}_{1}P \\ \hline 0 & 1 \end{array}\right], \tag{2}$$

где ${}^{n}_{1}R$ – матрица вращения (2×2, а в 3D-пространстве 3×3), задаёт ориентацию координат при переходе между системами отсчёта ближайших сочленений, здесь все повороты от захвата q^n к базе q^1, ${}^{n}_{1}P$ – столбец переноса положения начала системы отсчёта следующего сочленения относительно предыдущего, в нашем случае, учтены все смещения сочленений. Данный столбец даёт нам в общем виде формулы перевода из координат сочленений манипулятора в глобальные координаты положения захвата:

$${}^{n}_{1}P = f^n(l,\theta) = \begin{pmatrix} \sum_{i=1}^{n-1} l_{i+1} cos(\sum_{j=1}^{i} \theta_j) \\ \sum_{i=1}^{n-1} l_{i+1} sin(\sum_{j=1}^{i} \theta_j) \\ \sum_{i=1}^{n} \theta_i \end{pmatrix} \begin{matrix} -x \\ -y \\ -\text{ориентация} \end{matrix} \tag{3}$$

Параметры:

1. Углы $\theta = (\theta_1, \ldots, \theta_n)$ – поворотные соединения в порядке от основания к захвату (не все сочленения могут быть поворотными). θ_1 – ориентация сочленения у основания, θ_n – ориентация захвата. Углы могут иметь строгие ограничения – $\Theta_{k1} < \theta_k <= \Theta_{k2}$, диапазон измерения.

2. Длины $= (l_{k_1}, \ldots, l_{k_z})$ – некоторые сочленения могут быть призматическими и менять свою длину l_k в диапазоне от l_{k1} до l_{k2}.

Если функция $f^n(l,\theta)$ даёт положение захвата и его ориентацию в глобальных координатах, то функции $f^i(l,\theta), i = \overline{1,n}$ – положения промежуточных i-х точек-сочленений в глобальных координатах.

Т.е. формулы $f^n(l,\theta)$ достаточно, чтобы управлять захватом манипулятора относительно координат сочленений. На практике обычно цели манипуляторам ставятся в оперативном пространстве, в координатах положения захвата. В этом случае приходится решать более сложную задачу обратной кинематики манипулятора.

ОБРАТНАЯ КИНЕМАТИКА МАНИПУЛЯТОРОВ

Обратное отображение – это отображение из пространства сочленений в рабочее пространство робота:

$$q = f^{-1}(x)$$

Решения обратной кинематики:
1. Аналитические методы — решение, полученное напрямую в виде уравнения, предпочтительнее, но в общем виде неразрешимо для достаточно сложных реально существующих механизмов
2. Численные методы — сложно-вычислимы, неточны, являются приближениями, но это единственный практичный вариант решения в общем случае.

Распространённые методы численных решений обратной кинематики манипулятора:
1. Циклический координатный спуск (CCD – Cyclic Coordinate Descent)
2. Псевдо-обратный расширенный Якобиан

Якобиан или Матрица Якоби - инструмент, который широко используется в робототехнике и теории управления, определяет динамические отношения между двумя различными представлениями системы.

$$\dot{x} = \dot{T}(q)\ \dot{q}, \quad \dot{x} = J\ \dot{q}, \quad J = \dot{T},$$

Якобиан является матрицей частных производных, позволяет рассчитывать управление в силах (крутящих моментах) для обеспечения необходимых положения и скорости захвата. Может связывать не только скорости, но и ускорения

$$\ddot{x} = \dot{J(q)}\ \dot{q} + J(q)\ddot{q}$$

Обратный Якобиан определяет изменения в q относительно x:

$$\dot{q} = J^{-1}\dot{x}$$

Манипулятор с пространством сочленений q, рабочим пространством x, кинематическим отображением $f : q \to x$ *кинематически избыточен*, если $\mathbf{dim}\ q > \mathbf{dim}\ x$. Существует много путей в пространстве сочленений, соответствующих одному пути захвата в рабочем пространстве.
$dim\ q$ - размерность пространства сочленений, каждое сочленение плоского робота добавляет 1 размерность.
$dim\ x$ - размерность рабочего пространства, позиция и ориентация=2+1

Почти все реальные роботы манипуляторы кинематически избыточны - имеют большее количество сочленений, чем минимально требуется для выполнения их задач. Данное свойство вызывает проблемы при

выводе функции управления роботом - обратной кинематики, которая приобретает строгие ограничения применимости и неизбежно включает в себя кинематические особенности, а выполняемые траектории становятся «неповторяемы».

Точка называется *кинематической особенностью* робота, если Якобиан в ней имеет не полный ранг: $rang\ J < \min(dim\ q, dim\ r)$. Когда конфигурация манипулятора близка к кинематической особенности, для умеренных скоростей захвата требуются очень большие скорости соединений. Таким образом, нужно определять траектории для сочленений, избегающие с достаточным запасом особенностей.

Если пространство сочленений избыточно по сравнению с пространством захвата, т.е. система переопределена или слабо обусловлена, то Якобиан не является квадратной матрицей, не имеет обратной матрицы. Нет гарантий, что Якобиан обратим, вместо обратного Якобиана необходимо взять *псевдо-обратный*.

$$J^+ = (J^T J)^{-1} J^T$$

$J^T J$ по определению квадратный, т.е., возможно, *обратимый*.

Если Якобиан – это линейное приближение задачи, а псевдообратный Якобиан – приближение решения неразрешимой задачи, т.е. приближение приближения, почему же просто не использовать J^T, который легко вычислять, для нахождения смещения $\dot{q}$? Потому что использование J^+ ускоряет сходимость к цели.

Виртуальными перемещениями стремимся к правильному решению, совершая небольшие шаги, в сторону минимизации ошибки:

$$Error = |(I - J^+ J)\dot{x}|$$

КОНКРЕТНАЯ МОДЕЛЬ ПЛОСКОГО МАНИПУЛЯТОРА

В модели руки, которую мы моделируем и тестируем есть максимум 4 сочленения: призматическое и три вращательных сочленения, обозначим:

$$q = (q_1, q_2, q_3, q_4) = (\phi, \psi, \theta, l_3)$$

Сочленения манипулятора имеют ограничения:

$$\begin{array}{ll} l_0 \approx 0.07 & -58^\circ \leq \varphi \leq 47^\circ \\ l_1 \approx 0.9575 & 56^\circ \leq \psi \leq 191^\circ \\ l_2 \approx 0.8846 & -45^\circ \leq \theta \leq 55^\circ \\ l_3 \approx 0 \div 0.4 & \end{array}$$

Возможность отключать и подключать сочленения и приводы позволит нам более фактурно рассмотреть данную проблему обучения. Можно записать матрицы перехода к локальным координатам сочленений:

$$A_1 = \begin{pmatrix} 1 & 0 & l_3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \text{ключица}$$

$$A_2 = \begin{pmatrix} c_\phi & -s_\phi & l_2 c_\phi \\ s_\phi & c_\phi & l_2 s_\phi \\ 0 & 0 & 1 \end{pmatrix} - \text{плечо}$$

$$A_2 = \begin{pmatrix} c & -s_\psi & l_1 c \\ s & c & l_1 s \\ 0 & 0 & 1 \end{pmatrix} - \text{локоть}$$

$$A_2 = \begin{pmatrix} c_\theta & -s_\theta & l_0 c_\theta \\ s_\theta & c_\theta & l_0 s_\theta \\ 0 & 0 & 1 \end{pmatrix} - \text{запястье}$$

Матрица прямого преобразования [2] – перевод точки из координат сочленений q в координаты захвата x:

$$T = A_1 A_2 A_3 A_4$$

$$T = \begin{pmatrix} c_{\phi\psi\theta} & -s_{\phi\psi\theta} & l_0 c_{\phi\psi\theta} + l_1 c_{\phi\psi} + l_2 c_\phi + l_3 \\ s_{\phi\psi\theta} & c_{\phi\psi\theta} & l_0 s_{\phi\psi\theta} + l_1 s_{\phi\psi} + l_2 s_\phi \\ 0 & 0 & 1 \end{pmatrix}$$

[2] Сокращённые обозначения: $c_\phi = \cos\phi$; $s_\phi = \sin\phi$; $c_{\phi\psi} = \cos(\phi + \psi)$; $s_{\phi\psi} = \sin(\phi + \psi)$; $c_{\phi\psi\theta} = \cos(\phi + \psi + \theta)$; $s_{\phi\psi\theta} = \sin(\phi + \psi + \theta)$

ВЕЗДЕХОД С ДВУМЯ ГУСЕНИЦАМИ

1. *Особая точка* — центр мобильного робота.
2. *Привод* — вращение гусеницы либо вперёд, либо назад.
3. **n=3** точки, **2** сочленения, **m=4** привода.

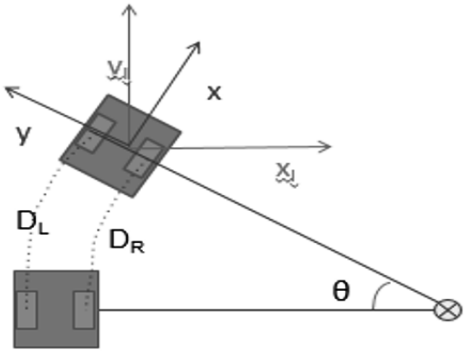
Модель движения идеализирована:
1. Без проскальзывания приводов
2. Без изгибания осей приводов
3. Приводы не сжимаются
и т.д.

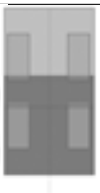
Не важно, колёса или гусеницы, главное, чтобы их можно вращать в обе стороны. Для устойчивости может быть добавлено несколько пассивных поддерживающих колёс.

Избыточность мобильного робота проявляется наиболее явно, кроме того, что есть бесконечное число траекторий проезда до целевой точки, находясь центром в целевой точке, приводы робота могут находиться в любом положении.

Рис. 3. Модель мобильного робота

КИНЕМАТИКА МОБИЛЬНОГО РОБОТА



			
$V_L = +k$, $V_R = 0$	$V_L = +k$, $V_R = +k'$	$V_L = +k$, $V_R = +k$	$V_L = +k$, $V_R = -k$
кружится вокруг сочленения	кружится вокруг внешней точки	едет прямо	кружится вокруг центра

Кинематика мобильных роботов вычисляется от скорости и позиции робота. Влияние каждой гусеницы на скорость робота рассматривается отдельно. Поворот робота осуществляется с помощью разности управляющих воздействий. Преобразование усилия приводов в движение:

$$\Delta x = \frac{v}{\omega}\sin(\omega t);\ \Delta y = \frac{-v}{\omega}\cos(\omega t) + \frac{v}{\omega}$$

$$\Delta\theta = \frac{(V_L - V_R)t}{d}$$

ОБРАТНАЯ КИНЕМАТИКА МОБИЛЬНОГО РОБОТА

В данном конкретном случае получить аналитически обратную кинематику просто. Управление задаём в терминах углов вращения колёс: V_L=$\dot{\Phi}_L$ r и V_R=$\dot{\Phi}_R$ r.

$$(\dot{x}, \dot{y}, \dot{\theta}) = \left(\frac{\dot{\Phi}_L r + \dot{\Phi}_R r}{2},\ 0\ ,\ \frac{\dot{\Phi}_L r - \dot{\Phi}_R r}{2}\right)$$

$$\begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta} \end{pmatrix} = \underbrace{\begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\dot{\xi}_I = T(\theta)\dot{\xi}_R} \begin{pmatrix} \frac{r\dot{\Phi}_r}{2} + \frac{r\dot{\Phi}_l}{2} \\ 0 \\ \frac{r\dot{\Phi}_r}{d} - \frac{r\dot{\Phi}_l}{d} \end{pmatrix}$$

$$\downarrow$$

$$T^{-1}(\theta)\dot{\xi}_I = T^{-1}(\theta)T(\theta)\dot{\xi}_R \qquad T^{-1} = \begin{pmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\downarrow$$

$$\dot{\xi}_R = T^{-1}(\theta)\dot{\xi}_I$$

ГОЛОНОМНАЯ КИНЕМАТИКА СИСТЕМЫ

В то время, как положение захвата манипулятора однозначно определяется углами сочленений, которые доступны в реальном времени, у мобильного робота значения датчиков просто возвращают обороты колёс, они должны быть интегрированы вместе со временем, что является источником большой неопределенности.

Кинематика робота голономная, если замкнутые траектории в пространстве сочленений q переводятся в замкнутые траектории в рабочем пространстве x.

Кинематика неголономная система, если замкнутые траектории в конфигурационном пространстве q могут не возвращать робота в начальное положение в рабочем пространстве x.

Манипулятор голономен, т.к. каждое положение сочленения соответствует единственному местоположению в пространстве. Мобильный робот, движущийся по рельсам тоже голономен.

Машинка или мобильный робот с гусеницами, которые могут вращаться с разной скоростью (differential-wheel robot) — неголономная система, потому что возвращение такого робота в начальное положение требует не только перемотки гусениц на нужное число оборотов назад, но ещё и правильного соотношения скоростей на них.

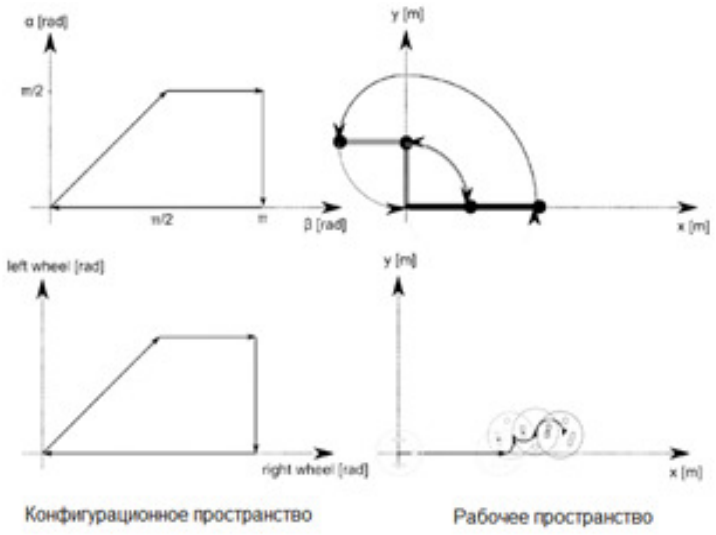


Рис. 4. Неголономная кинематика мобильного робота

На рисунке 5 показано, что мобильный робот сначала движется по прямой линии (обе гусеницы проворачиваются одинаковое число раз в секунду). Затем левая гусеница остаётся неподвижной, и только правая вращается вперёд. Потом только левая гусеница вращается назад. Наконец, правая гусеница вращается назад, приводя систему в изначальное состояние, но мобильный робот не вернулся к началу в координатах x.

Выполняя те же команды, на рисунке 5 видно, манипулятор с двумя вращательными сочленениями возвращается в исходное состояние.

Голономность или неголономность кинематики системы не должна влиять на качество работы алгоритма обучения и позиционирования. В том числе и эту зависимость предстоит изучить алгоритму.

КООРДИНАТЫ СОЧЛЕНЕНИЯ И ПРИВОДА В МОДЕЛИ

Состояние системы $s(t) = (x(t), \dot{x(t)})$ задано в неподвижных координатах рабочего пространства робота. Для робота это пространство является вычислимым, но не является непосредственно измеряемым. Доступные датчики, установленные в сочленениях, возвращают состояние приводов в локальных системах координат их сочленений. Информация с датчиков достаточно дискретизирована и точна. Функция вычисления статического состояния системы x хорошо изучена [3],[4],[5] по данным датчиков q (прямая кинематика) — простая и точная и не имеет особенностей:

$$f : q(t) \to x(t)$$

Обратная связь при управлении роботом может быть представлена в обоих этих пространствах q и x. Если представление x понятно и естественно, необходимо рассмотреть, что представляет из себя пространство q:

$$q(t) = (\ q_1(t), \ldots, q_{n-1}(t)\ )$$

где $q_k(t)$ - это положения сочленений, они могут быть выражены в метрах или градусах поворота, в зависимости от сочленения.

Хоть конкретный график изменения положения сочленения во времени неизвестен, но можно предположить примерную функцию работы пары приводов-антагонистов сочленения. Самое основное - функция каждого привода гладкая, всюду существует первая и вторая производные, хотя, возможно, как мы покажем ниже, производные слева и справа могут не совпадать.

Более того, первый привод из пары имеет неубывающую функцию $Q_{k_1}(t)$ в пространстве q, а второй — невозрастающую $Q_{k_2}(t)$.

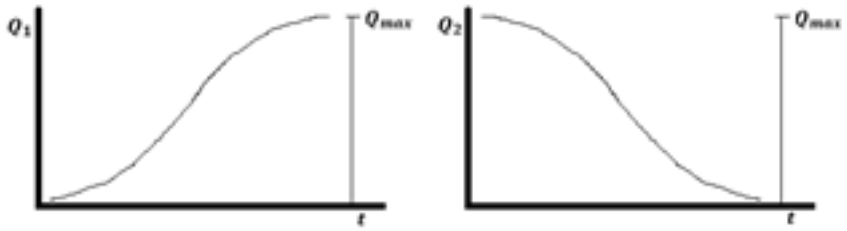


Рис. 5. Полное раскрытие и смыкание сочленения робота-манипулятора, показана функция действия приводов-антагонистов сочленения.

Тогда

$$\dot{q}(t) = (\dot{q_1}(t), \ldots, \dot{q_{n-1}}(t)),$$

где $\dot{q}(t)$ – это величины изменений положений каждого сочленения в заданный момент времени в локальных координатах соответствующего сочленения.

Влияния в каждый момент времени обоих приводов накладываются и общее смещение сочленения вычисляется

$$q_k(t)=q_k(t-1) + \dot{Q_{k_1}}(t)+\dot{Q_{k_2}}(t), \tag{4}$$

при этом за счёт возрастания одного и убывания второго, смещения приводов будут взаимообратными.

$$\dot{q_k}(t)=\dot{Q_{k_1}}(t)+\dot{Q_{k_2}}(t) \tag{5}$$

Вообще говоря, величина смещения привода в одном направлении необязательно конечна. На рисунке показано, как себя ведут приводы в модели мобильного робота:

Можно видеть, что хотя на отметке Q_{max} гусеница вернулась в положение, с которого начиналось движение, смещение привода вперёд, а вместе с ним всего объекта, может продолжаться, что будет приводить к увеличению смещения в пространстве x. Можно бесконечно применять усилие в одном направлении (раскручивать колесо) и продолжать движение: будь то кружиться на одном месте или двигаться, пока не произойдёт столкновение с объектом среды. Величина смещения внутри одного привода в мобильном роботе неограничена.

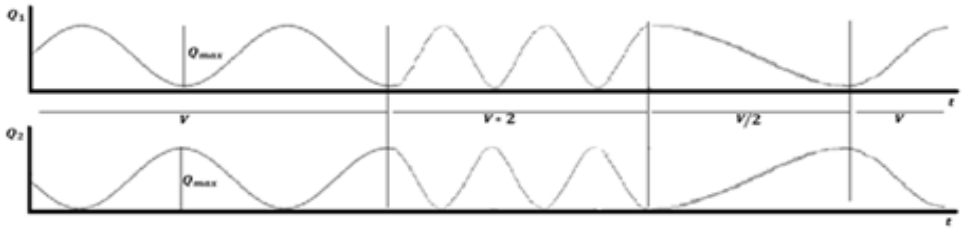


Рис. 6. Величина смещения от времени в приводах мобильного робота.

На том же рисунке 7 демонстрируется зависимость величины пройденного пути во времени для обоих приводов антагонистов, начинаем двигаться со скоростью v_k=$\dot{Q_{k_1}}(t)$=$\dot{Q_{k_2}}(t)$, затем ускоряемся до скорости $2\times v$, замедляемся до скорости $v/2$, и в конце стабилизируем скорость на начальном уровне v.

Для робота манипулятора можно допустить, что сочленения прокручиваются целиком на 360°, не задевая звенья опоры, т.е. если устроены так, что в них нет конструктивных ограничителей.

В нашей модели манипулятора величина смещения внутри его приводов ограничена $0 \leq q_k \leq Q_{kmax}$ (см. рис. 6), т.е. робот может раскрывать и смыкать сустав до каких-то пределов с обоих сторон. Это условие обобщает алгоритм обучения, мы стараемся в нём учесть разное поведение, которое может быть свойственно различным реальным системам.

По причине отсутствия информации о процессах, происходящих в приводах и сочленениях, нельзя полагаться на идентичность протекания процесса от попытки к попытке, от чего отсутствует повторяемость результата действия. Кроме того, во время движения на систему кроме усилий приводов действуют и другие силы, что нужно учитывать при генерации управления. Для симуляции такого поведения модели в внутреннюю функцию динамики системы будут введены *случайные отклонения*, научиться обрабатывать которые и есть конечная цель данного исследования.

Любое движение является в принципе *обратимым*, т.к. у каждого привода есть антагонист, смещающий q_k в противоположную сторону. Но обратное действие привода не гарантирует возвращения в изначальную точку. Теперь становится ясна причина наличия кнопки **reset**, она позволяет начинать любое действие с известного состояния системы.

ПРОЦЕССЫ В ПРИВОДАХ ПРИ УПРАВЛЕНИИ

При задании действия в эпизоде *варьировать можно только время пуска привода относительно остальных, длительность активности каждого привода* $u_k(t)$ (см. рис. 8). Длительность определяет на величину пройденного пути $q_k(t)$ сочленением.

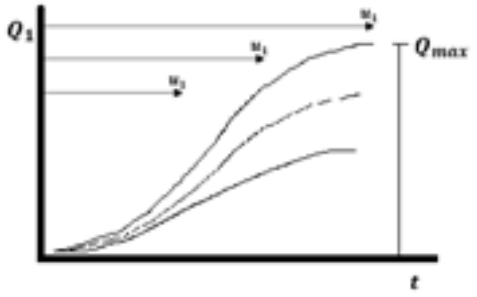


Рис. 7. Управление траекторией движения особой точки.

При работе приводы будут действовать по своим внутренним законам. Поскольку активация привода — не мгновенный процесс, существуют влияния инерции и иных сил на механизм системы, то для того, чтобы произошло видимое изменение известной части состояния системы $x(t)$ в виде сдвигов точек системы, приводы должны быть активны в течение нескольких тактов времени подряд. Отбрасывая управляющие сигналы приводов, фактически не меняющие состояние системы, управление $u(t)$ эпизода обучения можно представить, как кусочно-постоянную дискретную вектор-функцию, или как разряжённую матрицу размера $N \times m$ с несколькими последовательностями подряд идущих единиц в столбцах, где m столбцов - число приводов, а N строк - максимальное число тактов для эпизода, $(t_1 - t_0) < N$, где t_0 – начало эпизода, а t_1 – конец эпизода. *Траектория движения особой точки системы* в течении эпизода есть положения n-й точки в каждый такт времени:

$$q_n(t_0, t_1) = \{q_n(t_0), q_n(t_0 + 1), \ \ldots, q_n(t_i), \ \ldots, q_n(t_1)\}, \qquad (6)$$

$t_0 < t_i < t_1$

ФУНКЦИЯ ПЕРЕМЕЩЕНИЯ СОЧЛЕНЕНИЯ

Используя (4) и (5) запишем в пространстве q формулу перемещения k-го сочленения в момент t, начиная эпизод с момента t_0:

$$q_k(t) = q_k(t_0) + \sum_{t=t_0}^{t} \dot{q_k}(t) = q_k(t_0) + \sum_{t=t_0}^{t} \left(\dot{Q_{k1}}(t) + \dot{Q_{k2}}(t) \right)$$

Усилие привода — одна из неизвестных функций, задаёт законы внутреннего функционирования системы, напрямую ей управлять нельзя. Обозначим смещение по усилию привода $F_{kz} = F_{kz}(\dot{q}_k(t-1))$, $k=\overline{1, n-1}$, $z=1, 2$. Усилие действует, когда управление $u_{kz}(t)=1$ и зависит от скорости своего сочленения $\dot{q}_k(t-1)$ в предыдущий момент времени. Тогда общий прирост смещения внутри привода за такт есть:

$$\dot{Q_{kz}}(t) = (-1)^z \left[u_{kz}(t)\ F_{kz}(\dot{q_k}(t-1)) + I_{kz}(\dot{q_k}(t-1)) \right]$$

где $F_{kz} \geq 0$ — смещение по усилию (Force) привода, передаваемое связанным точкам системы, $I_{kz} \geq 0$ — смещение, вызванное инерцией (Inertia) после придания скорости.

Пример, как может выглядеть *функция усилия*:

$$F_{kz} = \dot{q_k}(t-1) + V,$$

где $V=const$.

Пример, как могут выглядеть *функции инерции*:

$$I_{kz} = \frac{\dot{q_k}(t-1)}{\sum_{j=1}^{k} m_j}$$

– для манипулятора, и

$$I_{kz} = \frac{\dot{q_k}(t-1)}{m_k + m_n}$$

– для вездехода, где $m_j=const$, $j=\overline{1, n}$.

ВЗАИМНЫЕ ВЛИЯНИЯ ПРИВОДОВ И СОЧЛЕНЕНИЙ

Как уже было сказано, любое движение имеет обратное, т.к. у каждого привода есть антагонист, смещающий q_k в противоположную сторону. В силу того, что разные по направлению силы могут быть активны одновременно, если они сравняются по модулю, то они будут *компенсировать усилия друг друга*, плотно фиксируя сочленение.

В чем смысл отдельно выделять по сути два одинаковых состояния: *состояние покоя* сочленения, когда на её приводы-антагонисты не подаётся никакое управление $u(0,0)$, и *состояние взаимной блокировки* $u(1,1)$, когда на оба привода подаётся сигнал выработки усилия, если в обоих состояниях сочленение не движется?

В случае, когда на это сочленение не действуют и приводы иных сочленений, действительно, разницы нет. Но если рассмотреть случай, показанный на рисунке 9:

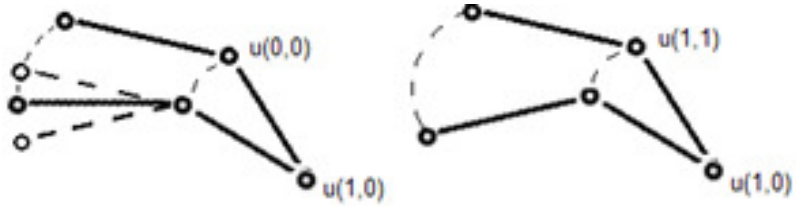


Рис. 8. (a) состояние покоя и (b) состояние взаимоблокировки локтя руки робота

Можно видеть, что в результате движений получаются разные положения захвата. В случае (a) левой части рисунка захват не доходит часть пути, относительно случая (b) в правой части. Пунктирные позиции показывают какое расстояние последнее звено не дошло – в нашей модели оно принято за половину.

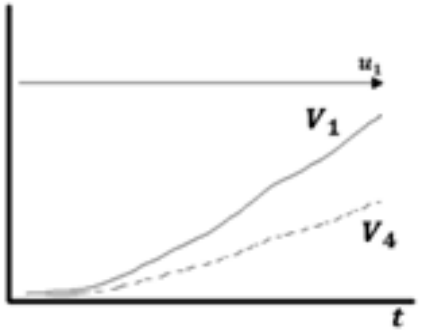


Рис. 9. Взаимное влияние сочленений. Закономерность скоростей

Т.е. сила, с которой сочленение влияет на звено следующее, за тем, которое связано непосредственно с этим сочленением, равна половине от исходной силы и противоположна по направлению. Сила, с которой со-

членение влияет на звено, через одно от непосредственно связанного с данным сочленением равна четверти исходной силы в обратную сторону. Рассмотрим рисунок 10, подаём управление на привод-смыкатель плеча, плечо начинает движение, развивая скорость полного усилия, и этим создаёт в покоящемся локте обратное усилие в половину величины, т.е. в приводе-размыкателе локтя имеем скорость – $V_4{=}V_1/2$. Добавим новый член в формулу усилия привода, который будет моделировать данное свойство:

$$\widetilde{\dot{Q_{kz}}(t)}=(-1)^z \left[u_{kz}(t)\ F_{kz}(\dot{q_k}(t-1))+I_{kz}(\dot{q_k}(t-1))+\sum_{\substack{p=0 \\ p\neq k}}^{n} \frac{\dot{q_p}(t-1)}{(p-k)^2} \right]$$

Ещё раз смотрим на рисунок 9, часть (b) – сочленение фиксировано силами, не может свободно двигаться, таким образом, плечо перемещает всю руку, как единое звено. А в части (a), наоборот, в покоящемся сочленении есть свобода и неприкреплённый конец пытается сохранить своё положение при движении плеча, локоть раскрывается в противоположную сторону от движения плеча, вместо всего пути последнее звено проходит только половину.

Аналогичное эффект можно смоделировать для вездехода, рис. 11.

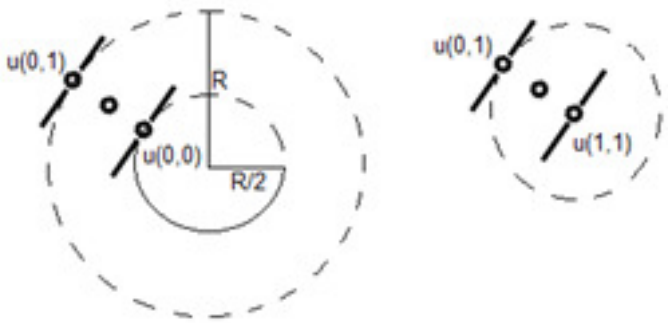


Рис. 10. (a) состояние покоя и (b) состояние взаимоблокировки левой гусеницы вездехода

МОДЕЛЬ ОЖИДАЕМЫХ ИЗМЕНЕНИЙ В ДИНАМИЧЕСКОМ СОСТОЯНИИ СИСТЕМЫ ПРИ ЭКСПЛУАТАЦИИ ПОСЛЕ ОБУЧЕНИЯ

Допустим, что система обучена в предположительно статических условиях и теперь занимается решением ставящихся перед ней задач. В процессе работы механизма в среде функционирования могут происходить изменения условий *наблюдаемые* (перемещение в новое помещение, установка под углом, навешивание груза) и *ненаблюдаемые* (изменение вязкости рабочего тела пневматики и пр.). Это означает, что с течением времени внутренние законы действия системы могут "гладко" изменяться в небольших пределах. Ниже даём определение этим изменениям. Таким образом, от повторения к повторению конечное состояние эпизода может смещаться. Следовательно, запомненное в базе данных действие, успешно достигшее цели, на практике уже не всегда её достигает с нужной точностью. Динамическое состояние системы и изменения в нём задаются разностными дифференциальными уравнениями, проводя эксперименты с переобучением системы, уточнением управления важно выяснить границы применимости адаптации для разных видов уравнений.

1. Мгновенные изменения

Происходят быстро (порядка нескольких тактов), редко (1 эпизод из 1000), тем самым вносят шум, который нужно уметь *отфильтровывать*. Примеры таких изменений: порыв ветра, задевание недопустимого объекта в рабочем пространстве и др.

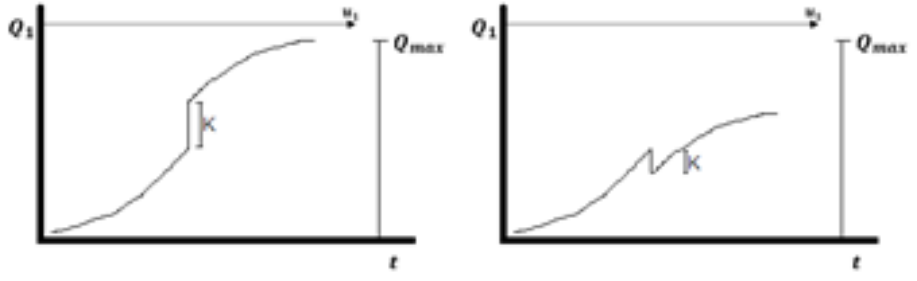


Рис. 11. Мгновенные изменения в модели системы

Для симуляции такого шума в закон функционирования привода добавим следующий член:

$$K(t_k) = \left\{ \begin{array}{c} K, t \geq t_k \\ 0, t < t_k \end{array} \right. , \quad K{=}const$$

Запишем полученный закон для **k**-го привода:

$$\widetilde{\dot{Q}_{kz}}(t){=}(-1)^z \left[u_{kz}(t)\ F_{kz}(\dot{q_k}(t{-}1)){+}I_{kz}(\dot{q_k}(t{-}1)){+}\right.$$
$$\left.{+}\sum_{\substack{p=0 \\ p\neq k}}^{n} \frac{\dot{q_p}(t{-}1)}{(p{-}k)^2} + K(t_k)\right] \quad (7)$$

2. Системные изменения

Происходят постепенно нарастающие изменения, которые действуют на робота длительное время, к ним нужно уметь *приспособляться* (нарастающие, в смысле, если новая функция динамики устоялась в течении 1000 тактов, постепенный возврат к исходной функции также является нарастающим изменением). Примеры системных изменений функционирования системы: атмосферное давление, влажность воздуха, температура помещения, износ подшипников и т.д.

