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

Интеллектуальные Системы.

Теория и приложения

ТОМ **26** ВЫПУСК 2 * **2022**

МОСКВА

ISSN 2411–4448

УДК 519.95; 007:159.955
ББК 32.81

Издается с 1996 г.*

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 имени М. В. Ломоносова, Научного Совета по комплексной проблеме «Кибернетика» РАН, Отделения «Математическое моделирование технологических процессов» МАТН, 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: 119991, Москва, ГСП-1, Ленинские Горы, д. 1, 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».

ОГЛАВЛЕНИЕ

Часть 1.
Общие проблемы теории интеллектуальных систем

Когнитивный уровень искусственного интеллекта

А. В. Чечкин[1]

Цель. Обсуждение когнитивного уровня и обосновать основной принцип искусственного интеллекта умных систем, человекоцентричность — подчинение и служение человеку (хозяину).

Методы. Подчеркивается центральная роль радикальной (избыточной) модели театра действий умной системы как ядра искусственного интеллекта. Обсуждаются этапы создания ультрасистемы обеспечения информационно-системной безопасности умной системы, ее интеллектуального планирования действий и ситуационного управления поведением в театре ее действий, включая развитие самой системы и разумное преобразование театра ее действий.

Результаты. Выделяются пять основных уровней развития искусственного интеллекта. Изучаются особенности первой и второй сигнальных систем (первичного и языкового сенсориумов). Выделяется информационная и системная нештатности умной системы как точки роста ее интеллектуального развития на базе машинного обучения. Исследуется коллективный искусственный интеллект группы умных систем в рамках их единого театра действий и подчинения человеку–хозяину группировки. Обосновывается принцип человеко-центрического целеполагания любой технической умной системы, определяемый безоговорочному подчинению ее своему человеку–хозяину. Обсуждаются соответствующие требования к ультрасистеме и ее языковой операционной системе интеллектуального планирования и ситуационного управления поведением отдельной умной системы или коллектива умных систем. Рассматриваются проблемы развития отдельной умной системы, реинжениринга коллектива таких систем и проблема разумного преобразовния театра действий такого коллектива.

Ключевые слова: когнитивность естественного интеллекта, автоматизированная умная система, коллективный искусственный интеллект, радикал, язык, интернет вещей, интерфейс, информационно-системная безопасность, тактическое и оперативное планирование действий умной системы.

[1] *Чечкин Александр Витальевич* — доктор физ.-мат. наук, профессор, Военная академия РВСН имени Петра Великого & Финансовый университет при правительстве Российской Федерации e-mail: a.chechkin@mail.ru

Chechkin Alexander Vitalievich — Doctor of Phys.-Math. Sci., Professor, Strategic Missile Forces Military Academy named after Peter the Great & Financial University under the Government of the Russian Federation.

1. Введение. Роль рефлексии в искусственном интеллекте

В статье [1] был выдвинут и обоснован тезис о необходимом и достаточном условии наличия искусственного интеллекта (ИИ) в отдельной умной системе (УС). В тезисе подчеркивалось, что ***для наличия ИИ в УС*** требуется присутствие следующих двух частей: 1) ***необходима*** *избыточность (радикальность) образной (чувственной) модели, включающей модель самой УС и модель всего театра действий УС в форме первичного сенсориума УС (первой сигнальной системы УС),* и 2) ***достаточно*** *выделение части этого первичного сенсориума в особой форме вторичной сигнальной системы УС, в (осознанной) форме языкового сенсориума УС.* Языковый сенсориум УС выступает дубликатом *выделенной, осознанной части* первичного сенсориума УС. Именно языковая форма сенсориума УС имеет легко доступный для самой УС дополнительный языковой (символьный) способ его активации и этим делает языковый сенсориум механизмом познания осознанной части первичного сенсориума. Языковый сенсориум позволяет самой УС *структурировать, изучать, исследовать* и *развивать* осознанную часть первичного сенсориума УС. При этом, одновременно развивается сам язык УС. Кроме того, благодаря именно наличию языкового сенсориума, происходит углубленное развитие элементов искусственного интеллекта (ИИ) УС и, как следствие, происходит ускоряющееся развитие как самой УС так и разумное преобразование театра ее действий.

В работе [1] было рассмотренно развитие языкового сенсориума вплоть до образования **ядра ИИ УС**, т.е до образования **единой радикальной модели УС** в рамках ее театра действий в виде распределенной БД и БЗ УС. Ядро ИИ УС является избыточным информационно-системным ресурсом УС в форме среды радикалов. В настоящей статье обсудим подробно верхний иерархический уровень языкового сенсориума, надстройку ядра ИИ УС, которая является ультрасистемой с языковой операционной системой интеллектуального планирования (целеполагания) и ситуационного управления поведением УС в рамках театра действий УС. Этот уровень реализует **когнитивные свойства языкового сенсориума** и, следовательно, когнитивные свойства ИИ УС. Именно на этом уровне со стороны УС появляется *мировозренческая* оценка *осознанной части* первичного сенсориума УС в форме картины познанного мира УС и признания наличия *неосознанной (непознанной, подсознательной) части* первичного сенсориума УС. Именно на этом когнитивном уровне ИИ УС осуществляется процесс осознанного целеполагания УС и интеллектуального управления поведенческой активностью

УС в рамках театра ее действий. В результате чего языковый сенсориум формируется в автоматизированную *ультрасистему УС*, рис. 1.

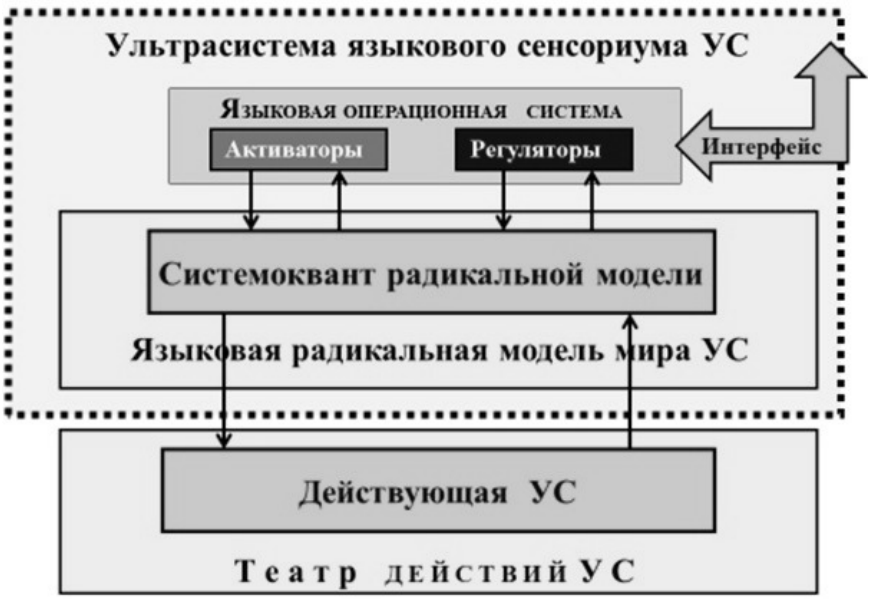


Рис. 1. Языковый сенсориум УС как ультрасистема интеллектуального планирования и ситуационного управления поведением УС

Исходным для образования и функционирования когнитивного уровня УС является свойство *рефлексии* УС в рамках ее театра действий. Рефлексия УС означает явное выделение собственного «Я» и включение его доминантой в языковую единую радикальную модель УС. Рефлексия УС начинается тогда, когда среди объектов своего театра действий УС вводит особый объект этого театра действий, а именно саму себя, УС, в виде специфического объекта театра действий. Это проявляется в форме уникального языкового радикала, «Я-УС» в единой радикальной модели УС. Семантика такого языкового радикала отождествляется со всей УС, включая весь ее театр действий, ее обоих сенсориумов УС, первичного и языкового, ее ядро и будущую надстройку ядра ИИ УС.

Такое самовыделение со стороны УС самой себя в единой радикальной языковой модели УС, когда среди разнообразных конкретных объектов осознанной части первичного сенсориума УС в некоторый момент появился объект «Я». Осознанный объект самой себя УС, носит название *рефлексивность искусственного интеллекта.*

ВЫВОД. Объект, самовыделенная «Я-УС» в рамках театра действия УС, должен обладать в будущей надстройке ядра ИИ УС, по крайней мере, тремя качествами: ***оценочное «Я-УС»*** *(Какая* «Я»?), ***когнитивное «Я-УС»*** *(Что мне делать и зачем?),* ***поведенческое «Я-УС»*** *(Как мне делать?).* С этого момента в языковом сенсориуме появляется и развивается верхний когнитивный иерархический уровень ИИ УС в форме языковой ***операционной системы*** планирования и управления поведением УС. Такая операционная система завершает формирование всего языкового сенсориума в автоматизированную ультрасистему *интеллектуального планирования и ситуационного управления поведением УС под руководством человека-хозяина УС,* [2, 4, 5], рис. 1.

Напомним, что в языковом сенсориуме УС развиваются следующие пять иерархических структурных интеллектуальных уровня развития.

Первый уровень — это *первичный сенсориум УС*, как постоянно расширяющаяся избыточная система сбора и хранения чувственных образов мониторинга самой УС себя и своего театра действий, безусловных и условных рефлексов УС, система приобретения и хранения навыков жизнеобеспечения УС, технологий жизнеобеспечения УС и правил действий УС в рамках ее театра действий [3].

Второй уровень — это выделение и развитие в первичном сенсориуме УС двух частей, *осознанной* и *подсознательной.* Выделение осознанной части происходит благодаря появлению в УС второй сигнальной системы в форме языка. Сначала это происходит путем наименования, номинирования языковыми символами–именами осознанной части особо значимых объектов и отношений первичного сенсориума. Номинированные объекты и отношения в первичном сенсориуме образуют его осознанную (основную) часть, которая в дальнейшем постоянно расширяется за счет подсознательной части и изучается. В результате происходит постоянное расширение и углубление познания осознанной части. При этом одновременно продолжается постоянное расширение первичного сенсориума в силу появления нового для УС внутри и вокруг нее, в силу изменчивости самой УС и окружающего мира УС в рамках театра ее действий [3].

Третий уровень — это появление слов-понятий или языковое выделение классов объектов, отношений этих объектов и классов навыков УС, развитие ***лексики*** языка. Лексика является созданием *языковой координатной системы понятий, языковой классификацией* объектов, отношений и навыков поведения УС в осознанной части театра ее действий, путем выделения классов объективного сходства, эквивалентности, порядка и присвоеия этим классам языковых символов слов-понятий. В результате появления лексики, терминологии языка в языковом сенсори-

уме организуется информационный уровень ультрасистемы, рис. 1, который постоянно расширяется вширь, вглубь и становится *опорной сетью* будущей ***единой языковой радикальной модели мира УС*** [2, 3].

Четвертый уровень — это развитие ***ядра ИИ УС*** в форме информационного описания объектов, отношений, и навыков в опорной сети в форме языкового информационно-системного ресурса данных, знаний, умений и технологий УС. Путем постоянной реструктуризации опорной координатной сети понятий лексика преобразуется в форму *распределенной БД и БЗ УС. Такая БД и БЗ* является ядром ИИ УС, информационно-системным ресурсом или языковой единой радикальной моделью мира УС в рамках ее театра действий. Этот ресурс появляется и развивается в результате поведенческой активности УС, путем самообучения и приобретения опыта УС. В УС на четвертом иерархическом уровне ИИ УС постоянно происходят два параллельных процесса ***ультраоснащенний*** языковой опорной координатной сети понятий, ***ультрамножественное*** (создание распределенной БД) и ***ультраоператорное*** (согздание распределенной БЗ), [2, 4, 5].

Пятый когнитивный уровень — это высший уровень ультрасистемы, рис. 1, результат появления рефлексии УС, когда в осознанной части первичного сенсориума УС происходит выделение специфического объекта театра действий УС самой себя УС и номинирования ее словом «Я». В результате в языковом сенсориуме появляется и развивается особый иерархический уровень, когда в единой радикальной модели УС появляются сначала понимание о наличии объекта — самой себя УС, затем ***оценивание*** свойств этого объекта, понимание о своих данных, своих знаниях, своих умениях и своих технологиях, о своей языковой единой радикальной модели мира УС в рамках театра своего действия и, наконец, появляется языковая операционная система, рис. 1.

В настоящей статье далее рассмотрим подробно как, благодаря рефлексии, когнитивный уровень УС обрастает особого нового типа «данными, знаниями, умениями и технологиями о своих данных, о своих знаниях, о своих умениях», которые далее УС широко использует для интеллектуального планирования и ситуационного управления поведением УС в рамках своего театра действий посредством языковой операционной системы.

2. Основной принцип целеполагания умной технической системы

Сначала отметим, что естественный интеллект (ЕИ) человека не передается по наследству. Каждый раз ЕИ возникает и развивается заново с

рождением человека, благодаря и вместе с освоением второй сигнальной системы, с обучения естественному языку человека. Каков язык человека, таков его ЕИ. При этом уже на самых ранних этапах развития человека у него наблюдаются появление и далее постоянное и бурное развитие всех аналогичных пяти языковым иерархическим уровням, которые были рассмотрены в п. 1. При этом в технических УС все пять уровней определяют интеллектуальную надстройку УС в форме *ультрасистемы* УС над *опорной системой* (над самой УС вместе с ее театром действий) [2, 4, 5], являющейся ИИ УС, рис. 1.

С появлением пятого когнитивного уровня ИИ УС, языковый сенсориум УС приобретает новые важные качества, понимание тройственного качества самой УС, *оценочное* «Я» (Какая «Я»?), *когнитивное* «Я» (Что мне делать? Какая моя цель и почему?), *поведенческое* «Я» (Как мне это делать?). Теперь для ИИ УС в мире имеется объект «Я» и все остальное «вне Я». Причем понимание «вне Я» определяется семантикой, которая находится в единой радикальной модели УС, включающая первичный сенсориум УС и языковый сенсориум УС. Более того эта единая модель мира определяет теперь не только то, что «вне Я», но и собственные качества самой себя, семантику «Я», рис. 1. Обсудим это подробнее.