T тактов, достаточно много, для плавного перехода от одной функции динамического состояния системы к другой:

$$\dot{q_k}(t) \to \left(\ (T{-}1)\dot{q_k}(t) + 1\ \dot{\tilde{q}_k}(t) \ \right)/T \to \ldots$$
$$\to \left(\ (T{-}i)\dot{q_k}(t) + i\ \dot{\tilde{q}_k}(t) \ \right)/T \to \ldots$$
$$\to \left(\ 1\ \dot{q_k}(t) + (T{-}1)\dot{\tilde{q}_k}(t) \ \right)/T \to \dot{\tilde{q}_k}(t), \quad (8)$$

Происходят в виде постепенной замены функций усилия приводов, изменения конструкционных параметров (масс, направления гравитации и пр.). Значит, просто заменяем функцию динамики в каждый момент t на новую:

$$\widetilde{\dot{Q}_{kz}}(t) = (-1)^z \left[u_{kz}(t)\check{F}_{kz}\left(\dot{\tilde{q}_k}(t)(t{-}1)\right) {+}\check{I}_{kz}\left(\dot{\tilde{q}_k}(t{-}1)\right) +\right.$$
$$\left.{+}\sum_{\substack{p=0 \\ p\neq k}}^{n} \frac{\dot{q_p}(t{-}1)}{(p{-}k)^2} + K(t_k)\right] \quad (9)$$

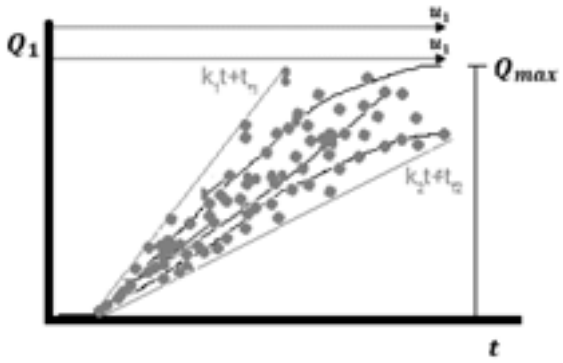


Рис. 12. Диапазон допустимых системных изменений в функциях усилия приводов

На рисунке ниже показаны границы, внутри которых могут быть выбраны функции усилия приводов:

Запишем формулу:

$$(k_{z2}\times t + t_{zf2}) \leq \dot{\widetilde{Q}}_{kz}(t) \leq (k_{z1}\times t + t_{zf1}), \tag{10}$$

где $k_{z1} > k_{z2}$ – константы уклона, t_{zf1}, t_{zf2} – моменты времени, спустя которые усилие привода превосходит силу трения внутри сочленения. Аналогичный конус для противоположного убывающего привода, также расширяющийся с течением времени, только направленный вниз. Модель функционирует так, что чем меньше по длительности работает привод, тем меньше будет случайный разброс значений.

Чтобы более детально определить характеристики конкретной случайной функции из всех возможных нужно воспользоваться *теорией случайных процессов*. В рамках определённого здесь диапазона нужно задать *ансамбль реализаций случайного процесса* – правило всех реализаций этого процесса. Задать такое однопараметрическое семейство случайных величин (от параметра времени t), заданных на одном и том же пространстве элементарных событий Ω, мы можем достаточно произвольно, нужно только вспомнить, что для соответствующего привода функции в его семействе должны быть либо неубывающими, либо невозрастающими.

3. Граничные условия

В этом разделе представлены виды граничных возмущений на концах доступного хода привода, т.е. в первых тактах самого начала движения, а также при принудительной остановке системы в следствие некорректного управления – при ударах о границы свободного хода сочленений, о границы рабочей области. На рисунке 15 (a) показаны этапы работы одного сочленения целиком – от крайнего до крайнего момента, здесь внимание будет заострено на первой Δt_1 и последней Δt_3 частях движения сочленения.

3.1. Если не предотвращать столкновение робота с ограничениями (пре-

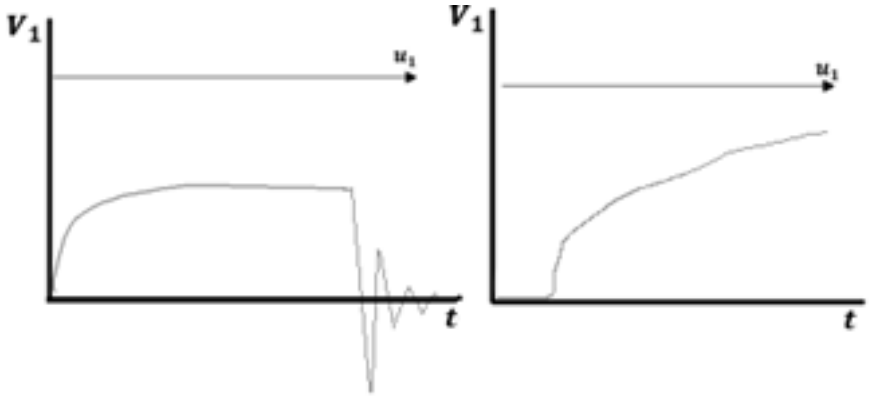
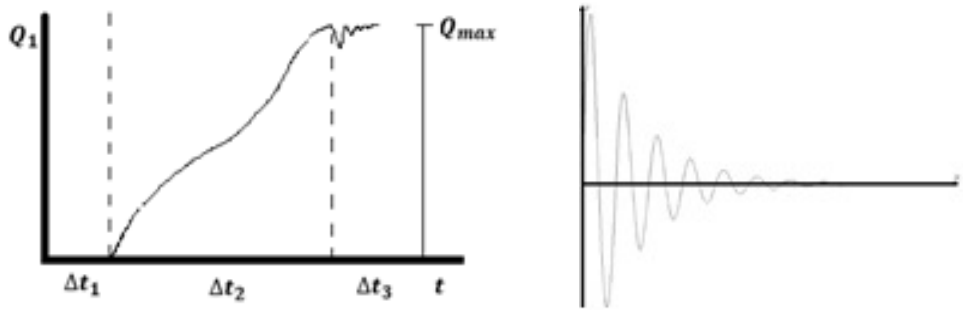


Рис. 13. Виды граничных изменений: (a) реакция смещения на столкновение, (b) преодоление трения на старте

делом движения сустава манипулятора или наезд на объект вездеходом), заблаговременно снизив скорость, можно получить сильные упругие колебания всей системы (рис. 16 (a)). Вид получаемых быстро затухающих колебаний можно описать следующей формулой (см. рис. 15 (b)):

$$f(x) = \frac{\sin x}{\sqrt{x}}$$

3.2. Колебания, возникающие на этапе *старта привода* (рис. 16 (b)), для преодоления трения внутри привода и сустава, или трения поверхности, требуется накопление импульса, за которым происходит моментальное ускорение особой точки системы и незначительная её раскачка. Располага



(a) (b)

Рис. 14. (a) работа одного сочленения целиком, (b) функция колебаний столкновения

Ограничения суставов следует избегать с максимально возможным запасом при любой траектории пространства сочленений. В случае с проблемой на старте движения, можно понимать так, что максимальной эффективности позиционирования роботов с данными законами движения можно достичь на движениях средней длины относительно хода привода, т.е. мы принимаем во внимание тот факт, что не нужно располагать целевые точки очень близко к начальной позиции, чтобы получать нормальный результат позиционирования.

ОГРАНИЧЕНИЯ СКОРОСТИ И УСКОРЕНИЯ ПРИ УПРАВЛЕНИИ

Траектории суставов должны быть как можно более плавными (в реальном роботе грубые и резкие движения могут повредить механизм), и общее движение суставов должно быть не больше, чем необходимо для получения заданного движения захвата.

Даже идеальная модель системы неизбежно сталкивается с проблемами при использовании простого пропорционально-дифференцирующего управления в оперативном пространстве:

$$u_x = k_p(x^* - x) - k_v\dot{x}$$

где x и $\dot{x}$ – позиция и скорость системы в оперативном пространстве, x^* - целевая позиция, а k_p и k_v – коэффициенты усиления пропорциональной и дифференцирующей состовляющих.

Если так определить управляющий сигнал в пространстве x и преобразовать его в точные крутящие моменты сочленений q, то траектория достаточно длинного движения особой точки $q_n(t)$ не будет прямой лини-

ей, т.к. либо двигатели не смогут фактически выдать нужные мгновенные крутящие моменты, либо будут недостаточны дискретизации управления или обратной связи. Некоторые из приводов связаны с меньшей массой, чем другие, в случае с кинематической цепью, и могут выдавать больший разброс по крутящим моментам в своём сочленении. Естественным представляется замедлить робота, ограничить скорость и ускорение для его сочленений. Введём желаемую скорость взяв её величину достаточно маленькой, чтобы крутящие моменты всех сочленений могли её обеспечить.

$$\widetilde{x} = (x - x^*)$$

$$u_x = -k_v \left(\dot{x} + sgn(\widetilde{x}) max \left(V_{max}, \frac{k_p}{k_v} |\widetilde{x}| \right) \right) \tag{11}$$

$$u_x = \begin{cases} -k_v(\dot{x} + sgn(\widetilde{x}) V_{max}) \\ -k_v \dot{x} + k_p \widetilde{x} \end{cases}$$

Остаётся сделать не одинаковое

1. Ограничение скорости

В случае с нашей моделью всё немного сложнее, потому что нет возможности задавать точные крутящие моменты на сочленения. Отсекать скорость можно только с некоторой задержкой, постфактум.

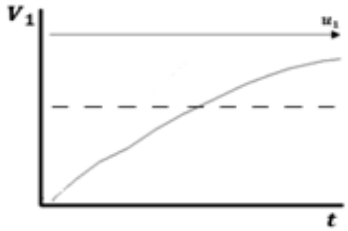


Рис. 15. Ограничение скорости

$$|\dot{q_k}(t)| < C_{kv}$$

где C_{kv}=$const$

2. Ограничение ускорения

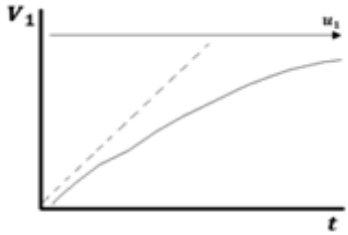


Рис. 16. Ограничение ускорения

Вопрос обстоит ещё сложнее с ограничением ускорений соединений роботов.

$$|\ddot{q}_k(t)| < C_{ka}$$

где C_{ka}=$const$ – некоторая константа,

Упругие деформации создают колебательность в приводах при их активном разгоне и торможении, т.е. тогда, когда появляются ускорения [3]. Нежелательны колебания, возникающие *при торможении* привода, они затягивают процесс остановки робота, следовательно, снижают быстродействие системы. Колебания, возникающие *на этапе разгона* привода обычно успевают затухнуть к моменту его торможения и поэтому менее вредны. Однако, если длительность этапа движения с постоянной скоростью мала, оба эти типа колебаний могут наложиться, что сильнее затруднит процесс торможения.

РЕЗУЛЬТАТЫ

Разработан алгоритм обучения систем с дискретным управлением в задаче позиционирования действовать своими приводами. Для первого и второго вопросов поставленных перед исследованием получены ответы, построены этапы алгоритма обучения, результаты сложности и времени обучения для двух моделей реальных устройств с нелинейными законами движения приведены в таблицах 1-4. В данном тексте третий вопрос исследования сформулирован как гипотеза для дальнейшей разработки, практические результаты ещё не получены.

Ответом на первый вопрос является двухэтапный алгоритм, который на первом этапе задействуя разные приводы и варьируя длительности их работы, даёт понимание разброса возможных состояний системы, направления работы приводов, локализует положение относительно робота интересующей области рабочего пространства в координатах управлений $u(t)$. Вторым этапом алгоритм создаёт плотную равномерную решётку конечных положений покрывающую все целевые точки интересующей области, подбор значений аргументов неизвестной функции, чтобы выдерживать равные дистанции между её значениями осуществляется методами Монте-Карло и МНК на основе действий, хранящихся в базе данных обучения.

Простые эксперименты показали, что случайная генерация бинарных матриц управлений за адекватное время не даёт такого же направленного эффекта, как данный двухэтапный алгоритм, по траекториям действий которого получается хорошая интерполяция: конечные положения и траектории равномерно распределены по интересующей области, без сгустков и пустот.

Сложность (в числе траекторий) алгоритма заметания решётки зависит от установленного шага (см. таблицы), чем меньше размер шага, тем выше сложность. Имея решётку конечных положений, дальше действовали двумя способами.

Первый способ – это приближаться к любым целевым точкам в изученной области попытками градиентного спуска. В среднем получается 8 попыток на 1 новую целевую точку. Выбираются несколько ближайших конечных точек к искомой и по ним строится промежуточное управление усреднением, приближающее к цели. Не всякие два управления подходят для получения промежуточного между ними, но если точек, рядом с целью достаточно много и дискретизация обратной связи и управления достаточно малы, алгоритм сходится к цели. Этот метод хорош тем, что легко применяет новый полученный опыт в процессе работы, а так-

же толерантен к шуму. Не требователен к вычислительным ресурсам, но имеет слишком большую сложность (число попыток). Алгоритм локальный, даже если учитывать обращение к базе данных, работает за константную асимптотику.

Второй способ – строить глобальную аппроксимацию случайными функциями[6] 2-функционала обратной кинематики системы. В среднем 1 попытка на новую целевую точку для данного метода, что очень хорошо. Но аппроксимация вычисляется для фиксированного набора произведённых траекторий. Пока не найден способ частичного пересчёта полной интерполяции (при добавлении каждой новой траектории нужно пересчитывать с нуля), очень высокие требования к оборудованию для пересчёта и квадратичная вычислительная сложность относительно записей в базе данных по памяти и CPU. Деградирует от зашумлённых данных.

Сложностью алгоритма назовём общее число движений системы при работе алгоритма обучения (изначальная категория, которую мы минимизируем). Метод оценки времени: если 1 такт дискретного времени = 1 миллисекунда (5 мкс), тогда 1 эпизод, максимальная длительность которого N тактов, в среднем длится $N/10\times$(5 мкс)$\approx$0,5 с., ещё столько же нужно, чтобы вернуться в начальную точку. Получается, что одно позиционирование туда (эпизод) и обратно (reset) в среднем занимает 1 секунду. Всего целевых точек расположенных в интересующей области рабочего пространства робота 400 штук.

Таблица 1. Глобальная интерполяция для манипулятора

Этапы	Сложность	Шаг заметания	Среднее число движений на 1 целевую точку	% попаданий в целевые точки	Время
1.1	255	71.43 мм	1	-	14 мин.
1.2	1352	8.93 мм	1	-	23 мин.
2	400	-	1	96%	7 мин.
Итого	2007	-	-	-	44 мин.

Далее есть намерение смешать описанные два подхода ко второму вопросу исследования, чтобы сгладить достоинствами одного недостатки другого. Применяя один алгоритм для разных систем показываем высокую абстрактность подхода, выявляем минимальные достаточные ограничения для успешного применения алгоритма для разных устройств.

Таблица 2. Глобальная интерполяция для вездехода

Этапы	Сложность	Шаг заметания	Среднее число движений на 1 целевую точку	% попаданий в целевые точки	Время
1.1	1359	18.75 мм	1	-	23 мин.
1.2	4288	5.62 мм	1	-	72 мин.
2	400	-	1	98%	7 мин.
Итого	6047	-	-	-	1.68 ч.

Таблица 3. Приближение попытками градиентного спуска манипулятора

Этапы	Сложность	Шаг заметания	Среднее число движений на 1 целевую точку	% попаданий в целевые точки	Время
1.1	255	71.43 мм	1	-	14 мин.
1.2	1352	8.93 мм	1	-	23 мин.
2	2468	-	7	100%	42 мин.
Итого	4075	-	-	-	1.19 ч.

Таблица 4. Приближение попытками градиентного спуска вездехода

Этапы	Сложность	Шаг заметания	Среднее число движений на 1 целевую точку	% попаданий в целевые точки	Время
1.1	1359	18.75 мм	1	-	23 мин.
1.2	4288	5.62 мм	1	-	72 мин.
2	3044	-	8	100%	51 мин.
Итого	8691	-	-	-	2.26 ч.

Продолжение исследования в направлении, которое можно описать так: память обширная, предоставляет возможности по построению адекватных предсказаний, но её данные со временем устаревают и перестают верно отображать обстановку вещей, а для того, чтобы переобучаться времени нет, уточнение действий должно производиться в реальном времени. После успешной фазы обучения происходит эксплуатация обученной системы, опыт постоянно дополняется. Так как записи действий сохраняются в базе данных навсегда, важно не запутаться в произведённых действиях с разными результатами и одной целевой точкой.

Список литературы

[1] Яблонский С.В. *Введение в дискретную математику.* — М.:Высшая школа, 2006.

[2] Гасанов Э.Э., *Кудрявцев В.Б. Интеллектуальные системы. Теория хранения и поиска информации* — 2-е изд. — М.:Юрайт, 2017.

[3] Юревич Е.И. *Основы робототехники.* – 2-е изд. – СПб.:БВХ-Петербург, 2005. – 416 с.: ил.

[4] Зенкевич С.Л., Ющенко А.С. *Основы управления манипуляционными роботами.* М.:Изд-во МГТУ им. Н.Э.Баумана, 2004. — 480 с.

[5] Кокс Д., Литтл Дж., О'Ши Д. *Идеалы, многообразия и алгоритмы. Введение в вычислительные аспекты алгебраической геометрии и коммутативной алгебры.* Пер. с англ. — М.:Мир, 2000. — 687 с.

[6] Бахвалов Ю.Н., Малыгин Л.Л., Черкасс П.С. *Многомерная интерполяция и аппроксимация на основе случайных функций* // Вестник Череповецкого ГУ, 2012. №2(39) [url: https://cyberleninka.ru/article/v/metod-mashinnogo-obucheniya-na-osnove-algoritma-mnogomernoy-interpolyatsii-i-approksimatsii-sluchaynyh-funktsiy]

[7] Jan Peters, Stefan Schaal *Learning Operational Space Control* — University of Southern California, 2008. [url: https://researchgate.net/publication/216053678_Learning_to_Control_in_Operational_Space]

[8] Travis DeWolf *IK and Operational space control* — University of Waterloo, 2013. [url: https://studywolf.wordpress.com/site-index/]

[9] John Baillieul, D. P. Martin, Bruce R. Donald *Robotics. Resolution of kinematic redundancy.* Proceedings of symposia in applied mathematics, volume 41 — American Mathematical Soc., 1990. — 196 pp.

[10] D. Lu *Kinematics of mobile robots* — University of Maryland, 2016. [url: http://cfar.umd.edu/ fer/cmsc498F-828K/lectures/Kinematics.pptx]

[11] Jason Clark *IK - Essential Math for Games Programmers*, 2005. [url: https://essentialmath.com/InverseKinematics.pps]

[12] Welman, Chris, *Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation* — Simon Fraser University. 1993. [url: http://run.usc.edu/cs520-s15/ik/welman.pdf]

[13] Meredith, Michael and Maddock, Steve, *Real-Time Inverse Kinematics: The Return of the Jacobian* — University of Sheffield, 2006. [url: https://staffwww.dcs.shef.ac.uk/people/S.Maddock/publications/MeredithMaddock2004_CS0406.pdf]

Learning algorithm of systems with discrete control
Golikov K.A.

The learning algorithm was developed for the problem of positioning systems with discrete control, it is based on a method of generalizing using a global interpolation and a gradient descent of trials and fails that stored in the database. The algorithm is optimized by the criterion of reducing the learning time (number of attempts). The algorithm was tested on a simulator for models of systems operating on a plate of two different types: for a mobile differential-drive robot and for an open kinematic chain with rotational and prismatic joints.

Keywords: positioning, learning algorithm, robot, interpolation, approximation.

Объектная модель правил дорожного движения

Менькин М. И.

Данная работа относится к области семантического анализа юридических документов. Под семантикой нормативно-правовых актов будем понимать отображение, на вход которого поступает текст нормативно-правового акта, а на выходе получается его формальная модель. В статье описывается один из возможных подходов построения формальной модели правил дорожного движения.

Ключевые слова: семантический анализ нормативно-правовых актов, прагматический анализ, построение формальных моделей.

1. Введение

Существует актуальная проблема автоматического извлечения знаний из текстов на естественном языке (ЕЯ). Целью настоящей работы является разработка аппарата извлечения знаний из нормативно-правовых актов (НПА) на примере Постановления Правительства «О правилах дорожного движения» [1] (далее — ПДД) для решения разного рода прикладных и теоретических задач. Будем считать, что мы правильно поняли смысл НПА, если по его тексту мы смогли построить некоторую формальную модель, описывающей этот НПА.

Для достижения цели необходимо сделать следующее:

1) разработать формальную модель, в которую будет преобразовываться текст;

2) автоматизировать процесс построения формальной модели по тексту;

3) решить практическую задачу (например, проверить непротиворечивость и полноту текста НПА на смоделированных данных) для демонстрации работоспособности модели;

4) получить теоретические оценки эффективности и адекватности модели.

Причиной рассмотрения именно юридического документа является формализованность структуры предложений и текста, а также ограниченность лексики (по сравнению с произвольным текстом на ЕЯ).

Примеры возможных практических применений результатов работы:

- создание вопросно-ответных систем для приёма или сдачи теоретического экзамена на знание ПДД;
- мониторинг дорожных ситуаций в реальном времени.

Если говорить о произвольных НПА, то их модели могут быть использованы при решении следующих задач:

- проверка реальных ситуаций и определение их «законности», «правильности» (т.е. их непротиворечивость НПА);
- проверка полноты знаний НПА при его применении к реальным ситуациям;
- проверка непротиворечивости частей одного НПА или разных НПА друг другу;
- синтез новых НПА (путём задания классов, их атрибутов, методов и правил взаимодействия).

Процесс программной реализации механизма извлечения знаний из текста можно условно разделить на несколько взаимосвязанных частей:

1) Морфологический (значимые компоненты слов);
2) Синтаксический (структурные отношения слов);
3) Семантический (содержательная компонента слов, смысл);
4) Прагматический (семантика по отношению к целям высказывания).

Иногда также выделяют фонетический этап (применительно к устной речи) и дискурсивный этап (анализ текста как акта коммуникации либо как текста конкретной предметной области) [2, 3].

В статье приводится прагматическая модель ПДД, в которую в дальнейшем будет преобразовываться текст из [1]. Данная часть работы заключается в определении основных сущностей документа, их атрибутов и правил взаимодействия. С точки зрения объектно-ориентированного программирования — это этап определения классов, их атрибутов и методов. С юридической точки зрения классы — это объекты и субъекты права, а методы классов — нормы права [4]. Дальнейшая автоматизация процесса извлечения знаний (через морфо-синтаксический анализ) будет основываться на приведении текста к модели, определённой на прагматическом этапе.

Следует отметить, что прагматический этап в некотором смысле независим от остальных ввиду уникальности каждой предметной области.

Идейно текущая работа пересекается с работами Е. М. Перпера по анализу юридических документов [5–8].

Автор выражает благодарность Э. Э. Гасанову за постановку задачи, бурное обсуждение и помощь в процессе работы.

2. Объекты ПДД

2.1. Основные понятия

Основные объекты ПДД — **дорога**, **транспортное средство** (ТС) и **манёвр**. Ввиду отсутствия в [1] определения термина «манёвр», дадим ему своё определение. Далее приведём необходимые нам определения по тексту ПДД:

Дорога — обустроенная или приспособленная и используемая для движения ТС полоса земли либо поверхность искусственного сооружения. Дорога включает в себя одну или несколько проезжих частей, а также трамвайные пути, тротуары, обочины и разделительные полосы при их наличии.

Проезжая часть — элемент дороги, предназначенный для движения безрельсовых ТС.

Полоса движения — любая из продольных полос проезжей части, обозначенная или не обозначенная разметкой и имеющая ширину, достаточную для движения автомобилей в один ряд.

Транспортное средство — устройство, предназначенное для перевозки по дорогам людей, грузов или оборудования, установленного на нем.

Преимущество (приоритет) — право на первоочередное движение в намеченном направлении по отношению к другим участникам движения.

Уступить дорогу (не создавать помех) — требование, означающее, что участник дорожного движения не должен начинать, возобновлять или продолжать движение, осуществлять какой-либо манёвр, если это может вынудить других участников движения, имеющих по отношению к нему преимущество, изменить направление движения или скорость.

Теперь введём свои определения.

Участок дороги (УД) — часть полосы движения, инвариантная относительно действующих на ней правил движения. На рис. 1 приведён пример УД: знак «Ограничение максимальной скорости» вносит новое правило движения (не превышать скорость), и поэтому этот знак определяет новый УД. Следующее событие, изменяющее правила движения (в нашем случае это знак «Падение камней»), определит новый УД, и т.д.

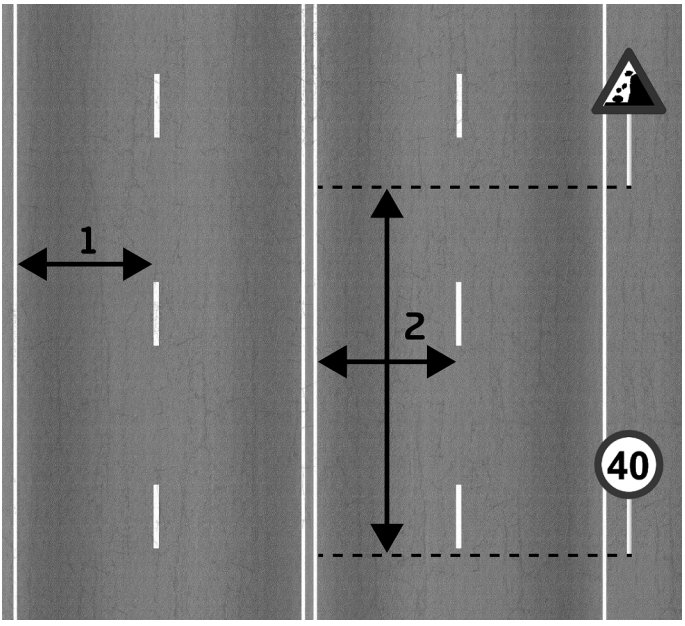


Рис. 1. Полоса движения (1) и участок дороги (2).

Граф дороги — раскрашенный граф $D = \big(V, E, P(i)\big)$, где V — множество вершин, каждому элементу которого соответствует один УД, E — множество рёбер, каждому элементу которого соответствует одно отношение соседства «УД»–«УД» (т.е. ребром соединяем только примыкаю-

щие друг к другу УД), $P(i) = P_v(i) \times P_e(i)$ — множество цветов, $P_v(i)$ — атрибуты вершин в момент времени i, $P_e(i)$ — атрибуты рёбер в момент времени i. Рёбра из E, которые соединяют УД, принадлежащие одной полосе, ориентированы по направлению движения на этой полосе. На рис. 2 приведён пример разбиения дороги на УД (числами «1», «4», «9», «12» обозначены обочины, остальными — участки проезжей части), на рис. 3 — граф, соответствующий этому разбиению.

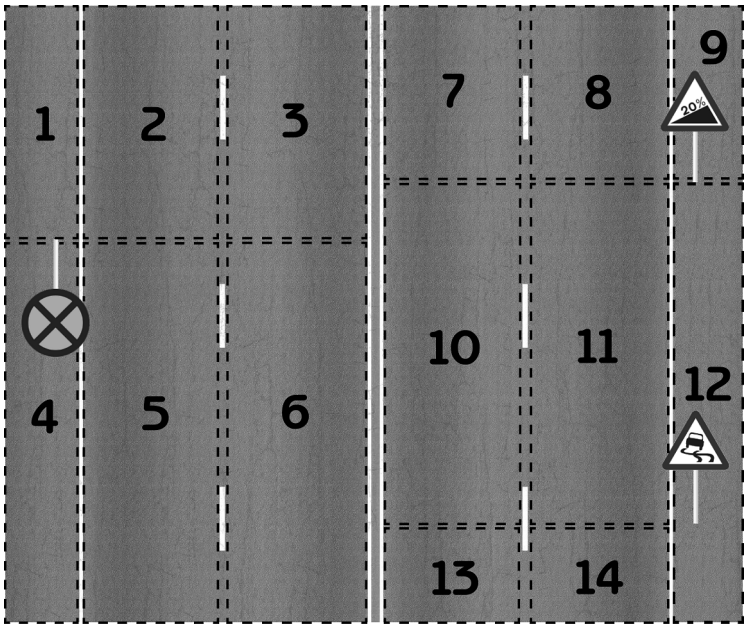


Рис. 2. Разбиение дороги на участки дорог.

Существование рёбер, и, как следствие, необходимость рассмотрения именно графовой модели, можно обосновать тем, что между парами УД возникают атрибуты, которые не существуют у УД по отдельности (например, горизонтальная разметка на дороге между двумя УД — это атрибут их отношения, или, иначе, цвет ребра графа дороги).

Манёвр — потенциально возможное действие ТС, в результате которого оно изменяет УД (например, меняет полосу движения, выезжает на перекрёсток или на УД с новым действующим знаком, и т.д.) или заметно изменяет свою скорость. В зависимости от дорожной ситуации, манёвр имеет статус «обязателен для исполнения» (например, манёвр «остановка» перед красным сигналом светофора), либо «допустим» (например, манёвр «опережение», если его выполнение не запрещено в данной до-

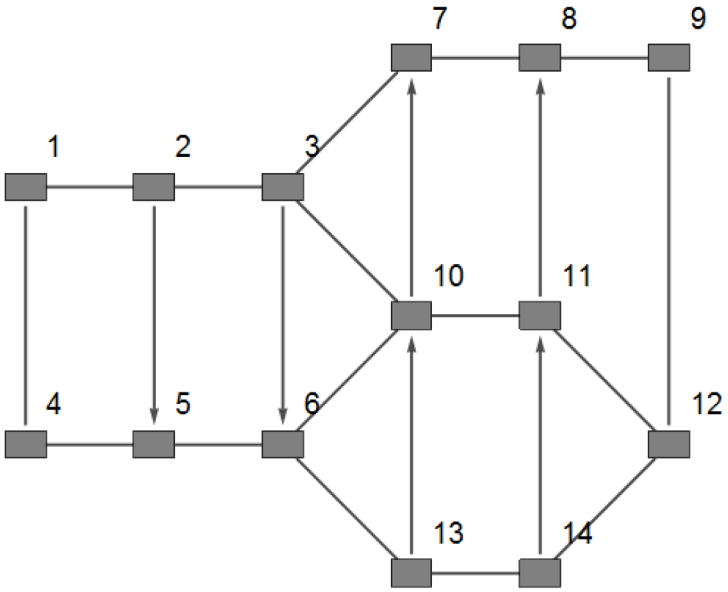


Рис. 3. Граф дороги.

рожной ситуации), либо «запрещён» (например, манёвр «разворот» в тоннеле).

Пусть $D = \big(V, E, P(i)\big)$ — граф дороги, T — множество ТС, M — множество манёвров. Тогда *дорожная ситуация* (ДС) — это пятёрка $S = \big(D, M, T, f(i), g(i)\big)$, где $f\colon T \times i \to V$ — функция позиционирования, которая в каждый момент времени i каждому ТС из T ставит в соответствие один УД из V, $g\colon T \times i \to M \cup \{*\}$ — функция назначения манёвра, которая в каждый момент времени i каждому ТС из T ставит в соответствие один манёвр из M. Если функция назначения манёвра принимает значение $*$, то это понимается как «манёвр неопределён». Множество T не обязательно должно содержать все ТС, которые находятся на дороге. Множество M не обязательно должно содержать все возможные манёвры.

Разрешение дорожной ситуации для данного транспортного средства — определение всех допустимых манёвров для данного ТС в данной ДС.

Разрешение дорожной ситуации — разрешение ДС для тех ТС, у которых значение манёвра неопределено.

Корректная дорожная ситуация — ДС, в которой для каждого ТС, у которого определён некоторый манёвр, этот манёвр входит в перечень

допустимых манёвров (т.е. таких манёвров, которые не противоречат в данной ДС существующим ПДД).

Задачами ПДД является разрешение либо определение корректности произвольных ДС.

В следующих подразделах перечислим объекты модели и их атрибуты, выделенные из ПДД (списки неполные и будут дополняться). Для атрибутов используются следующие условные обозначения:

- bool — булев тип;
- int / int[] — целочисленный тип / список целочисленных типов;
- enum — перечисляемый тип.

При программной реализации перечисленные ниже атрибуты будут являться переменными класса.

2.2. Дорога

В [1] дано слишком общее определение термину «дорога». На практике для разрешения ДС используется лишь небольшая часть всей дороги. Поэтому мы под дорогой будем понимать ту часть дороги, которая имеет, либо не имеет горизонтальной разметки, и которая содержит фиксированное количество полос движения для ТС.

При перечислении атрибутов в скобках укажем номер пункта ПДД, в котором встречается данный атрибут (номер пункта приводится в качестве примера, использование данного атрибута в тексте ПДД может быть не единственным). В случае если атрибут напрямую не проговаривается в документе, но подразумевается и важен для моделирования ДС, в скобках вместо пункта пишем фразу *модельный атрибут*.

- int: количество проезжих частей (п. 1.2)
- int[]: количество полос движения на каждой проезжей части (п. 1.2)
- bool: наличие реверсивных полос (п. 9.8)
- bool: наличие горизонтальной разметки (п. 1.2)
- bool: наличие тротуара (п. 1.2)
- bool: наличие обочины (п. 1.2)
- bool: наличие трамвайных путей (п. 1.2)

- bool: наличие пешеходной дорожки (п. 1.2)
- bool: наличие велосипедной дорожки (п. 1.2)
- bool: наличие железнодорожного переезда (п. 15.1)
- enum: движение правостороннее / левостороннее (в России — правостороннее, п. 1.4)

2.3. Участок дороги

Перечислим некоторые атрибуты УД. Атрибуты, которые изменяются в течение времени, будем называть состояниями.

- int: скорость максимальная (на данном участке дороги), км/ч (модельный атрибут, зависит от атрибутов УД и ТС)
- int: длина, м (модельный атрибут)
- int: ширина, м (модельный атрибут)
- enum: (перекрёсток / поворот) принадлежность одной / двум дорогам (модельный атрибут)
- enum: тип УД (проезжая часть, тротуар, обочина, трамвайные пути, железнодорожный переезд, велосипедная дорожка, пешеходная дорожка) (п. 1.2)
- bool: перекрёсток (п. 8.8)
- enum: (перекрёсток) регулируемый / нерегулируемый (п. 8.8)
- bool: пешеходный переход (п. 8.11)
- bool: тоннель (п. 8.11)
- bool: мост (на / под) (п. 8.11)
- bool: путепровод (на / под) (п. 8.11)
- bool: эстакада (на / под) (п. 8.11)
- bool: железнодорожный переезд (п. 8.11)
- bool: остановка маршрутного ТС (п. 8.11)

- bool: реверсивная полоса (п. 9.8)
- bool: жилая зона (п. 10.2)
- bool: населённый пункт (п. 10.2)
- bool: главная дорога (п. 11.4)
- bool: участок с ограниченной видимостью (конец подъёма / опасный поворот) (п. 11.4)
- bool: до железнодорожного переезда менее 100 м (п. 11.4)
- bool: уклон (п. 11.7)
- int[ ]: действующие знаки (п. 11.7)
- bool: наличие трамвайных путей (п. 12.4)
- bool: автомагистраль (п. 16.1)

Состояния:

- bool: наличие пешеходов (модельный атрибут; пешеходы, наравне с ТС, являются участниками дорожного движения, но т.к. у них нет своих атрибутов, кроме как принадлежать либо не принадлежать УД, мы представили их как атрибут УД)
- enum: интенсивность движения (п. 10.1)
- bool: наличие затора (п. 13.2)
- и т.д.

2.4. Транспортное средство

Некоторые атрибуты ТС:

- int: длина, м (модельный атрибут)
- int: ширина, м (модельный атрибут)
- int: высота, м (модельный атрибут)
- enum: тип ТС (велосипед, мопед, гужевое, мотоцикл, легковое, грузовое, автобус, троллейбус, трамвай) (п. 1.2)

- bool: механическое ТС (п. 1.2)
- bool: гибридный автомобиль (п. 1.2)
- bool: электромобиль (п. 1.2)
- bool: (автобус) междугородний (п. 10.3)
- bool: (автобус) маломестный (п. 10.3)
- int: скорость максимальная, км/ч (техническая характеристика) (п. 10.5)
- bool: тихоходное (п. 11.6)
- bool: безрельсовое (п. 13.4)
- int: (грузовой) разрешённая максимальная масса, т (п. 16.1)
- bool: маршрутное (п. 18.2)

Состояния:

- int[]: запланированная траектория (важный модельный атрибут, необходим для разрешения ДС; в непротиворечивой и полной модели этот атрибут должен совпасть с прогнозируемой траекторией ТС)
- int[]: местоположение на УД (модельный атрибут)
- int: скорость текущая, км/ч (модельный атрибут)
- enum: подача сигнала поворота направо / налево (п. 8.1)
- bool: движение задним ходом (п. 8.12)
- bool: осуществление организованной перевозки групп детей (п. 10.3)
- bool: (на уклонах) движение на спуск / подъём (п. 11.7)
- bool: буксировка механического ТС (п. 20.1)
- bool: (грузовой) перевозка людей в кузове (п. 22.1)
- и т.д.

2.5. Отношения «УД»–«УД»

Перечислим атрибуты рёбер графа дороги.

- bool: (для последовательных УД проезжей части) наличие сужения либо расширения проезжей части (модельный атрибут)
- bool: (для последовательных УД проезжей части) наличие светофора (п. 6)
- bool: (для последовательных УД проезжей части) наличие регулировщика (п. 6)
- enum: (при наличии светофора) тип светофора (автомобильный, с дополнительной стрелочной секцией / секциями, для маршрутных ТС с бело-лунными или цветными сигналами, для железнодорожного переезда, для реверсивной полосы, для велосипедистов, для пешеходов) (п. 6)
- bool: наличие шлагбаума (п. 15.2)
- bool: наличие поднимающегося барьера (на железнодорожных переездах) (модельный атрибут)
- bool: (между параллельными УД встречного движения) наличие разделительной полосы (п. 9.9)
- bool: (между параллельными УД) попутное / встречное направление движения (п. 13.9)
- bool: (между параллельными УД) наличие бордюра (приложение 2)
- int[ ]: знаки (приложение 1)
- int[ ]: разметка вертикальная / горизонтальная (приложение 2)

Состояния:

- enum: (при наличии светофора или регулировщика) сигнал светофора / регулировщика (п. 6)
- bool: (при наличии шлагбаума) шлагбаум поднят / опущен (п. 15.2)
- bool: (при наличии барьера) барьер поднят / опущен (модельный атрибут)

2.6. Манёвры

Под манёвром будем понимать некоторую процедуру, на вход которой подаём ТС (относительно которого решается вопрос допустимости манёвра) и ДС. Выход метода: «обязателен для исполнения», либо «допустим», либо «запрещён». При программной реализации манёвром будет являться метод класса ТС.

Вопреки данному в [1] определению, *уступить дорогу* будем считать манёвром.

Перечень манёвров:

- начало движения (п. 8.1)
- перестроение (п. 8.1)
- поворот (п. 8.1)
- разворот (п. 8.1)
- остановка (п. 8.1)
- объезд (п. 9.2)
- обгон (п. 9.2)
- повышение скорости (п. 10)
- снижение скорости (п. 10)
- резкое торможение (п. 10.5)
- опережение (п. 11.5)
- уступить дорогу (п. 11.7) (сложный / условный манёвр; является комбинацией других манёвров, которая зависит от ДС; другими словами, этот манёвр — это совокупность других манёвров с условиями их применимости)

3. Формулы правовых норм ПДД

В зависимости от ДС у ТС и УД отличаются наборы атрибутов и допустимые манёвры ТС. Введём формулы двух основных типов правовых норм, определяющие ограничения на значения атрибутов и область определения манёвров.

3.1. Формула «Конструктор атрибутов»

Правовая норма вводит ограничения на значения атрибутов и состояний классов ТС и УД. Формула определяет новое значение атрибута в зависимости от других существующих атрибутов объектов ДС.

$$\textbf{\textit{if}}\ (\{\textit{Имя класса.Атрибут класса} == \textit{Значение атрибута}\})$$
$$\textbf{\textit{then}}\ (\textit{Имя класса.Атрибут класса} := \textit{Значение атрибута})$$

Пример. Текст: (п. 10.3) «Вне населенных пунктов разрешается движение: <...>

2. междугородним и маломестным автобусам на всех дорогах - не более 90 км/ч; <...>».

Формула:

if (УД.Населённый пункт == *False*) **and** (Автобус.Маломестный == *True* **or** Автобус.Междугородний == *True*) **then** (Автобус. Максимальная скорость, км/ч := 90)

3.2. Формула «Конструктор манёвров»

Правовая норма определяет допустимость выполнение манёвра в данной ДС. Формула порождает новый метод, который мы добавляем в коллекцию методов. Порядок аргументов метода: (ТС; ДС), где ТС — это ТС, относительно которого определяется допустимость данного манёвра.

$$\textit{Название манёвра}\,(\textit{ТС}; \textit{ДС}) := z,$$

где $z \in \{$*«обязателен для исполнения», «допустим», «запрещён»*$\}$

Пример. Текст: (п. 13.6)

«Если сигналы светофора или регулировщика разрешают движение одновременно трамваю и безрельсовым транспортным средствам, то трамвай имеет преимущество независимо от направления его движения. <...>.»

Формула (дорожная ситуация упрощена):

Уступить дорогу(ТС.Безрельсовое; ТС.Трамвай, УД.Регулируемый перекрёсток) := *обязательно для исполнения*

Уступить дорогу(ТС.Трамвай; ТС.Безрельсовое, УД.Регулируемый перекрёсток) := *запрещено*

4. Заключение

Предполагается, что в дальнейшем работа будет развиваться в двух направлениях. Первое — автоматическая обработка текста, второе — выявление непротиворечивости и полноты ПДД для разрешения ДС.

Список литературы

[1] Постановление Правительства РФ от 23.10.1993 N 1090 (ред. от 04.12.2018) «О правилах дорожного движения» // КонсультантПлюс

[2] Новая философская энциклопедия в 4-х томах / Научно-ред. совет: Стёпин В. С., Гусейнов А. А., Семигин Г. Ю., Огурцов А. П. — М.: Мысль, 2000. — Том 1.

[3] Jurafsky D., Martin J. H. Speech and Language Processing, 2nd Edition. — Prentice Hall, 2008.

[4] Теория государства и права. Часть 2. Теория права: Учебник / под ред. М. Н. Марченко. — М.: Зерцало-М, 2011.

[5] Перпер Е. М. Автоматическое построение модели нормативно-правового акта по его тексту // Тезисы докладов XXI Международной научной конференции студентов, аспирантов и молодых учёных «Ломоносов 2014» — М.: 2014.

[6] Кудрявцев В. Б., Гасанов Э. Э., Перпер Е. М. Автоматическая генерация программы, моделирующей нормативно-правовой акт // Интеллектуальные системы — М.: 2014. — Том 18, выпуск 2.

[7] Перпер Е. М. О синтаксическом анализе юридических текстов // Интеллектуальные системы — М.: 2016. — Том 20, выпуск 2.

[8] Перпер Е. М., Гасанов Э. Э., Кудрявцев В. Б. О семантическом анализе юридических текстов // Интеллектуальные системы — М.: 2018. — Том 22, выпуск 3.

Objects model of rules of the road

Mikhail Menkin

The present article deals with semantic analysis of legal documents. By legal document semantics we mean the mapping from legal document text to formal model. In this article, we describe one possible approach to formal modeling of rules of the road.

Keywords: semantic analysis of legal documents, pragmatic analysis, formal modeling.

Часть 2.
Специальные вопросы теории интеллектуальных систем

О конечных заданиях логических систем

Боков Г. В.

В работе рассматривается задача конечного представления логических систем пропозициональными исчислениями. Исследуются три типа логических систем: линейные, монотонные и импликативные. Для каждого из этих типов логических систем доказаны достаточные условия их конечного задания. Кроме того, доказан критерий конечного задания произвольной логической системы множества классических тавтологий.

Ключевые слова: логические системы, пропозициональные исчисления, конечное задание, правила вывода.

1. Введение

Пропозициональные исчисления являются мощным средством задания логических систем и процессов [8]. Заложенный в них инструментарий позволяет решать алгоритмические проблемы для широкого класса логических систем [1, 3, 6, 18, 19, 21], включая системы с нестандартными правилами вывода [2, 5, 20].

При моделировании логических систем часто возникает вопрос: какие логические системы допускают «простое» задание пропозициональными исчислениями, а какие нет? Поскольку каждую логическую систему можно рассматривать как обобщённое пропозициональное исчисление, то естественно ограничиться рассмотрением только таких заданий, которые в некоторой степени проще исходной логической системы. Как правило, простое задание основано на выделении систем образующих или порождающих элементов логической системы. Рассмотрим это более подробно.

Пусть дана логическая система $\mathcal{L} = \langle \mathbf{M}, \mathbf{Q} \rangle$ над множеством формул $\mathbf{M}$, замкнутом относительно правил $\mathbf{Q}$.

Определение 1. Правило вывода $(F_1, \ldots, F_n)\,/F_0$ *допустимо* в логической системе $\mathcal{L}$, если

$$\sigma F_1 \in \mathbf{M},\ \ldots,\ \sigma F_n \in \mathbf{M} \quad \Longrightarrow \quad \sigma F_0 \in \mathbf{M}$$

для любой подстановки σ. Множество всех допустимых в $\mathcal{L}$ правил вывода обозначим через $\mathbf{R}_{\mathcal{L}}$.

Определение 2. Исчисление $\mathcal{P} = \langle \mathbf{A}, \mathbf{R} \rangle$ *задаёт* логическую систему $\mathcal{L}$, если выполнены следующие условия:

1) $\mathbf{A} \subseteq \mathbf{M}$;

2) $\mathbf{Q} \subseteq \mathbf{R} \subseteq \mathbf{R}_{\mathcal{L}}$;

3) $[\mathcal{P}] = \mathbf{M}$, т.е. эквивалентность

$$\mathbf{A} \vdash_{\mathbf{R}} A \;\Leftrightarrow\; A \in \mathbf{M}$$

выполнена для любой формулы A. В этом случае, аксиомы $\mathbf{A}$ играют роль порождающих элементов, а $\mathbf{R}$ — роль порождающих правил вывода. Исчисление $\mathcal{P}$, задающее логическую систему $\mathcal{L}$, назовём *порождающим* для $\mathcal{L}$.

Накладывая ограничения на множества $\mathbf{A}$ и $\mathbf{R}$ можно получать порождающие исчисления для логической системы $\mathcal{L}$, обладающие заданными свойствами. Например, если $\mathbf{A}$ и $\mathbf{R}$ — конечные множества, то говорят, что $\mathcal{L}$ допускает *конечное* задание. Такие задания представляют наибольший интерес, как с практической [7, 9], так и с теоретической [4] точки зрения. В основу исследования положены результаты работы [17].

2. Основные определения и обозначения

Введем ограничения на исходный пропозициональный язык, состоящий из логических связок $\mathcal{F}$ и счетного множества пропозициональных переменных $\mathcal{V}$. Будем считать, что всем логическим связкам из $\mathcal{F}$ в стандартной интерпретации соответствуют булевы функции, существенно зависящие от всех своих переменных, причём разным логическим связкам соответствуют разные булевы функции. Для простоты будем отождествлять логические связки с их стандартными интерпретациями, т.е. будем считать, что $\mathcal{F} \subseteq \mathbf{P}_2$.

Следуя [16], определим несколько классов булевых функций.

Определение 3. Функция $f(x_1, \ldots, x_n) \in \mathbf{P}_2$ называется

$$\alpha\text{-функцией, если } f(x, \ldots, x) = x,$$
$$\beta\text{-функцией, если } f(x, \ldots, x) = 1,$$
$$\gamma\text{-функцией, если } f(x, \ldots, x) = 0,$$
$$\delta\text{-функцией, если } f(x, \ldots, x) = \overline{x}.$$

Обозначим через $\mathcal{F}_\alpha$, $\mathcal{F}_\beta$, $\mathcal{F}_\gamma$, $\mathcal{F}_\delta$ соответственно множества всех α-, β-, γ-, δ-функций в $\mathcal{F}$.

Определение 4. Функция $f(x_1, \ldots, x_n) \in \mathbf{P}_2$ называется *линейной*, если для неё имеет место соотношение

$$f(x_1, \ldots, x_n) = c_0 \oplus c_1 x_1 \oplus \ldots \oplus c_n x_n,$$

где $c_0, c_1, \ldots, c_n \in \{0, 1\}$ и $\oplus$ — сумма по модулю 2. Множество всех линейных функций в $\mathbf{P}_2$ обозначим через $\mathbf{L}$.

Определение 5. Функция $f(x_1, \ldots, x_n) \in \mathbf{P}_2$ называется *монотонной*, если для любых наборов $\widetilde{\alpha}, \widetilde{\beta} \in \{0, 1\}^n$ таких, что $\widetilde{\alpha} \leq \widetilde{\beta}$, имеет место соотношение

$$f(\widetilde{\alpha}) \leq f(\widetilde{\beta}).$$

Множество всех монотонных функций в $\mathbf{P}_2$ обозначим через $\mathbf{M}$.

Определение 6. Функция $f(x_1, \ldots, x_n, y) \in \mathbf{P}_2$ называется *импликативной*, если соотношение

$$f(a_1, \ldots, a_n, a_0) = 0 \;\Leftrightarrow\; a_0 < \min(a_1, \ldots, a_n)$$

выполнено для любых $a_1, \ldots, a_n, a_0 \in \{0, 1\}$. Множество всех импликативных функций в $\mathbf{P}_2$ обозначим через $\mathbf{I}$.

3. Линейные логические системы

Пусть $\mathcal{F} \subseteq \mathbf{L}$. Формулы, тавтологии и исчисления над связками $\mathcal{F}$ будем называть *линейными*. Обозначим через $\nu_A(a)$ число вхождений символа $a \in \mathcal{F} \cup \mathcal{V}$ в формулу A. Для линейных тавтологий верна лемма.

Лемма 1. *Любая переменная линейной тавтологии имеет чётное число вхождений.*

Доказательство. Линейная формула A от переменных $x_1, \ldots, x_n$ однозначно задает линейную функцию $f_A(x_1, \ldots, x_n)$, которая по определению 4 представима в виде:

$$f_A(x_1, \ldots, x_n) = c_0 + c_1 x_1 + \ldots + c_n x_n,$$

где $c_0, c_1, \ldots, c_n \in \{0, 1\}$. Несложно убедиться, что для $1 \leq i \leq n$ коэффициент $c_i = 0$ тогда и только тогда, когда $\nu_A(x_i)$ — чётное. Следовательно, каждая переменная x_i, для которой $\nu_A(x_i)$ нечётно, является существенной для функции f_A. Напомним, что мы предположили, что все логические связки из $\mathcal{F}$, задают булевы функции, существенно зависящие от всех своих переменных. Поэтому, если A является тавтологией, то функция f_A не имеет существенных переменных и, следовательно, любая переменная имеет чётное число вхождений. □

3.1. Линейное исчисление для произвольных связок

Определим множество правил вывода $\mathbf{R}_L$, состоящее из правил вида

$$\frac{A}{B}$$

для всех $A, B \in \mathbf{Fm}$ таких, что $A \models B$ и A, B содержат не более 3 логических связок из $\mathcal{F}$. Формулы A и B, такие что $A \vdash_{\mathbf{R}_L} B$ и $B \vdash_{\mathbf{R}_L} A$, будем называть *эквивалентными* и обозначать через $A \sim B$. Если при этом A и B эквивалентны как алфавитные деревья, то будем говорить, что A и B эквивалентны *сильно*. Заметим, что в силу определения изоморфизма алфавитных деревьев число вхождений символов из $\mathcal{F} \cup \mathcal{V}$ в сильно эквивалентные формулы A и B совпадают.

Логическая связка $f \in \mathcal{F}$ имеет *корневое вхождение* в формуле A, если $\mathtt{t}_A(\varepsilon) = f$. Вхождение $\alpha \in \mathbf{Pos}(A)$ связки f в формулу A назовем *сводимым*, если для любого собственного непустого начала $\beta \prec \alpha$ выполнено $\mathtt{t}_A(\beta) = g$, где g — логическая связка арности больше 1. Следующая лемма показывает, что некорневые сводимые вхождения логических связок в линейных формулах можно опустить ближе в корню дерева, представляющего эту формулу.

Лемма 2. *Для любой линейной формулы A и любого неконстантного символа $f \in \mathcal{F}$, имеющего некорневое сводимое вхождение в A, существует строго эквивалентная ей формула B такая, что $Head(A) = Head(B)$ и $\mathtt{t}_B(k) = f$ для некоторого $k \in \mathbb{N}$.*

Доказательство. Рассмотрим некорневое сводимое вхождение α символа f в формулу A. Докажем утверждение леммы индукцией по длине слова α.

Базис индукции: $|\alpha| = 1$. Тогда для $B = A$ утверждение леммы верно.

Шаг индукции: пусть утверждение верно для всех таких $\alpha \in \mathbf{Pos}(A)$, что $1 \leq |\alpha| < l$. Докажем его для α, имеющего длину l. Поскольку $|\alpha| > 1$, то найдутся такие $k, i \in \mathbb{N}$ и $\beta \in \mathbb{N}^*$, что $\alpha = ki\beta$. Возможны следующие случаи.

(1) $|\beta| = 0$. Тогда формула A представима в виде

$$h(A_1, \ldots, g(B_1, \ldots, f(C_1, \ldots, C_j, \ldots, C_s), \ldots, B_m), \ldots, A_n),$$

где $n,\ m$ и s — арности логических связок $h,\ g$ и f соответственно, $j \in \{1, \ldots, s\}$. Поскольку $s \geq 1$, то такое j всегда найдется. Применением правил $\mathbf{R}_L$ можно вывести строго эквивалентную формулу B вида

$$h(A_1, \ldots, f(C_1, \ldots, g(B_1, \ldots, C_j, \ldots, B_m), \ldots, C_s), \ldots, A_n),$$

для которой $Head(B) = Head(A)$ и $\mathfrak{t}_B(k) = f$, поэтому для случая $|\beta| = 0$ утверждение леммы верно.

(2) $|\beta| > 0$. Тогда найдется такое $j \in \mathbb{N}$ и $\gamma \in \mathbb{N}^*$, что $\beta = j\gamma$ и $\alpha = kij\gamma$. В этом случае формула A представима в виде

$$h(A_1, \ldots, g(B_1, \ldots, f'(C_1, \ldots, C_{j'}, \ldots, C_s), \ldots, B_m), \ldots, A_n),$$

где $n,\ m$ и s — арности логических связок $h,\ g$ и f' соответственно, $j' \in \{1, \ldots, s\} \setminus \{j\}$. Поскольку $s > 1$, то такое j' всегда найдется. Применением правил $\mathbf{R}_L$ можно вывести строго эквивалентную формулу C вида:

$$h(A_1, \ldots, f'(C_1, \ldots, g(B_1, \ldots, C_{j'}, \ldots, B_m), \ldots, C_s), \ldots, A_n).$$

Для формулы C имеем $Head(A) = Head(C)$ и $\mathfrak{t}_C(kj\gamma) = f$ и $|kj\gamma| = l - 1$. По индуктивному предположению для формулы C найдется такая строго эквивалентная формула B, что $Head(C) = Head(B)$ и $\mathfrak{t}_B(k') = f$ для некоторого $k' \in \mathbb{N}$. Поскольку отношение строгой эквивалентности является транзитивным, то B строго эквивалентна A и $Head(A) = Head(B)$.

Разбор всех случаев завершает доказательство. □

Следствие 1. *Для любой линейной формулы A и любого неконстантного символа $f \in \mathcal{F}$, имеющего некорневое сводимое вхождение в A, существует строго эквивалентная ей формула B такая, что $Head(B) = f$.*

Доказательство. Согласно Лемме 2 найдется такая строго эквивалентная формула C, что $Head(A) = Head(C)$ и $\mathfrak{t}_C(k) = f$ для некоторого $k \in \mathbb{N}$. Формулу C можно представить в виде:

$$h(A_1, \ldots, f(B_1, \ldots, B_j, \ldots, B_m), \ldots, A_n),$$

где n и m — арности логических связок h и f соответственно, $j \in \{1, \ldots, m\}$. Поскольку $m \geq 1$, то такое j всегда найдется. Применением правил $\mathbf{R}_L$ можно вывести строго эквивалентную формулу B вида

$$f(B_1, \ldots, h(A_1, \ldots, B_j, \ldots, A_n), \ldots, B_m),$$

для которого $\mathfrak{t}_B(\varepsilon) = f$. □

Лемма 3. *Для любой линейной формулы A существует такая эквивалентная ей формула B, что выполнены следующие условия:*

1) B не содержит унарных α-связок;

2) B содержит не более одной унарной δ-связки.

Доказательство. Обозначим через $n_\alpha(A)$ и $n_\delta(A)$ число вхождений в A унарных α-связок и унарных δ-связок соответственно. Докажем утверждение леммы индукцией по $n_\alpha(A)$ и $n_\delta(A)$.

Базис индукции: $n_\alpha(A) = 0$ и $n_\delta(A) \leq 1$. Тогда формула A удовлетворяет условиям 1, 2 и утверждение верно.

Шаг индукции: пусть $n_\alpha(A) > 0$ или $n_\delta(A) > 1$. Рассмотрим наименьшее вхождение $\alpha \in \mathbf{Pos}(A)$ унарной связки $f \in \mathcal{F}_\alpha \cup \mathcal{F}_\delta$. Поскольку нет одноместных β-,γ-функций, существенно зависящих от одной переменной, то для любого собственного подслова $\beta \prec \alpha$ символ $\mathfrak{t}_A(\beta)$ имеет арность, большую 1. Поэтому вхождение α символа f является сводимым. По следствию 1 найдется такая строго эквивалентная формула $B \in \mathbf{Fm}$, для которой $Head(B) = f$. Так как $n_\alpha(B) = n_\alpha(A)$ и $n_\delta(B) = n_\delta(A)$, то возможны два случая.

(1) $f \in \mathcal{F}_\alpha$. Тогда применяя правило

$$\frac{f(x)}{x}$$

к формуле B получим, что $A \sim A|_1$. Для формулы $C = A|_1$ имеем $n_\alpha(C) < n_\alpha(A)$ и $n_\delta(C) = n_\delta(A)$.

(2) $f \in \mathcal{F}_\delta$, тогда рассмотрим некорневое сводимое вхождение β унарного символа $g \in \mathcal{F}_\alpha \cup \mathcal{F}_\delta$ в формулу A. Поскольку $n_\delta(A) > 1$, то такое вхождение всегда найдется. По Лемме 2 найдется такая строго эквивалентная формула $C \in \mathbf{Fm}$, что $Head(B) = Head(C)$ и $\mathfrak{t}_C(k) = g$ для некоторого $k \in \mathbb{N}$. Тогда формула C имеет вид $f(g(D))$, для некоторой формулы $D \in \mathbf{Fm}$, причём $n_\alpha(C) = n_\alpha(A)$ и $n_\delta(C) = n_\delta(A)$.

Если $g \in \mathcal{F}_\alpha$, то с помощью правила

$$\frac{f(g(x))}{f(x)}$$

из C можно вывести $f(D)$, рассмотрение которой сводится к предыдущему случаю. Если же $g \in \mathcal{F}_\delta$, то применяя правило

$$\frac{f(g(x))}{x}$$

к формуле C получим, что $C \sim D$. Для формулы D имеем $n_\alpha(D) = n_\alpha(A)$ и $n_\delta(D) < n_\delta(A)$.

Во всех случаях для формулы A найдется такая формула B, что $A \sim B$ и либо $n_\alpha(B) < n_\alpha(A)$, либо $n_\delta(B) < n_\delta(A)$. Тогда по предположению индукции лемма доказана. □

Лемма 4. *Для любой линейной формулы существует эквивалентная ей формула, не содержащая α-связок.*

Доказательство. Рассмотрим линейную формулу A с l α-связками. Докажем индукцией по $l \geq 0$, что существует эквивалентная формула B, не содержащая α-связок.

Базис индукции: $l = 0$. Тогда для $B = A$ утверждение верно.

Шаг индукции: пусть утверждение верно для всех $0 \leq l' < l$, докажем его для l. Поскольку $l > 0$, то найдется вхождение символа $f \in \mathcal{F}_\alpha$

в формулу A. Согласно Лемме 3 можно считать, что A не содержит одноместных символов из $\mathcal{F}_\alpha$ и содержит не более одного символа из $\mathcal{F}_\delta$. Без ограничения общности будем считать, что $\mathcal{F}$ содержит только одну 0-местную β-функцию и одну 0-местную γ-функцию, которые обозначим через 1 и 0, соответственно. Возможны два случая:

(1) Если A не содержит одноместных символов из $\mathcal{F}_\delta$, тогда по следствию 1 найдется такая строго эквивалентная формула B, что $B = f(B_1, \ldots, B_n)$, где n — арность символа f и $n > 1$. Поскольку B не имеет одноместных связок, то последовательным применением правил $\mathbf{R}_L$ из B можно вывести формулу C вида

$$f(\underbrace{x_1, \ldots, x_1}_{n_1}, \underbrace{x_2, \ldots, x_2}_{n_2}, \ldots \underbrace{x_k, \ldots, x_k}_{n_k}, \underbrace{1, \ldots, 1}_{n_{k+1}}, \underbrace{0, \ldots, 0}_{n_{k+2}}, D), \quad (1)$$

где $\{x_1, \ldots, x_k\}$ — все переменные формулы A, $n_1 \in \mathbb{N}$, $n_i \in \mathbb{N}_+$, $i = 2, \ldots, k+2$ и D — это формула вида $g(D_1, \ldots, D_m)$, где m — арность g и

$$D_1 = \begin{cases} x_i, \text{ если } n_{i-1} \neq 0 \text{ и } 0 \leq n_i < \nu_A(x_i), \\ 1, \text{ если } n_k = \nu_A(x_k) \text{ и } 0 \leq n_{k+1} < \nu_A(1), \\ 0, \text{ если } n_{k+1} = \nu_A(1). \end{cases}$$

Причем, для каждого $1 \leq i \leq k+2$, если $n_i = 0$, то $n_{i+1} = 0$, а также, если $n_{i+1} \neq 0$, то $n_i = \nu_A(x_i)$. По сути A в этом случае представляет собой дерево, внутренние вершины которого имеют хотя бы два выходных ребра, а правил $\mathbf{R}_L$ перебалансируют это дерево.

(2) Если A содержит одноместный символов $h \in \mathcal{F}_\delta$, тогда по следствию 1 найдется такая строго эквивалентная формула B, что $B = h(f(B_1, \ldots, B_n)$, где n — арность символа f и $n > 1$. Поскольку B_i не имеют одноместных связок, то последовательным применением правил $\mathbf{R}_L$ из B можно вывести формулу C вида

$$h(f(\underbrace{x_1, \ldots, x_1}_{n_1}, \underbrace{x_2, \ldots, x_2}_{n_2}, \ldots \underbrace{x_k, \ldots, x_k}_{n_k}, \underbrace{1, \ldots, 1}_{n_{k+1}}, \underbrace{0, \ldots, 0}_{n_{k+2}}, D)),$$

где $\{x_1, \ldots, x_k\}$ — все переменные формулы A, $n_1 \in \mathbb{N}$, $n_i \in \mathbb{N}_+$, $i = 2, \ldots, k+2$ и D — это формула вида $g(D_1, \ldots, D_m)$, где m — арность

g и

$$D_1 = \begin{cases} x_i, \text{ если } n_{i-1} \neq 0 \text{ и } 0 \leq n_i < \nu_A(x_i), \\ 1, \text{ если } n_k = \nu_A(x_k) \text{ и } 0 \leq n_{k+1} < \nu_A(1), \\ 0, \text{ если } n_{k+1} = \nu_A(1). \end{cases}$$

Причем, для каждого $1 \leq i \leq k+2$, если $n_i = 0$, то $n_{i+1} = 0$, а также, если $n_{i+1} \neq 0$, то $n_i = \nu_A(x_i)$. Из этой формулы с помощью правил $\mathbf{R}_L$ получим формулу

$$f(\underbrace{x_1, \ldots, x_1}_{n_1}, \underbrace{x_2, \ldots, x_2}_{n_2}, \ldots \underbrace{x_k, \ldots, x_k}_{n_k}, \underbrace{1, \ldots, 1}_{n_{k+1}}, \underbrace{0, \ldots, 0}_{n_{k+2}}, D'),$$

где $D' = g(D_1, \ldots, h(D_m))$.

В обоих случаях найдется формула B вида 1, строго эквивалентная A. Теперь рассмотрим случаи:

1) Если $n_i \neq 0$ и $n_{i+1} = 0$, где $i \leq k+1$, тогда по Лемме 1 $n_j = 2m_j,\ j = 1, \ldots, i-1$. Так как $f \in \mathcal{F}_\alpha$, то $n = 2l+1$, поэтому $n_i = 2m_i$, где $m_i = l - \sum_{j=1}^{i-1} m_j \in \mathbb{N}$. Применяя правило из $\mathbf{R}_L$, заданное схемой

$$\frac{f(\underbrace{x_1, \ldots, x_1}_{2m_1}, \underbrace{x_2, \ldots, x_2}_{2m_2}, \ldots \underbrace{x_i, \ldots, x_i}_{2m_i}, y)}{y}$$

получим, что $A \sim D$.

2) Если $n_{k+1} = 2m_{k+1}$, тогда по Лемме 1 $n_j = 2m_j,\ j = 1, \ldots, k$. Следовательно, $n_{k+2} = 2m_{k+2}$, где $m_{k+2} = l - \sum_{j=1}^{k+1} m_j \in \mathbb{N}$. Применяя правило из $\mathbf{R}_L$, заданное схемой

$$\frac{f(\underbrace{x_1, \ldots, x_1}_{2m_1}, \underbrace{x_2, \ldots, x_2}_{2m_2}, \ldots \underbrace{x_{k+2}, \ldots, x_{k+2}}_{2m_{k+2}}, y)}{y}$$

получим, что $A \sim D$.

3) Если $n_{k+1} = 2m_{k+1}+1$, тогда $n_{k+2} \neq 0$ и $D_1 = 0$, поэтому применяя правила из $\mathbf{R}_L$ можно получить формулу

$$f(\underbrace{x_1, \ldots, x_1}_{n_1}, \underbrace{x_2, \ldots, x_2}_{n_2}, \ldots \underbrace{x_k, \ldots, x_k}_{n_k} \underbrace{1, \ldots, 1}_{n_{k+1}-1}, \underbrace{0, \ldots, 0}_{n_{k+2}+1}, D'),$$

строго эквивалентную A, для которой $D' = g(D'_1, \ldots, D_m)$ и $D'_1 = 1$. Откуда применяя правила $\mathbf{R}_L$ можно вывести формулу D'.

В любом случае найдется такая формула $C \in \mathbf{Fm}$, что $A \sim C$ и число вхождений α-связок в C на единицу меньше, чем в A. Лемма доказана. □

Лемма 5. *Для любой линейной формулы существует эквивалентная ей формула, содержащая не более одного символа из $\mathcal{F}_\sigma$, где $\sigma \in \{\beta, \gamma, \delta\}$.*

Доказательство. Утверждение леммы будем доказывать индукцией по числу вхождений n символов из $\mathcal{F}_\sigma$ в формулу A.

Базис индукции: $n \leq 1$. Тогда для $B = A$ утверждение верно.