2.1. Оценочное «Я» (Какая «Я»?). Единая радикальная модель УС, являющаяся ядром ИИ УС, является избыточным информационно-системным ресурсом этого «Я» [2]. Качество такого ресурса становится для «Я» качеством самой себя — УС. Теперь для УС единая модель мира УС — это «мой ресурс». УС может оценивать свою единую модель и квалифицировать ее, давать оценку своих данных, своих знаний, своих рефлексов, своих навыков, своего театра действий, т.е. детальную оценку самого «Я». При этом такая оценка должна относиться как к своим собственным данным, знаниям, рефлексам всего осознанного и подсознательного в первичном сенсориуме, относящегося к самой УС, но так же оценку всего своего театра действий. Этим в ультрасистеме определяется первое качество рефлексии УС, ***оценочное «Я»***, «свои возможности в рамках своего театра действий».

2.2. Когнитивное «Я» (Что мне делать и зачем?) — Целеполагание УС. Рассмотрим концептуальные вопросы *поведения УС в рамках своего театра действий.* Так как техническая УС предполагается искусственной целенаправленной и автоматизированной системой, Т.Е. УС является человеко-ориентированной. В силу тактико-технического задания (ТТЗ) при своем проектировании и создании УС имеет в каждый момент своего функционирования свое индивидуальное предназначение (Зачем нужна эта система? Какая нужна эта система? Для чего такая система?). Дерево целей всякой целенаправленной системы имеет обычно нечеткую структуру частично упорядоченную по важности, по

значимости целей. При этом такое дерево целей обычно удобно разбить на три нечеткие части, стратегическую, тактичческую и оперативную, с размытыми границами. ***Стратегическая часть*** дерева целей — это дальнесрочные цели, опирающиеся на *концептуальную мировозренческую картину мира* целенаправленной системы в рамках ее театра действий, на *идеологию и миропонимание*, целесообразности существования такой системы. Стратегическая часть включает цели, относящиеся к *обновлению и реорганизации самой системы*, а также к *расширению театра действий системы* и к *преобразованию театра действий*. **Вторая часть** — *тактическая часть* дерева целей. Это — среднесрочные задачи, которые призваны реализовать стратегические цели. Они включают задачи, относящиеся к *обучению и развитию самой системы*, а также к некоторым *изменениям* и к *модификации театра действий системы*. **Третья часть** — *оперативная часть* дерева целей. Это — ближнесрочные, ситуационные задачи. Они включают задачи, относящиеся к текущим обновлениям системы, а также к текущим изменениям в театре действий системы.

Если стратегические цели УС отвечают на вопрос «Что собирается сделать УС в будущем?», то тактические задачи отвечают на вопрос «Как сделать то, что желает сделать УС?» Средневековый китайский полководец Сунь Цзы считал: «Стратегия без тактики — это самый медленный путь к победе. Тактика без стратегии — это просто суета перед поражением». Наконец, оперативные задачи отвечают на вопрос «Что делать УС здесь и сейчас?».

Перейдем к формулировке ***основного человеко-центрического принципа ИИ УС — подчинение человеку — своему хозяину.*** *Любая техническая УС является участником (агентом) некоторой группировки умных систем с бинарным деревом полного иерархического подчинения каждого участника группировки своему одному* ***лидеру****. Без лидера техническая УС не функционирует. Цели и задачи, поставленные лидером перед УС являются для данной УС безприкословными к исполнению, непререкаемыми, обязательными к выполнению. Лидер формулирует и контролирует исполнение своих целевых заданий для подчиненной УС, которые определяют верхнюю стратегическую часть дерева целей этой УС. Каждая техническая УС в группировке, исходя из целевого задания, сама планирует свою тактическую часть и оперативную часть своего дерева целей. Любой лидер в группировке имеет подчиненные УС и является или технической УС или человеком. Если человек в группировке подчиняется лидеру, то такой лидер обязательно человек. Если лидером технической УС является человек, то он называется хозяином для данной УС и для всех других УС, которые подчиняются этой УС. Группировка УС является многоагентной*

системой. Хозяин группировки УС имеет право изменять вплоть до отмены целевое задание любой УС, для которой он является хозяином.

Следствия из основного человеко-центрического принципа ИИ технической УС.

– **Стратегические цели технической УС** всегда определяются или контролируются человеком — хозяином такой УС. Человек-хозяин определяет идеологию и мировоззрение УС, он отвечает за системный анализ УС и за системный синтез УС, отвечает за стратегические цели УС и всей группировки, за обучение методам оперативного целеполагания отдельных УС — участников группировки.

– **Тактические задачи УС** — это дело и обязанность человека-хозяина при формулировки ТТЗ для УС для когнитивного уровня ИИ технической УС. Планирование и управление реализацией тактических задач — это дело языковой операционной системы, точнее интеллектуального планирования и далее ситуационного управления поведением самой УС по реализации целевого задания для УС, полученного от своего лидера.

– **Оперативные задачи УС** — это дело и обязанность целиком языковой операционной системы интеллектуального планирования и ситуационного управления поведением самой УС, включая оперативное планирование УС с учетом текущей ситуации в каждой УС и в театре ее действий, а так же ситуационное управление коррекцией тактических и оперативных задач УС по принципу «здесь и сейчас», но с согласия человека-хозяина этой УС.

2.3. Поведенческое «Я-УС» (Как мне делать?) Главное качество выстроенной в рамках языкового сенсориума ядра ИИ УС, как распределенной структуры языковой единой радикальной модели УС, состоит в том, что модель является избыточным распределенным информационно-системным ресурсом УС. Благодаря языковой, символьной природе и радикальной форме, избыточный распределенный информационно-системный ресурс УС [1] обладает *навигационно эффективным доступом к любой локальной зоне* (ЛБД, ЛБЗ) для его активирования и использования через канал языковой связи при реализации интеллектуального планирования и ситуационного управления поведением УС. Обсудим подробнее проблему эффективности локального активирования запросами соответствующих ЛБД и ЛБЗ единой радикальной модели мира УС через каналы языковой связи УС, рис. 1.

Во-первых, ультрасистема языкового сенсориума, рис. 1, отвечает за: постоянное развитие УС, саморасширение, саморазвитие обоих сенсориумов УС, первичного и языкового, самообновления, дополнения и модификации информационно-системного ресурса УС, постоянное введение новых объектов, новых классов объетов, новых задач, новых методов их

решения, новых отношений между ними, т.е. постоянное расширение и развитие ИИ УС. И все это опирается, с одной стороны, на существо функционирования языковой операционной системы, но, одновременно, это связано с уточнением, расширением и саморазвитием самого языкового сенсориума в рамках ультрасистемы УС.

Во-вторых, развитие ультрасистемы УС не только постоянно сопровождается *обновлением*: расширением языковой опорной координатной сети, появлением новых классов объектов или отношений, или навыков (технологий). Еще в ультрасистеме УС постоянно происходят одновременно и параллельно *ультраоснащения* новых опорных множеств и операторов. Тем самым в ультрасистеме происходит *развитие* языковой распределенной сети новых ЛБД и ЛБЗ, т.е. пополнения, дополнения и модификация информационно-системного ресурса УС [1]. Постоянно появляются новые классы задач, методов, алгоритмов, программ и далее новые разработки соответствующих им ЛБД и ЛБЗ. Под интеллектуальным планированием и ситуационным управлением языковой операционной системы ультрасистемы непрерывно происходит постоянное *развитие первичного и языкового сенсориумов УС* и, как следствие, коррекция и *развитие языковой единой радикальной модели мира УС*.

Ультрасистема интеллектуального планирования и ситуационного управления поведением УС, исходя из своих целевых заданий УС, которые формулируются лидером УС и контролируются человеком — хозяином УС, планирует для операционной подсистемы тактические задачи УС, реализует их с учетом ситуации в театре действий и с учетом спланированного порядка. Далее операционная подсистема осуществляет когнитивное управление поведением УС в рамках театра действий УС при постоянном самоконтроле и учете появляющихся конфликтов в УС и во всем театре действий УС. Если стратегические цели требуют, то операционная система с согласия человека-хозяина организует и реализует проведение разумной *перестройки театра действий УС*.

3. Человеко–центрические требования к ИИ УС

Обсудим главные требования к ультрасистеме интеллектуального планирования и ситуационного управления поведением технической УС в рамках театра действий УС, а значит к ИИ УС, рис. 1.

Стратегическая часть требований технической УС всегда будет за человеком - хозяином: это идеология и мировоззрение, мышление и познание, учет соотношения подсознательной и осознанной частей сенсориума, метафизическая и научная картина мира, прогнозирование будущего и стратегическое целеполагание.

3.1. Качества ИИ, которые обязательны для технической УС:

Когнитивность УС (оперативная часть, «здесь и сейчас») — системный анализ и системный синтез, интеллектуальное планирование и ситуационное управление поведением технической УС, выделение нештатных ситуаций, изучение нештатных задач, поиск и разработка новых алгоритмов, приобретение новых навыков, поиск и открытие новых технологий;
Информативность УС — средства понимания сведений о чем-либо;
Коммуникативность УС — средства связи при общении с другими УС;
Активационность УС — средства воздействия сведений на другие УС;
Оценочность УС — средства сбора данных и оценивания чего-либо;
Рефлексивность УС — самообщение, целеполагание, саморазвитие;
Номинативность УС — средства уникально именовать что-либо.

3.2. Три одновременных процесса в ультрасистеме УС. ИИ технической УС должен быть *открытой, развивающейся* системой. Все тактические задачи УС на разных этапах жизненного цикла УС делятся на три больших класса: задачи *целевые* (назначения), *сенсорные* (ситуационные) и *сертификационные* (гомеостатические). *Целевые задачи УС* нацелены непосредственно на планирование и выполнение тактических задач по основному назначению УС, *сенсорные задачи УС* обеспечивают сервисный мониторинг, сбор информации об УС и обо всем театре действий УС (принцип «*здесь и сейчас*»), *сертификационные задачи УС* имеют своей целью тестирование, проверку на целостность и нормативную готовность всех подсистем УС (принцип «гештальта»).

На базе избыточного информационно-системного ресурса УС языковая операционная система планирования и управления поведением УС в рамках распределенной радикальной БД и БЗ модели УС осуществляет одновременно три следующих информационных процесса: *текущий целевой* (решение очередной тактической задачи во исполнение целевого указания лидера) по принципу «целесообразность поведения», *текущий сенсорный* (непрерывный сбор ситуационной информации по принципу «здесь и сейчас» и коммуникационному принципу «*что нового сообщают другие УС*») и, наконец, *текущий сертификационный процесс* по принципу «закрывание гештальта» (обеспечение гомеостаза УС и всего театра действий УС), рис. 2.

ВЫВОДЫ.

1) При выполнении текущего целевого процесса в УС должен проводиться постоянный мониторинг на появление в УС и в театре его дествий новых конфликтов по принципу «*не навреди себе*». Организация устранения этих конфликтов.

Рис. 2. Активационные поцессы в языковой единой радикальной модели мира УС: целевой, сенсорный и сертификационный

2) При проведении текущего сенсрного процесса в УС и в театре ее действий должны во время вноситься и учитываться ситуационные изменения в единую модель УС.

3) При проведении текущих сертификационных коррекций и восстановления целостности систем, требуется следить за рабочим состоянием всех необходимых для эффективного поведения УС сервисных систем.

3.3. Штатные и нештатные ситуции и тактические задачи УС.

Нештатные ситуации и задачи являются основными *точками роста, обновления, развития УС* и *преобразования* театра действий УС. В случае успешного преодоления нештатности ее переводят в класс штатных ресурсов УС. При таком развитии сам ИИ УС обновляется, обучается, расширяет класс своих штатных ситуаций, задач и средств их решения. В результате функционирования УС в нештатных ситуациях происходит *самообучение УС, саморазвитие УС и целенаправленное преобразование театра действий УС.*

Штатные ситуации и задачи УС — это ситуации и задачи, регламентированные тактико-техническими требованиями к УС. Штатные ситуации и задачи — массовые и типовые для УС. Они должны быть хорошо изучены, заранее определенные и формализованные в единой радикальной моделе УС. В языковой радикальной модели УС для всех штатных задач должны быть заранее разработаны и представлены устойчивые методы, конструктивные алгоритмы и эффективные программно-

технические средства (ПТС) их решения на разных этапах жизненного цикла УС. Следует отметить, что именно к *штатным задачам УС* в первую очередь предъявляются требования *оперативного (немедленного)* их решения с *минимальными ресурсозатратами*. Для штатных задач *фактор времени и фактор малозатратности* самые важные. Штатные ситуации и задачи должны быть заранее формализованы и оптимизированы по сложности, ресурсоемкости, быстродействию. Решение штатных задач должно быть доведено до навыков, до «автоматизма», до «решения с закрытыми глазами», т.е. методы их решения не должны постоянно требовать модификации и трудоемкой настройки этих методов при применении.

Нештатные ситуации и задачи УС — это ситуации и задачи, нерегламентированные для данной УС, непредставленные заранее в единой радикальной модели данной УС. Для таких ситуаций и задач в радикальной модели УС нет готовых описаний, методов и тем более нет готовых алгоритмов и программно-технических средств их решения. В рамках кибернетических (автоматных) систем, которые функционируют по заранее предусмотренным в них алгоритмам, нештатные задачи не относятся к области определения этих автоматов. Нештатные задачи автоматами не могут быть решены. Однако для умных (интеллектуальных) поведенческих систем, к которым относятся УС со специальной языковой ультрасистемой интелектуального планирования и ситуационного управления в рамках ИИ УС (рис. 1), некоторые нештатные задачи решаются. Это происходит благодаря избыточности среды радикалов и когнитивности ее специализированного оснащения. При этом степень интеллектуальности языковой надстройки ИИ УС определяется именно теми нештатными задачами для данной УС, которые эта надстройка сможет преодолеть.

Нештатность — точки роста и развития УС. Идея преодоления нештатности в УС следующая — это либо поиск обновления радикальной модели в режиме обучения (с учителем) через запросы к лидеру и другим УС или обращения к поисковым системам по имеющемуся виртуальному глобальному пространству данных, знаний и умений и, далее, *закрепление* такого опыта *в режиме развития* самой УС, либо *организация попыток* преодоления нештатнсти (метод проб и ошибок) в рамках имеющейся радикальной модели, *оценивание этих попыток*. Наконец, возможна организация необходимого *преобразования* театра действий УС при согласовании со своим лидером.

Подчеркнем, что в случае нештатных ситуаций и нештатных задач на одно из первых мест для УС выдвигается *фактор риска*, связанный с неизвестностью последствий такого решения. Это отличает нештатные ситуации и задачи от штатных ситуаций и задач, для которых на пер-

вом месте для УС выступает *фактор времени* и *ресурсозатратности* Поэтому при попытках учесть нештатные ситуации или решать нештатные задачи в УС должен быть предусмотрен анализ рисков в рамках ее радикальной модели, учет результатов проведенных попыток, так называемый, режим *самообучения*.

Удобно класс нештатных ситуаций и задач разбить как минимум на следующие два подкласса, а именно — ***слабо нештатные*** *и* ***сильно нештатные*** *ситуации и задачи. К слабо нештатным ситуациям и задачам для данной УС отнесем ситуации и задачи с частными (небольшими) отклонениями от штатных ситуаций и задач для УС. Например, когда появляются нештатные элементы в штатных ситуации* или *нештатные ограничения в штатной задаче* и др. *Все остальные нештатные ситуации и задачи для данной УС отнесем к сильно нештатным ситуациям и задачам этой УС*. Слабо нештатные ситуации и задачи требуют *модификации известных подходов и методов*. Например, к слабо нештатным задачам отнесем штатные задачи с *неполной исходной информацией*, или штатные задачи с *малой устойчивостью решения*, или штатные задачи с *неудовлетворительно высокой сложностью имеющихся алгоритмов* их решения и многие другие.

Типичным примером *слабо нештатных задач* является класс *некорректных* по Адамару задач. Для них разработаны эффективные подходы к их эффективному решению А. Н. Тихоновым, В. К. Ивановым, М. М. Лаврентьевым и др. Например, подход А. Н. Тихонова, который назван *регуляризацией некорректных задач (1963 г.)*, основан на введении специальных дополнительных ограничений в условия задачи, введения дополнительной, специфической для каждой задачи, информации о решении задачи. Причем, для каждой некорректной задачи вводятся свои, индивидуальные дополнительные ограничения, например, в форме компактных множеств [2].

Примером *сильно нештатной задачи* является *новая, во многом неизвестная* для ИИ УС задача, требующая *разработки принципиально* новой модели, нового математического аппарата, нового метода и тем самым часто нового алгоритма ее решения. Сильно нештатная задача обычно требует не только нового метода и алгоритма, но еще разработки новых программно-технических средств реализации ее решения, ее верификации и т.д. Например, в финансовой математике задача определения стоимости опциона являлась сильно нештатной задачей до 1973 года, когда была создана модель Блэка-Шоулза в 1973 году и выведена формула для стоимости опционов CALL и PUT европейского типа [2].

4. Информационно-системная безопасность УС

Все когнитивные требования к ИИ ультрасистемы УС удобно объединить в одном едином обобщеном глобальном требовании: «Обеспечения *информационно-системной безопасности* УС» [2].

4.1. Информационно-системная безопасность (ИСБ) УС. Требование ИСБ УС реализует целевую установку лидера данной УС. Понятие ИСБ УС является глобальным двуединым требованием, которое имеет тесно связанные между собой две стороны безопасности, *информационную* и *системную*. ИСБ УС включает в себя все частные случаи безопасности УС от экологической, энергетической и т.д. до функциональной безопасности, рис. 3.

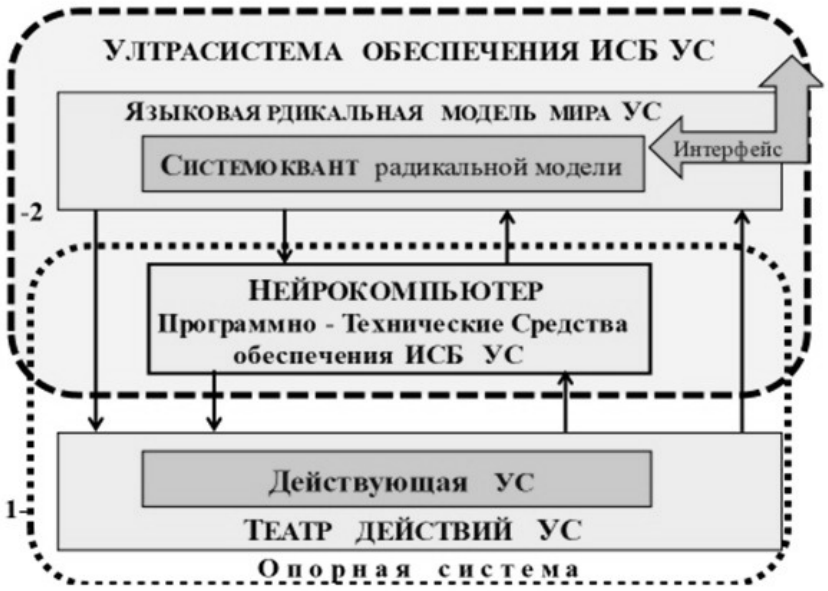


Рис. 3. Ультрасистема обеспечения ИСБ УС

Сформулируем и обсудим обе стороны ИСБ УС.

Информационная сторона ИСБ УС. *Каждая штатная задача жизненного цикла УС*, включая целевые, сенсорные и сертификационные задачи, должна быть безусловно и эффективно решена, независимо от формы и полноты оперативной информации, от наличия помех, путем создания и логической обработки избыточной модели театра действий УС (языкового сенсориума широкой проблемной области УС), включающей в частности модель самой УС, модель театра действий УС. Избыточная модель должна являться необходимым условием обеспечения

решаемой штатной задачи и достаточным информационным и системным ресурсом для ее решения. Она должна быть надежно защищена от несанкционированного доступа (НСД) и т.п.

Каждая нештатная задача жизненного цикла УС должна быть в центре внимания УС, исследована на необходимость и возможность ее решения с согласия лидера в рамках его идеологии, мировозрения и стратегических целей. При этом УС может использовать ресурсы избыточной модели, или воспользоваться запросами к глобальному информационно-системному (виртуальному) пространству за обновлениями, или организовать попытки решения задачи в режиме «проб и ошибок», или другим каким–либо способом самообучения, развития самой УС, вплоть до преобразований театра действий УС с согласия своего лидера.

Системная сторона ИСБ УС. *Каждая задача штатная или нештатная жизненного цикла УС*, в процессе своего решения, должна сопровождаться постоянным учетом всех последствий решения для самой УС и для театра ее действий путем постоянного сертифицирования избыточной модели, тестирования самой УС, анализа театра действий УС и устранения конфликтов для сохранения системной целостности, гомеостаза самой УС и театра ее действий в полном соответствии с целевыми заданиями УС от своего лидера.

4.2. Необходимость нейрокомпьютера для ультрасистемы УС. Обсудим программно-технические средства (ПТС) ультрасистемы ИИ УС, рис. 3. В вычислительной технике различают два принципиально отличных друг от друга типа процессоров, к которым относятся *аналоговые* и *цифровые*. Аналоговый процессор (нейросетевой, квантовый и др.), использует интерференцию физических полей. Он используется в следящих системах и ориентирован на максимальное *распараллеливание* вычислительного процесса, а, тем самым, на максимальную его *оперативность*. Техническая и программная реализация решения задач при использовании аналоговых процессоров опирается на *базис, который для краткости будем именовать нейросетевым*. Именно в этом базисе, например, с большой эффективностью и оперативностью можно решать задачи *распознавания образов, задачи экспресс-анализа*, многие *прямые задачи*. Часто аналоговые процессоры на основе сверточных нейронных сетей являются базовыми в следящих системах управления.

В отличии от аналогового, цифровой процессор сегодня использует *позиционную* систему записи чисел, *последовательный* вычислительный процесс, который опирается на базис, который будем называть *логическим*. Цифровой процессор архитектуры Дж. Фон Неймана основан на использовании линейной позиционной системы записи чисел и требует реализации (методологии) последовательной обработки информации, которая базируется на логическом базисе. Напомним, что основная опера-

ция в цифровом процессоре, *суммирование чисел*, происходит исключительно последовательно от меньших разрядов к старшим, с прерыванием и задержкой.

В логическом базисе цифрового процессора более эффективно решаются задачи, требующие учета и анализа *рисков неблагоприятных последствий. К числу таких задач относятся, например, задачи принятия решений, задачи синтеза, обратные задачи, алгоритмически неразрешимые задачи.*

Дуализм нейрокомпьютера. Двухкорневой термин *нейрокомпьютер (НК)*, своим названием как бы подчеркивает полезный *дуализм* нейрокомпьютерной реализации радикальной модели УС в форме среды радикалов. ПТС УС должно в перспективе быть устроено по типу распределенной вычислительной среды, образованной двумя видами процессоров аналоговыми (Нейро) и цифровыми (Комп), рис. 4. В перспективе, именно *нейрокомпьютеры*, сегодня видятся той распределенной вычислительной средой, которая базируется одновременно на двух типах процессоров. С одной стороны, такая среда имеет возможности нейросетевого базиса, на аналоговых процессорах, т.е. на нейросетевых или квантовых процессорах. С другой стороны, нейрокомпьютеры сохраняют возможности логического базиса цифровых процессоров фон-неймановского типа.

Рассмотрим более подробно идею эффективного сочетания, соотношения, распределения, аппаратной координации при нейрокомпьютерной реализации решения задач на разных этапах ЖЦ УС.

- Штатные задачи *следует решать оперативно и малозатратно. В штатных задачах на первый план выступает фактор времени и эффективности. Поэтому рекомендуется их решать, по возможности, максимально используя механизмы распараллеливания, т.е. решать на аналоговых процессорах, в нейросетевом базисе. На языке технической реализации информационных процессов это означает, что при решении штатных задач желателен переход от логического базиса к нейросетевому базису. Интересно, что в физиологии это соответствует переводу процессов решения штатных задач в оперативный режим рефлексов, навыков, а в технике к следящим системам,* рис. 3.

- Слабо нештатные задачи *рекомендуется решать в смешанном логическом и нейросетевом базисе, эффективно сочетая последовательную и параллельную обработку информации на аналоговых и цифровых процессорах. При этом рекомендуется использовать элементы режима поиска в глобальном виртуальном пространстве, или обучения с учителем, или в режиме самообучения и принципа увеличения доли параллельной обработки.*

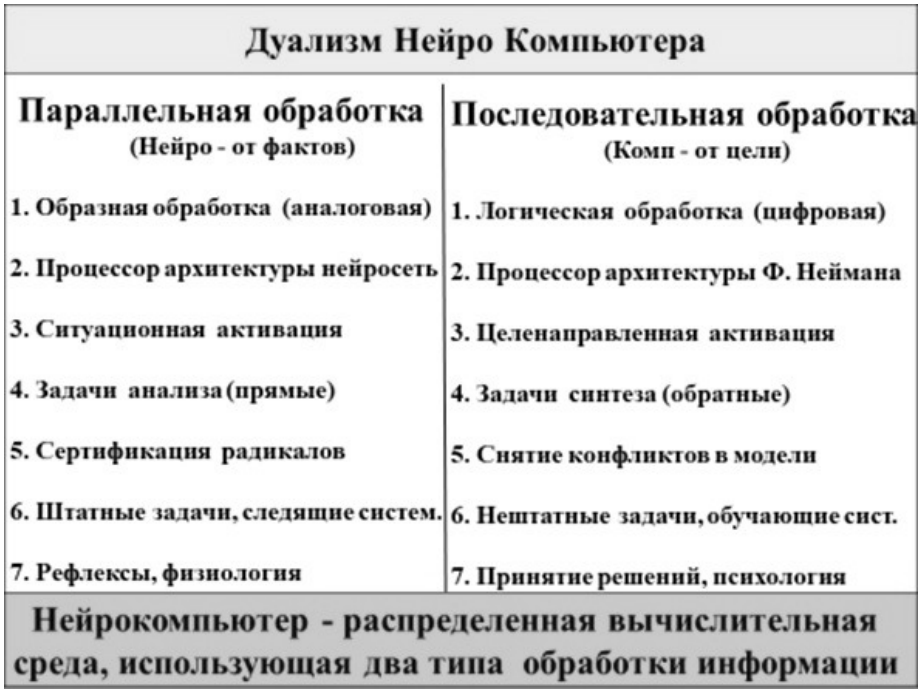


Рис. 4. Аналоговый (нейросетевой) и цифровой (логический) базисы нейрокомпьютера

- Сильно нештатные задачи *рекомендуется решать преимущественно на цифровых процессорах, в логическом базисе, в режиме поиска в глобальном виртуальном пространстве или, обучения с учителем или, в режиме самообучения, проб и ошибок. В таких задачах на первом месте выступает фактор риска.*

В дальнейшем при успешном решении нештатной задачи предполагается перевод такой задачи в класс штатных и далее организовывать максимальное распараллеливание разработанного алгоритма. При этом нештатные задачи являются, своего рода, точками роста, развития самой УС и преобразования театра действий УС.

Обсуждение. 1) В теории алгоритмов имеются задачи алгоритмически разрешимые и неразрешимые. Для алгоритмически разрешимых задач различают три вида алгоритмов по сложности их реализации — алгоритмы *степенной сложности*, алгоритмы *экспоненциальной сложности* и задачи *NP-полной сложности*. Класс задач алгоритмически неразрешимых (у них нет общего алгоритма решения) относится, естественно, к сильно нештатным задачам. Класс нештатных задач, явля-

ющихся NP-полной сложности, являются, как правило, классом слабо нештатных задач.

2) Обратим особое внимание на выделение программно-технических средств технической УС в специфическую систему «нейрокомпьютер». Нейрокомпьютер особая двойственная системная сущность, ПТС УС. Она присутствует в *опорной системы* УС и она присутствует в *ультрасистеме обеспечения ИСБ* УС, рис. 2.

3) Для повышения эффективности УС необходимо сочетать преимущества параллельной и последовательной обработки информации, следящих (аналоговых) и логических (цифровых) процессоров нейрокомпьютерной парадигмы ПТС опорной системы УС и ультрасистемы УС, рис. 2.

5. Заключение о глобальном интеллектуальном виртуальном пространстве ноосферы Земли

С появлением множества искусственных технических УС различного назначения и с учетом постоянного их развития возникает возможность и потребность объединить оба вида интеллекта, ЕИ людей и ИИ УС, в единое интеллектуальное пространство, в новую *сетевую ноосферу Земли* по терминологии академика В. И. Вернадского. Главное требование к такому пространству — это организация коммуникационного взаимодействия и информационно-системного обмена между людьми, между техническими УС и людьми, между техническими УС различного назначения между собой, с требованием ко всем участникам этого пространства соблюдения ИСБ каждым участником и субординаций, рис. 3. Несмотря на геополитические общеземные проблемы, элементы такого взаимодействия и обмена в настоящее время уже появились и бурно развиваются в форме *интернета людей*, в форме языковых средств *компьютерных интерфейсов, в форме интернета вещей (IoT)*. По-видимому, пришло время единого для техники и человека делового языка интеллектуального общения, своего рода *универсального языка сетевой ноосферы Земли*.