Шаг индукции: пусть утверждение леммы верно для n, такого что $1 \leq n < k$. Покажем, что оно верно и для $n = k$. Согласно Лемме 3 можно считать, что A не содержит одноместных символов из $\mathcal{F}_\alpha$ и содержит не более одного символа из $\mathcal{F}_\delta$. Применяя те же рассуждения, что и в Лемме 4, можно считать, что A не содержит одноместных символов из $\mathcal{F}_\delta$ и $\mathcal{F}$ содержит только одну 0-местную β-функцию и одну 0-местную γ-функцию, которые также обозначим через 1 и 0, соответственно.

Поскольку $k > 1$, то найдутся вхождения символов $f, g \in \mathcal{F}_\sigma$ в формулу A. Тогда по Лемме 2 и следствию 1 найдется такая строго эквивалентная формула B вида $B = f(gB_1, \ldots, B_n)$, в которой расстановка скобок для связки g определена её арностью. Поскольку B не имеет одноместных связок, то последовательным применением правил $\mathbf{R}_L$ из B можно вывести формулу C вида

$$f(g\underbrace{x_1, \ldots, x_1}_{n_1}, \underbrace{x_2, \ldots, x_2}_{n_2}, \ldots \underbrace{x_k, \ldots, x_k}_{n_k}, \underbrace{1, \ldots, 1}_{n_{k+1}}, \underbrace{0, \ldots, 0}_{n_{k+2}}, D),$$

где $\{x_1, \ldots, x_k\}$ — все переменные формулы A, $n_1 \in \mathbb{N}$, $n_i \in \mathbb{N}_+$, $i = 2, \ldots, k+2$ и D — это формула вида $h(D_1, \ldots, D_m$, где m — арность h и

$$D_1 = \begin{cases} x_i, \text{ если } n_{i-1} \neq 0 \text{ и } 0 \leq n_i < \nu_A(x_i), \\ 1, \text{ если } n_k = \nu_A(x_k) \text{ и } 0 \leq n_{k+1} < \nu_A(1), \\ 0, \text{ если } n_{k+1} = \nu_A(1). \end{cases}$$

Причем, для каждого $1 \leq i \leq k+2$, если $n_i = 0$, то $n_{i+1} = 0$, а также, если $n_{i+1} \neq 0$, то $n_i = \nu_A(x_i)$. Рассмотрим случаи:

1) Если $n_i \neq 0$ и $n_{i+1} = 0$, где $i \leq k+1$, тогда по Лемме 1 $n_j = 2m_j$, $j = 1, \ldots, i-1$. Так как $f, g \in \mathcal{F}_\sigma$ и $\sigma \in \{\beta, \gamma, \delta\}$, то $n = 2l + 1$,

поэтому $n_i = 2m_i$, где $m_i = l - \sum_{j=1}^{i-1} m_j \in \mathbb{N}$. Применяя правило из $\mathbf{R}_L$, заданное схемой

$$\frac{f(g\underbrace{x_1,\ldots,x_1}_{2m_1},\underbrace{x_2,\ldots,x_2}_{2m_2},\ldots\underbrace{x_i,\ldots,x_i}_{2m_i},y)}{y}$$

получим, что $A \sim D$.

2) Если $n_{k+1} = 2m_{k+1}$, тогда по Лемме 1 $n_j = 2m_j,\ j = 1,\ldots,k$. Следовательно, $n_{k+2} = 2m_{k+2}$, где $m_{k+2} = l - \sum_{j=1}^{k+1} m_j \in \mathbb{N}$. Применяя правило из $\mathbf{R}_L$, заданное схемой

$$\frac{fg\underbrace{x_1\ldots x_1}_{2m_1}\underbrace{x_2\ldots x_2}_{2m_2}\ldots\underbrace{x_{k+2}\ldots x_{k+2}}_{2m_{k+2}}y}{y}$$

получим, что $A \sim D$.

3) Если $n_{k+1} = 2m_{k+1}+1$, тогда $n_{k+2} \neq 0$ и $D_1 = 0$, поэтому применяя правила из $\mathbf{R}_L$ можем получить формулу

$$f(g\underbrace{x_1,\ldots,x_1}_{n_1},\underbrace{x_2,\ldots,x_2}_{n_2},\ldots\underbrace{x_k,\ldots,x_k}_{n_k},\underbrace{1,\ldots,1}_{n_{k+1}-1},\underbrace{0,\ldots,0}_{n_{k+2}+1},D'),$$

строго эквивалентную A, где $D' = g(D'_1,\ldots,D_m)$ и $D'_1 = 1$. Откуда применяя правила $\mathbf{R}_L$ можно вывести формулу D'.

В любом случае найдется такая линейная формула C, что $A \sim C$ и число вхождений символов из $\mathcal{F}_\sigma$ в C меньше, чем в A. □

Если n — максимальная арность связок из $\mathcal{F}$, то обозначим через $\mathbf{A}_L$ множество всех тавтологий над $\mathcal{F}$, не содержащих символов из $\mathcal{F}_\alpha$, содержащие не более одного символа из $\mathcal{F}_\sigma$ для каждого $\sigma \in \{\beta,\gamma,\delta\}$ и не более $3n$ вхождений переменных. Определим линейное пропозициональное исчисление $\mathcal{P}_L = \langle \emptyset, \mathbf{A}_L \cup \mathbf{R}_L \rangle$ над схемами аксиом $\mathbf{A}_L$ и правилами вывода $\mathbf{R}_L$. Для исчисления $\mathcal{P}_L$ верна следующая лемма.

Лемма 6. *Исчисление $\mathcal{P}_L$ является конечным заданием логической системы $\langle \mathbf{Cl}, \mathbf{R} \rangle$ для любого $\mathbf{R} \subseteq \mathbf{R}_L$.*

Доказательство. Согласно леммам 4 и 5 для любой линейной тавтологии A существует эквивалентная ей формула $B \in \mathbf{A}_L$. Поэтому

$$\vdash_{\mathcal{P}_L} A$$

для любой тавтологии $A \in \mathbf{Cl}$ и, следовательно, исчисление $\mathcal{P}_L$ является конечным заданием системы $\langle \mathbf{Cl}, \mathbf{R} \rangle$ для любого $\mathbf{R} \subseteq \mathbf{R}_L$. □

При наложении ограничения на связки множества $\mathcal{F}$ можно значительно упростить исчисление $\mathcal{P}_L$, порождающее множество линейных тавтологий. Следующие два раздела демонстрируют примеры таких исчислений.

3.2. Линейное исчисление для эквивалентности

Предположим, что функция $x \oplus y \oplus 1 \in [\mathcal{F}]$. Прежде, чем приступить к определению исчисления, напомним некоторые вспомогательные понятия.

Определение 7. *Системой тождеств* над логическими связками $\mathcal{F}$ — это подмножество пар формул $\mathbf{S} \subseteq \mathbf{Fm} \times \mathbf{Fm}$. Каждая система тождеств $\mathbf{S}$ задает отношение эквивалентности $\sim_{\mathbf{S}}$ на множестве формул $\mathbf{Fm}$, являющееся рефлексивным, симметричным и транзитивным замыканием множества пар

$$\mathbf{S}^* = \{(A, A[B]_\alpha) \mid \alpha \in \mathbf{Pos}(A) \ \text{ и } \ (A|_\alpha, B) \in \mathbf{S}\}.$$

При этом для каждой такой пары $(A, A[B]_\alpha)$ говорят, что формула $A[B]_\alpha$ получена из формулы A заменой подформулы $A|_\alpha$ на эквивалентную ей формулу B. Будем говорить, что формулы A и B эквивалентны относительно $\mathbf{S}$, если $A \sim_{\mathbf{S}} B$.

Система тождеств $\mathbf{S}$ называется *полной*, если любые эквивалентные формулы в $\mathbf{Fm}$ эквивалентным относительно $\mathbf{S}$, т.е. для любых $A, B \in \mathbf{Fm}$ выполнено

$$f_A \equiv f_B \quad \Longrightarrow \quad A \sim_{\mathbf{S}} B.$$

Согласно теореме Линдона [13] для любой системы связок $\mathcal{F} \subseteq P_2$ существует конечная полная система тождеств $\mathbf{S}$. Стоит отметить, что Линдон не только доказал, что для каждого замкнутого класса в P_2 система формул в его базисе имеет конечную полную систему тождеств, но и привел пример замкнутого класса в $\mathbf{P}_7$, для системы формул которого

не существует такой системы тождеств [14]. Позже было показано, что данные классы существуют и в логиках меньшей значности: Вишин [10] привел пример в $\mathbf{P}_4$, а Мурский [15] в $\mathbf{P}_3$.

Пусть $\mathbf{S}$ — конечная полная система тождеств для $\mathcal{F}$ и $x \leftrightarrow y$ — формула, выражающая функцию $x \oplus y \oplus 1 \in [\mathcal{F}]$. Тогда определим конечное множество $\mathbf{R}_L^1$, состоящее из правил трех типов:

$$\frac{x_1 \leftrightarrow y_1,\ \ldots,\ x_n \leftrightarrow y_n}{f(x_1, \ldots, x_n) \leftrightarrow f(y_1, \ldots, y_n)} \tag{2}$$

для каждого $f \in \mathcal{F}$,

$$\frac{x \leftrightarrow y}{y \leftrightarrow x} \tag{3}$$

и

$$\frac{x,\ x \leftrightarrow y}{y}. \tag{4}$$

Определим линейное исчисление $\mathcal{P}_L^1 = \langle \emptyset, \mathbf{A}_L^1 \cup \mathbf{R}_L^1 \rangle$ с множеством схем аксиом

$$\mathbf{A}_L^1 = \{x \leftrightarrow x\} \cup \{A \leftrightarrow B \mid (A, B) \in \mathbf{S}\}$$

и множеством правил вывода $\mathbf{R}_L^1$. Ясно, что исчисление $\mathcal{P}_L^1$ конечно. Докажем вспомогательную лемму.

Лемма 7. *Если* $(A \leftrightarrow B) \in [\mathcal{P}_L^1]$, *то* $(C[A] \leftrightarrow C[B]) \in [\mathcal{P}_L^1]$ *для любой формулы* C.

Доказательство. Заметим, что для любого $\alpha \in \mathbf{Pos}(C)$ формула $C|_\alpha \leftrightarrow C|_\alpha$ является подстановочным вариантом аксиомы $x \leftrightarrow x$ из $\mathbf{A}_L^1$. Так как формула $C[A] \leftrightarrow C[B]$ выводима из формулы $A \leftrightarrow B$ и формул $C|_\alpha \leftrightarrow C|_\alpha$ для $\alpha \in \mathbf{Pos}(C)$ с помощью применения конечного числа правил вида (2), то $C[A] \leftrightarrow C[B]$ выводима в $\mathcal{P}_L^1$. □

Покажем, что исчисление $\mathcal{P}_L^1$ является конечным заданием множества тавтологий.

Лемма 8. *Исчисление* $\mathcal{P}_L^1$ *является конечным заданием логической системы* $\langle \mathbf{Cl}, \mathbf{R} \rangle$ *для любого* $\mathbf{R} \subseteq \mathbf{R}_L^1$.

Доказательство. Рассмотрим произвольную тавтологию $A \in \mathbf{Cl}$. Так как $\mathbf{S}$ — конечная полная система тождеств для $\mathcal{F}$, то $A \sim_{\mathbf{S}} B$ для $B = (x \leftrightarrow x)$. Докажем индукцией по определению отношения $\sim_{\mathbf{S}}$, что $A \leftrightarrow B \in [\mathcal{P}_L^1]$.

Базис индукции: состоит из следующих случаев:

1) $A = B$. Тогда $A \leftrightarrow B$ является подстановочным вариантом аксиомы $x \leftrightarrow x$ из $\mathbf{A}^1_L$;

2) $(A, B) \in \mathbf{S}^*$. Тогда для $B = A[C]_\alpha$, для некоторой формулы C и позиции $\alpha \in \mathbf{Pos}(A)$ таких, что $(A|_\alpha, C) \in \mathbf{S}$. Так как $A|_\alpha \leftrightarrow C$ — аксиома из $\mathbf{A}^1_L$, то $A \leftrightarrow B$ выводима в $\mathcal{P}^1_L$ по Лемме 7.

Шаг индукции: состоит из следующих случаев:

1) $B \sim_{\mathbf{S}} A$. По предположению индукции

$$\vdash_{\mathcal{P}^1_L} B \leftrightarrow A.$$

Так как $A \leftrightarrow B$ выводима из $B \leftrightarrow A$ с помощью правила (3), то $A \leftrightarrow B$ выводима в $\mathcal{P}^1_L$;

2) $A \sim_{\mathbf{S}} C$ и $C \sim_{\mathbf{S}} B$. По предположению индукции

$$A \leftrightarrow C,\ C \leftrightarrow B \in [\mathcal{P}^1_L].$$

По Лемме 7

$$(A \leftrightarrow C) \leftrightarrow (A \leftrightarrow B) \in [\mathcal{P}^1_L].$$

Так как $A \leftrightarrow B$ выводима из $A \leftrightarrow C$ и $(A \leftrightarrow C) \leftrightarrow (A \leftrightarrow B)$ с помощью правила (4), то $A \leftrightarrow B$ выводима в $\mathcal{P}^1_L$.

Из доказанного следует, что $(x \leftrightarrow x) \leftrightarrow A \in [\mathcal{P}^1_L]$. Поскольку $x \leftrightarrow x$ — аксиома из $\mathbf{A}^1_L$, то A выводима в $\mathcal{P}^1_L$ с помощью правила (4). Таким образом, $[\mathcal{P}^1_L] = \mathbf{Cl}$ и, следовательно, $\mathcal{P}^1_L$ является конечным заданием логической системы $\langle \mathbf{Cl}, \mathbf{R} \rangle$ для любого $\mathbf{R} \subseteq \mathbf{R}^1_L$. □

3.3. Линейное исчисление для унарных связок

Предположим, что функция $x \oplus y \oplus 1 \notin [\mathcal{F}]$, тогда $\mathcal{F}$ состоит только из унарных линейных функций и констант. Определим конечное множество схем аксиом

$$\mathbf{A}^2_L = \{c \mid c \in \mathcal{F}_\beta\} \cup \{f(c) \mid f \in \mathcal{F}_\delta,\ c \in \mathcal{F}_\gamma\}$$

и конечное множество правил вывода $\mathbf{R}^2_L$, состоящее из правил:

$$\frac{x}{f(x)}, \quad \frac{g(x)}{g(f(x))}, \quad \frac{x}{g(h(x))}$$

для всех $f \in \mathcal{F}_\alpha$ и $g, h \in \mathcal{F}_\delta$. Пусть $\mathcal{P}_L^2 = \langle \emptyset, \mathbf{A}_L^2 \cup \mathbf{R}_L^2 \rangle$ — линейное пропозициональное исчисление над схемами аксиом $\mathbf{A}_L^2$ и правилами вывода $\mathbf{R}_L^2$. Покажем, что $\mathcal{P}_L^2$ является конечным заданием множества тавтологий.

Лемма 9. *Исчисление $\mathcal{P}_L^2$ является конечным заданием логической системы $\langle \mathbf{Cl}, \mathbf{R} \rangle$ для любого $\mathbf{R} \subseteq \mathbf{R}_L^2$.*

Доказательство. Так как $\mathcal{F}$ состоит из унарных связок и констант, то каждая тавтология $A \in \mathbf{Cl}$ представима в виде

$$A = f_1(f_2(\dots f_n(c))),$$

где $f_i \in \mathcal{F}_\alpha \cup \mathcal{F}_\delta$ и $c \in \mathcal{F}_\beta \cup \mathcal{F}_\gamma$, причём

- если $c \in \mathcal{F}_\beta$, то A содержит четное число вхождений символов из $\mathcal{F}_\delta$;
- если $c \in \mathcal{F}_\gamma$, то A содержит нечетное число вхождений символов из $\mathcal{F}_\delta$.

Докажем индукцией по n, что $A \in [\mathcal{P}_L^2]$.

Базис индукции: $n = 0$ и $c \in \mathcal{F}_\beta$, либо $n = 1$ и $c \in \mathcal{F}_\gamma$. Тогда A является аксиомой из $\mathbf{A}_L^2$ и, следовательно, $A \in [\mathcal{P}_L^2]$.

Шаг индукции: пусть $f_1 \in \mathcal{F}_\alpha \cup \mathcal{F}_\delta$ и $B = f_2(f_3(\dots f_n(c)) \in [\mathcal{P}_L^2]$. Рассмотрим случаи

1) если $f_1 \in \mathcal{F}_\alpha$, то A выводима из $B = f_2(f_3(\dots f_n(c))$ с помощью $\mathbf{R}_L^2$;
2) если $f_1 \in \mathcal{F}_\delta$ и $f_2 \in \mathcal{F}_\alpha$, то A выводима из $B = f_1(f_3(\dots f_n(c))$ с помощью $\mathbf{R}_L^2$;
3) если $f_1 \in \mathcal{F}_\delta$ и $f_2 \in \mathcal{F}_\delta$, то A выводима из $B = f_3(f_4(\dots f_n(c))$ с помощью $\mathbf{R}_L^2$.

В любом случае A выводима из некоторой формулы $B \in [\mathcal{P}_L^2]$ с помощью правил $\mathbf{R}_L^2$. Поэтому $A \in [\mathcal{P}_L^2]$. Таким образом, $[\mathcal{P}_L^2] = \mathbf{Cl}$ и, следовательно, $\mathcal{P}_L^2$ является конечным заданием логической системы $\langle \mathbf{Cl}, \mathbf{R} \rangle$ для любого $\mathbf{R} \subseteq \mathbf{R}_L^2$. □

4. Монотонные логические системы

Пусть $\mathcal{F} \subseteq \mathbf{M}$. Определим конечное множество схем аксиом

$$\mathbf{A}_M = \{c \mid c \in \mathcal{F},\ c \equiv 1\}$$

и конечное множество правил вывода $\mathbf{R}_M$, состоящее из правил вида:

$$\frac{x_{i_1},\ \ldots,\ x_{i_k}}{f(x_1, \ldots, x_n)}$$

для всех $k, n \geq 0$, $f \in \mathcal{F}$ арности n и $i_1, \ldots, i_k \in \{1, \ldots, n\}$ таких, что

$$x_{i_1}, \ldots, x_{i_k} \models f(x_1, \ldots, x_n).$$

Данные правила основаны на том, что сокращенные дизъюнктивные нормальные формы (д.н.ф.) монотонных функций не содержат отрицаний [16]. Поэтому, если $f(x_1, \ldots, x_n) \in \mathbf{M}$ и её сокращенная д.н.ф. имеет вид

$$\bigvee_{i=1}^{k} x_{i_1} \wedge \ldots \wedge x_{i_{n_i}}$$

где все $i_j \in \{1, \ldots, n\}$, то формула $f(A_1, \ldots, A_n)$ всегда является логическим следствием формул $A_{i_1}, \ldots, A_{i_k}$.

Рассмотрим пропозициональное исчисление $\mathcal{P}_M = \langle \emptyset, \mathbf{A}_M \cup \mathbf{R}_M \rangle$ над схемами аксиом $\mathbf{A}_M$ и правилами вывода $\mathbf{R}_M$. Покажем, что $\mathcal{P}_M$ является конечным заданием множества тавтологий.

Лемма 10. *Исчисление $\mathcal{P}_M$ является конечным заданием логической системы $\langle \mathbf{Cl}, \mathbf{R} \rangle$ для любого $\mathbf{R} \subseteq \mathbf{R}_M$.*

Доказательство. Если $1 \notin \mathcal{F}$, то $\mathbf{Cl} = \emptyset$, то утверждение верно. Иначе, $\mathbf{A}_M \neq \emptyset$. Докажем индукцией по глубине формул, что

$$A \in \mathbf{Cl} \quad \Longrightarrow \quad A \in [\mathcal{P}_M]$$

для любой формулы A.

Базис индукции: $A = c$ для $c \in \mathcal{F}$. Тогда $A \in \mathbf{A}_M$.

Шаг индукции: пусть утверждение верно для $A_1, \ldots, A_n$, докажем его для $A = f(A_1, \ldots, A_n)$ с $f \notin \mathcal{F}$. Рассмотрим сокращенную д.н.ф. функции f:

$$\bigvee_{i=1}^{k} x_{i_1} \& \ldots \& x_{i_{n_i}}$$

Так как $A \in \mathbf{Cl}$, то для некоторого конъюнкта $x_{i_1}\& \ldots \& x_{i_{n_i}}$ выполнено $A_{i_1}, \ldots, A_{i_{n_i}} \in \mathbf{Cl}$. По индуктивному предположению $A_{i_1}, \ldots, A_{i_{n_i}}$ выводимы в $\mathcal{P}_M$. Тогда применяя правило

$$\frac{x_{i_1}, \ldots, x_{i_{n_i}}}{f(x_1, \ldots, x_n)}$$

из $\mathbf{R}_M$ можно вывести формулу A. Поэтому A также выводима в $\mathcal{P}_M$.

Так как $\mathbf{A}_M \subseteq \mathbf{Cl}$ и для каждого правила

$$\frac{x_{i_1}, \ldots, x_{i_{n_i}}}{f(x_1, \ldots, x_n)}$$

из $\mathbf{R}_M$ выполнено $x_{i_1}, \ldots, x_{i_{n_i}} \models f(x_1, \ldots, x_n)$, то всякая формула A, выводимая в $\mathcal{P}_M$, является тавтологией. Следовательно, исчисление $\mathcal{P}_M$ является конечным заданием логической системы $\langle \mathbf{Cl}, \mathbf{R}_M \rangle$. □

5. Импликативные логические системы

Пусть $[\mathcal{F}] \cap \mathbf{I} \neq \emptyset$. Логические системы над данными связками будем называть *импликативными*. Л. Хенкин в своей работе [22] показал, что каждое пропозициональное исчисление, содержащее классическую импликацию, конечно-порождено относительной modus ponens. Покажем, что это верно и для любых импликативных логических систем. Отметим, что для доказательства полноты системы аксиом мы будем использовать известный метод Кальмара [11, 23]

Рассмотрим произвольную функцию $f(x_1, \ldots, x_n, y) \in [\mathcal{F}] \cap \mathbf{I}$ и произвольную формулу $F \in \mathbf{Fm}$ от переменных $x_1, \ldots, x_n, y$, выражающей функцию f. В этом случае формулу $F(x_1, \ldots, x_n, y)$ будем обозначать через $x_1 \ldots x_n \to y$. Если $x = x_1 = \ldots = x_n$, то для краткости будем писать $x \to y$ вместо $\underbrace{x \ldots x}_{n} \to y$. Определим множество правил вывода $\mathbf{R}_I(f)$, состоящее из единственного правила

$$\frac{x_1,\ \ldots,\ x_n,\ x_1 \ldots x_n \to y]}{y},$$

которое будем называть *обобщенным правилом modus ponens* или просто правилом modus ponens, когда вид правило определяется контекстом. По определению 6, выполнено

$$x_1,\ \ldots,\ x_n,\ x_1 \ldots x_n \to y \models y.$$

Теперь определим схемы аксиом. Без ограничения общности будем считать, что $\mathcal{F} = \{\to, \varphi\}$, где $\to$ — импликативная функция $x_1x_2 \to y$, а φ — произвольная m-арная функция из $\mathbf{P}_2$. Рассмотрим множество схем аксиом $\mathbf{A}_I$, состоящее из тавтологий:

1. $x \to (y_1y_2 \to x)$
2. $x_1x_2 \to x_i$
3. $(x_1x_2 \to (y_1y_2 \to z)) \to ((x_1x_2 \to y_1)(x_1x_2 \to y_2) \to (x_1x_2 \to z))$
4. $(x_i \to y) \to ((x_1x_2 \to z) \to y) \to y)$

где $i \in \{1, 2\}$.

К этой системе добавим 2^m аксиом, определяющих значения функции φ на всевозможных значения своих переменных. Пусть $x_1, \ldots, x_m$ — различные переменные, если $(x'_1, \ldots, x'_m) \in E_2^m$, то положим $\varphi' = \varphi(x'_1, \ldots, x'_m) \in E_2$. Пусть y — новая переменная, не встречающаяся в $x_1, \ldots, x_m$. Если $x'_i = 1$, то через x_i^* обозначим формулу $(x_i \to y) \to y$. Если же $x'_i = 0$, то через x_i^* обозначим формулу $x_i \to y$. По аналогии, если $\varphi' = 1$, то через φ^* обозначим формулу $(\varphi(x_1, \ldots, x_m) \to y) \to y$, если же $\varphi' = 0$, то через φ^* обозначим формулу $\varphi(x_1, \ldots, x_m) \to y$. Таким образом, к уже определенным аксиомам добавим еще 2^m аксиом вида

$$x_1^* \to (x_2^* \to \ldots (x_m^* \to \varphi^*))$$

В дальнейшем, если встретиться формула $A \in \mathbf{Fm}$ и $x_1, \ldots, x_n$ — различные переменные в A, $(x'_1, \ldots, x'_n) \in E_2^n$ и $A' = f_A(x'_1, \ldots, x'_n)$, то через A^* будем обозначать формулу $(A \to y) \to y$, если $A' = 1$, и $A \to y$, если $A' = 0$.

Рассмотрим пропозициональное исчисление $\mathcal{P}_I(f) = \langle \emptyset, \mathbf{A}_I \cup \mathbf{R}_I(f) \rangle$ над схемами аксиом $\mathbf{A}_I$ и правилами вывода $\mathbf{R}_I(f)$. Покажем, что $\mathcal{P}_I(f)$ является конечным заданием множества тавтологий. В данном разделе под $\vdash$ будем понимать $\vdash_{\mathbf{P}_I(f)}$.

Для исчисления $\mathcal{P}_I$ верна так называемая Теорема дедукции [12]: если $\Gamma,\ A_1,\ A_2 \vdash B$, то $\Gamma \vdash A_1A_2 \to B$, для любых формул $A_1,\ A_2$ и B и любого множества формул Γ.

Лемма 11. *Если* $\Gamma,\ A_1,\ A_2 \vdash B$, *то* $\Gamma \vdash A_1A_2 \to B$, *для любых формул* $A_1,\ A_2$ *и* B *и любого множества формул* Γ.

Доказательство. Доказательство будем вести индукцией по длине вывода $F_1, \ldots, F_n = B$ формулы B из $\Gamma,\ A_1,\ A_2$.

Если B - аксиома, либо $B \in \Gamma$, тогда доказательство следует из аксиомы 1. Если $B = A_i$, то доказательство следует из аксиомы 2.

Пусть B - это результат применения обобщенной операции modus ponens к формулам $F_{i_1},\ F_{i_2},\ F_j$, причем F_j имеет вид $F_{i_1}F_{i_2} \to B$. По предположению индукции

$$\Gamma \vdash A_1A_2 \to F_{i_1}$$
$$\Gamma \vdash A_1A_2 \to F_{i_2}$$
$$\Gamma \vdash A_1A_2 \to (F_{i_1}F_{i_2} \to B)$$

Объединим выводы этих формул и добавим к ним следующие три формулы:

$$(A_1A_2 \to (F_{i_1}F_{i_2} \to B)) \to ((A_1A_2 \to F_{i_1})(A_1A_2 \to F_{i_2}) \to (A_1A_2 \to B))$$

$$(A_1A_2 \to F_{i_1})(A_1A_2 \to F_{i_2}) \to (A_1A_2 \to B)$$

$$A_1A_2 \to B$$

Первая получена из аксиомы 3, две последние выводятся из первой с помощью обобщенной операции modus ponens. Образованная последовательность формул является выводом формулы $A_1A_2 \to B$ из Γ. $\square$

Лемма 12. *Пусть $A \in \mathbf{Fm}$ и $x_1, \ldots, x_n$ — различные переменные в A. Пусть $(x'_1, \ldots, x'_n) \in E_2^n$ и $A' = f_A(x'_1, \ldots, x'_n)$. Тогда*

$$x_1^*, \ldots, x_n^* \vdash A^*$$

Доказательство. Будем доказывать индукцией по длине формулы A.

Если A — переменная x_i, то утверждение леммы следует из определения выводимости $\vdash$.

Если A — это формула $\varphi(A_1, \ldots, A_m)$, то по индуктивному предположению утверждение леммы верно для формул $A_1, \ldots, A_m$, т.е.

$$x_1^*, \ldots, x_n^* \vdash A_i^*,\ i = 1, \ldots, m.$$

Кроме того, из аксиомы для логической связки φ имеем

$$A_1^*, \ldots, A_m^* \vdash A^*$$

Следовательно,

$$x_1^*, \ldots, x_n^* \vdash A^*$$

Пусть A — это формула $B_1B_2 \to C$ и утверждение леммы верно для формул $B_1,\ B_2$ и C:

$$x_1^*, \ldots, x_n^* \vdash B_i^* \qquad (*)$$

$$x_1^*, \ldots, x_n^* \vdash C^* \qquad (**)$$

Рассмотрим несколько подслучаев. Если $B_i' = 0$, то $A' = 1$. Тогда $B_i^* = B_1 \to y,\ A^* = (A \to y) \to y$ и по аксиоме 4 имеем

$$\vdash (B_i \to y) \to ((B_1B_2 \to C) \to y) \to y)$$

Если $C' = 1$, то $A' = 1$. Тогда $C^* = (C \to y) \to y$ и $A^* = (A \to y) \to y$. Имеют место следующие соотношения

$$\begin{gathered} C \vdash B_1B_2 \to C \\ C,\ (B_1B_2 \to C) \to y \vdash y \\ (B_1B_2 \to C) \to y \vdash C \to y \\ (B_1B_2 \to C) \to y,\ (C \to y) \to y \vdash y \\ (C \to y) \to y \vdash ((B_1B_2 \to C) \to y) \to y \end{gathered}$$

где первый вывод есть использование аксиомы 1, второй и четвертый — применение modus ponens, остальные получены по теореме дедукции.

Если $B_1', B_2' = 1,\ C' = 0$, то $A' = 0$. Тогда $B_i^* = (B_i \to y) \to y$, $C^* = C \to y$ и $A^* = A \to y$. Имеют место следующие соотношения

$$\begin{gathered} B_1,\ B_2,\ B_1B_2 \to C \vdash C \\ B_1,\ B_2,\ B_1B_2 \to C,\ C \to y \vdash y \\ B_1B_2 \to C,\ C \to y \vdash B_1B_2 \to y \\ B_1B_2 \to C,\ C \to y,\ (B_1B_2 \to y) \to y \vdash y \\ C \to y,\ (B_1B_2 \to y) \to y \vdash (B_1B_2 \to C) \to y \end{gathered}$$

где первый, второй и четвертый вывод являются применением modus ponens, остальные получены по теореме дедукции.

Тогда из (*) и (**) следует

$$x_1^*, \ldots, x_n^* \vdash A^*$$

Разбор всех случаев завершает доказательство леммы. $\square$

Поскольку каждая схема аксиом является тавтологией и обобщенное правило modus, будучи примененным к тавтологиям, выводит тавтологию, то оказывается справедливой лемма.

Лемма 13. *Для любой формулы $A \in \mathbf{Fm}$, если $\vdash A$, то A — тавтология.*

Докажем теперь обратную лемму.

Лемма 14. *Для любой формулы $A \in \mathbf{Fm}$, если A — тавтология, то $\vdash A$.*

Доказательство. Пусть $x_1, \ldots, x_n$ — различные переменные, встречающиеся в A. Так как A тавтология, то для любого набора $(x'_1, \ldots, x'_n) \in E_2$ значение $A' = f_A(x'_1, \ldots, x'_n) = 1$. По Лемме 12 для каждого из 2^n возможных множеств $\Gamma_n = \{x_1^*, \ldots, x_n^*\}$ имеем

$$\Gamma_n \vdash (A \to y) \to y,$$

Это означает, что для каждого из 2^{n-1} возможных множеств $\Gamma_{n-1} = \{x_1^*, \ldots, x_{n-1}^*\}$ верно

$$\Gamma_{n-1},\ x_n \to y \vdash (A \to y) \to y$$
$$\Gamma_{n-1},\ (x_n \to y) \to y \vdash (A \to y) \to y$$

Откуда по Теореме дедукции

$$\Gamma_{n-1} \vdash (x_n \to y) \to (A \to y) \to y$$
$$\Gamma_{n-1} \vdash ((x_n \to y) \to y) \to (A \to y) \to y$$

Но

$$\vdash \Big((x_n \to y) \to \big((A \to y) \to y\big)\Big) \to$$
$$\to \Bigg(\Big(\big((x_n \to y) \to y\big) \to \big((A \to y) \to y\big)\Big) \to \big((A \to y) \to y\big)\Bigg)$$

выводимо из аксиомы 4. С помощью двукратного применения правила modus ponens получим

$$\vdash (A \to y) \to y$$

Подставляя вместо y формулу A, имеем

$$\vdash (A \to A) \to A$$

Поскольку, $A \vdash A$, то по Теореме дедукции $\vdash A \to A$. Тогда, применяя правило modus ponens, окончательно получаем, что $\vdash A$. $\square$

Согласно Леммам 13 и 14 исчисление $\mathcal{P}_I(f)$ порождает множество тавтологий $\mathbf{Cl}$. Таким образом, верна следующая лемма.

Лемма 15. *Исчисление $\mathcal{P}_I(f)$ является конечным заданием логической системы $\langle \mathbf{Cl}, \mathbf{R} \rangle$ для любого $\mathbf{R} \subseteq \mathbf{R}_I$.*

6. Критерий конечного задания множества тавтологий

Теперь докажем критерий конечного задания множества тавтологий $\mathbf{Cl}$.

Теорема 1. *Исчисление $\mathcal{P}$ является конечным заданием множества тавтологий $\mathbf{Cl}$ тогда и только тогда, когда выполнено одно из условий:*

1. $\mathcal{F} \subseteq \mathbf{L}$ *и* $\mathcal{P}_L \leq \mathcal{P}$;
2. $\mathcal{F} \subseteq \mathbf{M}$ *и* $\mathcal{P}_M \leq \mathcal{P}$;
3. $[\mathcal{F}] \cap \mathbf{I} \neq \emptyset$ *и* $\mathcal{P}_I(f) \leq \mathcal{P}$ *для некоторой* $f \in [\mathcal{F}] \cap \mathbf{I}$;
4. $\mathbf{Cl} = \emptyset$.

Доказательство. Если $[\mathcal{P}] = \mathbf{Cl}$, то $\mathcal{P}$ является расширением любого непротиворечивого пропозиционального исчисления. Следовательно, необходимость условий выполнена.

Достаточность условий 1, 2 и 3 следует из Лемма 6, 10 и 15. Достаточность условия 4 следует из того, что пустое множество допускает задание пустым исчислением, т.е. исчислением с пустыми множествами аксиом и правил вывода. □

Список литературы

[1] *Боков Г. В.* Проблема полноты в исчислении высказываний // Интеллектуальные системы, т. 13, вып. 1–4, 2009, сс. 165–182.

[2] *Боков Г. В.* Итеративные пропозициональные исчисления // Интеллектуальные системы. Теория и приложения, т. 18, вып. 4, 2014, сс. 99–106.

[3] *Боков Г. В.* Об алгоритмической неразрешимости некоторых проблем распознавания для пропозициональных исчислений // Интеллектуальные системы. Теория и приложения, т. 18, вып. 4, 2014, сс. 207–214.

[4] *Боков Г. В.* О некоторых свойствах решетки пропозициональных исчислений // Интеллектуальные системы. Теория и приложения, т. 19, вып. 2, 2015, сс. 47–64.

[5] *Боков Г. В.* Разрешимость одно-переменных итеративных пропозициональных исчислений // Интеллектуальные системы. Теория и приложения, т. 19, вып. 2, 2015, с. 125–134.

[6] *Боков Г. В.* Неразрешимое суперинтуиционисткое пропозициональное исчисление от трех переменных // Интеллектуальные системы. Теория и приложения, т. 19, вып. 3, 2015, сс. 95–100.

[7] *Боков Г. В.* Об одной системе Фреге // Интеллектуальные системы. Теория и приложения, т. 19, вып. 4, 2015, сс. 155–168.

[8] *Боков Г. В.* Пропозициональные исчисления как средство задания логических процессов // Интеллектуальные системы. Теория и приложения, т. 20, вып. 3, 2016, сс. 24–36.

[9] *Боков Г. В.* От булевых схем к доказательству теорем // Интеллектуальные системы. Теория и приложения, т. 22, вып. 1, 2018, сс. 123–130.

[10] *Вишин В. В.* Тождественные преобразования в четырехзначной логике // ДАН СССР, т. 150, № 4, 1963, сс. 719–721.

[11] *Клини С. К.* Математическая логика // Москва, Мир, 1973.

[12] *Колмогоров А. Н., Драгалин А. Г.* Введение в математическую логику // Москва, Изд-во Моск. ун-та, 1982.

[13] *Линдон Р. К.* Тождества в двузначных исчислениях // Кибернетический сборник, т. 1, 1959, сс. 234–245.

[14] *Линдон Р. К.* Тождества в конечных алгебрах // Кибернетический сборник, т. 1, 1959, сс. 246–248.

[15] *Мурский В. Л.* Существование в трехзначной логике замкнутого класса с конечным базисом, не имеющего конечной полной системы тождеств // ДАН СССР, т. 163, № 4, 1965, сс. 815–818.

[16] *Яблонский С. В., Гаврилов Г. П., Кудрявцев В. Б.* Функции алгебры логики и классы Поста // Москва, Наука, 1966.

[17] *Bokov G. V.* Criterion for propositional calculi to be finitely generated // Discrete Mathematics and Applications, Volume 23, Issue 5-6, 2014, Pages 399–427.

[18] *Bokov G. V.* Undecidability of the problem of recognizing axiomatizations for propositional calculi with implication // Logic Journal of the IGPL, Volume 23, Issue 2, 2015, Pages 341–353.

[19] *Bokov G. V.* On the number of variables in undecidable superintuitionistic propositional calculi // Logic Journal of the IGPL, Volume 24, Issue 5, 2016, Pages 774–791.

[20] *Bokov G. V.* Undecidable Iterative Propositional Calculus // Algebra Logic, Volume 55, Issue 4, 2016, Pages 274–282.

[21] *Bokov G. V.* Undecidable problems for propositional calculi with implication // Logic Journal of the IGPL, Volume 24, Issue 5, 2016, Pages 792–806.

[22] *Henkin L.* Fragments of the propositional calculus // J. Symb. Logic, vol. 14, 1949, pp. 42–82.

[23] *Kalmar L.* Über die Axiomatisierbarkeit des Aussagenkalkül // Acta Scientiarum Mathematicarum, vol. 7, 1934, pp. 222–243.

On the finite representation of logical systems
Bokov G. V.

In this paper, we consider a problem of finite representation for logical systems. We research three types of logical systems: linear, monotone and implicational. For each type of logical systems we prove sufficient conditions of finite representation. Moreover, we prove a criterion for logical system of classical tautologies to be finitely generated.

Keywords: logical systems, propositional calculus, finite representation, inference rules.

Полиномиальная полнота конечных квазигрупп

Галатенко А.В., Панкратьев А.Е., Родин С.Б.

Приводится обзор результатов, связанных с проверкой полиномиальной полноты конечных квазигрупп. Работа подготовлена по материалам доклада на семинаре “Теория автоматов”.

Ключевые слова: квазигруппа, латинский квадрат, полиномиальная полнота, простота, аффинность

Памяти Михаила Михайловича Глухова

1. Введение

В последние годы наблюдается интерес к построению перспективных криптосистем, основанных на различных алгебраических структурах, в том числе неассоциативных. Одной из первых работ, в которых раскрываются возможности применения квазигрупп в криптографии, является статья М.М. Глухова [1]. Впоследствии был предложен ряд криптосистем на основе конечных квазигрупп, или, что то же самое, латинских квадратов (см., например, [2, 3, 4]).

Желательным с точки зрения стойкости свойством при этом является полиномиальная полнота, в некотором смысле гарантирующая вычислительную невозможность атаки методом решения системы уравнений на биты ключа [5]. Критерий полиномиальной полноты для квазигрупп порядка 4 получен в работе [6]; для квазигрупп простого порядка критерий и полиномиальный от порядка алгоритм проверки приведены в [7]; обобщение на случай произвольного порядка сделано в [8].

2. Основные определения

Конечной квазигруппой порядка $k \in \mathbb{N}$ называется множество $Q = \{q_1, \ldots, q_k\}$ с бинарной операцией $f : Q \times Q \to Q$, такой что для любых $a, b \in Q$ уравнения $f(x, a) = b$ и $f(a, y) = b$ однозначно разрешимы.

Таблица умножения квазигруппы, то есть матрица $k \times k$ с окаймляющей строкой и окаймляющим столбцом, является латинским квадратом — в силу разрешимости уравнений каждая строка и каждый столбец матрицы является записью некоторой перестановки на множестве Q.

Квазигруппы (Q, f_1) и (Q, f_2) называются изотопными, если найдутся перестановки α, β, γ на множестве Q, такие что выполнено тождество

$$f_1(x, y) = \gamma^{-1}\left(f_2(\alpha(x), \beta(y)\right).$$

В терминах латинских квадратов это означает, что один квадрат может быть преобразован в другой перестановкой строк и столбцов, а также переименованием элементов.

Заметим, что функция f может естественным образом рассматриваться как элемент k-значной логики P_k. Благодаря этому в нашем распоряжении оказывается операция суперпозиции, а также знания о предполных классах. Используемые в дальнейшем понятия и факты из k-значной логики описаны, например, в [9].

Квазигруппа называется полиномиально полной, если система из функции f и всех констант полна: $\left[\{f\} \cup P_k^0\right] = P_k$. Квазигруппа называется простой, если операция f не сохраняет никакое нетривиальное отношение эквивалентности. Несложно заметить, что квазигрупповая операция может сохранять только равномерное разбиение (то есть порождающее равномощные классы эквивалентности), поэтому все квазигруппы простого порядка являются простыми.

Квазигруппа аффинна, если на множестве Q можно задать структуру абелевой группы $(Q, +)$, относительно которой найдутся автоморфизмы α и β и константа $c \in Q$, такие что $f(x, y) = \alpha(x) + \beta(y) + c$. В работе [10] по сути показано, что простая квазигруппа аффинна тогда и только тогда, когда f является квазилинейной функцией.

Известно ([11, 12]), что полиномиальная полнота эквивалентна одновременной простоте и неаффинности (невложенности f ни в один из классов $\mathfrak{U}$, $\mathfrak{L}$ в обозначениях [9]). Несложно увидеть, что при $k = 2$ или 3 все квазигрупповые операции линейны, поэтому полиномиально полных квазигрупп нет. Примеры полиномиально полных квазигрупп порядка 4 приведены в работе [6].

3. Случай квазигрупп простого порядка

В случае квазигрупп простого порядка отсутствие полиномиальной полноты эквивалентно одновременной линеаризуемости всех одноместных

функций вида $f(x,a)$ и $f(a,y)$ при $a \in Q$. Более формально, верен следующий факт.

Теорема 1 ([7]). *Пусть p — простое число, $p \geq 5$, (Q,f) — квазигруппа порядка p. Тогда следующие условия эквивалентны:*

1) *Q не является полиномиально полной;*

2) *существует биективное отображение множества $\{q_1,\ldots,q_p\}$ на множество $\mathbb{Z}_p$, при котором квазигрупповая операция становится линейной функцией;*

3) *существует биективное отображение множества $\{q_1,\ldots,q_p\}$ на множество $\mathbb{Z}_p$, при котором все строки и столбцы матрицы, задающей квазигрупповую операцию, становятся линейными функциями, то есть линейными перестановками набора $(0,1,\ldots,p-1)$.*

Используя это утверждение, несложно построить полиномиальный от порядка квазигруппы алгоритм, проверяющий полиномиальную полноту. Достаточно заметить, что после умножения всех перестановок-строк латинского квадрата на перестановку, обратную первой строке, в случае неполноты все строки примут вид $x+d_i$, $d_i \in Q$, и для восстановления линеаризующего отображения по сути достаточно перебрать всевозможные варианты для константы, соответствующей второй строке. Для каждого варианта потребуется найти коэффициенты линейного представления (с константной сложностью) и проверить истинность найденного представления (со сложностью, квадратичной от порядка). Таким образом, верно следующее утверждение.

Теорема 2 ([7]). *Задача проверки полиномиальной полноты квазигрупп простого порядка решается за время, кубическое от порядка квазигруппы.*

Заметим, что уже в случае порядка 4 описанная идеология перестает работать, так как полиномиально полные квазигруппы порядка 4 существуют, но все перестановки квазилинейны.

Из теоремы 1 вытекает ряд несложных следствий.

Следствие 1 ([7]). *Почти все квазигруппы простого порядка полиномиально полны и не изотопны полиномиально неполным квазигруппам.*

Следствие 2 ([7]). *Для любой квазигруппы простого порядка, не являющейся полиномиально полной, существует изотопная ей полиномиально полная квазигруппа.*

Из теоремы 1 также следует, что в случае отсутствия полиномиальной полноты строки и столбцы латинского квадрата, задающего операцию f, при некоторой кодировке соответствуют линейным функциям вида $ax + b \mod p$, где параметр a одинаков для всех строк (столбцов). Как известно, линейная перестановка либо является тождественной, либо представляет из себя циклический сдвиг (случай $a = 1$, $b \neq 0$), либо состоит из одной неподвижной точки и s циклов длины t, $st = p-1$. Для удобства в первых двух случаях будем считать, что $s = 1$. Изменение кодировки действует на перестановки как сопряжение, то есть сохраняет цикловую структуру.

Следствие 3 ([7]). *Если в латинском квадрате простого порядка есть строка (столбец) с нелинейной цикловой структурой, или же две строки (два столбца) с линейной цикловой структурой, но различными значениями параметра s, то соответствующая квазигруппа является полиномиально полной.*

Заметим, что условия следствия 3 могут быть проверены со сложностью, квадратичной от порядка квазигруппы, поэтому разумно добавить такую проверку в начало процедуры распознавания полиномиальной полноты, имеющей, как было указано выше, кубическую сложность.

4. Случай порядка, не представимого в виде степени простого числа

Из результатов работы [10] следует, что если порядок квазигруппы не представим в виде p^t ни для какого простого p и натурального t, то полиномиальная полнота эквивалентна простоте. Проверка простоты квзигруппы может быть проведена, например, так. Для каждой пары (q_1, q_i), где q_1 — выделенный элемент Q, а q_i — произвольный элемент Q, отличный от q_1, строится транзитивное замыкание отношения $q_1 \sim q_i$ относительно операции f. Такое замыкание несложно вычислить со сложностью, кубической от порядка квазигруппы; при умножении на число пар, которые требуется рассмотреть, получается итоговая сложность алгоритма $O\left(k^4\right)$.

Теорема 3 ([8]). *Существует процедура проверки простоты квазигруппы порядка k, имеющая сложность $O\left(k^4\right)$.*

Следствие 4. *Пусть $k \in \mathbb{N}$ не представимо в виде p^t ни для какого простого p и натурального t. Тогда существует процедура проверки полиномиальной полноты квазигрупп, имеющая сложность $O\left(k^4\right)$.*

5. Общий случай

Для построения алгоритма определения полиномиальной полноты в общем случае остается рассмотреть процедуру проверки аффинности. Несложно заметить, что для аффинных квазигрупп домножение всех строк-перестановок соответствующего латинского квадрата L на перестановку, обратную первой строке, с последующей перестановкой строк позволяет получить таблицу Кэли L' для абелевой группы $(Q, +)$ из определения аффинности; после этого искомое представление $f(x, y) = \alpha(x) + \beta(y) + c$ может быть легко восстановлено. Заметим, что самым трудоемким этапом здесь является проверка ассоциативности операции, задаваемой матрицей L', то есть процедура кубическая по сложности.

Аккуратное описание алгоритма, кратко представленного выше, приведено в работе [8].

Теорема 4 ([8])**.** *Существует процедура проверки аффинности квазигруппы порядка k, имеющая сложность $O\left(k^3\right)$.*

Таким образом, верен следующий факт.

Теорема 5 ([8])**.** *Существует процедура проверки полиномиальной полноты квазигруппы порядка k, имеющая сложность $O\left(k^4\right)$.*

6. Обобщение на случай n-квазигрупп

Пусть $n, k \in \mathbb{N}$, $n > 2$; n-квазигруппой порядка k называется множество $Q = \{q_1, \ldots, q_k\}$ с n-арной операцией f, такой что для любых элементов $a_1, a_2, \ldots, a_n, b \in Q$ все уравнения

$$\begin{aligned} f(x, a_2, a_3, \ldots, a_n) &= b, \\ f(a_1, x, a_3, \ldots, a_n) &= b, \\ &\vdots \\ f(a_1, a_2, \ldots, a_{n-1}, x) &= b \end{aligned}$$

однозначно разрешимы в Q. Понятие полиномиальной полноты для n-квазигрупп вводится аналогично случаю квазигрупп. Критерии полиномиальной полноты и алгоритмы распознавания также естественным образом переносятся на n-квазигруппы. Отметим, что наиболее трудоемкие операции (вычисление транзитивного замыкания и проверка ассоциативности) не зависят от значения параметра n, так что сложностные характеристики не меняют порядок при переходе от квазигрупп к 3-квазигруппам. Итоговый результат примет следующий вид.

Теорема 6 ([8]). *Пусть $n \in \mathbb{N}$, $n > 2$ —фиксированный параметр. Тогда полиномиальная полнота n-квазигруппы порядка k может быть установлена со сложностью $O\left(k^{n+1}\right)$.*

Список литературы

[1] М.М. Глухов, "О применениях квазигрупп в криптографии", *ПДМ*, 2008, 2, 28–32.

[2] V. Shcherbacov, "Quasigroup based crypto-algorithms", arXiv:1201.3016v1.

[3] Y. Wu, Y. Zhou, J.P. Noonan, S. Agaian, C.L.P. Chen, "A Novel Latin Square Image Cipher", arXiv:1204.2310v1.

[4] A. Mileva, S. Markovski, "Quasigroup String Transformations and Hash Function Design", in: D. Davcev, J.M. Gómez (eds) *ICT Innovations*, Springer, 2009, 367–376.

[5] G. Horváth, C.L. Nehaniv, Cs. Szabó, "An assertion concerning functionally complete algebras and NP-completeness", *Theoret. Comput. Sci.*, **407** (2008), 591–595.

[6] V.A. Artamonov, S. Chakrabarti, S. Gangopadhyay, S.K. Pal, "On Latin squares of polynomially complete quasigroups and quasigroups generated by shifts", *Quasigroups and Related Systems*, **21**:2 (2013), 117–130.

[7] A.V. Galatenko, A.E. Pankratiev, S.B. Rodin, "Polynomially Complete Quasigroups of Prime Order", *Algebra and Logic*, **57**:5, (2018), 327–335.

[8] А.В. Галатенко, А.Е. Панкратьев, "О сложности проверки полиномиальной полноты конечных квазигрупп", *Дискрет. матем.*, **30**:4 (2018), 3–11.

[9] D. Lau, *Function algebras on finite sets: a basic course on many-valued logic and clone theory*, Springer, 2006.

[10] V.A. Artamonov, S. Chakrabarti, S.K. Pal, "Characterizations of highly non-associative quasigroups and associative triples", *Quasigroups and Related Systems*, **25** (2017), 1–19.

[11] J. Hagemann, C. Herrmann, "Arithmetical locally equational classes and representation of partial functions", *Universal Algebra, Esztergom (Hungary)*, **29** (1982), 345–360.

[12] В.Л. Югай, "Об одном критерии полиномиальной полноты квазигрупп", *Интеллектуальные системы. Теория и приложения*, **21**:3 (2017), 131–135.

Polynomial completeness of finite quasigroups

Galatenko A.V., Pankratiev A.E., Rodin S.B.

We give a survey of results connected to deciding polynomial completeness of finite quasigroups. The paper is based on a report presented at the seminar "Automata theory".

Keywords: quasigroup, Latin square, polynomial completeness, simplicity, affinity

О прогрессивном представлении периодических семейств с ограничениями на начало и шаг

Дергач П.С., Данилевская Е.Д.

В статье изучается множество $K(n) := \mathbb{N} \setminus (n, n)$, исследуется его представление в виде объединения как можно меньшего количества арифметических прогрессий с ограничением на начало или шаг. В каждом из двух случаев найдены соответствующие точные оценки.

Ключевые слова: арифметическая прогрессия, натуральный ряд, проблема минимизации, типы ограничений.

Введение

Статья написана в соавторстве Дергача П. С. с его ученицей Данилевской Е. Д. и является переработанным результатом ее дипломной выпускной работы в филиале МГУ имени М. В. Ломоносова в городе Ташкенте. Рассматривается периодическое семейство натуральных чисел $K(n) := \mathbb{N} \setminus (n, n)$, взятое из работы [1]. Необходимо представить это семейство минимальным по количеству объединением арифметических прогрессий, на которые наложены следующие типы ограничений. Либо это ограничение сверху на начало прогрессий, либо это ограничение сверху на шаг прогрессий. В обоих случаях приводится точная реализация, доставляющая соответствующий минимум. Доказывается неулучшаемость полученных оценок. Читатели, желающие познакомиться с аналогичными интересными результатами, отсылаются к статьям [3-20].

Основные определения и результаты

Множество натуральных чисел обозначаем через $\mathbb{N}$, а множество целых неотрицательных чисел — через $\mathbb{N}_0$. Множество натуральных чисел от 1 до s обозначаем через E_s. Для $a \in \mathbb{N}$, $b \in \mathbb{N}$ обозначаем

$$(a, b) := \{a + ib \mid i \in \mathbb{N}_0\}$$

и называем это множество *арифметической прогрессией с началом a и шагом b*. Для произвольного $n \in \mathbb{N}$ положим

$$K(n) := \mathbb{N} \setminus (n, n).$$

Множество арифметических прогрессий обозначаем через $\mathbb{P}$. Для произвольного $k \in \mathbb{N}$ рассматриваем множества

$$\mathbb{B}(k) := \{(a, b) \in \mathbb{P} \mid a \leq k\},$$

$$\mathbb{S}(k) := \{(a, b) \in \mathbb{P} \mid b \leq k\}.$$

Для $* \in \{\mathbb{B}, \mathbb{S}\}$ пишем, что

$$K(n) \in U_k(*),$$

если множество $K(n)$ можно представить конечным объединением элементов из $*(k)$. Наконец, для произвольного $n \in \mathbb{N}$ вводим обозначения

$$f_1(n) = \min_k\{K(n) \in U_k(\mathbb{B})\},$$

$$f_2(n) = \min_k\{K(n) \in U_k(\mathbb{S})\}.$$

Теорема 1. *Пусть $n \in \mathbb{N}$ и $n = p_1^{a_1} p_2^{a_2} \dots p_s^{a_s}$ — его разложение на простые множители. Тогда*

$$f_1(n) = \max_{i \in E_s} p_i^{a_i - 1}(p_i - 1).$$

Теорема 2. *Пусть $n \in \mathbb{N}$ и $n = p_1^{a_1} p_2^{a_2} \dots p_s^{a_s}$ — его разложение на простые множители. Тогда*

$$f_2(n) = \max_{i \in E_s} p_i^{a_i}.$$

Доказательство вспомогательных утверждений

Лемма 1. Критерий пересечения. *Для любых* $a, c \in \mathbb{N}_0$ *и* $b, d \in \mathbb{N}$ *верно*

$$(a, b) \cap (c, d) \neq \emptyset \Longleftrightarrow a \equiv c \ (mod \ НОД(b, d)).$$

Доказательство леммы см. в [2].

Доказательство основных утверждений

Теорема 1. *Пусть* $n \in \mathbb{N}$ *и* $n = p_1^{a_1} p_2^{a_2} \dots p_s^{a_s}$ *— его разложение на простые множители. Тогда*

$$f_1(n) = \max_{i \in E_s} p_i^{a_i - 1}(p_i - 1).$$

Доказательство.

Обозначим через h_n число $\max\limits_{i \in E_s} p_i^{a_i - 1}(p_i - 1)$ и докажем верхнюю оценку, то есть что

$$f_1(n) \leq h_n. \tag{1}$$

Для этого достаточно показать, что множество $K(n)$ можно представить конечным объединением элементов из $\mathbb{B}(h_n)$, то есть арифметических прогрессий с началом не больше h_n. Это можно сделать, например, следующим образом. Для всех

$$0 \leq i \leq a_1 - 1, \tag{2}$$

$$p_1^i \leq j \leq p_1^i(p_1 - 1) \tag{3}$$

рассматриваем прогрессии

$$(a_i^j, b_i^j) := (j, p_1^{i+1}). \tag{4}$$

Ясно, что для каждого фиксированного i из (2) объединение прогрессий (a_i^j, b_i^j) по всем j из (3) дает нам множество всех натуральных чисел, которые делятся на p_1^i, но не делятся на p_1^{i+1}. Если теперь объединить эти конструкции по всем i из (2), то получим множество всех натуральных чисел, которые не делятся на $p_1^{a_1}$, то есть множество $K(p_1^{a_1})$. При этом, начала всех использованных в (4) прогрессий не превосходят $p_1^{a_1 - 1}(p_1 - 1)$, а значит не превосходят и h_n. Поэтому множество $K(p_1^{a_1})$

лежит в $U_{h_n}(\mathbb{B})$. Аналогично можно показать, что при всех $i \in E_s$ множество $K(p_i^{a_i})$ лежит в $U_{h_n}(\mathbb{B})$. Для этого достаточно в приведенном выше рассуждении заменить p_1 на p_i. Значит и множество $\bigcup\limits_{i \in E_s} K(p_i^{a_i})$ лежит в $U_{h_n}(\mathbb{B})$. Осталось заметить, что

$$\bigcup_{i \in E_s} K(p_i^{a_i}) = K(p_1^{a_1} p_2^{a_2} \dots p_s^{a_s}) = K(n).$$

Неравенство (1) доказано.

Докажем теперь нижнюю оценку, то есть что

$$f_1(n) \geq h(n). \tag{5}$$

Для этого достаточно показать, что множество $K(n)$ нельзя представить конечным объединением элементов из множества $\mathbb{B}(h_n - 1)$, то есть арифметических прогрессий с началом не больше $h_n - 1$. Это можно сделать, например, следующим образом. Будем доказывать утверждение от противного. Допустим, такое представление существует. Без ограничения общности, будем считать, что

$$h_n = p_1^{a_1 - 1}(p_1 - 1). \tag{6}$$

Обозначим через a число, которое по модулю $p_1^{a_1}$ дает остаток $h(n)$, а по модулям $p_2^{a_2}, \dots, p_s^{a_s}$ дает остаток 0, то есть делится нацело. Из китайской теоремы об остатках известно, что найдется (и при том ровно одно) такое число на промежутке E_n. Очевидно, что $a \neq n$. Значит $a \in K(n)$ и поэтому содержится хотя бы в одной из прогрессий представления. Выберем любую из них. Пусть эта прогрессия имеет начало x. Рассмотрим тогда ее подпрогрессию $(x, a - x)$. Эта подпрогрессия все еще лежит в $K(n)$ и, значит, не пересекается с прогрессией (n, n). Поэтому из леммы 1 заключаем, что

$$x \not\equiv n \ (mod \ \text{НОД}(a - x, n)).$$

Значит найдутся такие $i \in E_s$ и $j \in E_{a_i}$, для которых

$$x \not\equiv n \ (mod \ p_i^j), \ x \equiv a \ (mod \ p_i^j).$$

Однако, оба числа a и n делятся нацело на

$$p_1^{a_1 - 1} p_2^{a_2} \dots p_s^{a_s} = \frac{n}{p_1}.$$

Значит,

$$x \not\equiv n \ (mod\ p_1^{a_1}),\ x \equiv a \ (mod\ p_1^{a_1}). \tag{7}$$

Так как число a по модулю $p_1^{a_1}$ дает остаток $h(n)$, то из (7) следует, что

$$x \equiv h(n) \ (mod\ p_1^{a_1}). \tag{8}$$

Однако, каждое начало прогрессий нашего представления не превосходит $h_n - 1$ и значит

$$x < h_n. \tag{9}$$

Условия (8) и (9) противоречат друг другу. Неравенство (5) доказано. Вместе с неравенством (1) это завершает доказательство теоремы.

■

Теорема 2. *Пусть $n \in \mathbb{N}$ и $n = p_1^{a_1} p_2^{a_2} \dots p_s^{a_s}$ — его разложение на простые множители. Тогда*

$$f_2(n) = \max_{i \in E_s} p_i^{a_i}.$$

Доказательство.

Обозначим через r_n число $\max\limits_{i \in E_s} p_i^{a_i}$ и докажем верхнюю оценку, то есть что

$$f_2(n) \le r_n. \tag{10}$$

Для этого достаточно показать, что множество $K(n)$ можно представить конечным объединением элементов из множества $\mathbb{S}(r_n)$, то есть арифметических прогрессий с шагом не больше r_n. Это можно сделать, например, следующим образом. Для всех

$$1 \le i \le p_1^{a_1} - 1 \tag{11}$$

рассматриваем прогрессии

$$(a_i, b_i) := (i, p_1^{a_1}). \tag{12}$$

Ясно, что объединение прогрессий (a_i, b_i) по всем i из (11) дает нам множество всех натуральных чисел, которые не делятся на $p_1^{a_1}$. Шаги всех использованных в (12) прогрессий равны $p_1^{a_1}$, а значит не превосходят r_n. Поэтому множество $K(p_1^{a_1})$ лежит в $U_{r_n}(\mathbb{S})$. Аналогично можно показать, что при всех $i \in E_s$ множество $K(p_i^{a_i})$ лежит в $U_{r_n}(\mathbb{S})$. Для этого

достаточно в приведенном выше рассуждении заменить p_1 на p_i. Значит и множество $\bigcup\limits_{i\in E_s} K(p_i^{a_i})$ лежит в $U_{r_n}(\mathbb{S})$. Осталось заметить, что

$$\bigcup_{i\in E_s} K(p_i^{a_i}) = K(p_1^{a_1}p_2^{a_2}\dots p_s^{a_s}) = K(n).$$

Докажем теперь нижнюю оценку, то есть что

$$f_1(n) \geq h(n). \tag{13}$$

Для этого достаточно показать, что множество $K(n)$ нельзя представить конечным объединением элементов из множества $\mathbb{S}(r_n-1)$, то есть арифметических прогрессий с шагом не больше $r_n - 1$. Это можно сделать, например, следующим образом. Будем доказывать утверждение от противного. Допустим, такое представление существует. Без ограничения общности, будем считать, что

$$r_n = p_1^{a_1}. \tag{14}$$

Тогда рассмотрим ту прогрессию представления, которая содержит число $\frac{n}{p_1} = p_1^{a_1-1}p_2^{a_2}\dots p_s^{a_s}$. Обозначим шаг этой прогрессии через t. Тогда прогрессия

$$(p_1^{a_1-1}p_2^{a_2}\dots p_s^{a_s}, t) \tag{15}$$

будет в ней лежать, поэтому лежит и в множестве $K(n)$. Значит прогрессия (15) не пересекается с прогрессией (n, n). По лемме 1 получаем отсюда, что

$$p_1^{a_1-1}p_2^{a_2}\dots p_s^{a_s} \not\equiv n \ (mod \ \text{НОД}(n, t)).$$

Отсюда с неизбежностью следует, что t делится нацело на $p_1^{a_1} = r_n$. Но это противоречит тому, что все элементы представления лежат в множестве $\mathbb{S}(r_n - 1)$. Неравенство (13) доказано. Вместе с неравенством (10) это завершает доказательство теоремы.

■

Список литературы

[1] П. С. Дергач, Э. С. Айрапетов. *О прогрессивном разбиении некоторых подмножеств натурального ряда.* Интеллектуальные системы, 2015. Т.19, вып. 3, М., Сс. 79-86.

[2] П. С. Дергач. *О каноническом регулярном представлении S-тонких языков.* Интеллектуальные системы, 2014. Т.18, вып. 1, М., Сс. 211-242.

[3] П. С. Дергач. *О проблеме вложения допустимых классов.* Интеллектуальные системы, 2015. Т.19, вып. 2, М., Сс. 143-174.

[4] П. С. Дергач. *О двух размерностях спектров тонких языков.* Интеллектуальные системы, 2015. Т.19, вып. 3, М., Сс. 155-174.

[5] П. С. Дергач, Э. С. Айрапетов. *О прогрессивном разбиении последовательности натуральных чисел, имеющей пропуск длины 2.* Интеллектуальные системы, 2016. Т.20, вып. 2, М., Сс. 67-86.

[6] П. С. Дергач, Е. Д. Данилевская. *О покрытиях и разбиениях натуральных чисел, имеющих два последовательных пропуска длины 1.* Интеллектуальные системы, 2017. Т.21, вып. 1, М., Сс.192-237.

[7] П. С. Дергач. *О структуре вложения прогрессивных множеств сложности два.* Интеллектуальные системы, 2017. Т.21, вып. 2, М., Сс.117-162.

[8] П. С. Дергач, Ж. И. Раджабов. *О длине минимальной алфавитной склейки для класса линейных регулярных языков.* Интеллектуальные системы, 2017. Т.21, вып. 3, М., Сс.120-130.

[9] Д. Е. Александров. *Эффективные методы реализации проверки содержания сетевых пакетов регулярными выражениями.* Интеллектуальные системы, 2014. Т.18, вып. 1, М., Сс. 37-60.

[10] Д. Н. Бабин. *Частотные регулярные языки.* Интеллектуальные системы, 2014. Т.18, вып. 1, М., Сс. 205-210.

[11] Д. Е. Александров. *Об оценках автоматной сложности распознавания классов регулярных языков.* Интеллектуальные системы, 2014. Т.18, вып. 4, М., Сс. 161-190.

[12] В. М. Дементьев. *О звездной высоте регулярного языка и циклической сложности минимального автомата.* Интеллектуальные системы, 2014. Т.18, вып. 4, М., Сс. 215-222.

[13] И. Е. Иванов. *О сохранении периодических последовательностей автоматами с магазинной памятью с однобуквенным магазином.* Интеллектуальные системы, 2015. Т.19, вып. 1, М., Сс. 145-160.

[14] А. А. Петюшко. *О контекстно-свободных биграмных языках.* Интеллектуальные системы, 2015. Т.19, вып. 2, М., Сс. 187-208.

[15] И. Е. Иванов. *Нижняя оценка на максимальную длину периода выходной последовательности автономного автомата с магазинной памятью.* Интеллектуальные системы, 2015. Т.19, вып. 3, М., Сс. 175-194.

[16] В. А. Орлов. *О конечных автоматах с максимальной степенью различимости состояний.* Интеллектуальные системы, 2016. Т.20, вып. 1, М., Сс. 213-222.

[17] П. С. Дергач. *О проблеме проверки однозначности алфавитного декодирования в классе регулярных языков с полиномиальной функцией роста.* Интеллектуальные системы, 2016. Т.20, вып. 2, М., Сс. 147-202.

[18] А. М. Миронов. *Основные понятия теории вероятностных автоматов.* Интеллектуальные системы, 2016. Т.20, вып. 2, М., Сс. 283-330.

[19] А. А. Петюшко, Д. Н. Бабин. *Классификация Хомского для матриц биграммных языков.* Интеллектуальные системы, 2016. Т.20, вып. 2, М., Сс. 331-336.