Список литературы

[1] Чечкин А. В., “Тезис о наличии искусственного интеллекта.”, *Интеллектуальные системы. Теория и приложения*, **25**:1 (2021), 29–49.

[2] Потюпкин А. Ю., Чечкин А. В., *Искусственный интеллект на базе информационно – системной избыточности*, «КУРС», Москва, 2019, 382 pp.

[3] Воронков Г. С., Чечкин А. В., “Проблемы моделирования сенсориума и языковой системы естественного интеллекта индивидуума”, *Интеллектуальные системы*, **2**:1–4 (1997), 23–34

[4] Чечкин А. В., *Математическая информатика*, «Наука», Москва, 1991, 412 pp.

[5] Чечкин А. В., “Классификация базовых ультрасистем.”, *Труды V Международной научно-практической конференции «Современная математика и концепции инновационного математического образования» Москва 1 июня 2018 года – М.: Изд. Дом МФО*, 2018, 104–119

Cognitive level of artificial intelligence

Chechkin A.V.

Goal. To discuss the cognitive level and substantiate the basic principle of artificial intelligence of smart systems, human-centricity - subordination and service to a person (master).

Methods. The central role of the radical (redundant) model of the theater of operations of a smart system as the core of artificial intelligence is emphasized. The stages of creating an ultrasystem to ensure the information and system security of a smart system, its intelligent planning of actions and situational behavior management in the theater of its actions, including the development of the system itself and the reasonable transformation of the theater of its actions, are discussed.

Results. There are five main levels of artificial intelligence development. The features of the first and second signaling systems (primary and language sensoriums) are studied. The information and system irregularities of a smart system are highlighted as points of growth of its intellectual development based on machine learning. The collective artificial intelligence of a group of smart systems is being investigated within the framework of their unified theater of action and subordination to the human master of the grouping. The principle of human-centric goal-setting of any technical intelligent system, determined by its unconditional subordination to its human owner, is substantiated. The corresponding requirements for the ultrasystem and its language operating system of intelligent planning and situational management of the behavior of a separate smart system or a team of smart systems are discussed. The problems of the development of a separate smart system, the reengineering of a collective of such systems and the problem of reasonable transformation of the theater of actions of such a collective are considered.

Keywords: cognitiveness of natural intelligence, automated smart system, collective artificial intelligence, radical, language, Internet of Things, interface, information system security, tactical and operational planning of smart system actions.

References

[1] Chechkin A. V., "Thesis about the availability of artificial intelligence. (in Russian)", *Intelligent Systems. Theory and Applications*, **25**:1 (2021), 29–49

[2] Potyupkin A. Yu., Chechkin A. V., *Artificial intelligence based on information and system redundancy (in Russian)*, «KURS», Moscow, 2019, 382 pp.

[3] Voronkov G. S., Chechkin A. V., "Problems of modeling the sensorium and the language system of natural intelligence of the individual (in Russian)", *Intelligent Systems*, **2**:1–4 (1997), 23–34

[4] Checkin A. V., *Mathematical informatics (in Russian)*, «Nauka», Moscow, 1991, 412 pp.

[5] Chechkin A. V., "Classification of basic ultrasystems. (in Russian)", *Proceedings of the V International Scientific and Practical Conference "Contemporary Mathematics and the Concepts of Innovative Mathematical Education" Moscow, June 1, 2018 - Moscow: MFO Publishing House.*, 2018, 104–119

Часть 2.
Специальные вопросы теории интеллектуальных систем

Методы анализа данных в задаче прогнозировани спортивных результатов

А. Д. Журавлев[1]

В данной статье рассматривается задача возможности предсказывать спортивные результаты с помощью методов анализа данных и машинного обучения и определения качества такого прогноза. Также приводитя сравнение построенной модели с моделью используемой букмекерскими конторами.

Ключевые слова: Ключевые слова: машинное обучение, прогнозирование спортивных результатов, анализ данных, классификация.

1. Введение

Данная статья продолжает работу по построению модели, описывающей некоторые параметры хокккейного матча, начатую в [1]. Будет рассмотрен один из возможных подходов к моделированию, использующий в качестве основного инструмента методы машинного обучения. Весь процесс исследования будет происходить на реальных исторических данных.

2. Подготовка данных

2.1. Описание данных

Различные исторические данные по хоккейным матчам предоставляются на официальных сайтах турниров. Например, на сайте www.khl.ru Континентальной Хоккейной Лиги можно найти информацию об игроках, результатах матчей. В частности, можно получить такие данные, как количество забитых шайб командами как в матче целиком, так и за определенный период.

В качестве набора данных была получена статистическая информация о матчах Континентальной Хоккейной Лиги сезонов 2014-2020. Доступными оказались следующие признаки: домашняя и гостевая коман-

[1] *Журавлев Артем Дмитриевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: artemzhuravlev.msu@gmail.com.

Zhuravlev Artem Dmitrievich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

ды, дата матча, итоговый счет, количество бросков, результат каждого из 3-х периодов, заброшенные шайбы в большинстве и меньшинстве, число численных преимуществ, число выигранных вбрасываний, штрафное время и различный коэффициенты БК.

2.2. Отбор переменных и обработка данных

Чтобы улучшить качество переменных, можно использовать генерирование накопленной статистики. Например, подходят следующие признаки: средняя реализация бросков, среднее количество нанесенных бросков, среднее количество заброшенных шайб, среднее количество пропущенных шайб, среднее количество пропущенных бросков. Такие значения накапливаются для каждой команды, исходя из ее предыдущих результатов, и обновляются после каждого сыгранного матча.

Распределение признаков с заброшенными шайбами будет хорошо моделироваться распределением Пуассона - распределение дискретного типа случайной величины, представляющей собой число событий, произошедших за фиксированное время, при условии, что данные события происходят с некоторой фиксированной средней интенсивностью и независимо друг от друга. Введем определение распределения Пуассона:

Выберем фиксированное число $\lambda > 0$ и определим дискретное распределение, задаваемое следующей функцией вероятности:

$$\mathrm{p(k)} \equiv \mathbb{P}(Y = k) = \frac{\lambda^k}{k!}\,\mathrm{e}^{-\lambda},$$

где k! обозначает факториал числа k, e = 2.718281... — основание натурального логарифма. Тот факт, что случайная величина Y имеет распределение Пуассона с математическим ожиданием λ, записывается: $Y \sim \mathrm{P}(\lambda)$.

Таким образом, для каждого матча можем получить значения различных вероятностей, которые его описывают.

Также в признаках присутствуют коэффициенты БК на события, при которых в матче будет забито больше или меньше некоторого значения суммарного количества забитых шайб обеими командами, и само количество - тотал.

В качестве целевой переменной будем рассматривать суммарное количество голов, забитых обеими командами в матче. Целевая переменная будет принимать значение 1, если реальное суммарное количество голов больше, чем тотал, предложенный БК, и значение 0, если меньше, соответственно.

Временно́й ряд (или ряд динамики) — собранный в разные моменты времени статистический материал о значении каких-либо параметров (в простейшем случае одного) исследуемого процесса. Каждая единица

статистического материала называется измерением или отсчётом, также допустимо называть его уровнем на указанный с ним момент времени. Во временном ряде для каждого отсчёта должно быть указано время измерения или номер измерения по порядку.

Так как у нас имеются статистические показатели, которые формируются на основе предыдущих результатов команды, то такие данные необходимо анализировать с помощью временных рядов - для построения алгоритма необходимо использовать только уже прошедшие события до некоторого время t, а для оценки события, которые происходят после.

3. Постановка задачи классификации

Пусть X — множество описаний объектов, Y — множество номеров (или наименований) классов. Существует неизвестная целевая зависимость — отображение $y^*\colon X \to Y$, значения которой известны только на объектах конечной обучающей выборки
$X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$. Требуется построить алгоритм $a\colon X \to Y$, способный классифицировать произвольный объект $x \in X$, сопоставив объект $y \in Y$.

Нам необходимо отличать хороший алгоритм классификации от плохого, для этого нам нужно ввести метрики для оценки качества модели. Модель также будем оценивать по тому, сколько денег мы получим или потеряем, если будем ставить согласно ее предсказаниям. Для этого введем следующую метрику: *ROI (return on investment)* - $ROI = \frac{P_n * 100}{s * n}$, где P_n - прибыль на дистанции в n матчей, s - сумма одной ставки, n - количество ставок. То есть ROI является отношением заработанных денег к потраченным. ROI - основной показатель эффективности прогностической модели. Таким образом, будем "играть в плюс"в том случае, если $ROI > 0$.

3.1. Перекресная проверка

Кросс-валидация (cross-validation) - метод оценки аналитической модели и её поведения на независимых данных. При оценке модели имеющиеся в наличии данные разбиваются на k частей. Затем на $k-1$ частях данных производится обучение модели, а оставшаяся часть данных используется для тестирования. Процедура повторяется k раз; в итоге каждая из k частей данных используется для тестирования. В результате получается оценка эффективности выбранной модели с наиболее равномерным использованием имеющихся данных.

Улучшим качество с помощью кросс-валидации. Так как мы будем строить предсказательную модель с помощью временных рядов, то необ-

ходимо правильно использовать кросс-валидацию. Будем сдвигать обучающую выборку на n шагов, а не увеличивать ее, добавляя прошедшие матчи, чтобы не учитываять слишком старые результаты, которые уже потеряли свою информативность и могут только помешать при обучении.

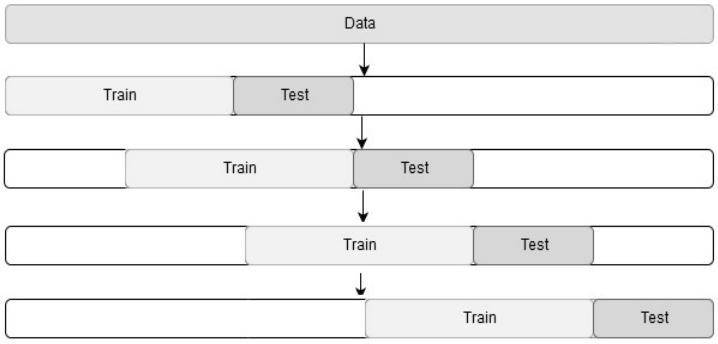


Рис. 1. Кросс-валидация

После проведенного исследования для выбора оптимальных t и n , основываясь на размере доступных исторических данных - 2770 матчей, оказались значения 600 и 100 соответственно - Рис.2. Их и будем использовать для построения модели.

3.2. Предсказание модели

Так как в исходных признаках у нас есть коэффициенты на целевые события от БК - коэффциенты k_0 и k_1 для исхода "Тотал меньше" и "Тотал больше" какого-то количества шайб соответственно, то будем строить вероятностные модели для улучшения предсказательной способности алгоритма, то есть модель возвращает вероятность принадлежности к классу 0 или 1 - p_0 и p_1 соответственно. Введем величины $b_i = p_i * k_i$ для $i = 0, 1$ - отношение вероятности наступления исхода i, предсказанной нашей моделью, к вероятности, которую дает БК. Данные величины - математическое ожидание наших ставок.

Теперь можно рассмотреть различные пороговые значения b для отбора значений b_i, при которых хотим ставить на матч. Такой порог b должен быть больше 1, так как иначе мы будем играть с отрицательным математическим ожиданием. В предположении, что p_i - вероятности событий, то изменяя порог b принятия решений, мы будем влиять на выигрыш.

	train_size	test_size	train_roi
0	500	50	-0.0161334
1	500	100	-0.0295533
2	500	150	0.0188796
3	500	200	-0.033247
4	600	50	-0.0224978
5	600	100	0.0209019
6	600	150	0.0154973
7	600	200	0.0127306
8	700	50	-0.0425073
9	700	100	-0.0155961
10	700	150	-0.0233171
11	700	200	-0.0658967
12	800	50	-0.00506778
13	800	100	-0.00514799
14	800	150	-0.0229847
15	800	200	-0.0341113

Рис. 2. Поиск оптимальных t и n

Поэтому и хотим искать именно вероятности, а не просто построить бинарный классификатор. При верной гипотезе, можно искать подходящие коэффициенты от БК для того чтобы увеличить свой выигрыш.

Перейдем к доказательству гипотезы. Для потенциального улучшения предсказательной способности алгоритма рассмотрим введение следующих пороговых значений: 1, 1.01, 1.02, 1.03, 1.04, 1.05.

4. Техническая реализация

4.1. Поиск наилучшего алгоритма

Разделим выборку на 2 множества, предварительно отсортировав по дате: $X_1 = \{\text{первые 2000 матчей}\}, X_2 = \{\text{оставшиеся 770 матчей}\}$. Модель будет обучаться с помощью ранее введенной кросс-валидации на множестве X_1, а после с соответствующими окнами оценена на множестве X_2, где на первом шаге выборка для обучения будет составлять последние 600 матчей множества X_1. В качестве моделей выбраны сле-

дующие алгоритмы: LightGBM, XGBoost - одни из сильных алгоритмов машинного обучения. Также был использован алгоритм случайного поиска лучших параметров классификатора по сетке - Random Search вместо перебора всевозможных вариантов - Grid Search по причине его трудоемкости.

4.2. Анализ результатов

Для каждого алгоритма проводим перебор гиперпараметров, перебор различных порогов и выбираем лучшую модель, согласно выбранной метрике. Получили следующие результаты - Рис.3

D	E	F	G	H	I	J	K	L
model_ty	feats	bord	test_money	test_pa	test_lo	test_w	test_be	test_roi
lgbm	default	1	-4.37000000000001	189	294	288	582	-0.007508591065292
lgbm	default	1.01	1.07999999999999	235	269	267	536	0.002014925373134
lgbm	default	1.02	8.07	295	236	240	476	0.016953781512605
lgbm	default	1.03	4.55	339	216	216	432	0.010532407407407
lgbm	default	1.04	10.78	386	190	195	385	0.028
lgbm	default	1.05	13.05	426	169	176	345	0.037826086956522

Рис. 3. Поиск лучшего порога - LGBM

Таким образом, можно сделать вывод, что:

1) LightGBM с порогом 1.05 - наилучший алгоритм с точки зрения метрики ROI

2) итоговый ROI на X_2 составил 3.8%- это означает, что если бы мы ставили по предсказниям нашей модели, то переиграли бы БК.