[20] С. Б. Родин. *О связи линейно реализуемых автоматов и автоматов с максимальной вариативностью относительно кодирования состояний.* Интеллектуальные системы, 2016. Т.20, вып. 2, М., Сс. 337-348.

Сведения об авторах

Дергач Петр Сергеевич
Младший научный сотрудник МГУ имени М. В. Ломоносова
e-mail: dergachpes@mail.ru,
Данилевская Екатерина Дмитриевна
Выпускница факультета ПМиИ филиала МГУ имени М. В. Ломоносова в городе Ташкенте
e-mail: katirina_@mail.ru.

On the progressive representation of periodic sets with restrictions on the beginning and the step
Dergach P.S., Danilevskaya E.D.

In the article the set $K(n) := \mathbb{N} \setminus (n, n)$ is being examined, it's presentation as union of as few arithmetic progressions as possible with constraints to the begining or step is being investigated. In both two cases appropriate accurate estimations have been found.

Key words: arithmetic progression, natural series, minimization problem, types of constraints.

Равномерная V-реализуемость принципа Маркова в V-перечислимой области

Коновалов А. Ю.

Определяются различные варианты понятия V-реализуемости для формул языка логики предикатов, основанные на использовании функций из множества V для интерпретации импликации и квантора всеобщности. Устанавливается, что принцип Маркова является слабо V-реализуемым, не является равномерно V-реализуемым и является V-реализуемым равномерно в области M, если множество $M \subseteq \mathbb{N}$ является V-перечислимым.

Ключевые слова: конструктивная семантика, реализуемость, абсолютная реализуемость, обобщенная реализуемость, принцип Маркова.

Принцип конструктивного подбора, или *принцип Маркова*, выражаемый предикатной формулой

$$\forall x\,(P(x) \vee \neg P(x)) \to (\neg\neg\exists x\,P(x) \to \exists x\,P(x)), \qquad \text{(MP)}$$

является одним из основных законов конструктивной логики, отличающим ее от интуиционистской логики. В контексте рекурсивной реализуемости принцип Маркова означает, что формула (MP) является реализуемой, если областью возможных значений переменной x является множество всех натуральных чисел $\mathbb{N}$. Возможность варьирования предметной области полностью исследована в работе [1]. Там доказано, что формула (MP) реализуема в некоторой области M тогда и только тогда, когда множество M рекурсивно-перечислимо.

В работах [2, 3] определены обобщения понятия рекурсивной реализуемости, в которых вместо индексов частично-рекурсивных функций в качестве реализаций используются индексы частичных функций из более широких классов. Представляет интерес исследовать вопрос о корректности принципа Маркова относительно различных вариантов обобщенной реализуемости.

Пусть V — некоторое счетное множество частичных функций натурального аргумента. Элементы множества V назовем V-функциями. Будем считать, что для каждого натурального числа n имеется нумерация всех n-местных V-функций. А именно, определено множество индексов $\mathsf{I}_n^V \subseteq \mathbb{N}$ вместе с отображением, которое каждому натуральному числу $z \in \mathsf{I}_n^V$ ставит в соответствие n-местную V-функцию $\varphi_z^{V,n}$, и при этом всякая n-местная V-функция есть $\varphi_z^{V,n}$ для некоторого $z \in \mathsf{I}_n^V$. Будем говорить, что натуральное число z есть V-*индекс* n-местной V-функции φ, если $z \in \mathsf{I}_n^V$ и $\varphi = \varphi_z^{V,n}$. При записи выражений вида $\varphi_z^{V,n}(t_1, \ldots, t_n)$ обычно будем опускать верхний индекс n. Будем считать, что множество V вместе с вышеописанной нумерацией обладает следующими свойствами:

C1) множество V содержит все частично-рекурсивные функции;

C2) если ψ есть n-местная V-функция, s — перестановка на множестве $\{1, \ldots, n\}$, то функция ψ', определенная условным равенством

$$\psi'(x_1, \ldots, x_n) \simeq \psi(x_{s(1)}, \ldots, x_{s(n)}),$$

является V-функцией;

C3) если ψ есть n-местная V-функция, то функция ψ', определенная условным равенством

$$\psi'(x_1, \ldots, x_n, x_{n+1}) \simeq \psi(x_1, \ldots, x_n),$$

является V-функцией;

C4) если ψ есть $(n+m)$-местная V-функция, $a_1, \ldots, a_m$ — натуральные числа, то функция ψ', определенная условным равенством

$$\psi'(x_1, \ldots, x_n) \simeq \psi(x_1, \ldots, x_n, a_1, \ldots, a_m),$$

является V-функцией;

C5) если ψ есть $(n+1)$-местная V-функция, то функция ψ', определенная условным равенством

$$\psi'(x_1, \ldots, x_n) \simeq \mu x\,[\psi(x_1, \ldots, x_n, x) = 0],$$

является V-функцией (μ — оператор минимизации);

C6) если ψ есть m-местная V-функция, $\psi_1, \ldots, \psi_m$ суть n-местные V-функции, то функция ψ', определенная условным равенством

$$\psi'(x_1, \ldots, x_n) \simeq \psi(\psi_1(x_1, \ldots, x_n), \ldots, \psi_m(x_1, \ldots, x_n)),$$

является V-функцией.

Кроме этого, потребуем эффективного выполнения свойств С2–С6. А именно, будем считать, что существует такая V-функция (своя для каждого из свойств С2–С6), которая по V-индексу функции ψ (V-индексам функций $\psi, \psi_1, \ldots, \psi_m$) находит некоторый V-индекс функции ψ'.

Отметим, что свойствам С1–С6 удовлетворяет множество всех частично-рекурсивных функций с подходящей нумерацией. В работах [2, 3] приводятся такие нумерации всех арифметических и всех гиперарифметических функций, что свойства С1–С6 выполняются для соответствующих классов функций.

Предикатные формулы строятся обычным образом из атомов $P(v_1, \ldots, v_n)$, где P есть n-местная предикатная переменная, а $v_1, \ldots, v_n$ — предметные переменные, при помощи логических констант $\top$, $\bot$, связок $\wedge$, $\vee$, $\to$ и кванторов $\forall$, $\exists$.

Пусть M — непустое подмножество натурального ряда. Следуя [4], n-местным обобщенным предикатом на множестве M будем называть всякую функцию типа $M^n \to 2^{\mathbb{N}}$. Пусть A — предикатная формула, f — отображение, которое каждой предикатной переменной из A валентности n ставит в соответствие n-местный обобщенный предикат на множестве M. В этом случае отображение f будем называть M-оценкой формулы A. Временно введем в язык логики предикатов константы для обозначения элементов множества M. Формулы с этими константами будем называть предикатными формулами расширенного языка.

Пусть фиксированы примитивно-рекурсивные двухместная функция c, которая взаимно однозначно нумерует все пары натуральных чисел, и одноместные обратные функции p_1 и p_2, так что выполняются соотношения $\mathsf{p}_1(\mathsf{c}(x, y)) = x$ и $\mathsf{p}_2(\mathsf{c}(x, y)) = y$. В выражениях вида $\mathsf{p}_1(t)$, $\mathsf{p}_2(t)$ обычно будем опускать скобки.

Для натурального числа e, замкнутой предикатной формулы A расширенного языка и M-оценки f формулы A определим отношение $e\ \mathbf{r}_f^V\ A$ (число e V-реализует формулу A при оценке f):

1) неверно $e\ \mathbf{r}_f^V\ \bot$;

2) верно $e\ \mathbf{r}_f^V\ \top$;

3) $e\ \mathbf{r}_f^V\ P(a_1, \ldots, a_n) \rightleftharpoons e \in f(P)(a_1, \ldots, a_n)$, если P есть n-местная предикатная переменная;

4) $e\ \mathbf{r}_f^V\ (\Phi \wedge \Psi) \rightleftharpoons \mathsf{p}_1 e\ \mathbf{r}_f^V\ \Phi$ и $\mathsf{p}_2 e\ \mathbf{r}_f^V\ \Psi$;

5) $e\ \mathbf{r}_f^V\ (\Phi \vee \Psi) \rightleftharpoons (\mathsf{p}_1 e = 0$ и $\mathsf{p}_2 e\ \mathbf{r}_f^V\ \Phi)$ или $(\mathsf{p}_1 e = 1$ и $\mathsf{p}_2 e\ \mathbf{r}_f^V\ \Psi)$;

6) $e\ \mathbf{r}_f^V\ (\Phi \to \Psi) \rightleftharpoons e \in \mathsf{I}_1^V$ и для всех натуральных чисел s, если имеет место $s\ \mathbf{r}_f^V\ \Phi$, то определено $\varphi_e^V(s)$ и верно $\varphi_e^V(s)\ \mathbf{r}_f^V\ \Psi$.

7) $e\ \mathbf{r}_f^V\ \exists x\, \Phi(x) \rightleftharpoons \mathsf{p}_1 e \in M$ и $\mathsf{p}_2 e\ \mathbf{r}_f^V\ \Phi(\mathsf{p}_1 e)$;

8) $e\ \mathbf{r}_f^V\ \forall x\, \Phi(x) \rightleftharpoons e \in \mathsf{I}_1^V$ и для всех $a \in M$ определено $\varphi_e^V(a)$ и имеет место $\varphi_e^V(a)\ \mathbf{r}_f^V\ \Phi(a)$.

Есть несколько способов определить V-реализуемость замкнутых предикатных формул на основании отношения $e\ \mathbf{r}_f^V\ A$:

- замкнутую предикатную формулу A назовем *слабо V-реализуемой* (обозначение: $\mathbf{r}^V\ A$), если для любого непустого множества $M \subseteq \mathbb{N}$ и произвольной M-оценки f найдется такое натуральное число e, что имеет место $e\ \mathbf{r}_f^V\ A$;

- замкнутую предикатную формулу A назовем *V-реализуемой равномерно в области M* (обозначение: $\mathbf{ur}_M^V\ A$), если найдется такое натуральное число e, что для любой M-оценки f имеет место $e\ \mathbf{r}_f^V\ A$;

- замкнутую предикатную формулу A назовем *равномерно V-реализуемой* (обозначение: $\mathbf{ur}^V\ A$), если найдется такое натуральное число e, что для любого непустого множества $M \subseteq \mathbb{N}$ и произвольной M-оценки f имеет место $e\ \mathbf{r}_f^V\ A$.

Верны следующие теоремы.

Теорема 1. *Формула* (MP) *является слабо V-реализуемой.*

Теорема 2. *Формула* (MP) *не является равномерно V-реализуемой.*

Теорема 3. *Пусть множество M непусто и V-перечислимо. Тогда формула* (MP) *является V-реализуемой равномерно в области M.*

Список литературы

[1] Заславский И. Д., Цейтин Г. С. К вопросу об обобщениях принципа конструктивного подбора // Тр. МИАН СССР. 1964. **72**. 344—347.

[2] Коновалов А. Ю., Плиско В. Е. О гиперарифметической реализуемости // Мат. зам. 2015. **98**, №5. 725—746.

[3] Коновалов А. Ю. Арифметическая реализуемость и базисная логика // Вестн. Моск. ун-та. Матем. Механ. 2016. №1. 52–56.

[4] Плиско В. Е. Абслолютная реализуемость предикатных формул // Изв. АН СССР. Сер. матем. 1983. **47**. №2. стр. 315–334.

Markov's Principle is uniformly V-realizable in any V-enumerable domain.

Konovalov A. Yu.

Various variants of the notion of the V-realizability for predicate formulas are defined, where indexes of functions in the set V are used for interpreting the implication and the universal quantifier. It is proved that Markov's Principle is weakly V-realizable, not uniformly V-realizable, and uniformly V-realizable in any V-enumerable domain $M \subseteq \mathbb{N}$.

Keywords: constructive semantics, realizability, absolute realizability, generalized realizability, Markov's Principle.

Часть 3.
Математические модели

О минимальной Шефферовой функции в классе кусочно-параллельных функций, определенных над двоично-рациональными числами

Агафонова М.В.

В настоящей статье рассматривается класс нейронных кусочно-параллельных функций с двоично-рациональными коэффициентами(ВРР). Приводится доказательство существования в нем минимальной Шефферовой функции. Под минимальной Шефферовой функцией в рассматриваемом классе, понимается функция этого класса, порождающая этот класс по операциям суперпозиции и содержащая минимально возможное количество переменных и пороговых функций. Было установлено, что в данном классе минимальная Шефферова функция содержит две переменных и одну пороговую функцию. Также в статье приводится одно из необходимых условий Шефферовости функции, принадлежащей ВРР.

Ключевые слова: класс, кусочно-параллельные функции, нейронные функции, двоично-рациональные коэффициенты, операции суперпозици, Шефферова функция.

1. Введение.

Исследование класса нейронных кусочно-параллельных функций берет свое начало из работы Половникова В.С [1]. В указанной работе, вводится понятие нейронной схемы над элементами, реализующими линейные функции и нелинейные функции активации. Над нейронными схемами выполняются операции суперпозиции: добавление фиктивного входа, изъятие фиктивного входа, склеивание входов, переименование входов без склеивания, последовательное соединение. Таким образом, рассматривается функциональная система [2] нейронных схем. Автором работы [1], были изучены и описаны некоторые свойства нейронных схем, методы их построения,а также произведены доказательства эквивалентности

между множеством функций реализуемых нейронными схемами без памяти и множеством кусочно-линейных функций (PL), а также между множеством функций, реализуемых нейронными схемами модели Мак-Каллока-Питтса[3], и множеством кусочно-параллельных функций (PP). Изучение класса PL было продолжено в работах А. Кана. [4][5]. Рассмотрение же класса PP, было развито в работе [6], а в настоящей работе рассматривается его подкласс, состоящий из кусочно-параллельных функций с двоично-рациональными коэффициентами.

Класс кусочно-параллельных функций порождается множеством состоящим, из всех вещественных констант, сумматора, умножителя на вещественную константу и функции Хэвисайда. В работе [6] вещественные коэффициенты в PP заменяются на приближающие их двоично-рациональные. Учитывая, что любое двоично-рациональное число может быть получено из константы $\frac{1}{2}$ и формул $\frac{1}{2}x, -x, x+y$, при использовании операций суперпозиции, рассматриваемый в этой работе класс, имеет конечный базис, в отличие от PP. Учитывая, что константа $\frac{1}{2}$ получается из суперпозиции: $\frac{1}{2}(\theta(\theta(x)) = \frac{1}{2}$, где $\theta(x)$ - функция Хэвисайда:

$$\theta(x) \quad = \quad \begin{cases} 1 & \text{при } x \geq 0 \\ 0 & \text{при } x < 0 \end{cases},$$

имеем базис

$$B = \{\frac{1}{2}x, -x, x + y, \theta(x)\}.$$

Замыкание функций $B = \{\frac{1}{2}x, -x, x + y, \theta(x)\}$, по операциям суперпозиции (добавление фиктивной переменной, удаление фиктивной переменной, отожествление переменных, переименование переменных без отожествления, последовательная подстановка функции вместо переменной) образует множеством кусочно-параллельных функций с двоично-рациональными коэффициентами и обозначается BPP,

$$[B] = BPP.$$

Конечность базиса в классе BPP существенно отличает его от класса PP.

Ранее было показано, что любая функция из класса PP аппроксимируется с любой наперед заданной точностью элементом из множества BPP, зависящим от заданной точности. [6].

Следует отметить, что в работе [6] приводится доказательство того, что для BPP существует Шефферова [7] функция, которая имеет вид:

$$F(x,y,z) = x - \frac{1}{2}(y) - \frac{1}{2}\theta'(z) + \frac{1}{2},$$

где

$$\theta'(z) \quad = \quad \begin{cases} 1 & \text{при } z > 0 \\ 0 & \text{при } z \leq 0 \end{cases},$$

а также, доказательство существования в классе BPP для произвольного натурального числа k базиса, состоящего из k элементов.

В данной же работе, найдена Шефферова функция минимальная в классе BPP а также некоторые условия Шефферовости функции в этом классе. Под минимальной Шефферовой функцией понимается функция этого класса, порождающая этот класс и имеющая минимальное количество переменных и пороговых элементов.

2. Полученные результаты

Сначала рассмотрим, полученное необходимое условие Шефферовости функции, принадлежащей классу BPP, с тем условием, что данная функция будет минимальной Шефферовой в нем. Не трудно увидеть, что Шефферова функция содержит хотя бы один нелинейный элемент и зависит не менее чем от двух переменных. Поэтому Шефферовы функции будем искать в следующем виде:

$$g(x,y) = ax + by + c + r\theta(dx + ey + f).$$

Очевидно количество переменных не может быть уменьшено, так как из суперпозиций функций, зависящих от одной переменной, получаются функции, также зависящие от одной переменной. А удаление пороговой функции приведет к тому, что мы будем получать только непрерывные линейные функции. *Определение* Пусть для функции $g(x,y) \in BBP$,

$$g(x,y) = ax + by + c + r\theta(dx + ey + f),$$

выполнено:

$$a + b - 1 \neq 0$$

Тогда, если функция g является Шеферовой, тогда выполнены следующие неравенства:

$$\begin{cases} d(-\frac{c}{a+b-1}) + e(-\frac{c}{a+b-1}) + f \geq 0, \\ d(-\frac{c+r}{a+b-1}) + e(-\frac{c+r}{a+b-1}) + f < 0. \end{cases}$$

Доказательство: Это утверждение следует из того факта, что Шеффера функция, не должна сохранять ни одну из констант. Рассмотрим обратное, подставим какую-либо произвольную константу k:

$$g(k,k) = ak + bk + c + r\theta(dk + ek + f) = k$$

Возможны 2 случая для $\theta(dk + ek + f)$:
1)При $dk + ek + f < 0$, при этом $\theta(dk + ek + f) = 0$,
Тогда функция $g(k,k)$ примет вид:

$$g(k,k) = ak + bk + c = k.$$

Откуда найдем выражение для k:

$$ak + bk - k = -c,$$

$$k(a + b - 1) = -c.$$

$$k = -\frac{c}{(a + b - 1)}$$

при условии, что $a + b - 1 \neq 0$

Т.е. если система условий $\begin{cases} dk + ek + f < 0, \\ a + b - 1 \neq 0. \end{cases}$ выполнена, то функция g сохраняет константу k:

$$k = -\frac{c}{a + b - 1}.$$

Следовательно, для того чтобы такая константа k не существовала, т.е. функция $g(x,y)$ не сохраняла, ни одну из констант, необходимо чтобы:

$$d(-\frac{c}{a + b - 1}) + e(-\frac{c}{a + b - 1}) + f \geq 0.$$

Аналогично рассматривается второй случай:
2) При $dk + ek + f \geq 0$, при этом $\theta(dk + ek + f) = 1$,
Тогда функция $g(k,k)$ примет вид:

$$g(k,k) = ak + bk + c + r = k.$$

Откуда найдем выражение для k:

$$ak + bk - k = -c - r,$$

$$k(a + b - 1) = -c - r,$$

$$k = -\frac{c + r}{a + b - 1}$$

при условии, что $a + b - 1 \neq 0$.

Т.е. при выполнении системы условий $\begin{cases} dk + ek + f \geq 0, \\ a + b - 1 \neq 0. \end{cases}$, такая константа k существует:

$$k = -\frac{c + r}{a + b - 1}.$$

Т.е., для того чтобы такая константа k не существовала, необходимо, чтобы:

$$d(-\frac{c + r}{a + b - 1}) + e(-\frac{c + r}{a + b - 1}) + f < 0.$$

Следовательно, для того, чтобы g(x,y) не сохраняла ни какую из переменных необходимо чтобы ее коэффициенты удовлетворяли системе условий:

$$\begin{cases} d(-\frac{c}{a+b-1}) + e(-\frac{c}{a+b-1}) + f \geq 0, \\ d(-\frac{c+r}{a+b-1}) + e(-\frac{c+r}{a+b-1}) + f < 0. \end{cases}$$

Утверждение доказано.

Рассмотрим пример функции, удовлетворяющей описанным выше условиям и являющейся Шефферовой в классе BPP.

Такая функция имеет вид:

$$F(x, y) = -\frac{1}{2}x + y + \frac{1}{2}\theta(-x).$$

Теперь подставим коэффициенты рассматриваемой функции $F(x, y)$ в найденные уравнения.

$$a = -\frac{1}{2}, b = 1, c = 0, r = \frac{1}{2}, d = -1, e = 0, f = 0.$$

$$\begin{cases} -1(-\frac{0}{-\frac{1}{2}+1-1}) + 0(-\frac{0}{-\frac{1}{2}+1-1}) + 0 \geq 0, \\ -1(-\frac{0+-\frac{1}{2}}{-\frac{1}{2}+1-1}) + 0(-\frac{0+\frac{1}{2}}{-\frac{1}{2}+1-1}) + 0 < 0. \end{cases}, \begin{cases} 0 \geq 0, \\ -(-\frac{\frac{1}{2}}{-\frac{1}{2}+1-1}) < 0. \end{cases}, \begin{cases} 0 \geq 0, \\ -1 < 0. \end{cases}$$

Условия выполняются. Докажем теперь, что эта функция является Шефферовой в классе кусочно-параллельных функций с двоично-рациональными коэффициентами.

Теорема 1. *Функция:*

$$F(x,y) = -\frac{1}{2}x + y + \frac{1}{2}\theta(-x),$$

где

$$\theta(-x) \quad = \quad \begin{cases} 1 & при \ -x \geq 0 \\ 0 & при \ -x < 0 \end{cases},$$

является минимальной Шефферовой функцией в классе кусочно-параллельных функций с двоично-рациональными коэффициентами.

Доказательство: Очевидно функция является минимальной в BPP. Докажем теперь, что она Шефферова.

Для удобства доказательства проведем некоторые преобразования функции $F(x,y)$. Используя равенство:

$$\theta(-x) = 1 - \theta^{'}(x),$$

заменим $\theta(-x)$ на $\theta^{'}(x)$, где

$$\theta^{'}(x) \quad = \quad \begin{cases} 1 & \text{при } x > 0 \\ 0 & \text{при } x \leq 0 \end{cases}.$$

$$F(x,y) = -\frac{1}{2}x + y + \frac{1}{2}(1 - \theta^{'}(x)) = -\frac{1}{2}x + y - \frac{1}{2}\theta^{'}(x) + \frac{1}{2},$$

Теперь докажем, что $F(x,y)$ является Шефферовой в BPP.

Для начала, получим константу $\frac{1}{2}$ из функции $F(x,y)$. Для этого рассмотрим следующую суперпозицию:

$$F_1(x,y) = F(x, F(x,y)) = -\frac{1}{2}x - \frac{1}{2}x + y - \frac{1}{2}\theta^{'}(x) + \frac{1}{2} - \frac{1}{2}\theta^{'}(x) + \frac{1}{2},$$

$$F_1(x,y) = -x + y - \theta^{'}(x) + 1.$$

Отождествим переменные x и y:

$$F_2(x) = F_1(x,x) = -x + x - \theta^{'}(x) + 1 = -\theta^{'}(x) + 1 = 1 - \theta^{'}(x).$$

Применим операцию суперпозиции для F и $F_2(x)$:

$$F_3 = F(F_2(x), F_2(x)) = -\frac{1}{2}(1 - \theta^{'}(x)) + 1 - \theta^{'}(x) - \frac{1}{2}\theta^{'}(1 - \theta^{'}(x)) + \frac{1}{2}$$

$$F_3 = -\frac{1}{2} + \frac{1}{2}\theta^{'}(x) + 1 - \theta^{'}(x) - \frac{1}{2}\theta^{'}(1 - \theta^{'}(x)) + \frac{1}{2} = 1 - \theta^{'}(x) + \frac{1}{2}\theta^{'}(x) - \frac{1}{2}\theta^{'}(1 - \theta^{'}(x)).$$

Рассмотрим чему равно $\theta^{'}(1-\theta^{'}(x))$. Так как

$$1-\theta^{'}(x) \quad = \quad \begin{cases} 0 & \text{при } x>0 \\ 1 & \text{при } x\leq 0 \end{cases},$$

и одновременно

$$\theta^{'}(1-\theta^{'}(x)) \quad = \quad \begin{cases} 0 & \text{при } x>0 \\ 1 & \text{при } x\leq 0 \end{cases}.$$

Значит, $\theta^{'}(1-\theta^{'}(x))=1-\theta^{'}(x)$. Следовательно,

$$F_3=1-\theta^{'}(x)+\frac{1}{2}\theta^{'}(x)-\frac{1}{2}(1-\theta^{'}(x))=1-\theta^{'}(x)+\frac{1}{2}\theta^{'}(x)-\frac{1}{2}+\frac{1}{2}\theta^{'}(x)=\frac{1}{2}.$$

Получим теперь константу ноль. Для чего, подставим в функцию F_2 вместо переменной x, константу $\frac{1}{2}$:

$$F_2(\frac{1}{2})=-\theta^{'}(\frac{1}{2})+1=-1+1\equiv 0.$$

Получим функцию $y+1$ и константу 1:

$$F_4(y)=F_1(0,y)=0+y-\theta^{'}(0)+1=y+1,$$

$$F_4(0)\equiv 1.$$

Теперь получим функцию $-\theta^{'}(x)+2$, содержащую $\theta^{'}(x)$ и в тоже время всегда принимающую положительные значения:

$$-\theta^{'}(x)+2 \quad = \quad \begin{cases} 1 & \text{при } x>0 \\ 2 & \text{при } x\leq 0 \end{cases},$$

$$F_5(x)=F_4(F_2(x))=-\theta^{'}(x)+1+1=-\theta^{'}(x)+2>0.$$

Получим функцию $-x$.

Переименуем в функции $F_1(x,y)$ переменную x в переменную k и применив операцию суперпозиции для $F_1(x,y)$ и $F_1(k,y)$ получим:

$$F_6(x,k,y)=F_1(x,F_1(k,y))=-x-k+y-\theta^{'}(k)+1-\theta^{'}(x)+1=-x-k+y-\theta^{'}(k)-\theta^{'}(x)+2$$

Применим суперпозицию для $F_6(x,k,y)$ и $F_5(x)$, причем

$$F_5(x)>0:$$

$$F_7(x,y) = F_6(x, F_5(x), y) = -x + \theta'(x) - 2 + y - \theta'(-\theta'(x) + 2) - \theta'(x) + 2,$$

$$F_7(x,y) = -x + \theta'(x) - 2 + y - 1 - \theta'(x) + 2 = -x + y - 1,$$

$$F_8(x) = F_4(F_7(x,y)) = -x + y - 1 + 1 = -x + y,$$

$$F_9(x) = F_8(x,0) = -x.$$

Функцию $x + y$ получим следующим образом:

$$F_{10}(x,y) = F_8(F_9(x), y) = -(-x) + y = x + y.$$

Теперь получим функцию $\frac{1}{2}x$. Для этого переименуем в функции $F(x,y)$ переменную x в переменную k и применив операцию суперпозиции для $F_1(x,y)$ и $F(k,y)$ получим:

$$F_{11}(x,k,y) = F_1(x, F(k,y)) = -x - \frac{1}{2}k + y - \frac{1}{2}\theta'(k) + \frac{1}{2} - \theta'(x) + 1,$$

рассмотрим суперпозицию $F_{11}(x,k,y)$ и $F_5(x)$:

$$F_{12}(x,y) = F_{11}(x, F_5(x), y) = -x - \frac{1}{2}(\theta'(x)+2) + y - \frac{1}{2}\theta'(-\theta'(x)+2) + \frac{1}{2} - \theta'(x) + 1 =$$

$$= -x + \frac{1}{2}\theta'(x) - 1 + y - \frac{1}{2} + \frac{1}{2} - \theta'(x) + 1 = -x - \frac{1}{2}\theta'(x) + y.$$

И далее рассмотрим суперпозицию $F'(x,y) = F(x,-y)$ и $F_{12}(x,0)$:

$$F_{13}(x) = F'(x, F_1 2(x,0)) = -\frac{1}{2}x - (-x - \frac{1}{2}\theta'(x)) - \frac{1}{2}\theta'(x) + \frac{1}{2} =$$

$$= -\frac{1}{2}x + x + \frac{1}{2}\theta'(x) - \frac{1}{2}\theta'(x) + \frac{1}{2} = \frac{1}{2}x + \frac{1}{2}.$$

Из функции $-x$ и константы $\frac{1}{2}$ получим константу $-\frac{1}{2}$, и из полученной константы, функций $x + y$ и $F_{13}(x)$ получим $\frac{1}{2}x$:

$$F_{14}(x) = F_{10}(F_1 3(x), -\frac{1}{2}) = \frac{1}{2}x + \frac{1}{2} - \frac{1}{2} = \frac{1}{2}x.$$

И последнее, выведем $\theta(x)$:

$$\theta(x) = 1 - \theta'(-x).$$

Таким образом, получены все функции множества B,

$$B = \{\frac{1}{2}x, -x, x + y, \theta(x)\}$$

являющегося порождающим для класса кусочно-параллельных функций с двоично-рациональными коэффициентами. Следовательно функция

$$F(x,y) = -\frac{1}{2}x + y - \frac{1}{2}\theta'(x) + \frac{1}{2}$$

является Шефферовой в данном классе. Следовательно, и функция

$$F(x,y) = -\frac{1}{2}x + y + \frac{1}{2}\theta(-x),$$

является минимальной Шефферовой в BPP.
Теорема доказана.

Автор выражает искреннюю признательность Часовских А.А. за ценные обсуждения, советы и замечания в ходе работы над темой.

Список литературы

[1] Половников В. С. Об оптимизации структурной реализации нейронных сетей. Диссертация на соискание ученой степени кандидата физико-математических наук — Москва, 2006.

[2] Кудрявцев В. Б., Функциональные системы . — М.: Изд-во МГУ, 1982.

[3] Хайкин С. Нейронные сети: полный курс // 2-е издание. Вильямс, 2006. // Вестн. Моск. ун-та. Матем. Механ. — 2016. — № 4. — С. 12–17.

[4] Кан А. Н. Вопросы выразимости в классе нейронных функций //Интеллектуальные системы — том 19, — Выпуск 1. — 2015.

[5] Кан А. Н. Вопросы полноты в классе кусочно-линейных непрерывных функций. //Интеллектуальные системы — том 21, — Выпуск 2. — 2017.

[6] Агафонова М. В. О классе нейронных функций с двоично-рациональными параметрами. //Интеллектуальные системы — том 22, — Выпуск 1. — 2018.

[7] Яблонский С. В. Введение в дискретную математику. . — М.: Изд-во Наука, 1986.

On a minimal Scheffer function in the class of partial-parallel functions defined over binary rational numbers.
Agafonova M.V.

This article discusses the class of neural partial-parallel functions with binary rational coefficients (BPP-class). It is proven that a minimal Scheffer function exists in this class. By a minimal Scheffer function in the class, we mean a function of this class that generates this class by superposition operations and contains the minimum possible number of variables and threshold functions. It was established that a minimal Scheffer function contains two variables and one threshold function. The article also provides one of the necessary conditions for the Scheffer-type function to be contained in the BPP-class.

Keywords: class, partial-parallel functions, neural functions, binary coefficients, superposition operations, Scheffer function.

Верхняя оценка энергопотребления в классе объемных схем

Ефимов А.А.

В данной работе рассматриваются объёмные схемы, являющиеся обобщением плоских схем в пространстве. Был рассмотрен класс схем, реализующих булевы функции. Для этого класса получена верхняя оценка потенциала — меры мощности, равной количеству элементов схемы, выдающих единицу на данном входном наборе. Показано, что любую функцию от n переменных можно реализовать объемной схемой, потенциал которой не превосходит $\mathcal{O}(2^{n/3})$.

Ключевые слова: схемы из функциональных элементов, объёмные схемы, мощность схемы, потенциал.

1. Введение

В ряде работ исследовалась сложность схем из функциональных элементов, реализующих функции алгебры логики от n аргументов. Однако, зачастую в них рассматривались схемы, в которых не накладывалось никаких ограничений на размещение элементов схемы, способ соединения и т.п. На самом деле в любой схеме, когда она располагается в пространстве, функциональные элементы имеют определенную длину, ширину и соединяются проводниками, размеры которых следует учитывать.

Данная работа посвящена кубическим схемам, которые определяются аналогично плоским схемам, но в пространстве. Впервые понятие плоской схемы было введено Кравцовым в 1967 году [1]. Развитие теории плоских схем было связано с развитием технологии производства и укладки реальных микросхем. Идея о том, что схемы можно укладывать друг на друга в пространстве была также известна давно, но не находила широкого применения вплоть до недавнего времени. Лишь несколько лет назад подобная технология начала использоваться, так как у инженеров закончились способы выжать лучшие характеристики из чипов прежне-

го размера. В частности, речь идёт о том, чтобы в будущем использовать многослойные чипы.

Основной целью данной работы является обобщение результатов Калачева [2, 3] на объёмные схемы. Как и в его работах, автор использует такое понятие сложности схемы, как максимальный потенциал. Он равен максимальному значению количества единиц на всех внутренних узлах схемы, взятому по всем входным наборам. Неформально говоря, потенциал показывает количество «энергии» схемы, необходимой для её функционирования. В данной работе была получена верхняя оценка потенциала для класса булевых функций.

2. Основные понятия и формулировка результатов

Кубическим элементом будем называть булев оператор, у которого в сумме не более шести входов и выходов, причем каждому его входу и выходу сопоставлена некоторая метка из множества $\{l, t, r, b, f, a\}$, причём метки не повторяются.

Метки будем называть сторонами элемента:

- l – левая сторона;
- r – правая сторона;
- t – верхняя сторона;
- b – нижняя сторона;
- f – передняя сторона;
- a – задняя сторона.