3) При увеличении порога отсечения происходит увеличение метрики ROI

5. Заключение

Итоги проведенного исследования показывают, что методы машинного обучения можно применять для анализа спортивных исторических данных. Также они подтверждают сформулированную гипотезу о том,

что можно увеличить ROI с помощью использования пороговых значений для отбора вероятностных предсказаний.

Полученную модель можно использовать для игры на рынке спортивных ставок против БК. Более того, стоит заметить, что в качестве коэффициентов БК в данных использовались коэффициенты на момент закрытия приема ставок, которые значительно ниже коэффициентов на live-ставки, то есть прибыль может быть увеличена.

Список литературы

[1] Журавлев А.Д., "Возможный подход к задаче прогнозирования спортивных результатов методами анализа данных", *Интеллектуальные системы: Теория и приложения том 25, № 1,*, 2021, 63-69.

[2] https://www.sports.ru/tribuna/blogs/teilnahme/1098481.html, "Хоккейная аналитика".

Data analysis methods in the problem predicting sports results

Zhuravlev A. D.

This article discusses the problem of predicting sports results using data analysis and machine learning methods and determining the quality of such a forecast. Also giving a comparison of the constructed model with the model used by bookmakers.

Keywords: machine learning, sports performance prediction, data analysis, classification.

References

[1] Zhuravlev A. D., "Possible approach to the problem of predicting sports results using data analysis methods", 2021, № Intelligent Systems: Theory and Applications Volume 25, No. 1,, 63-69.

[2] https://www.sports.ru/tribuna/blogs/teilnahme/1098481.html, "Hockey analytics".

Автоассоциативные нейронные сети в задаче классификации с усеченным множеством

А. А. Хусаенов[1]

Рассматривается модель оценки риска неблагоприятного клинического исхода [2]. Предлагается применение метода обучения без учителя для задачи бинарной классификации с обучающим множеством, имеющим единственный ответ. Под усеченным множеством понимается не только малое количество примеров одного из случаев (благоприятного или неблагоприятного), но и данные полученные после очистки исходной таблицы. Некоторые результаты применения данной модели представлены в совместном исследовании [1], выполненном сотрудниками Национального медицинского исследовательского центра терапии и профилактической медицины МЗ РФ и механико-математического факультета МГУ имени М.В. Ломоносова.

Ключевые слова: нейронные сети, обучение без учителя, автоассоциативные нейронные сети, неблагоприятный клинический исход.

1. Введение

В исследовании [1] была сформулирована задача бинарной классификации клинических событий на основе анализов пациента. Рассматриваются события 2-х типов: благоприятное и неблагоприятное. Данные были представлены таблицей, содержащей 5062 строки и 66 столбцов.

Основной проблемой являлось малое количество данных неблагоприятных событий. Из 5062 доступных событий:

- 84 неблагоприятных
- 4978 благоприятных

[1] *Хусаенов Артем Азатович* — аспирант кафедры математической теории интеллектуальных систем мех.-мат. ф-та МГУ; e-mail: a.khusaenov@mail.ru

Khusaenov Artem Azatovich — postgraduate student, Moscow State University, faculty of Mechanics and Mathematics, Mathematical Theory of Intelligent Systems department

[2] работа выполнена при поддержке РФФИ грант № 19-29-01051 «Разработка алгоритмов принятия решений для управления рисками неблагоприятных клинических событий в высокотехнологичной медицинской организации на основе технологии data mining»

Так как неблагоприятных событий мало, то при обучении модели на данном множестве возникает высокая вероятность ошибки первого рода (больной пациент признан здоровым). Возможно обучить модель со смещением наибольшей ошибки в зону второго рода (здоровый пациент признан больным). При этом ошибка останется допустимой. В таком случае допустимая доля здоровых пациентов будет лишний раз показана врачу.

Модель обучается на благоприятных событиях. При обучении выделяется главная компонента множества, а разделяющая поверхность сдвигается к границам этого множества. В случае возникновения отклонения модели выше некоторого порогового значения, событие признается неблагоприятным.

2. Постановка задачи и модель

Набор признаков, методы предобработки данных и устранения пропусков подробно описаны в исследовании [1]. Обозначим:

- $Y = (y_1, y_2, ..., y_m)$ - множество ответов о событиях
- $X_i^n = (x_{1_i}, x_{2_i}, ..., x_{n_i})$ - множество из n признаков события i
- $Y^0 = \{y_i = 0 \mid y_i \in Y, i = \overline{1, m}\}$ - класс благоприятных событий
- $Y^1 = \{y_i = 1 \mid y_i \in Y, i = \overline{1, m}\}$ - класс неблагоприятных событий

Объем данных после устранения и обработки пропусков:

- $n = 23$
- $|Y^0| = 2893$
- $|Y^1| = 66$

Модель
Применялась автоассоциативная нейронная сеть [2] (рис. 1):

- 23 нейрона входного слоя ($n = 23$)
- 12 нейронов скрытого слоя
- 23 нейрона выходного слоя ($n = 23$)

В качестве ответа (учителя) используется $X_i' = (x'_{1_i}, x'_{2_i}, ..., x'_{23_i})$ - дубликат множества признаков X_i. То есть вместо решения задачи классификации $F : X_i \to y_i$ имеем $F : X_i \to X_i'$.

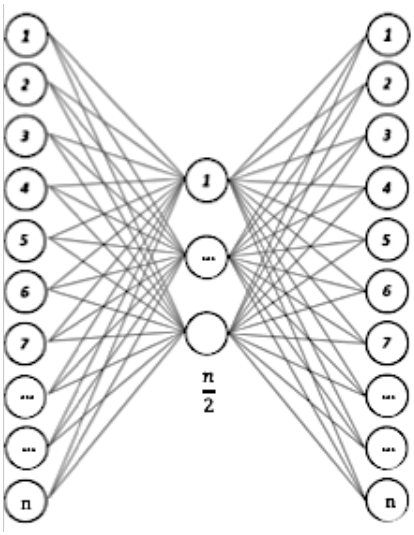


Рис. 1. архитектура нейронной сети

Нейронная сеть обучалась на множестве Y^0 и минимальное отклонение обучения по каждому выходному нейрону принималось как пороговое значение классификации. После обучения сети некоторое тестовое событие y_k оценивалось как:

- $y_k = 1$, если возникло отклонение выше порогового в одном из выходных нейронов
- $y_k = 0$, если отклонения по всем выходным нейронам не превышало порогового

Функции активации - сигмоидальные, функция потерь - разность по каждому выходному нейрону. Обучение производилось методом обратного распространения ошибки.

Результаты

Тестовое множество состояло из 132 событий (все события Y^1 и, соизмеримо, 66 событий из Y^0). Обучающее множество состояло из 2827 событий (все остальные события Y^0). На тестовом множестве достигнуто 100% точности.

При добавлении 12 несущественных признаков [1] достигнута точность 97%, где 3% - ошибка второго рода (4 здоровых пациента признаны больными).

Необходимо учитывать, что результаты получены на относительно небольшом тестовом множестве. Для объективной проверки необходим доступ к большему объему неблагоприятных событий и различные этапы повторных тестирований.

3. Смещение рода ошибки

Пусть A - множество нейронов скрытого слоя. Легко заметить, что в указанной модели (рис.2), для $F_1 : X_i \to A$ и $F_2 : A \to X_i'$ выполняется

$$F_2(A) = F_1^{-1}(X) + E^n, \tag{1}$$

где $E^n = (e_1, e_2, ..., e_n)$ - ошибка обучения

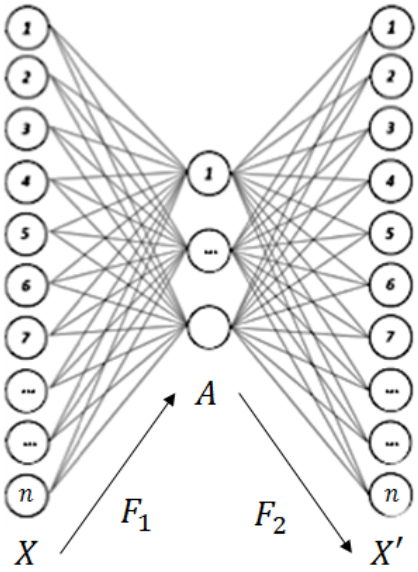


Рис. 2. Отображения в автоассоциативной нейронной сети

Будем считать, что нейронная сеть произвела обучение, если $E^n = \sigma^n$, где $\sigma^n = (\sigma_1, \sigma_2, \sigma_3, ..., \sigma_n)$ - неустранимая ошибка обучения [3].

Скрытый слой A значительно меньшей размерности (рис.2) усиливает значения, позволяющие осуществить обратное отображение $F_2(A)$ с наименьшей ошибкой $E^n \to \sigma^n$, и ослабляет значения вносящие шум. Тогда $F_2(A)$ при $E^n = \sigma^n$ будем называть *главной компонентой* множества X^n (рис. 3).

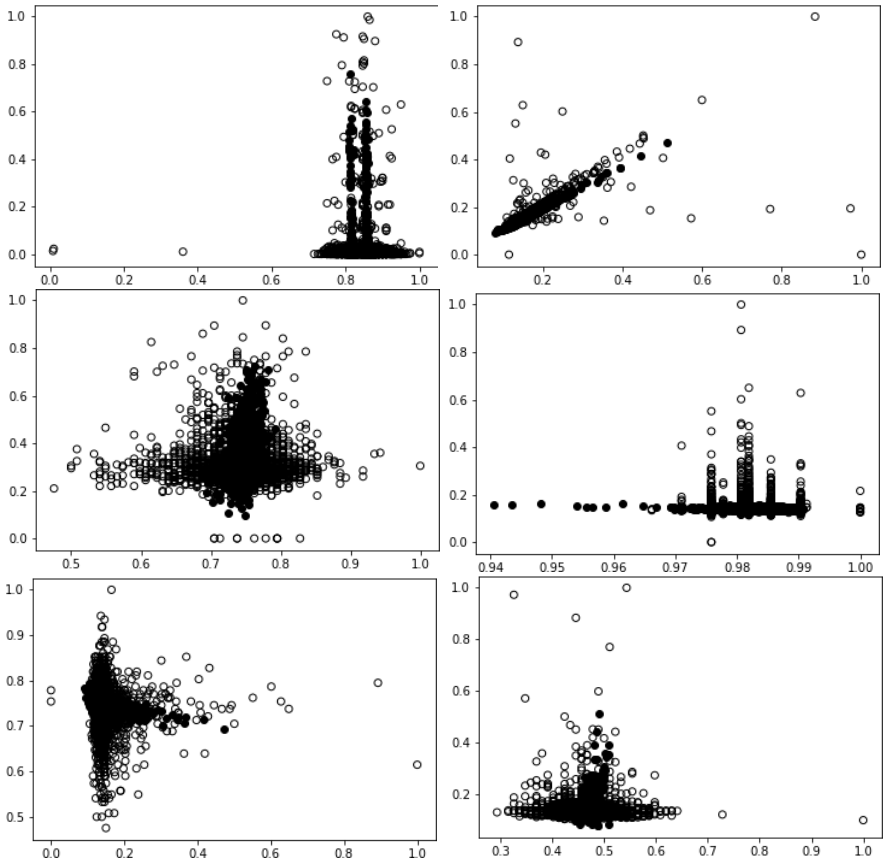
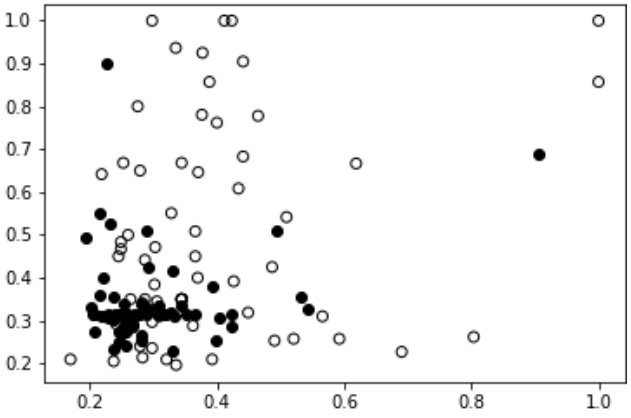


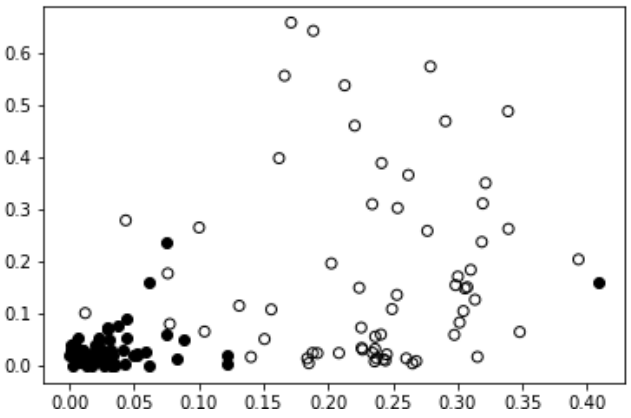
Рис. 3. главная компонента множества X^n в 2-мерных срезах (*черные точки* - $F_2(A)$, *белые точки* - X)

Классификация тестового события y_k возможна в пространстве ошибок за счет оценки отклонений от главной компоненты $F_2(A)$. Поскольку обучение производилось на множестве Y^0, то предполагается, что отклонение для событий Y^1 будет выше.