Кубический элемент будем изображать в виде единичного куба в пространстве. При этом входам и выходам элемента сопоставляются грани куба в соответствии с присвоенными им метками.

Метки, присвоенные входам (выходам) оператора будем называть *входами (выходами)* элемента. Метки, не присвоенные ни входам, ни выходам, будем называть *изоляторами*. Множество входов (выходов) элемента e будем обозначать $in(e)$ ($out(e)$). Входы и выходы элемента будем называть его *контактами*.

Если на всех выходах элемента реализуются тождественные функции, то будем называть элемент *коммутационным*, иначе – *логическим*.

Коммутационный элемент соответсвует либо проводнику в микросхеме, либо пересечению проводов, либо тождественной функции, служащей для усиления сигнала.

Описывать элемент можно уравнениями, которые задают его оператор, заменяя все переменные в них на сопоставленые им метки (l, t, r, b, f, a). Тогда в левой части каждого уравнения будет стоять выходная метка, а в правую часть будут входить только входные метки.

Всюду далее значок $:=$ будет обозначать «по определению равно».

За E обозначим множество всех кубических элементов.

Сетью из кубических элементов на множестве $M \subset \mathbb{Z}^3$ будем называть отображение $K : M \to E$.

Элемент $K(x, y, z)$ будем называть *элементом схемы K с координатами (x,y,z)*.

Левой, правой, верхней, нижней, передней и задней стороной элемента e с координатами (x, y, z) будем называть точки с координатами $(x - \frac{1}{2}, y, z), (x + \frac{1}{2}, y, z), (x, y, z + \frac{1}{2}), (x, y, z - \frac{1}{2}), (x, y + \frac{1}{2}, z), (x, y - \frac{1}{2}, z)$ соответственно.

Будем говорить, что сеть K из кубических элементов корректна, если для любых элементов x и y схемы K верно, что если сторона a элемента x совпадает со стороной b элемента y, то выполнено одно из условий:

- один из элементов x, y – изолирующий,
- стороны a и b являются изоляторами,
- среди них одна является входом, другая выходом, например, a – выход, а b – вход, в таком случае будем говорить, что выход a *подключен* ко входу b.

Множество M будем называть *носителем* сети K.

Введём понятие *графа корректной сети из кубических элементов K* (будем обозначать G_K). G_K – ориентированный граф, вершинами которого являются входы и выходы элементов схемы. Если выход одного элемента подключен ко входу другого, то им будет соответствовать одна и та же вершина графа (будем говорить, что эта вершина является выходом первого элемента и входом второго). Из вершины a в вершину b ведет ребро в том и только в том случае, когда существует элемент e такой, что a является его входом, b – выходом, причем функция, реализуемая на выходе b, существенно зависит от входа a.

Объёмной схемой или *схемой из кубических элементов* на множестве $M \subset \mathbb{Z}^3$ будем называть корректную сеть из кубических элементов, в графе которой нет ориентированных циклов. Множество M будем называть *носителем* схемы K.

Длиной схемы K будем называть длину наименьшего прямоугольного параллелепипеда, содержащего все непустые элементы схемы K, обозначается $l(K)$.

Шириной схемы K будем называть ширину наименьшего прямоугольного параллелепипеда, содержащего все непустые элементы схемы K, обозначается $w(K)$.

Высотой схемы K будем называть высоту наименьшего прямоугольного параллелепипеда, содержащего все непустые элементы схемы K, обозначается $h(K)$.

Если вход (выход) элемента не подключен к выходу (входу) другого элемента, будем его называть *входом (выходом)* схемы. *Контактами* схемы K будем называть её входы и выходы, и обозначать их $In(K), Out(K)$ соответственно.

Узлами схемы K будем называть вершины графа G_K.

Если M – носитель схемы K, то величину $|M|$, равную количеству элементов в множестве M, будем называть *объёмом* схемы K и обозначать $|K|$.

В графе G_K будем считать, что все ребра имеют вес 1. *Расстоянием между вершинами* в графе G_K будем считать длину наименьшего пути между этими вершинами. *Расстоянием между узлами* схемы будем называть расстояние между соответствующими вершинами в G_K. Расстояние от узла a до узла b на схеме K будем обозначать $\rho_K(a, b)$.

Каждой объёмной схеме K можно сопоставить схему их функциональных элементов $Circ(K)$ следующим образом:

1) каждой функции $f_{s,i}$, которую реализует i-й выход элемента s объёмной схемы, сопоставим функциональный элемент $e_{s,i}$, реализующий $f_{s,i}$; если i-й и j-й выходы являются выходами одной и той же функции, то им будет соответствовать один и тот же функциональный элемент;

2) если i-й выход элемента s_1 подключен к j-му входу элемента s_2, то соединим выход элемента $e_{s_1,i}$ с j-ми входами элементов $e_{s_2,k}$ для всех k, для которых $f_{s_2,k}$ существенно зависит от j-го аргумента;

3) удалим все тождественные функции, присоединив их вход ко всем их выходам.

Будем говорить, что схема K *реализует* булев оператор F, если схема из функциональных элементов $Circ(K)$ реализует F. Через $Impl(F)$ обозначим множество всех объёмных схем, реализующих оператор F.

Назовём схему K *минимальной*, если она обладает минимальным объёмом среди всех объёмных схем, реализующих F_K.

Через $V(F)$ обозначим объём минимальной схемы, реализующей оператор F.

Будем говорить, что объёмные схемы K_1 и K_2 *равны* и писать $K_1 = K_2$, если существует параллельный перенос пространства, который позволяет совместить схемы K_1 и K_2, иначе будем говорить, что они *различны*. Для каждой схемы K зафиксируем некоторую нумерацию её узлов. На i-м узле реализуется некоторая функция g_i от входных переменных схемы K (на входах схемы счиатем, что реализуются тождественные функции).

Везде далее будем считать, что схема K имеет n входов и l узлов. *Состоянием* схемы K на входном наборе x назовём вектор

$$s_K(x) := (g_1(x), ..., g_l(x)).$$

Если $v = (v_1, ..., v_q) \in \{0,1\}^q$, обозначим $|v| := v_1 + v_2 + ... + v_q$.

Потенциалом схемы K на входном наборе $x \in \{0,1\}^n$ назовём величину $u_K(x) := |s_K(x)|$.

Максимальным потенциалом схемы K назовём величину

$$\hat{U}(K) := \max_{x \in \{0,1\}^n} u_K(x).$$

Пусть $f : \{0,1\}^n \to \{0,1\}$ – булева функция. Тогда

$$\hat{U}(f) := \min_{K \in Impl(f)} \hat{U}(K).$$

Если $Impl(f)$ пусто, то формально полагаем $\hat{U}(f) = \infty$.

Теорема 1 (Основная теорема)**.** *Пусть дана булева функция $f(x_1, x_2, \ldots, x_n)$. Тогда существует объемная схема V_f со входами $x_1, x_2, \ldots, x_n$ на одном выходе которой реализуется функция $f(x_1, x_2, \ldots, x_n)$, причём схема V_f обладает следующими характеристиками:*

1) $l(V_f) = \mathcal{O}(2^{n/3}),\ w(V_f) = \mathcal{O}(2^{n/3}),\ h(V_f) = \mathcal{O}(2^{n/3})$.

2) $\hat{U}(V_f) = \mathcal{O}(2^{n/3})$.

3. Реализация булевой функции

3.1. Параметры основных блоков

Для реализации булевой функции нам потребуются несколько различных блоков. Опишем их характеристики.

1) Дешифратор D'_n (*Калачёв Г.В., [3, лемма 2.14]*):

$$l(D'_n) = 2^n,\ w(D'_n) \le n(n+3)/2,\ h(D'_n) = 1,\ \hat{U}(D'_n) = \mathcal{O}(n^2 \cdot 2^n).$$

2) Дешифратор D^1_n:

$$l(D^1_n) = \mathcal{O}(2^n),\ w(D^1_n) = \mathcal{O}(2^{n/2}),\ h(D^1_n) = 1,\ \hat{U}(D^1_n) = \mathcal{O}(2^n).$$

3) Блок дешифраторов $D'_{n,k}$ (*Калачёв Г.В., [3, лемма 2.19]*):

$$l(D'_{n,k}) = \mathcal{O}(k \cdot 2^n),\ w(D'_{n,k}) = \mathcal{O}(n^2) + \mathcal{O}(nk),\ h(D'_{n,k}) = 1,$$

$$\hat{U}(D'_{n,k}) = \mathcal{O}(kn^2 \cdot 2^n) + \mathcal{O}(k^2 n \cdot 2^n).$$

4) Левый обратный блок $D'^{-1}_{n,k}$ (*Калачёв Г.В., [3, лемма 2.20]*):

$$l(D'^{-1}_{n,k}) = \mathcal{O}(k \cdot 2^n),\ w(D'^{-1}_{n,k}) = \mathcal{O}(kn^2),\ h(D'^{-1}_{n,k}) = 1,$$

$$\hat{U}(D'^{-1}_{n,k}) = \mathcal{O}(k^2 n^2 \cdot 2^n).$$

5) Схема S_f, реализующая функцию f от n переменных (*Калачёв Г.В., [3, лемма 2.25]*):

$$l(S_f) = \mathcal{O}(2^{n/2}),\ w(S_f) = \mathcal{O}(2^{n/2}),\ h(S_f) = 1,\ \hat{U}(S_f) = \mathcal{O}(2^{n/2}).$$

6) Блок S^1_f:

$$l(S^1_f) = \mathcal{O}(2^{n/3}),\ w(S^1_f) = \mathcal{O}(2^{n/3}),\ h(S^1_f) = 1,\ \hat{U}(S^1_f) = \mathcal{O}(2^{n/3}).$$

7) Схема V^1_f, реализующая функцию f от n переменных:

$$l(V^1_f) = \mathcal{O}(2^{n/3}),\ w(V^1_f) = \mathcal{O}(2^{n/3}),\ h(V^1_f) = \mathcal{O}(2^{n/3}),\ \hat{U}(V^1_f) = \mathcal{O}(2^{n/3}).$$

3.2. Реализация вспомогательных блоков

В данном параграфе подробно опишем реализацию всех вспомогательных блоков. Будем считать, что если у нас есть плоская схема, то можно естественным образом построить объемную схему такой же длины, ширины, и единичной высоты. При этом ясно, что оценки потенциала такой объемной схемы будут совпадать. Отметим, что некоторые леммы из работы [3] мы переформулируем указанным образом, то есть будем считать, что плоские схемы — это объемные схемы единичной высоты.

Почти все блоки, которые мы будем использовать будут иметь так называемый *управляющий вход* z. Если $z = 0$ и значения других входов равны 0, то потенциал внутренней части блока равен 0. Отметим, что значения выходов в таком случае также равны 0, то есть реализуемая схемой функция от переменных $z, x_1, \ldots, x_n$ лежит в классе T_0. Таким образом, вход z играет роль «выключателя» блока. Наличие такого входа позволяет достаточно легко оценивать потенциал схем, состоящих из нескольких блоков.

Для удобства введём еще одно обозначение. Пусть $i \in \mathbb{Z}, 0 \leq i \leq 2^n - 1$. Тогда $\bar{i}_1, \bar{i}_2, \ldots \bar{i}_n$ — разложение числа i в двоичном виде, где $\bar{i}_1$ — младший бит разложения, а $\bar{i}_n$ — старший.

Дешифратор D'_n.

D'_n — плоский дешифратор.

Лемма 1. *(Калачёв Г.В., [3, лемма 2.14]) Существует объемная схема D'_n со входами $z, x_1, \ldots, x_n$ имеющая 2^n выходов, на i-м выходе которой на допустимых наборах $(z \geq x_1 \vee \ldots \vee x_n)$ реализуется функция*

$$z x_1^{\bar{i}_1} x_2^{\bar{i}_2} \ldots x_n^{\bar{i}_n},$$

причём схема D'_n обладает следующими характеристиками:

1) $l(D'_n) = \mathcal{O}(2^n),\ w(D'_n) = \mathcal{O}(n^2),\ h(D'_n) = 1.$

2) $\hat{U}(D'_n) = \mathcal{O}(n^2 \cdot 2^n).$

Дешифратор D^1_n.

D^1_n — плоский дешифратор, имеющий оптимальный потенциал.

Лемма 2. *Существует объемная схема D_n^1 со входами $z, x_1, \ldots, x_n$ имеющая 2^n выходов, на i-м выходе которой на допустимых наборах $(z \geq x_1 \vee \ldots \vee x_n)$ реализуется функция*

$$z x_1^{\bar{i}_1} x_2^{\bar{i}_2} \ldots x_n^{\bar{i}_n},$$

причём схема D_n^1 обладает следующими характеристиками:

1) $l(D_n^1) = \mathcal{O}(2^n),\ w(D_n^1) = \mathcal{O}(2^{n/2}),\ h(D_n^1) = 1.$

2) $\hat{U}(D_n^1) = \mathcal{O}(2^n).$

Доказательство. Будем полагать, что $n = 2k$. Построим дешифратор D_n^1 из двух дешифраторов D'_k и некоторого количества элементов & и &′ (см. рис. 1).

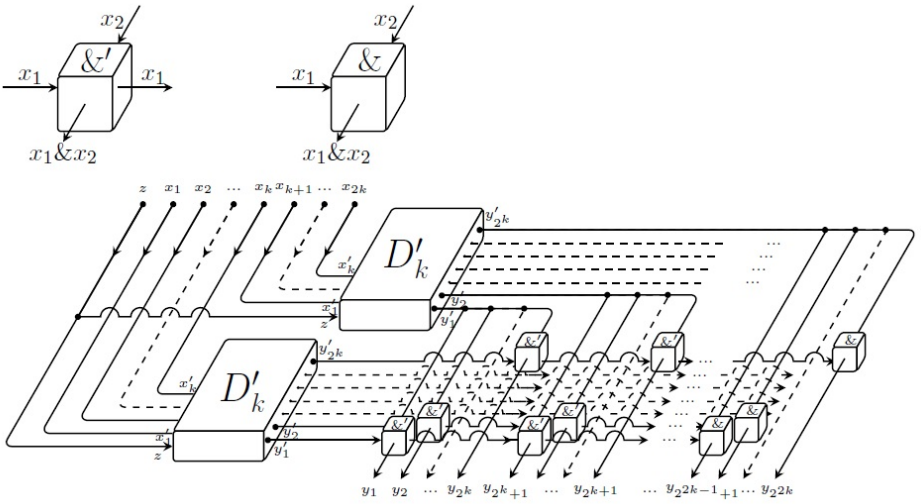


Рис. 1. Реализация дешифратора D_n^1.

Посчитаем характеристики схемы D_n^1.

$$l(D_n^1) = (k+1) + 2 \cdot \mathcal{O}(k^2) + 2^k \cdot 2^k = \mathcal{O}(2^{2k}) = \mathcal{O}(2^n).$$

$$w(D_n^1) = 2 \cdot l(D'_k) = \mathcal{O}(2^k) = \mathcal{O}(2^{n/2}).$$

$$h(D_n^1) = 1.$$

Оценим потенциал схемы. Заметим, что так как у каждого элемента схемы максимум 6 входов/выходов, то потенциал не превосходит объема схемы, умноженного на 6. Таким образом, потенциал всей «левой» части схемы (включая блоки D'_k) оценим через объем.

$$U_1 \leq 6 \cdot w(D^1_n) \cdot ((k+1) + 2 \cdot w(D'_k)).$$

Далее воспользуемся тем фактом, что при любом входном наборе из любой из 2-х схем D'_k выходит ровно один активный провод. Оценим длину активного провода нижнего блока D'_k.

$$U_2 \leq \mathcal{O}(2^{2k}).$$

Оценим длину активного провода верхнего блока D'_k.

$$U_3 \leq \mathcal{O}(2^{2k}).$$

Также заметим, что сигнал из верхнего блока D'_k в какой-то момент разветвится на 2^k сигналов, которые пойдут вниз и пересекутся с сигналом из нижнего блока D'_k. Оценим потенциал этой части схемы через объем.

$$U_4 \leq 6 \cdot 2l(D'_k) \cdot 2^k.$$

Сложим полученные результаты и получим оценку потенциала:

$$\hat{U}(D^1_n) = U_1 + U_2 + U_3 + U_4 \leq 6 \cdot w(D^1_n) \cdot ((k+1) + 2 \cdot w(D'_k)) + \mathcal{O}(2^{2k}) + \mathcal{O}(2^{2k}) +$$

$$+ 6 \cdot 2l(D'_k) \cdot 2^k = \mathcal{O}(2^k) \cdot \mathcal{O}(k^2) + \mathcal{O}(2^{2k}) + \mathcal{O}(2^{2k}) = \mathcal{O}(2^n).$$

В случае, когда $n = 2k + 1$ построим схему для $n = 2k + 2$ и подадим на последний вход x_{2n+2} константу 0. Получим схему с нужными характеристиками в силу того, что константы в оценках схемы увеличатся максимум в 2 раза, нам нужны оценки по порядку. □

Замечание: В данной работе мы используем дешифратор D^1_n, так как он имеет оптимальный потенциал. Дешифратор D'_n использовался в работе [3] потому, что у него была оптимальная «глубина», хотя и неоптимальный потенциал.

Блок дешифраторов $D'_{n,k}$.

Плоский блок дешифраторов $D'_{n,k}$. На вход подаются k групп переменных по n штук + отдельная переменная z. Переменные обозначаем x_j^i, где i — номер группы, а i — номер переменной в этой группе. Каждую группу переменных мы подаем на отдельный дешифратор D'_n, переменную z подаем на все дешифраторы. Объединение выходов всех дешифраторов есть выходы схемы. Основное свойство блока дешифраторов в том, что у него сравнительно небольшой потенциал, а при этом на выходе активны всегда ровно k выходов (по одному с каждого дешифратора).

Лемма 3. *(Калачёв Г.В., [3, лемма 2.19]) Существует объемная схема $D'_{n,k}$ со входами $z, x_1^1, \ldots, x_n^1, x_1^2 \ldots, x_n^2, \ldots x_n^k$ имеющая $k \cdot 2^n$ выходов, на (i,j)-м выходе которой реализуется функция*

$$(x_1^i)^{\bar{j}_1}(x_2^i)^{\bar{j}_2} \ldots (x_n^i)^{\bar{j}_n},$$

причём схема $D'_{n,k}$ обладает следующими характеристиками:

1) $l(D'_{n,k}) = \mathcal{O}(k \cdot 2^n),\ w(D'_{n,k}) = \mathcal{O}(n^2) + \mathcal{O}(nk),\ h(D'_{n,k}) = 1.$

2) $\hat{U}(D'_{n,k}) = \mathcal{O}(kn^2 \cdot 2^n) + \mathcal{O}(k^2 n \cdot 2^n).$

Обратный блок дешифраторов $D'^{-1}_{n,k}$.

Плоский левый обратный блок дешифраторов $D'^{-1}_{n,k}$ к блоку $D'_{n,k}$.

Лемма 4. *(Калачёв Г.В., [3, лемма 2.20]) Существует объемная схема $D'^{-1}_{n,k}$ со входами $z, x_1^1, \ldots, x_{2^n}^1, x_1^2 \ldots, x_{2^n}^2, \ldots x_{2^n}^k$ имеющая $k \cdot n$ выходов, причём схема $D'^{-1}_{n,k}$ обладает следующими характеристиками:*

1) $l(D'^{-1}_{n,k}) = \mathcal{O}(k \cdot 2^n),\ w(D'^{-1}_{n,k}) = \mathcal{O}(kn^2),\ h(D'^{-1}_{n,k}) = 1.$

2) $\hat{U}(D'^{-1}_{n,k}) = \mathcal{O}(k^2 n^2 \cdot 2^n).$

Блок S_f.

Плоский блок S_f. Блок, который выдает значения данной булевой функции f и имеет оптимальные параметры (на плоскости).

Лемма 5. *(Калачёв Г.В., [3, лемма 2.25]) Пусть дана функция $f(x_1, x_2, \ldots, x_n)$. Тогда существует объемная схема S_f со входами $z, x_1, x_2, \ldots, x_n$ на одном выходе которой на допустимых наборах ($z \geq x_1 \vee \ldots \vee x_n$) реализуется функция $zf(x_1, x_2, \ldots, x_n)$, причём схема S_f обладает следующими характеристиками:*

1) $l(S_f) = \mathcal{O}(2^{n/2}),\ w(S_f) = \mathcal{O}(2^{n/2}),\ h(S_f) = 1.$

2) $\hat{U}(S_f) = \mathcal{O}(2^n).$

3.3. Реализация булевой функции

Итак, пусть дана булева функция $f(x_1, x_2, \ldots, x_n)$. Будем считать, что $n = 6k$. Разложим функцию f по последним $2k$ переменным:

$$f(x_1, x_2, \ldots, x_{6k}) = \bigvee_{i=0}^{2^{2k}-1} x_{4k+1}^{\bar{i}_1} x_{4k+2}^{\bar{i}_2} \ldots x_{6k}^{\bar{i}_{2k}} f_i(x_1, \ldots, x_{4k}), \qquad (1)$$

где

$$f_i(x_1, \ldots, x_{4k}) = f(x_1, \ldots, x_{4k}, \bar{i}_1, \bar{i}_2, \ldots, \bar{i}_{2k}).$$

Для каждой функции f_i от $4k$ переменных построим вспомогательный блок $S^1_{f_i}$ (см. рис. 2), реализующий данную функцию. Особенностью данного блока является тот факт, что на вход ему мы подаем ему выходы из блока дешифраторов $D'_{k,4}$. Таким образом, если вход z неактивен, то потенциал всей схемы равен 4. Подробно посчитаем характеристики схемы.

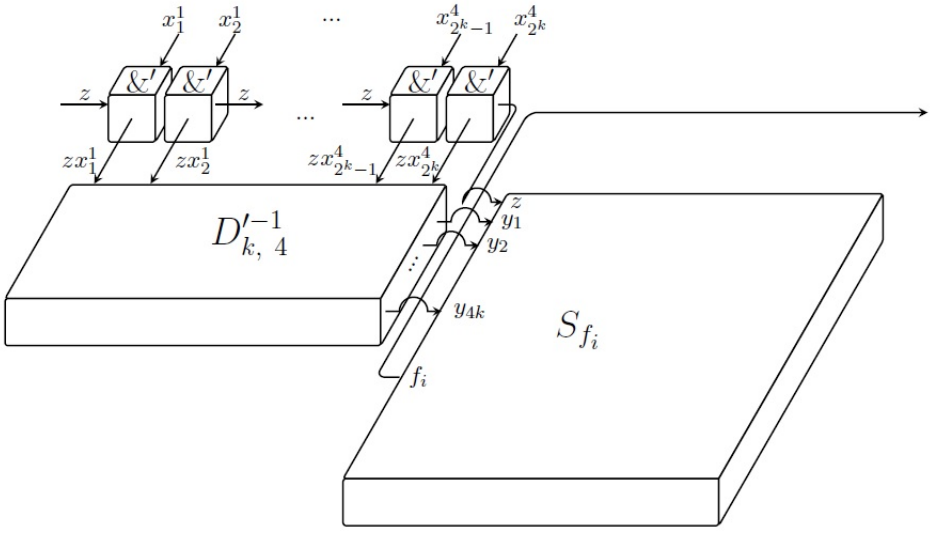


Рис. 2. Реализация блока $S^1_{f_i}$.

Лемма 6. *Пусть дана булева функция $f_i(x_1, x_2, \ldots, x_{4k})$. Тогда существует объемная схема $S^1_{f_i}$, такая, что схема $S^1_{f_i} \circ D'_{k,4}$ со входами*

$z, x_1, x_2, \ldots, x_n$ на одном выходе на допустимых наборах $(z \geq x_1 \vee \ldots \vee x_n)$ реализует функцию $zf_i(x_1, x_2, \ldots, x_{4k})$, причём схема $S^1_{f_i}$ обладает следующими характеристиками:

1) $l(S^1_{f_i}) = \mathcal{O}(2^{2k})$, $w(S^1_{f_i}) = \mathcal{O}(2^{2k})$, $h(S^1_{f_i}) = 1$.

2) $\hat{U}(S^1_{f_i}) = \mathcal{O}(2^{2k})$, *если* $z = 1$; *и* $\hat{U}(S^1_{f_i}) = 4$, *если* $z = 0$.

Доказательство.

$$l(S^1_{f_i}) = l(D^{-1}_{k,4}) + 1 + w(S_{f_i}) = \mathcal{O}(4 \cdot 2^k) + \mathcal{O}(2^{2k}) = \mathcal{O}(2^{2k}).$$

$$w(S^1_{f_i}) = l(S_{f_i}) + 1 = \mathcal{O}(2^{2k}).$$

$$h(S^1_{f_i}) = 1.$$

Оценим потенциал, если $z = 1$.

$$\hat{U}(S^1_{f_i}) = \mathcal{O}(4 \cdot 2^k) + \hat{U}(D^{-1}_{k,4}) + l(S_{f_i}) + w(S_{f_i}) + \hat{U}(S_{f_i}) =$$

$$= \mathcal{O}(4 \cdot 2^k) + \mathcal{O}(16k^2 \cdot 2^k) + \mathcal{O}(2^{2k}) + \mathcal{O}(2^{2k}) + \mathcal{O}(2^{2k}) = \mathcal{O}(2^{2k}).$$

□

Лемма 7 (Основная лемма)**.** *Пусть дана булева функция $f(x_1, x_2, \ldots, x_n)$. Тогда существует объемная схема V^1_f со входами $z, x_1, x_2, \ldots, x_n$ на одном выходе которой на допустимых наборах $(z \geq x_1 \vee \ldots \vee x_n)$ реализуется функция $zf(x_1, x_2, \ldots, x_n)$, причём схема V^1_f обладает следующими характеристиками:*

1) $l(V^1_f) = \mathcal{O}(2^{n/3})$, $w(V^1_f) = \mathcal{O}(2^{n/3})$, $h(V^1_f) = \mathcal{O}(2^{n/3})$.

2) $\hat{U}(V^1_f) = \mathcal{O}(2^{n/3})$.

Доказательство. Покажем, что схема V^1_f (см. рис. 3) реализует функцию f согласно формуле (1).

Дешифратор D^1_{2k} реализует все элементарные конъюнкции

$$y_i = x^{\bar{i}_1}_{4k+1} x^{\bar{i}_2}_{4k+2} \ldots x^{\bar{i}_{2k}}_{6k},$$

причем при любом значении переменных ровно один выход будет активным, а остальные нет. Это означает, что среди блоков $S^1_{f_i}$ активным будет только один. Оставшиеся переменные $x_1, \ldots, x_{4k}$ отправляются на

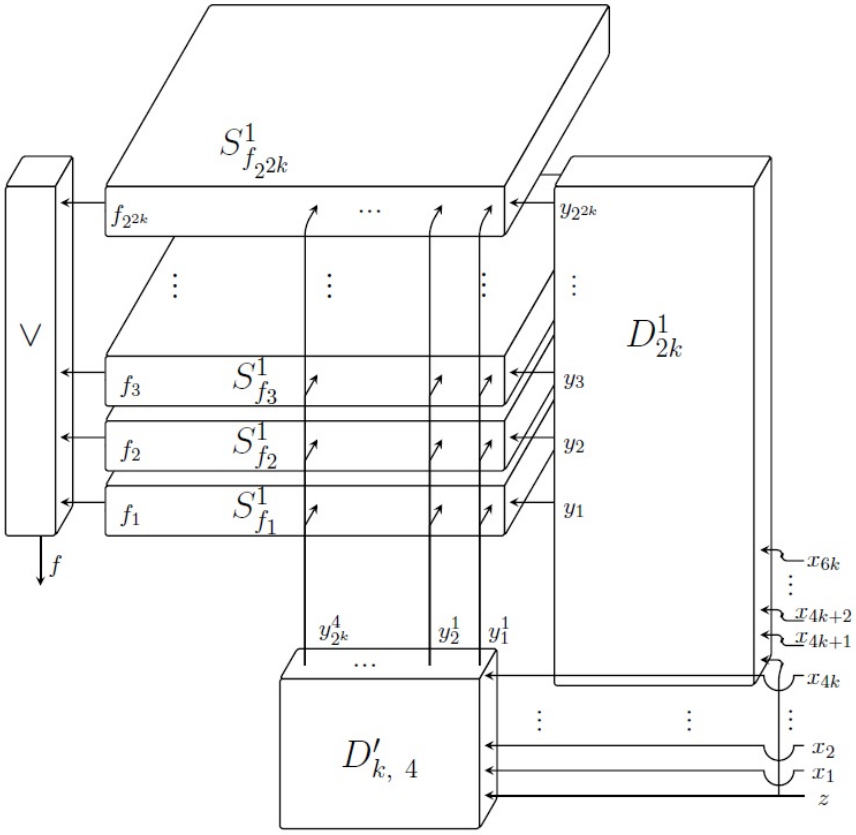


Рис. 3. Реализация основного блока V^1_f.

блок дешифраторов $D'_{k,4}$, где в «зашифрованном» виде отправляются на все блоки $S^1_{f_i}$. В каждом блоке $S^1_{f_i}$ они «расшифровываются», то есть преобразуются обратно в переменные $x_1, \ldots, x_{4k}$, после чего реализуется функция $f_i(x_1, \ldots, x_{4k})$. А так как управляющим входом в блок $S^1_{f_i}$ является $y_i = x^{\bar{i}_1}_{4k+1} x^{\bar{i}_2}_{4k+2} \ldots x^{\bar{i}_{2k}}_{6k}$, то фактически на выходе блока $S^1_{f_i}$ реализуется функция

$$x^{\bar{i}_1}_{4k+1} x^{\bar{i}_2}_{4k+2} \ldots x^{\bar{i}_{2k}}_{6k} f_i(x_1, \ldots, x_{4k}).$$

Далее берется дизъюнкция всех выходов блоков $S^1_{f_i}$, что полностью соответствует формуле (1).

Оценим параметры схемы V_f^1 в случае $n = 6k$.

$$l(V_f^1) = 1 + w(D_{2k}^1) + l(S_{f_i}^1) + 1 = \mathcal{O}(2^k) + \mathcal{O}(2^{2k}) = \mathcal{O}(2^{2k}).$$

$$w(V_f^1) = 1 + w(S_{f_i}^1) = \mathcal{O}(2^{2k}).$$

$$h(V_f^1) = w(D'_{k,4}) + l(D_{2k}^1) - 1 = \mathcal{O}(k^2) + \mathcal{O}(4k) + \mathcal{O}(2^{2k}) = \mathcal{O}(2^{2k}).$$

Оценим потенциал схемы. Входы $z, x_1, x_2, \ldots x_{4k}$ подводим к блоку $D'_{k,4}$. Эту часть схемы оцениваем через объем.

$$U_1 \leq 6 \cdot (4k + 1) \cdot (w(D_{2k}^1) + 1).$$

Оценим потенциал блока $D'_{k,4}$.

$$U_2 \leq \hat{U}(D'_{k,4}).$$

На выходах блока $D'_{k,4}$ будут активны ровно 4 провода, подводим их к блокам $S_{f_i}^1$ и оценим потенциал.

$$U_3 \leq 4 \cdot l(D_{2k}^1).$$

Подводим провода $z, x_{4k}, x_{4k+1}, \ldots, x_{6k}$ к дешифратору D_{2k}^1 и оценим потенциал.

$$U_4 \leq \mathcal{O}(2k).$$

Оценим потенциал дешифратора D_{2k}^1.

$$U_5 \leq \hat{U}(D_{2k}^1).$$

Так как среди выходов дешифратора D_{2k}^1 будет активным только один, и все его выходы будут подключены к управляющим входам блоков $S_{f_i}^1$, то только 1 из блоков будет активен, а остальные $2^{2k} - 1$ будут иметь потенциал 4.

$$U_6 \leq 4 \cdot (2^{2k} - 1) + \hat{U}(S_{f_i}^1).$$

Общую дизъюнкцию всех выходов $S_{f_i}^1$ оценим через объем.

$$U_7 \leq 6 \cdot \mathcal{O}(2^{2k}).$$

В итоге, имеем следующую оценку потенциала схемы V_f^1:

$$\hat{U}(V_f^1) = U_1 + U_2 + U_3 + U_4 + U_5 + U_6 + U_7 \leq 6 \cdot (4k+1) \cdot (w(D_{2k}^1) + 1) + \hat{U}(D'_{k,4}) +$$

$$+4\cdot l(D^1_{2k})+\mathcal{O}(2k)+\hat{U}(D^1_{2k})+4\cdot(2^{2k}-1)+\hat{U}(S^1_{f_i})+6\cdot\mathcal{O}(2^{2k})=\mathcal{O}(k\cdot 2^k)+$$
$$+\mathcal{O}(4k^2\cdot 2^k)+\mathcal{O}(16k\cdot 2^k)+\mathcal{O}(2k)+\mathcal{O}(2^{2k})+8\mathcal{O}(2^{2k})+\mathcal{O}(2^{2k})+\mathcal{O}(2^{2k})=$$
$$=\mathcal{O}(2^{2k}).$$

Таким образом, получаем верное утверждение теоремы в случае $n = 6k$. Если же $n = 6k + r$, где $r = 1 \dots 5$, то построим схему для $n = 6k + 6$ и на последние $6 - r$ входов подадим константу 0. Заметим, что в данном случае получим искомую схему и константы в оценках увеличатся не более, чем в 4 раза, а значит оценки по порядку останутся верными. □

В качестве следствия докажем основную теорему.

Теорема 2 (Основная теорема)**.** *Пусть дана булева функция $f(x_1, x_2, \dots, x_n)$. Тогда существует объемная схема V_f со входами $x_1, x_2, \dots, x_n$ на одном выходе которой реализуется функция $f(x_1, x_2, \dots, x_n)$, причём схема V_f обладает следующими характеристиками:*

1) $l(V_f) = \mathcal{O}(2^{n/3}),\ w(V_f) = \mathcal{O}(2^{n/3}),\ h(V_f) = \mathcal{O}(2^{n/3})$.