В пространстве ошибок события Y^0 будут сконцентрированы в зоне нуля, а Y^1 распределены по всему пространству. Произвести классификацию на пространстве ошибок будет проще, чем на пространстве признаков (рис. 4)



(а)



(б)

Рис. 4. (а) - пространство признаков в одном из 2-мерных срезов
(б) - пространство ошибок для среза (а)
(*черные точки* - Y^0, *белые точки* - Y^1)

Классификация в пространстве ошибок (рис.4(б)) может быть реализована отдельными методами. В данной задаче было достаточно порогового условия (рис.5) вида

$$y_k = \begin{cases} 1, & \exists e_j > \sigma_j,\ e_j \in E_k \\ 0, & \forall e_j < \sigma_j,\ e_j \in E_k \end{cases} \tag{2}$$

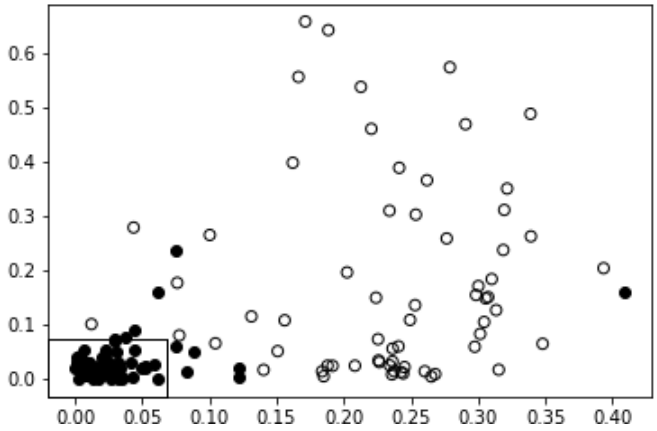


Рис. 5. Классификация в пространстве ошибок
(*черные точки* - Y^0, *белые точки* - Y^1)

Ошибки второго рода здесь - это выбросы класса Y^0. Уменьшение количества выбросов возможно за счет увеличения данных класса Y^0 (главная компонента множества будет сформирована более точно). При этом увеличить объем данных по здоровым пациентам проще, чем по больным.

Необходимо заметить, что на рис. 4 и рис. 5 представлены 2-мерные срезы. При классификации в многомерном виде ошибка значительно ниже (п.2 - результаты)

4. Отдельные замечания

Замечание 1. При обучении нейронной сети вида $F : X_i \to y_i$ на подобном множестве (п.1) риск переобучения крайне велик. Так как данных Y^1 значительно меньше Y^0, то разделяющие поверхности могут распределиться как показано на рис.6(б).

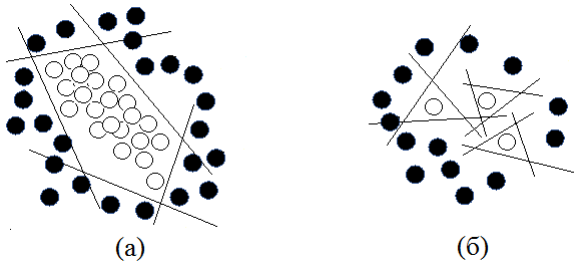


Рис. 6. (а) - допустимая обобщающая способность (б) - переобучение

Замечание 2. Данные должны быть нормализованы к области допустимых значений функций активации нейронов выходного слоя.

Замечание 3. Количество нейронов скрытого слоя A в данном случае подобрано эмпирически.

Выражаю благодарность проф. Рыжову А.П. и доц. Строгалову А.С. за обсуждения работы и ценные замечания.

Список литературы

[1] Горный Б.Э., Рыжов А.П., Строгалов А.С., Журавлев А.Д., Хусаенов А.А., Шергин И.А., Фещенко Д.А., Абдуллаев А.М., Концевая А.В., "Оценка риска неблагоприятного клинического исхода методами углубленного анализа данных", *Интеллектуальные системы. Теория и приложения*, **2**:25 (2021).

[2] Hinton G.E. , Salakhutdinov R.R., "Reducing the Dimensionality of Data with Neural Networks", *Science*, **313** (2006), 504–507.

[3] Stuart Geman, Bienenstock E., Doursat R., "Neural networks and the bias/variance dilemma", *Neural Computation*, **1**:4 (1992), 1–58.

Autoassociative neural networks in a classification problem with truncated dataset

Khusaenov A.A.

The adverse clinical outcome risk assessing model is considering. It is proposed the unsupervised learning method application for a binary classification problem with a single answer training set. A truncated set is a dataset with a small examples number of one of the classes (favorable or unfavorable). A truncated set is also the data obtained after original table clearing. Some results of this model application are presented in a research [1] conducted by the scientists of the National Research Center for Therapy and Preventive Medicine of the Ministry of Health of the Russian Federation and the scientists of the Faculty of Mechanics and Mathematics of the Lomonosov Moscow State University. It is proposed the general method for such problems.

Keywords: neural networks, unsupervised learning, Autoassociative neural networks, autoencoder, adverse clinical outcome.

References

[1] Gornyi B.E., Ryjov A.P., Strogalov A.S.,Zhuravlev A.D., Khusaenov A.A., Shergin I.A., Feshchenko D.A., Abdullaev A.M., Kontsevaya A.V., "The adverse clinical outcome risk assessment by in-depth data analysis methods", *Intelligent systems. Theory and applications*, **2**:25 (2021) (In Russian).

[2] Hinton G.E. , Salakhutdinov R.R., "Reducing the Dimensionality of Data with Neural Networks", *Science*, **313** (2006), 504–507.

[3] Stuart Geman, Bienenstock E., Doursat R., "Neural networks and the bias/variance dilemma", *Neural Computation*, **1**:4 (1992), 1–58.

О построении явной архитектуры нейронной сети, приближающей кусочно-линейные функции

В. Г. Шишляков[1]

В работе рассматривается вопрос о нахождении оценки сверху параметров архитектуры нейронной сети, которая хорошо приближает зависимости, описывающиеся кусочно-линейными функциями. Основной результат работы заключен в теореме, утверждающей, что любую наперед заданную кусочно-линейную функцию можно приблизить сколь угодно точно нейронной сетью с активационными функциями сигмоидного типа на достаточно объемном множестве. Доказательство данной теоремы конструктивно, то есть в ней строится архитектура нейронной сети, удовлетворяющая вышеописанным свойствам.

Ключевые слова: схемы функциональных элементов, нейронные сети, архитектура, аппроксимация функций, оценка сверху, кусочно-линейные функции.

1. Введение

Истоки подобных задач восходят еще к 1900 году, когда Д. Гильберт сформулировал список существенных проблем математики, в котором под номером 13 был вопрос о представимости функции n переменных в виде суперпозиции функций меньшего числа переменных.

В пятидесятых годах XIX века А.Н. Колмогоров [1], [2], [3] и В.И. Арнольд [4], [5] показали, что любую непрерывную функцию n переменных можно представить в виде суперпозиции одноместных функций и операции сложения.

В дальнейшем про представления, описанные в [1], [2], [3], [4], [5] вспомнили во время развития искусственных нейронных сетей. Однако данные представления были доказаны в неконструктивной форме – для каждой непрерывной функции n переменных требовалось искать новые одноместные функции, участвующие в суперпозиции. При этом явного алгоритма поиска таких функций представлено не было. Поэтому данные представления не нашли применения в области нейронных сетей, но

[1] *Шишляков Владимир Геннадьевич* — аспирант каф. общих проблем управления мех.-мат. ф-та МГУ, e-mail: bolotmaks@yandex.ru.

Shishlyakov Vladimir Gennad'evich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of General Problems of Control.

направили исследователей в нужное направление – обоснование способностей нейронных сетей восстанавливать или приближать определенные классы функций.

Исследования в области восстановления и аппроксимации функций нейронными сетями начались с работ У. Мак-Каллока и У. Питтса [6], [7], в которых давалось описание математической модели нейрона, а также было доказано, что булевские функции и конечные автоматы могут быть представлены нейронными сетями.

Однако на тот момент было не ясно, как производить обучение нейронных сетей, то есть как производить настройку их синаптических весов. Первый алгоритм обучения нейронных сетей был разработан Ф. Розенблаттом [8], [9]. Им была разработана модель нейронной сети, названная им перцептроном, и описан алгоритм обучения такой модели.

Было показано, что некоторые задачи могут быть решены перцептронами Розенблатта эффективнее, чем компьютерами с классической архитектурой. Однако в дальнейшем М. Минский и С. Пейперт [10] выяснили, что область применимости перцептронов имеет серьезные ограничения. В частности, ими было показано, что для некоторых задач, которые могут быть решены перцептроном Розенблатта, может потребоваться либо очень большое число нейронов, либо очень большое количество времени.

Ограничения [10] были сняты при замене функций активации нейронов с пороговых на сигмоидные. В конце восьмидесятых Г. Цыбенко [11], К. Фунахаши [12] и К. Хорник [13] независимо показали, что любую непрерывную на компакте $K \subset \mathbb{R}^n$ функцию f можно аппроксимировать в равномерной метрике многослойной нейронной сетью с линейными функциями активации в последнем слое и функциями сигмоидного типа во внутренних слоях. При этом в работах [11] и [13] были получены результаты не только для непрерывных функций, но и для функций пространства L_1.

Вместе с исследованиями [14], [15], [16], в которых был открыт и развит алгоритм обратного распространения ошибки, работы [11], [12], [13] дали теоретическое обоснование разумности использования нейронных сетей с сигмоидальными функциями активации и их автоматического обучения при помощи алгоритма обратного распространения ошибки.

Однако доказательства в работах [11], [12], [13] были неконструктивны, при этом не уточнялось, какое количество нейронов требуется взять в каждом слое имеющейся нейронной сети, чтобы приблизить заранее выбранную непрерывную функцию. Таким образом, вопрос о выборе разумной архитектуры нейронной сети оставался исследованным не до конца, так как в работах указывалось достаточное число слоев в нейронных сетях, но не количество нейронов в каждом слое.

Кроме того, в данных работах аппроксимации осуществлялись ступенчатыми функциями, которые позволяют с легкостью решать задачи классификации и управления, так как решения подобных задач основываются на аппроксимации кусочно-постоянных функций, но бывают крайне неудобными, например, в задачах регрессии. Так, к примеру, для приближения линейной функции на компакте при помощи ступенчатой функций с приемлемым качеством потребуется тем больше нейронов, чем точнее требуется приближение. К тому же, даже при малейшем выходе точки пространства входных данных за компакт K, качество приближения функции нейронной сетью, построенной подобным образом, может стремительно падать. Это бывает неудобно, если предполагается, что входные данные, на которых будет использоваться нейронная сеть, не имеют каких-либо ограничений.

В 2003 году в работе Д.В. Алексеева [17] было показано, что в интегральной метрике с весом Чебышева-Эрмита возможно приближение произвольной измеримой по Лебегу функции n переменных двухслойной нейронной сетью, причем функции активации первого слоя могут быть заданы заранее, а второго – линейны. Однако в работе [17] не указывалась какая-либо оценка числа нейронов в каждом слое нейронной сети, а интегральная метрика с весом Чебышева-Эрмита являлась более слабой по сравнению с равномерной метрикой.

Наконец, в 2009 году в работе В.С. Половникова [18], нейронные сети, построенные из нейронов модели Мак-Каллока и Питтса, были рассмотрены с точки зрения схем функциональных элементов [19]. Также в работе [18] было доказано, что любую кусочно-линейную функцию (не обязательно непрерывную) можно представить в виде схемы функциональных элементов над базисом $B_1 = \{c, \gamma \cdot x, \sum_n(x_1, ..., x_n), \theta(x), F(x, y)\}$, где $F(x,y) = \begin{cases} x, & y \geqslant 0 \\ 0, & y < 0 \end{cases}$, $\theta(x) = \begin{cases} 1, & x \geqslant 0 \\ 0, & x < 0 \end{cases}$. Доказательство в работе [18] было конструктивным, причем в доказательстве были даны оценки количества функциональных элементов, достаточных для восстановления любой кусочно-линейной функции.

Однако базис B_1 не подходил для обучения нейронных сетей градиентными методами из-за того, что в нем имелись функции со всюду нулевыми производными.

Изначальная цель данной работы - решить описанную проблему. Но в дальнейшем цель была расширена до исследования более обобщенного базиса $B_2 = \{c, \gamma \cdot x, \sum_n(x_1, ..., x_n), \prod_n(x_1, ..., x_n), \psi(x)\}$, где $\psi(x)$ - это некоторая произвольная функция сигмоидного типа [17].

Базис B_2 был выбран исходя из следующих умозаключений. Если возможно восстановление произвольной кусочно-линейной функции схемами функциональных элементов над базисом B_1, то, в силу того, что

$F(x, y) = x \cdot \theta(y)$, такое восстановление возможно и схемами функциональных элементов над базисом
$B_3 = \{c, \gamma \cdot x, \sum_n(x_1, ..., x_n), \prod_n(x_1, ..., x_n), \theta(x)\}$. При замене в базисе B_3 функции $\theta(x)$ на $\sigma(x) = \frac{1}{1+e^{-x}}$, получается базис, в котором все элементы являются дифференцируемыми функциями, и в котором можно надеяться хотя бы на приближение кусочно-линейных функций схемами функциональных элементов (так как функция $\sigma(x)$ является аппроксимацией функции $\theta(x)$). Но далее возникает интерес в рассмотрении еще более широкого класса базисов, получаемых из B_3 заменой функции $\theta(x)$ на произвольную функцию $\psi(x)$ сигмоидного типа, и исследования вопроса аппроксимации кусочно-линейных функций схемами функциональных элементов в полученных базисах.

Таким образом, базис B_2 является более близким к классическому базису Мак-Каллока и Питтса, а также более подходящим для общепринятого подхода обучения методом обратного распространения ошибки (в том случае, когда функция $\psi(x)$ является дифференцируемой).

Очевидно, что в базисе B_2 задача восстановления кусочно-линейной функции в общем виде (когда $\psi(x)$ - произвольная сигмоидная функция) невозможна. Поэтому в работе решается задача аппроксимации, а именно, доказывается, что в базисе B_2 любую кусочно-линейную функцию можно приблизить на некотором компакте нейронной сетью со сколь угодно большой точностью.

Причем доказательство конструктивно, а в построенной нейронной сети указывается оценка сверху количества нейронов в каждом слое сети. В силу того, что в данной работе вместо нелинейной сложности и глубины [18] используется другая оценка построенной нейронной схемы, потребовалось ввести несколько дополнительных определений, уточняющих понятия нейрона и слоя нейронной сети в терминах схем функциональных элементов.

Также стоит отметить, что в случае небольшого выхода точки пространства входных данных за пределы компакта, внутри которого производилось обучение нейронной сети, данная модель будет работать примерно с той же погрешностью, что и внутри компакта. Хотя погрешность нейронной сети постепенно увеличивается при удалении точки пространства входных данных от компакта, на котором оценивалась точность сети.