2) $\hat{U}(V_f) = \mathcal{O}(2^{n/3})$.

Доказательство. Подадим в схеме V^1_f на вход z константу 1. Полученная таким образом схема V_f реализует функцию $f(x_1, x_2, \dots, x_n)$ на всех наборах $x_1, x_2, \dots, x_n$ и её параметры остаются такими же по порядку, как и у схемы V^1_f. □

Список литературы

[1] Кравцов С. С. О реализации функций алгебры логики в одном классе схем из функциональных и коммутационных элементов. // Проблемы кибернетики. Вып. 19. — Наука, М., 1967. — С. 285–293.

[2] Калачев Г. В. Порядок мощности плоских схем, реализующих булевы функции // Дискретная математика. — 2014. — Т. 26, № 1. — С. 49–74.

[3] Калачев Г. В. Об одновременной минимизации площади, мощности и глубины плоских схем, реализующих частичные булевы операторы // Интеллектуальные системы. Теория и приложения (ранее: Интеллектуальные системы по 2014, № 2, ISSN 2075-9460). — 2016. — Т. 20, № 2. — С. 203–266.

The top assessment of energy consumption in a class of volume schemes

Efimov A.A. In this work volume schemes which are generalization of plane schemes in space are considered. The class of the schemes implementing boolean functions was considered. For this class upper assessment of potential — a measure of the power equal to quantity of the circuit elements giving unit on this input pattern is received. It is shown that any function from n of variables can be implemented the volume scheme which potential does not exceed $\mathcal{O}(2^{n/3})$.

Keywords: schemes from functional elements, volume schemes, scheme power, potential.

Условие корректности и полноты классической логики для семантики относительной V-реализуемости

Коновалов А. Ю.

Пусть L — некоторое расширение языка арифметики, V — некоторый класс числовых функций. Определяется понятие V-реализуемости для предикатных формул, основанное на оценке предикатных переменных формулами языка L. Устанавливается корректность и полнота классической логики относительно семантики V-реализуемости в случае, когда класс V содержит все функции, определимые в языке L.

Ключевые слова: конструктивная семантика, реализуемость, обобщенная реализуемость, формальная арифметика.

Пусть V — некоторое множество частичных функций натурального аргумента. Элементы множества V назовем V-функциями. Будем считать, что для каждого натурального числа n имеется нумерация всех n-местных V-функций. А именно, определено множество индексов $\mathsf{I}_n^V \subseteq \mathbb{N}$ вместе с отображением, которое каждому натуральному числу $z \in \mathsf{I}_n^V$ ставит в соответствие n-местную V-функцию $\varphi_z^{V,n}$, и при этом всякая n-местная V-функция есть $\varphi_z^{V,n}$ для некоторого $z \in \mathsf{I}_n^V$. Будем считать, что множество V вместе с вышеописанной нумерацией обладает следующими свойствами:

C1) V содержит все частично-рекурсивные функции;

C2) если ψ есть n-местная V-функция, s — перестановка на множестве $\{1,\ldots,n\}$, то функция ψ', определенная условным равенством $\psi'(x_1,\ldots,x_n) \simeq \psi(x_{s(1)},\ldots,x_{s(n)})$, является V-функцией;

C3) если ψ есть n-местная V-функция, то функция ψ', определенная условным равенством

$$\psi'(x_1,\ldots,x_n,x_{n+1}) \simeq \psi(x_1,\ldots,x_n),$$

является V-функцией;

C4) композиция V-функций есть V-функция;

C5) если ψ_1, ψ_2 суть $(n+1)$-местные V-функции, то функция ψ', определенная условным равенством

$$\psi'(x_1,\ldots,x_n) \simeq \mu x\,[\psi_1(x_1,\ldots,x_n,x) = \psi_2(x_1,\ldots,x_n,x)],$$

является V-функцией (μ — оператор минимизации);

C6) если ψ_1, ψ_2 суть n-местные V-функции, χ — всюду определенная n-местная V-функция, то функция ψ', определенная условным равенством

$$\psi'(x_1,\ldots,x_n) \simeq \begin{cases} {}_1(x_1,\ldots,x_n), & \text{если } \chi(x_1,\ldots,x_n)=1; \\ {}_2(x_1,\ldots,x_n), & \text{иначе,} \end{cases}$$

является V-функцией;

C7) для каждой $(n+m)$-местной V-функции ψ найдется всюду определенная m-местная V-функция ψ', что справедливо условное равенство

$$\varphi^{V,\,n}_{\psi'(x_{n+1},\ldots,x_{n+m})}(x_1,\ldots,x_n) \simeq \psi(x_1,\ldots,x_n,x_{n+1},\ldots,x_{n+m}).$$

Множество всех частично-рекурсивных функций обладает свойствами C1–C7. Другим примером функций, обладающих свойствами C1–C7, могут служить все арифметические функции или все гиперарифметические функции с подходящей нумерацией (см. [1], [2]).

Будем считать, что язык формальной арифметики LA содержит обозначения для всех примитивно рекурсивных функций, константы для обозначения всех натуральных чисел, а также логические константы $\top$ (истина) и $\bot$ (ложь), которые считаются атомарными формулами (атомами). Расширение LA' языка LA получается добавлением к LA предикатных символов P_i^n и функциональных символов f_i^n для всех $i \geq 0,\ n \geq 1$. Валентность символов P_i^n и f_i^n полагается равной n. Формулы языка LA' строятся из атомов при помощи логических связок $\wedge$, $\vee$, $\to$ и кванторов $\exists$, $\forall$, причем квантор $\forall$ используется следующим образом: если A и B — формулы, $\overline{x} = x_1,\ldots,x_n$ — список переменных, то выражение $\forall\overline{x}\,(A \to B)$ считается формулой. Такое определение формулы навеяно идеями из базисной арифметики (см. [3]). Выражение $\neg A$ условимся рассматривать как сокращение для формулы $A \to \bot$. Выражение $A(x_1,\ldots,x_n)$ означает, что все свободные переменные формулы A находятся в списке $x_1,\ldots,x_n$. Будем считать, что фиксированы расширение L языка LA и интерпретация $\mathcal{N}_L$ языка L такие, что L — подъязык

языка LA', и интерпретация $\mathcal{N}_L$ является продолжением стандартной интерпретацией языка LA. Заметим, что при этом $\mathcal{N}_L \models \top$ и $\mathcal{N}_L \not\models \bot$.

Понятие V-реализуемости для языка L определим по аналогии с рекурсивной реализуемостью Клини [4, §82].

Пусть фиксированы примитивно-рекурсивные двухместная функция c, которая взаимно однозначно нумерует все пары натуральных чисел, и одноместные обратные функции p_1 и p_2, так что выполняются соотношения $\mathsf{p}_1(\mathsf{c}(x,y)) = x$ и $\mathsf{p}_2(\mathsf{c}(x,y)) = y$. В выражениях вида $\mathsf{p}_1(t),\ \mathsf{p}_2(t)$ обычно будем опускать скобки.

Для каждого натурального числа e и произвольной замкнутой формулы Φ языка L определим отношение $e\,\mathbf{r}^V\,\Phi$ (e V-реализует Φ) индукцией по построению формулы Φ:

1) $e\,\mathbf{r}^V\,\Phi \rightleftharpoons \mathcal{N}_L \models \Phi$, если Φ — атом языка L;

2) $e\,\mathbf{r}^V\,(\Phi \wedge \Psi) \rightleftharpoons \mathsf{p}_1 e\,\mathbf{r}^V\,\Phi$ и $\mathsf{p}_2 e\,\mathbf{r}^V\,\Psi$;

3) $e\,\mathbf{r}^V\,(\Phi \vee \Psi) \rightleftharpoons (\mathsf{p}_1 e = 0$ и $\mathsf{p}_2 e\,\mathbf{r}^V\,\Phi)$ или $(\mathsf{p}_1 e = 1$ и $\mathsf{p}_2 e\,\mathbf{r}^V\,\Psi)$;

4) $e\,\mathbf{r}^V\,\exists x\,\Phi(x) \rightleftharpoons \mathsf{p}_2 e\,\mathbf{r}^V\,\Phi(\mathsf{p}_1 e)$;

5) $e\,\mathbf{r}^V\,\forall x_1,\ldots,x_n\,(\Phi(x_1,\ldots,x_n) \to \Psi(x_1,\ldots,x_n)) \rightleftharpoons e \in \mathsf{I}^V_{n+1}$ и для всех[1] натуральных чисел s, $a_1,\ldots,a_n$, если верно $s\,\mathbf{r}^V\,\Phi(a_1,\ldots,a_n)$, то определено $\varphi_e^{V,n+1}(a_1,\ldots,a_n,s)$ и $\varphi_e^{V,n+1}(a_1,\ldots,a_n,s)\,\mathbf{r}^V\,\Psi(a_1,\ldots,a_n)$.

Замкнутую формулу Φ языка L назовем *V-реализуемой* (обозначение: $\mathbf{r}^V\,\Phi$), если найдется такое натуральное число e, что $e\,\mathbf{r}^V\,\Phi$.

Предикатные формулы строятся обычным образом из атомов $P(v_1,\ldots,v_n)$, где P есть n-местная предикатная переменная, а $v_1,\ldots,v_n$ — предметные переменные, при помощи логических констант $\top$, $\bot$, связок $\wedge$, $\vee$, $\to$ и кванторов $\forall$, $\exists$.

Будем говорить, что замкнутая предикатная формула является *V-реализуемой относительно языка L* (обозначение: $\mathbf{r}^V_L\,\Phi$), если любой ее замкнутый L-пример оказывается V-реализуемым.

Семантики предикатных формул, основанные на понятии V-реализуемости для некоторых конкретных классов V, рассматривались в работах [1] и [2]. Там исследовались соотношения таких семантик с базисной и интуиционисткой логикой. Сейчас мы установим критерий совпадения семантик, основанных на понятии V-реализуемости, с классической логикой.

[1] Однако, если в списке $x_1,\ldots,x_n$ на некоторых позициях i и j стоят одинаковые переменные x_i и x_j, то мы не допускаем рассмотрение тех списков $a_1,\ldots,a_n$, в которых $a_i \neq a_j$.

Будем говорить, что n-местная частичная функция ψ определима в языке L формулой $\Phi(x_1, \ldots, x_n, y)$ этого языка, если имеет место

$$(k_1, \ldots, k_n) = k \Longleftrightarrow \mathcal{N}_L \models \Phi(k_1, \ldots, k_n, k)$$

для всех натуральных чисел $k_1, \ldots, k_n, k$. Множество всех функций, определимых в языке L, обозначим $\mathsf{F}(L)$.

Через CPC обозначим классическое исчисление предикатов. Верны следующие теоремы.

Теорема 1. *Пусть $V \supseteq \mathsf{F}(L)$. Тогда имеет место*

$$\mathbf{r}_L^V A \Longleftrightarrow CPC \vdash A$$

для всех замкнутых предикатных формул A.

Теорема 2. *Пусть $V \not\supseteq \mathsf{F}(L)$. Тогда формула $\forall z\,(P(z) \vee \neg P(z))$ не является V-реализуемой относительно языка L.*

Список литературы

[1] Коновалов А. Ю., Плиско В. Е. О гиперарифметической реализуемости // Мат. зам. 2015. **98**, №5. 725—746.

[2] Коновалов А. Ю. Арифметическая реализуемость и базисная логика // Вестн. Моск. ун-та. Матем. Механ. 2016. №1. 52–56.

[3] Provably total functions of basic arithmetic // Math. Log. Quart. 2003. **49**. N 3. 316—322.

[4] Клини С. К. Введение в метаматематику. М.: ИЛ, 1957.

The criterion of the soundness and the comleteness of the classical logic with respect to the V-realizability.

Konovalov A. Yu.

Let L be an extension of the language of arithmetic, V a class of number-theoretical functions. A notion of the V-realizability for predicate formulas is defined in such a way that predicate variables are substituted by formulas of the language L. It is proved that the classical logic is sound and complete with respect to the semantics of the V-realizability if V contains all L-definable functions.

Keywords: constructive semantics, realizability, generalized realizability, formal arithmetic.

О классификациях базисов в P_k по разрешимости полноты для автоматов

Кудрявцев В.Б., Бабин Д.Н.

Рассматривается проблема полноты систем автоматных функций с операциями суперпозиции и обратной связи вида $\Phi \cup \nu$, где $\Phi \subseteq P_k$, ν - конечно. При $k = 2$ решение этой задачи приводит к разделению решётки замкнутых классов Поста на сильные (наличие которых в исследуемой системе гарантирует разрешимость задачи полноты конечных базисов) и слабые (наличие которых в исследуемой системе этого не гарантирует). При $k = 2$ эта задача для систем автоматных функций произвольного вида была решена (Бабин Д.Н. 1998). В статье рассмотрены следствия и возможные обощения этой задачи, а также некоторые результаты для $P_k, k > 2$.
Ключевые слова: булева функция, конечный автомат, алгоритмическая разрешимосить.

1. Введение

Первый толчок к возникновению теории автоматов дала работа Э. Поста 1921 года [1]. В ней были получены фундаментальные результаты о строении решетки замкнутых классов булевых функций, которые были в дальнейшем методически переработаны и упрощены в книге Яблонского С.В., Гаврилова Г.П., Кудрявцева В.Б. "Функции алгебры логики и классы Поста"[2]. Последующие работы по изучению алгебр автоматов велись под большим влиянием известных статей А.В. Кузнецова [3] и С.В. Яблонского [4] по теории функций k-значной логики.

Функции k-значной логики P_k могут рассматриваться как автоматы без памяти, к которым применяются операции суперпозиции. Возникшие для таких функций постановки задач о выразимости, полноте, базисах, решетке замкнутых классов и другие оказались весьма действенными и для алгебр автоматных функций.

Основу результатов для функций из P_k составляет подход, опирающийся на понятие предполного класса. Для конечно-порожденных си-

стем таких функций семейство предполных классов образует критериальную систему. Множество этих предполных классов оказалось конечным и из их описания вытекает алгоритмическая разрешимость задачи о полноте. На этом пути С.В. Яблонским путем явного описания всех предполных классов была решена задача о полноте для функций трехзначной логики. После усилий многих исследователей при $k > 3$ в P_k были описаны все семейства предполных классов. Заключительные построения в этой задаче провел Розенберг [5].

Одновременно с изучением функций без памяти, были сделаны попытки применения аппарата предполных классов в задаче полноты для автоматов. Сначала для автоматов без обратных связей, называемых функциями с задержками была эффективно решена задача о полноте и ее естественных модификациях [6]. После этого было проведено рассмотрение общего случая и на этом пути был получен фундаментальный результат негативного характера, который показал континуальность множества предполных классов автоматных функций [7]. В дальнейшем, Кратко М.И. была показана алгоритмическая неразрешимость задачи о полноте для автоматных функций [8].

Еще Слупецким [9] была решена задача о полноте в P_k для систем, содержащих все одноместные функции. Для автоматов в 1961 А.А. Летичевским [10] был получен алгоритм решения задачи о полноте для конечных систем автоматов, выдающих номер своего состояния (автоматы Медведева), при наличии в исследуемой системе всех булевых функций. В 1986 В.А. Буевич [11] показал алгоритмическую разрешимость задачи А-полноты для конечных систем автоматов, содержащих все булевы функции. В 1992 г. было показано [12], что существует алгоритм распознавания полноты при наличии в рассматриваемой системе автоматов всех булевых функций.

В этой ситуации было предложено использовать разрешимость автоматной полноты как инструмент для исследования базисов функций, а именно, исследовать на полноту (A-полноту) системы вида $\Phi \cup \nu$, где Φ — замкнутый класс функций из P_k (его конечный базис), а ν - конечная система автоматных функций. Была построена классификация базисов в P_2 по их способности гарантировать разрешимость полноты конечных систем автоматов. Оказалось, что класс является сильным, точно тогда, когда в классе Φ содержится функция $[x \oplus y \oplus z] = L_4$, либо функция $[xy \cup xz \cup yz] = D_2$ [15] (смотри рисунок 1).

Если выбирать множество автоматных функций ν из дефинитных автоматов, то задача полноты для системы $\Phi \cup \nu$ разрешима точно то-

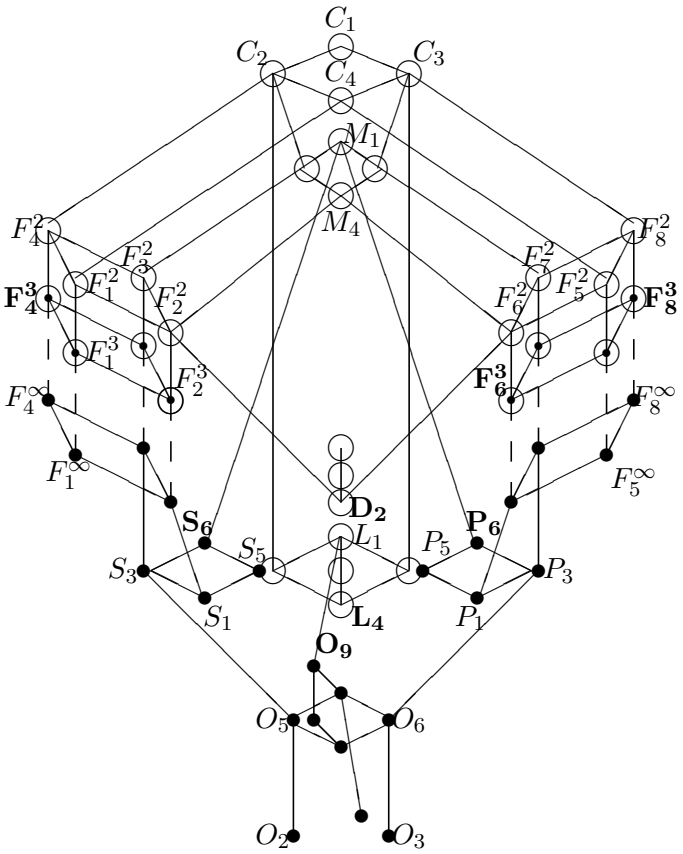


Рис.1 Белые кружки - сильные классы Поста,
черные кружки - слабые классы Поста.

гда, когда в классе Φ содержится функция $x \oplus y \oplus z$, либо функция $xy \cup xz \cup yz$, либо функции порождающие класс F_2^3, либо функции порождающие класс F_6^3 [16]. Заметим, что граница сильных классов в этом случае строго понижается (смотри рисунок 1). Оказывается, что (возможно не строгое) понижение границы сильных классов будет происходить при переходе к системам $\nu \subseteq M \subseteq P$. То есть, свойство классов быть сильными сохраняется при сужении множества $M \supseteq \nu$.

2. Обозначения и теоремы.

Пусть $E_k = \{0, 1, \dots k-1\}$, $g\colon E_k^n \to E_k^m$ вектор-функции, их множество обозначается через P_k. Пусть

$$E_k^\infty = \{a(1)a(2)\dots | a(j) \in E_k, j = 1, 2, \dots\}$$

— множество всех сверхслов, а

$$E_k^\tau = \{a(1)a(2)\dots a(\tau)|a(j) \in E_k, j = 1, 2, \dots \tau\}$$

— множество всех слов длины τ. Пусть

$$f\colon (E_k^\infty)^n \to (E_k^\infty)^m$$

— автоматная функция (a-функция), т.е. она задается рекуррентно соотношениями

$$\begin{cases} q(1) = q_1, \\ q(t+1) = \varphi(q(t), a_1(t), ..., a_n(t)), \\ b_j(t) = \psi_j(q(t), a_1(t), ..., a_n(t)), \quad j = 1, ..., m. \end{cases}$$

где $q \in Q = \{q_1, \dots, q_r\}$. Параметр q называется состоянием a-функции f, q_1 — ее начальным состоянием, вектор-буквы $a = (a_1, \dots, a_n)$ и $b = (b_1, \dots, b_m)$ называются входной и выходной буквами, а сверхслова $a(1)a(2)...$ и $b(1)b(2)...$ — входным и выходным сверхсловами, соответственно. Класс всех a-функций обозначим через P. В этом классе обычным образом введем операции суперпозиции и обратной связи. Пусть $\nu \subseteq P$, обозначим через $[\nu]$ множество всех a-функций, получающихся из ν с помощью операций суперпозиции и обратной связи. Множество ν называется полным, если $[\nu] = P$. Проблема полноты для P состоит в описании всех полных множеств ν.

Пусть τ — натуральное число, $f(x_1, \dots, x_n)$ — некоторая автоматная функция,

$$f^\tau\colon (E_k^\tau)^n \to (E_k^\tau)^m$$

— ограничение этой функции на множество слов длины τ. Скажем, что a-функции $f(x_1, \dots, x_n)$ и $g(x_1, \dots, x_n)$ τ-равны, если $f^\tau = g^\tau$. Обозначим через $[\nu]_\tau$ множество всех a-функций, τ-равных получающимся из ν с помощью операций суперпозиции и обратной связи, а через

$$[\nu]_A = \bigcap_{\tau=1}^{\infty} [\nu]_\tau.$$

Известно [11], что результат применения о. с. τ-равен τ применениям суперпозиции. Множество ν называется τ-полным, если $[\nu]_\tau = P$. Множество ν называется A-полным, если $[\nu]_\tau = P$ при всех τ. Проблема A-полноты для P состоит в описании всех A-полных множеств ν. Очевидно, что полное множество ν является A-полным.

Пусть $P^{(i)}$ — класс всех a-функций с не более, чем i состояниями, тогда

$$P = \bigcup_{i=1}^{\infty} P^{(i)}.$$

Известно [14], что

$$[P^{(2)}] = P, \qquad [P^{(1)}] = P^{(1)}.$$

A-функции из $P^{(1)}$ называются истинностными и мы будем отождествлять $P^{(1)}$ с множеством P_k функций k- значной логики, полагая

$$f(a(1)a(2)\dots) = f(a(1))f(a(2))\dots,$$

где $f\colon (E_k)^n \to (E_k)^m, f \in P_k$.

Множество функций $f\colon (E_k)^n \to E_k$, для которых выполнено одно из свойств: $n = 1$ или f не принимает всех значений из E_k, называется классом Слупецкого. Известно, что класс Слупецкого замкнут относительно суперпозиции, т.е. результат применения суперпозиции к функциям из этого класса являются функцией из этого же класса. Будем обозначать класс Слупецкого через $SLUP$.

Для $l \in \{0, 1, \dots, k-1\}$ функция $f \in P_k$, такая что $f(l, l, \dots, l) = l$ называется сохраняющей константу l , а множество всех таких функций - классом сохранения константы l, оно обозначается через U_l . Обозначим через

$$U = \bigcap_{i=0}^{k-1} U_l$$

класс сохранения всех констант. Известно, что классы $U, U_l, l = 1, 2, \dots, k-1$ замкнуты относительно суперпозиции, т.е. результат применения суперпозиции к функциям из этих классов являются функциями из этих же классов.

Функция

$$\mathbf{w}(x, y) = max(x, y) + 1 \ (mod\ k)$$

называется функцией Вебба. Известно, что $P_k = [\{\mathbf{w}\}]$, т.е. функция $\mathbf{w}$ образует полную систему в классе k -значных функций.

Автоматная функция $B\colon E_k^{\infty} \to E_k^{\infty}$, задаваемая уравнениями

$$\begin{cases} q(1) = k-1, \\ q(t+1) = x(t), \\ b(t) = q(t), \end{cases}$$

называется a-функцией задержки. Про нее известно [14], что

$$[\{B\} \cup \{\mathbf{w}\}] = P.$$

Дефинитным называется автомат, для которого найдётся натуральное t, что каждое входное слово длины t переводит автомат из любого состояния в одно и то же состояние, зависящего от этого входного слова.

Утверждение 1. *Пусть $F_1 \subseteq F \subseteq P_k$. Если не существует алгоритма, по конечному множеству $\nu \subseteq P$ решающего вопрос о полноте (A-полноте) множества $F \cup \nu$, то не существует и алгоритма, решающего вопрос о полноте (A-полноте) множества $F_1 \cup \nu$.*

Утверждение 2. *Пусть $F_1 \subseteq P_k$, а $F_2 \subseteq P_k$ двойственный к нему класс. Если не существует алгоритма, по конечному множеству $\nu \subseteq P$ решающего вопрос о полноте (A-полноте) множества $F_1 \cup \nu$, то не существует и алгоритма, решающего вопрос о полноте (A-полноте) множества $F_2 \cup \nu$.*

Утверждение 3. *Пусть $M \subseteq P$ некоторое подмножество автоматных функций. Если существует алгоритм, по конечному множеству $\nu \subseteq P$ решающий вопрос о полноте (A-полноте) множества $F \cup \nu$, то существует и алгоритм, решающий вопрос о полноте (A-полноте) множества $F \cup \mu$ по конечному множеству $\mu \subseteq M$.*

Несмотря на то, что Утверждение 3 кажется очевидным, это мощный инструмент решения проблемы разрешимости полноты при фиксированных автоматных добавках к базису. Имеет место

Следствие 1. *Пусть F_0, F_1 конечные подмножества автоматных функций и $[F_0] \supseteq F_1$, тогда из разрешимости задачи полноты систем $F_1 \cup \nu$ следует разрешимость полноты систем $F_0 \cup \nu$.*

Приведенные утверждения весьма полезны, потому что автоматически создают теоремы о разрешимости полноты для конкретных систем автоматов.

Имеют место следующие утверждения.

Теорема 1. [15]

Проблема полноты (А-полноты) системы $\Phi \cup \nu, \Phi \subseteq P_2$ разрешима точно тогда, когда функция $x \oplus y \oplus z \in \Phi$, либо функция $xy \cup xz \cup yz \in \Phi$.

Теорема 2. [16]

Проблема A-полноты системы $\Phi \cup \nu, \Phi \subseteq P_2$, где ν состоит из дефинитных автоматов, разрешима точно тогда, когда функция $x \oplus y \oplus z \in \Phi$, либо функция $xy \cup xz \cup yz \in \Phi$, либо $(F_2)^3 \subseteq \Phi$, либо $(F_6)^3 \subseteq \Phi$.

Согласно теореме 2 граница между сильными и слабыми классами решётки Поста опустилась по сравнению с границей, опеределяемой теоремой 1 (смотри рисунок). Известно [13], что для систем линейных автоматов (независимо от булевой части базиса) существует алгоритм определения полноты конечных систем. Этот факт подтверждает вывод из утверждения 3 и теоремы 1 о разрешимости полноты систем линейных автоматов при дополнительных условиях вхождения в них сильных булевых функций. Наконец, если системы автоматов слишком бедны, чтобы быть полными при наличии любых булевых функций, такими, например, являются автоматы с безусловными переходами, тогда задача полноты алгоритмически разрешима. Этот факт также подтверждает вывод из утверждения 3 и теоремы 1 о разрешимости полноты систем автоматов с безусловными переходами при дополнительных условиях вхождения в них сильных булевых функций. Имеют место

Теорема 3.[17]

Пусть $\Phi \subseteq SLUP$, не существует алгоритма, по конечному множеству $\nu \subseteq P$ решающего вопрос, верно ли, что $[\Phi \cup \nu] = P$.

Теорема 4.[17]

Пусть $\Phi \subseteq SLUP$, не существует алгоритма, по конечному множеству $\nu \subseteq P$ решающего вопрос, верно ли, что $[\Phi \cup \nu]_A = P$.

Теорема 5.[17] Для любого $\Phi \supseteq U$ существует алгоритм, по конечному множеству $\nu \subseteq P$ решающий, верно ли, что $[\Phi \cup \nu] = P$.

Теорема 6.[17] Для любого $\Phi \supseteq U$ существует алгоритм, по конечному множеству $\nu \subseteq P$ решающий, верно ли, что $[\Phi \cup \nu]_A = P$.

Из утверждений следует, что достаточно доказать теоремы 3-6 только для случая $\Phi = SLUP, \Phi = U$.

Список литературы

[1] Post E. Two-valued iterative systems of math logik. Printston 1941

[2] Яблонский С.В., Гаврилов Г.П., Кудрявцев В.Б. Функции алгебры логики и классы Поста М. Наука 1966

[3] Кузнецов А.В. О проблемах тождества и функциональной полноты для алгебраических систем. Труды третьего всесоюзного математического съезда/. 2 1956, 45–146, М. Изд. АН СССР.

[4] Яблонский С.В. Функциональные построения в k-значной логике. Труды Матем. ин-та им. В.А. Стеклова. 51,1958, 5–142., М. Изд. АН СССР.

[5] Rosenberg J. La structure des fonctions de plusiers variables sur un ensemble fini. Comptes Rendus Acad. 1965. 260, 3817–3819, Sci. Paris

[6] Кудрявцев В.Б. Теорема полноты для одного класса автоматов без обратных связей. Проблемы кибернетики. 1962, 8, 91–115, М,. Физматгиз.

[7] Кудрявцев В.Б., О мощностях множеств предполных классов некоторых функциональных систем, связанных с автоматами. ДАН СССР, 1963, 151, 3, 493–496, М. Изд. АН СССР

[8] Кратко М.И., Алгоритмическая неразрешимость проблемы распознавания полноты для конечных автоматов, ДАН СССР, 1964, 155, 1, 35–37, М. Изд. АН СССР.

[9] Slupecki J. Kriterium pelnosci wielowartosciowych systemow logiki zdan, Comptes Rendus des Seances de la Societe des Sciences et des Lottres de Varsovie, 1939, 32, 102–128, Cl. III

[10] Летичевский А.А. Условия полноты для конечных автоматов, Вычислительная математика и математическая физика, 1961, 4, 702–710

[11] Буевич В.А., Условия А-полноты для автоматов, изд. МГУ, 1986

[12] Бабин Д.Н., Разрешимый случай задачи о полноте автоматных функций, Дискретная математика, 1992, 4, 4, 41–56, М. Наука

[13] Часовских А.А., О полноте в классе линейных автоматов, Математическме вопросы кибернетики, 1995, 3, 140–166.

[14] Кудрявцев В.Б., Алешин С.В., Подколзин А.С., Введение в теорию автоматов, 1985, М. Наука.

[15] Бабин Д.Н., О классификации автоматных базисов Поста по разрешимости свойств полноты и А-полноты, ДОКЛАДЫ АКАДЕМИИ НАУК,367, 4,1999, 439–441.

[16] Жук Д.Н., О классификации автоматных базисов Поста по разрешимости свойств А-полноты для дефинитных автоматов, Дискретная математика, 22, 2, 2010, 80–95.

[17] Бабин Д.Н., О КЛАССИФИКАЦИИ БАЗИСОВ В P_k ПО РАЗРЕШИМОСТИ ПРОБЛЕМЫ ПОЛНОТЫ ДЛЯ АВТОМАТОВ, Фундаментальная и прикладная математика, 15, 3, 2010, 33–47, М. Интуит.

Classification of bases in P_k by the property of decidability of the completeness for automata.

Kudryavtsev V.B., Babin D.N.

We consider the problem of the completeness of systems of automaton functions with operations of superposition and feedback of the form $\Phi \cup \nu$, where $\Phi \subseteq P_k$, ν is finite. For $k = 2$, the solution of this problem leads to the separation of the lattice of closed Post classes into strong ones (whose presence in the system under study guarantees the solvability of the completeness problem of finite bases) and weak ones (which does not guarantee this in the system under study). For $k = 2$ this problem was solved for systems of automaton functions of arbitrary form (Babin DN 1998). In this paper, we investigate corollaries and possible extensions of this problem, as well as some results for $P_k, k > 2$.

Keywords: Boolean function, finite automaton, algorithmic solvalibility of functions by formulas.

К сведению авторов публикаций в журнале «Интеллектуальные системы. Теория и приложения»

В соответствии с требованиями ВАК РФ к изданиям, входящим в перечень ведущих рецензируемых научных журналов и изданий, в которых могут быть опубликованы основные научные результаты диссертаций на соискание ученой степени доктора и кандидата наук, статьи в журнал «Интеллектуальные системы. Теория и приложения» предоставляются авторами в следующей форме:

1. Статьи, набранные в пакете LaTeX, предоставляются к загрузке через WEB-форму http://intsysjournal.org/generator_form .

2. К статье прилагаются файлы, содержащие название статьи на русском и английском языках, аннотацию на русском и английском языках (не более 50 слов), список ключевых слов на русском и английском языках (не более 20 слов), информация об авторах: Ф.И.О. полностью, место работы, должность, ученая степень и/или звание (если имеется), контактные телефоны (с кодом города и страны), e-mail, почтовый адрес с индексом города (домашний или служебный).

3. Список литературы оформляется в едином формате, установленном системой Российского индекса научного цитирования.

4. За публикацию статей в журнале «Интеллектуальные системы. Теория и приложения» с авторов (в том числе аспирантов высших учебных заведений) статей, рекомендованных к публикации, плата не взимается. Оттиски статей авторам не предоставляются. Журнал распространяется по подписке, экземпляры журнала рассылаются подписчикам наложенным платежом. Условия подписки публикуются в каталоге НТИ «Роспечать», индекс журнала 64559.

5. Доступ к электронной версии последнего вышедшего номера осуществляется через НЭБ «Российский индекс научного цитирования». Номера, вышедшие ранее, размещаются на сайте http://intsysjournal.org, и доступ к ним бесплатный. Там же будут размещены аннотации всех публикуемых статей.

Подписано в печать: 20.03.2019
Дата выхода: 28.03.2019
Тираж: 200 зкз.
Цена свободная
Свидетельство о регистрации СМИ: ПИ № ФС77–58444 от 25 июня 2014 г., выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций(Роскомнадзор).