Таким образом, основной результат данной работы является конструктивным аналогом теоремы Цыбенко [11], но только для нейронных сетей, построенных над видоизмененным базисом, который удобен как для обучения нейронных сетей классическими градиентными методами (при выборе, например, $\psi(x) = \sigma(x) = \frac{1}{1+e^{-x}}$), так и для решения задач регрессии при помощи нейронных сетей. При этом в теореме, рассмот-

ренной в данной работе, дается оценка количества нейронов на каждом слое, при котором можно подобрать веса нейронной сети так, чтобы она приближала выбранную кусочно-линейную функцию с заданной точностью.

2. Основные понятия и формулировка результата

Для начала определим основные понятия, которые используются в данной статье. Нейронные сети можно рассматривать с двух точек зрения. С одной стороны на них можно смотреть, как на функции с большим количеством подбираемых параметров (весов), а с другой стороны – как на схемы, реализующие эти функции (тоже с большим количеством подбираемых параметров). Поэтому вполне логично при рассмотрении искусственных нейронных сетей со схематической точки зрения ввести понятие базиса нейронной сети. Помимо этого, следуя ссылкам [18], [19] и [20], напомним определения и обозначения основных объектов нейронных сетей.

Определение 1. Базисом будем называть некоторый набор функциональных элементов, где каждый функциональный элемент представляет из себя пару $(S, f(x_1, ..., x_n))$, в которой $f(x_1, .., x_n) : \mathbb{R}^n \longrightarrow \mathbb{R}$, а S - сопоставленный ей графический объект с n входными стрелками и одной выходной (кратко – входы и выход объекта S). Входам объекта S приписаны слева направо переменные $x_1, ..., x_n$ функции f, выходу приписан выход функции f.

Стоит отметить, что базис нейронной сети не является базисом с обычной математической точки зрения, так как он часто является избыточным (то есть существуют элементы базиса, выражаемые через другие элементы базиса).

В классической модели Мак-Каллока Питтса [6] принимается базис (1), приведенный ниже.

$$B_1 = \{c, \gamma \cdot x, \sum_n (x_1, ..., x_n), \theta(x)\} \tag{1}$$

В базисе (1) используются следующие классы функций:

1) Сумматор - каждая функция данного класса суммирует определенное количество входных аргументов и обозначается $\sum_n (x_1, ..., x_n)$.

2) Константа - каждая функция данного класса выдает константу (у данной функции нет входных аргументов, каждый раз, когда на схему приходят входные сигналы, константа выдает одинаковое заранее определенное значение).

3) Усилитель (умножение на константу) - в данном классе каждая функция умножает пришедший на вход аргумент x на фиксированную константу γ.

4) Функция активации - в модели Мак-Каллока Питтса эта функция единственна и выглядит так
$\theta(x) = \begin{cases} 1, & x \geqslant 0 \\ 0, & x < 0 \end{cases}$.

Графические изображения данных элементов приведены на рис. 1 (а, б, в, г).

Определение 2. *Функцию* $\psi(x) : \mathbb{R} \longrightarrow \mathbb{R}$ *будем называть сигмоидной [13], [17], если она не убывает на* $\mathbb{R}$ *и выполняется, что* $\lim_{x \longrightarrow -\infty} \psi(x) = 0$, $\lim_{x \longrightarrow +\infty} \psi(x) = 1$.

Рассмотрим следующий базис:

$$B_2 = \{c, \gamma \cdot x, \sum_n (x_1, ..., x_n), \prod_n (x_1, ..., x_n), \psi(x)\} \tag{2}$$

Базис (2) отличается от классического базиса (1) тем, что в нем добавлено семейство функций $\prod_n (x_1, ..., x_n)$, которое определяется аналогично семейству $\sum_n (x_1, ..., x_n)$, а функция активации $\theta(x)$ заменена на $\psi(x)$ - произвольную сигмоидную функцию.

Функции $\prod_n (x_1, ..., x_n)$ будем обозначать на схемах, как на рис. 1 (д), а $\psi(x)$ - как на рис. 1 (е).

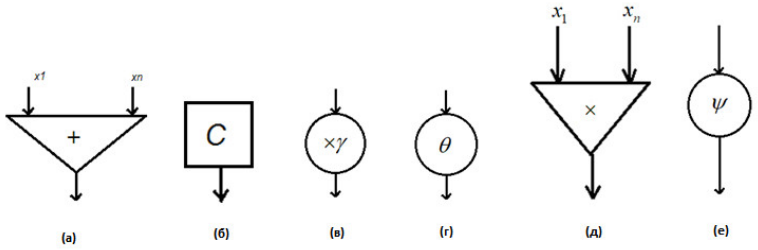


Рис. 1. Функциональные элементы рассматриваемых базисов

Подробнее о том, как строятся схемы функциональных элементов из элементов любого из рассматриваемых в данной работе базисов, описано в [18].

Определение 3. *Нейроном в базисе* (2) *будем называть всякую схему, вычисляющую одну из следующих функций*

$$\varphi(\sum_{i=1}^{n}(w_i \cdot x_i) + c) \tag{3}$$

или

$$\varphi(\prod_{i=1}^{n}(w_i \cdot x_i) + c) \tag{4}$$

В формулах (3) *и* (4) *функция* $\varphi(x)$ *называется активационной функцией. В качестве* $\varphi(x)$ *может быть выбрана либо* $\psi(x)$*, либо* x*.*

В дальнейшем, для краткости, нейроны в схемах будем обозначать, как на рис. 2 (а, б). Если же функция активации φ является тождественной, то такие нейроны будем обозначать, как на рис. 2 (в, г).

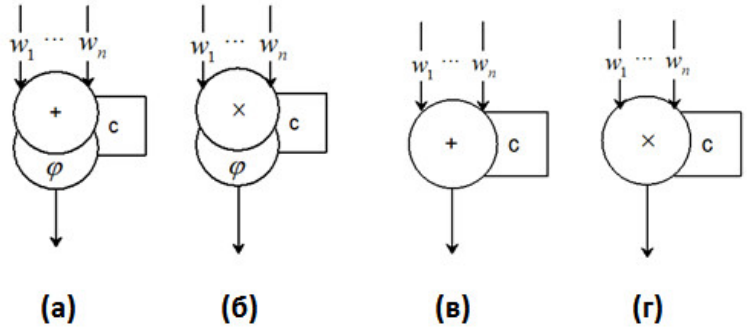


Рис. 2. Графические обозначения нейронов (3) и (4)

Следует отметить, что самой распространенной [20] схемой нейронов (3) и (4) является схема, изображенная на рис. 3. В дальнейших выкладках именно такие схемы будут заменяться на обозначения нейронов.

Определение 4. *Все нейроны вида* (3) *будем называть нейронами-сумматорами, а нейроны вида* (4) *- нейронами-продукторами.*

Определение 5. *Введем понятие слоя нейронной сети.*

1) *Множество нейронов, все входы которых не подсоединены ни к каким выходам каких-либо функциональных элементов, назовем нейронами первого слоя.*

2) *Пусть определено множество нейронов* n*-го слоя. Тогда* $n+1$*-ым слоем назовем все нейроны, для которых выполняются одновременно следующие условия:*

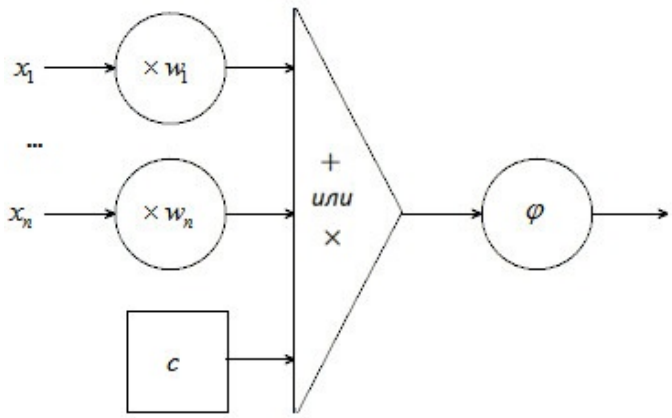


Рис. 3. Схема нейрона

а) Хотя бы один вход подсоединен к выходу нейрона n-го слоя

б) Все оставшиеся входы подсоединены либо к выходам нейронов из слоев $\{1, 2, ..., n\}$, либо не подсоединены ни к каким нейронам (тогда считается, что на вход принимаются входные данные).

Таким образом, можно комбинировать не отдельные элементы базиса схемы, а целые схемы, реализующие нейроны. При комбинации нейронов друг с другом, будут получаться различные функции, которые и будут исследоваться в данной работе. Особый интерес для данного исследования представляют так называемые кусочно-линейные функции. Дадим их определение, следуя определениям из [18].

Определение 6. *Пусть $\overline{x} = (x_1, ..., x_n)$ (входные сигналы), $l_1, ..., l_k$ - некоторые гиперплоскости, определяемые выражениями $l_i = \{x \in \mathbb{R}^n | \langle \overline{a_i}, \overline{x} \rangle + c_i = 0\}$ (здесь $\langle \overline{a_i}, \overline{x} \rangle = \sum_{j=1}^{n} a_{ij} \cdot x_j$ - скалярное произведение векторов, c_i константа, $\overline{a_i} \neq \overline{0}$). Также обозначим $l_i(\overline{x}) = \langle \overline{a_i}, \overline{x} \rangle + c_i$.*

Отметим, что все пространство $\mathbb{R}^n$ разбивается этими гиперплоскостями $l_1, ..., l_k$ на классы эквивалентности. Вектор-функция $\sigma(\overline{x}) = (sgn(\overline{a_1} \cdot \overline{x} + c_1), ..., sgn(\overline{a_k} \cdot \overline{x} + c_k))$ называется сигнатурой вектора $\overline{x}$ [18]. Каждая ее компонента показывает расположение точки $\overline{x}$ относительно соответствующей ей гиперплоскости из набора $\{l_1, ..., l_k\}$. Поэтому данная функция однозначно определяет, в каком куске пространства лежит точка $\overline{x} = (x_1, ..., x_n) \in \mathbb{R}^n$.

Сигнатурой класса будем называть сигнатуру любого вектора из этого класса. Определение корректно, так как у всех точек одного класса одинаковые сигнатуры [18].

Определение 7. *Пусть пространство $\mathbb{R}^n$ разбивается на классы $R^1, ..., R^s$ гиперплоскостями $l_1, ..., l_k$. Будем говорить, что $f(\overline{x}) \in PL$ (является кусочно-линейной), если $f(\overline{x})|_{R^j} = \overline{b_j} \cdot \overline{x} + d_j$ (то есть сужение на каждый из классов $R^1, ..., R^s$ является линейной функцией, $\overline{b_j}, d_j = const$).*

Также введем несколько полезных обозначений. Пусть $l_1, ..., l_k$ - гиперплоскости. Возьмем $\forall \xi > 0$ и рассмотрим множества
$L_{i,\xi} = \{\overline{x} \in \mathbb{R}^n | |l_i(\overline{x})| < \xi\}, i = 1, ..., k$. Обозначим $L_\xi = \bigcup\limits_{i=1}^{k} L_{i,\xi}$.

Выражением $O_R(\overline{x})$ будем обозначать окрестность радиуса R некоторой точки $\overline{x}$.

Основным результатом данной статьи является теорема, в которой утверждается, что любую кусочно-линейную функцию, заданную относительно гиперплоскостей $l_1, ..., l_k$, можно приблизить нейронной схемой над базисом (2) со сколь угодно большой точностью всюду вне множества L_ξ, где ξ может быть сколь угодно близким к нулю числом.

3. Основные результаты

Теорема 1 (о приближении кусочно-линейной функции)**.** *Пусть $l_1, ..., l_k$ - гиперплоскости, которые разбивают пространство $\mathbb{R}^n$ на s классов эквивалентности $R^1, ..., R^s$, из которых s' классов обладают такими сигнатурами $(sgn(\overline{a_1} \cdot \overline{x} + c_1), ..., sgn(\overline{a_k} \cdot \overline{x} + c_k))$, что $sgn(\overline{a_i} \cdot \overline{x} + c_i) \neq 0, i = 1, ..., k$, а $f(\overline{x})$ - кусочно-линейная функция, заданная над данными классами эквивалентности.*

Тогда $\forall \varepsilon > 0, \forall \xi > 0, \forall R > 0$ существует нейронная сеть $G(\overline{x})$ над базисом (2) такая, что выполняется $\sup\limits_{\overline{x} \in O_R(\overline{0}) \setminus L_\xi} |G(\overline{x}) - f(\overline{x})| < \varepsilon$. Причем данная нейронная сеть обладает следующей архитектурой:

1. На первом слое потребуется не более $2k$ нейронов-сумматоров, имеющих функцию активации $\varphi(x) = \psi(x)$;

2. На втором слое потребуется $2s'' \leqslant 2s'$ нейронов, из которых s'' нейронов имеют функцию активации $\varphi(x) = \psi(x)$, а остальные s'' нейронов – $\varphi(x) = x$. При этом каждый нейрон с функциями активации $\varphi(x) = x$ на данном слое принимает на вход кроме выходов нейронов предыдущего слоя вектор $\overline{x} = (x_1, ..., x_n)$, который является копией вектора, подающегося на нейроны входного слоя сети;

3. На третьем слое потребуется не более s' нейронов-продукторов с тождественной функцией активации;

4. На четвертом слое потребуется один нейрон-сумматор с тождественной функцией активации.

Доказательство. Зафиксируем произвольные $\xi > 0, R > 0$ и рассмотрим множество $O = O_R(\overline{0}) \setminus L_\xi$. Покажем, что $\exists G(\overline{x})$, реализуемая нейронной сетью над базисом (2), такая что $\sup\limits_{\overline{x} \in O} |G(\overline{x}) - f(\overline{x})| < \varepsilon$.

Зафиксируем произвольное $\varepsilon > 0$. Константу d положим такой, чтобы $\psi(x + d) > \frac{1}{2}$ при $x > 0$ и $\psi(x + d) \leqslant \frac{1}{2}$ при $x \leqslant 0$. Такая константа существует в силу того, что $\lim\limits_{x \to -\infty} \psi(x) = 0$, $\lim\limits_{x \to +\infty} \psi(x) = 1$ и ψ является неубывающей функцией. Положим $c > 1$ таким, чтобы $|\psi(c \cdot \xi + d) - \frac{1}{2}| \geq \varepsilon'$ и $|\psi(c \cdot (-\xi) + d) - \frac{1}{2}| \geq \varepsilon'$, где $\varepsilon' = \frac{1}{2} \cdot \frac{k}{(k+1)}$. Это всегда можно сделать, так как по условию $\lim\limits_{x \to -\infty} \psi(x) = 0$ и $\lim\limits_{x \to +\infty} \psi(x) = 1$.

Но тогда выполняется:

$$\psi(c \cdot l_i(\overline{x}) + d) \in \begin{cases} [\frac{1}{2} + \varepsilon', 1], & l_i(\overline{x}) \geqslant \xi \\ [0, \frac{1}{2} - \varepsilon'], & l_i(\overline{x}) \leqslant -\xi \end{cases} \tag{5}$$

$$\psi(c \cdot (-l_i(\overline{x})) + d) \in \begin{cases} [\frac{1}{2} + \varepsilon', 1], & l_i(\overline{x}) \leqslant -\xi \\ [0, \frac{1}{2} - \varepsilon'], & l_i(\overline{x}) \geqslant \xi \end{cases} \tag{6}$$

Рассмотрим теперь класс R^j, обладающий сигнатурой $(\sigma_1^j, ..., \sigma_k^j)$, где все $\sigma_i^j \neq 0$. Построим функцию $\sum\limits_{i=1}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x})$, где $\pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) = \begin{cases} \psi(c \cdot l_i(\overline{x}) + d), & \sigma_i^j = 1 \\ \psi(c \cdot (-l_i(\overline{x})) + d), & \sigma_i^j = -1 \end{cases} = \psi(\sigma_i^j \cdot c \cdot l_i(\overline{x}) + d)$.

Из (5) и (6) следует, что $\forall \overline{x} \in O$ выполняется:

$$\pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) \in [\frac{1}{2} + \varepsilon', 1], \text{если } sgn(l_i(\overline{x})) = \sigma_i^j \tag{7}$$

$$\pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) \in [0, \frac{1}{2} - \varepsilon'], \text{если } sgn(l_i(\overline{x})) \neq \sigma_i^j \tag{8}$$

Но тогда получаем, что $\forall \overline{x} \in O$, если $\overline{x} \in R^j$, то $(sgn(l_1(\overline{x})), ..., sgn(l_k(\overline{x}))) = (\sigma_1^j, ..., \sigma_k^j)$ и тогда из (7) получаем, что $\pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) \geqslant \frac{1}{2} + \varepsilon'$, $i \in \{1, ..., k\}$. А тогда $\sum\limits_{i=1}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) \geqslant \sum\limits_{i=1}^{k} (\frac{1}{2} + \varepsilon') = \frac{1}{2}k + \varepsilon' k$.

Если же $\forall \overline{x} \in O$, но $\overline{x} \notin R^j$, то вектор $(sgn(l_1(\overline{x}), ..., sgn(l_k(\overline{x})))$ отличается от вектора $(\sigma_1^j, ..., \sigma_k^j)$ хотя бы в одной компоненте. Пусть это компонента p, то есть $sgn(l_p(\overline{x})) \neq \sigma_p^j$. Тогда из (7) и (8) получаем, что $\pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) \leqslant 1,\ i \in \{1, ..., k\} \setminus \{p\}$ и $\pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) \leqslant \frac{1}{2} - \varepsilon',\ i = p$. Откуда $\sum\limits_{i=1}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) \leqslant \sum\limits_{i=1, i\neq p}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) + \pi_{sgn(l_p(\overline{x}))=\sigma_p^j}(\overline{x}) \leqslant k - 1 + \frac{1}{2} - \varepsilon' = k - \frac{1}{2} - \varepsilon'$.

Найдем условие на ε', при котором

$$\frac{1}{2}k + k\varepsilon' > k - \frac{1}{2} - \varepsilon' \tag{9}$$

$\frac{1}{2}k + k\varepsilon' > k - \frac{1}{2} - \varepsilon' \Leftrightarrow (k+1)\ \varepsilon' > \frac{1}{2}(k-1) \Leftrightarrow \varepsilon' > \frac{1}{2}\frac{(k-1)}{(k+1)}$. Но для взятого в начале доказательства ε' выполняется следующее неравенство:

$$\varepsilon' = \frac{1}{2}\frac{k}{(k+1)} > \frac{1}{2}\frac{(k-1)}{(k+1)}, \tag{10}$$

поэтому условие (9) выполняется.

Таким образом, взяв $\pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) = \psi(\sigma_i^j \cdot c \cdot l_i(\overline{x}) + d)$, получим, что $\forall \overline{x} \in O$ верно, что при $\overline{x} \in R^j$ сумма $\sum\limits_{i=1}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x})$ всегда больше той же суммы при $\overline{x} \notin R^j$.

Положим теперь $M = \frac{1}{2} \cdot ((\frac{1}{2}k + k\varepsilon') + (k - \frac{1}{2} - \varepsilon'))$. В силу (10) для выбранного ε' выполняется условие (9), из которого следует, что $\frac{1}{2}k + \varepsilon' k \neq k - \frac{1}{2} - \varepsilon'$. Поэтому $k - \frac{1}{2} - \varepsilon' < M < \frac{1}{2}k + \varepsilon' k$.

Рассмотрим функцию $\sum\limits_{i=1}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) - M$ и $\forall \overline{x} \in O$. Очевидно, что если $\overline{x} \in R^j$, то $\sum\limits_{i=1}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) - M > 0$, а если $\overline{x} \notin R^j$, то $\sum\limits_{i=1}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) - M < 0$.

Далее, взяв $m = min\left\{\frac{1}{2}k + k\ \varepsilon' - M, M - \left(k - \frac{1}{2} - \varepsilon'\right)\right\}$, получаем, что для $\forall \overline{x} \in O$ выполняется, что $\left|\sum\limits_{i=1}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) - M\right| \geqslant m$.

Теперь возьмем сколь угодно большое число $\mu > 0$ и $\forall \overline{x} \in O$. В силу того, что $\left|\sum\limits_{i=1}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) - M\right| \geqslant m$, получаем, что

$\frac{\mu}{m}\left(\sum\limits_{i=1}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) - M\right) \geqslant \mu$ при $\overline{x} \in R^j$ и

$\frac{\mu}{m}\left(\sum\limits_{i=1}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) - M\right) \leqslant -\mu$ при $\overline{x} \notin R^j$.

Обозначим $s(\xi) = \#\{R^j | R^j \cap O \neq \varnothing\}$. Очевидно, что $s(\xi) \in \mathbb{N} \cup \{0\}$: $s(\xi) \leqslant s' \leqslant s$, причем, функция $s(\xi)$ является не возрастающей. Другими словами, при уменьшении величины $\xi > 0$ соответствующее ей значение $s(\xi)$ не убывает.

Если $s(\xi) = 0$, то взятое $\xi > 0$ оказалось слишком большим и $O = \varnothing$. Поэтому любая функция подойдет в качестве $G(\overline{x})$, в том числе и реализуемая нейронной сетью с требуемой архитектурой и произвольными значениями ее параметров.

Поэтому здесь и далее будем полагать, что $s(\xi) \geqslant 1$, а, следовательно, $O \neq \varnothing$. Также, без ограничения общности, будем считать, что $\{R^j | R^j \cap O \neq \varnothing\} = \{1, ..., s(\xi)\}$.

Положим $\tau(\varepsilon, R, \xi) = \frac{\varepsilon}{s(\xi) \cdot \max\limits_{\overline{x} \in O} |f(\overline{x})| + 1} < \infty$. Обозначим

$\Psi_j(\overline{x}) = \psi\left(\frac{\mu}{m}\left(\sum\limits_{i=1}^{k} \pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) - M\right)\right)$, $j \in \{1, ..., s(\xi)\}$. После чего возьмем μ таким, чтобы для $\forall \overline{x} \in O$ и $\forall j \in \{1, ..., s(\xi)\}$ были выполнены следующие условия:

$$|\Psi_j(\overline{x}) - 1| < \tau(\varepsilon, R, \xi) \text{ при } \overline{x} \in R^j \tag{11}$$

и

$$|\Psi_j(\overline{x}) - 0| < \tau(\varepsilon, R, \xi) \text{ при } \overline{x} \notin R^j. \tag{12}$$

Далее рассмотрим функцию

$$G(\overline{x}) = \sum_{j=1}^{s(\xi)} (\overline{b_j} \cdot \overline{x} + d_j) \cdot \Psi_j(\overline{x}) \tag{13}$$

Возьмем $\forall \overline{x} \in O$. Очевидно, что $\exists p \in \{1, ..., s(\xi)\} : \overline{x} \in R^p$. Тогда верны следующие рассуждения:

$$|G(\overline{x}) - f(\overline{x})| = |\sum_{j=1}^{s(\xi)}(\overline{b_j}\cdot\overline{x}+d_j)\cdot\Psi_j(\overline{x}) - f(\overline{x})|_{R^p}| =$$

$$\left|\sum_{\substack{j=1,\\ j\neq p}}^{s(\xi)}(\overline{b_j}\cdot\overline{x}+d_j)\cdot\Psi_j(\overline{x}) + (\overline{b_p}\cdot\overline{x}+d_p)\cdot(\Psi_p(\overline{x})-1)\right| <$$

$$\left|\sum_{\substack{j=1,\\ j\neq p}}^{s(\xi)}\max_{\overline{x}\in O}|f(\overline{x})|\cdot\tau(\varepsilon,R,\xi)\right| + \left|\max_{\overline{x}\in O}|f(\overline{x})|\cdot\tau(\varepsilon,R,\xi)\right| =$$

$$\tau(\varepsilon,R,\xi)\cdot\sum_{j=1}^{s(\xi)}\max_{\overline{x}\in O}|f(\overline{x})| = \tau(\varepsilon,R,\xi)\cdot s(\xi)\cdot\max_{\overline{x}\in O}|f(\overline{x})| <$$

$$\tau(\varepsilon,R,\xi)\cdot(s(\xi)\cdot\max_{\overline{x}\in O}|f(\overline{x})|+1) = \varepsilon \quad (14)$$

Из (14) немедленно следует, что $\sup\limits_{\overline{x}\in O}|G(\overline{x}) - f(\overline{x})| < \varepsilon$.

Учитывая, что $\pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) = \psi(\sigma_i^j\cdot c\cdot l_i(\overline{x})+d)$, формулу (13) для $G(\overline{x})$ можно переписать в следующем виде:

$$\sum_{j=1}^{s(\xi)}(\overline{b_j}\cdot\overline{x}+d_j)\cdot\psi\left(\frac{\mu}{m}\left(\sum_{i=1}^{k}\pi_{sgn(l_i(\overline{x}))=\sigma_i^j}(\overline{x}) - M\right)\right) =$$

$$\sum_{j=1}^{s(\xi)}(\overline{b_j}\cdot\overline{x}+d_j)\cdot\psi\left(\frac{\mu}{m}\cdot\sum_{i=1}^{k}\psi(\sigma_i^j\cdot c\cdot l_i(\overline{x})+d) - \frac{\mu}{m}\cdot M\right) \quad (15)$$

Обозначив $\Delta = -\frac{\mu}{m}\cdot M = const$, $\delta_{i,j}^{+} = \begin{cases}\frac{\mu}{m}, & \text{если } \sigma_i^j = 1\\ 0, & \text{если } \sigma_i^j = -1\end{cases}$ и $\delta_{i,j}^{-} = \begin{cases}0, & \text{если } \sigma_i^j = 1\\ \frac{\mu}{m}, & \text{если } \sigma_i^j = -1\end{cases}$, в выражении (15), получаем следующее представление для функции $G(\overline{x})$:

$$G(\overline{x}) = \sum_{j=1}^{s(\xi)} (\overline{b_j} \cdot \overline{x} + d_j) \cdot \psi(\sum_{i=1}^{k} \frac{\mu}{m} \cdot \psi(\sigma_i^j \cdot l_i(\overline{x}) + d) + \Delta) =$$

$$\sum_{j=1}^{s(\xi)} (\overline{b_j} \cdot \overline{x} + d_j) \cdot \psi(\sum_{i=1}^{k} \delta_{i,j}^{+} \cdot \psi(c \cdot l_i(\overline{x}) + d) +$$

$$\sum_{i=1}^{k} \delta_{i,j}^{-} \cdot \psi(-c \cdot l_i(\overline{x}) + d) + \Delta) \quad (16)$$

Далее, делая следующие обозначения в (16):

$$c \cdot l_i(\overline{x}) + d = c \cdot (a_{i1}x_1 + ... + a_{in}x_n) =$$
$$(ca_{i1})x_1 + ... + (ca_{in})x_n + (c \cdot a_{i0} + d) = p_i(\overline{x})$$

$$-c \cdot l_i(\overline{x}) + d = -c \cdot (a_{i1}x_1 + ... + a_{in}x_n) =$$
$$(-ca_{i1})x_1 + ... + (-ca_{in})x_n + (-c \cdot a_{i0} + d) = q_i(\overline{x}),$$

получаем формулу (17):

$$G(\overline{x}) = \sum_{j=1}^{s(\xi)} (\overline{b_j} \cdot \overline{x} + d_j) \cdot \psi(\sum_{i=1}^{k} \delta_{i,j}^{+} \cdot \psi(p_i(\overline{x})) +$$

$$+ \sum_{i=1}^{k} \delta_{i,j}^{-} \cdot \psi(q_i(\overline{x})) + \Delta) \quad (17)$$

Обозначим для лаконичности $(c \cdot a_{ij}) = \alpha_{ij}^{+}, (-c \cdot a_{ij}) = \alpha_{ij}^{-}$, $i = 1, ..., k, j = 1, ..., n$, а также $c \cdot a_{i0} + d = \alpha_{i0}^{+}, (-c \cdot a_{i0} + d) = \alpha_{i0}^{-}$. Тогда верно, что:

$$p_i(\overline{x}) = \alpha_{i1}^{+}x_1 + ... + \alpha_{in}^{+}x_n + \alpha_{i0}^{+} \quad (18)$$

$$q_i(\overline{x}) = \alpha_{i1}^{-}x_1 + ... + \alpha_{in}^{-}x_n + \alpha_{i0}^{-} \quad (19)$$

Изобразим схему функциональных элементов для выражения, стоящего справа от знака равенства в (17), учитывая (18) и (19), а также заменяя группы функциональных элементов, которые можно объединить в нейроны, на обозначения этих нейронов (рис. 4).

Таким образом, имеем следующую архитектуру нейронной сети. Сеть состоит из четырех слоев (не считая входного слоя), так как если рассмотреть самые длинные пути от входа к выходу, то на каждом таком

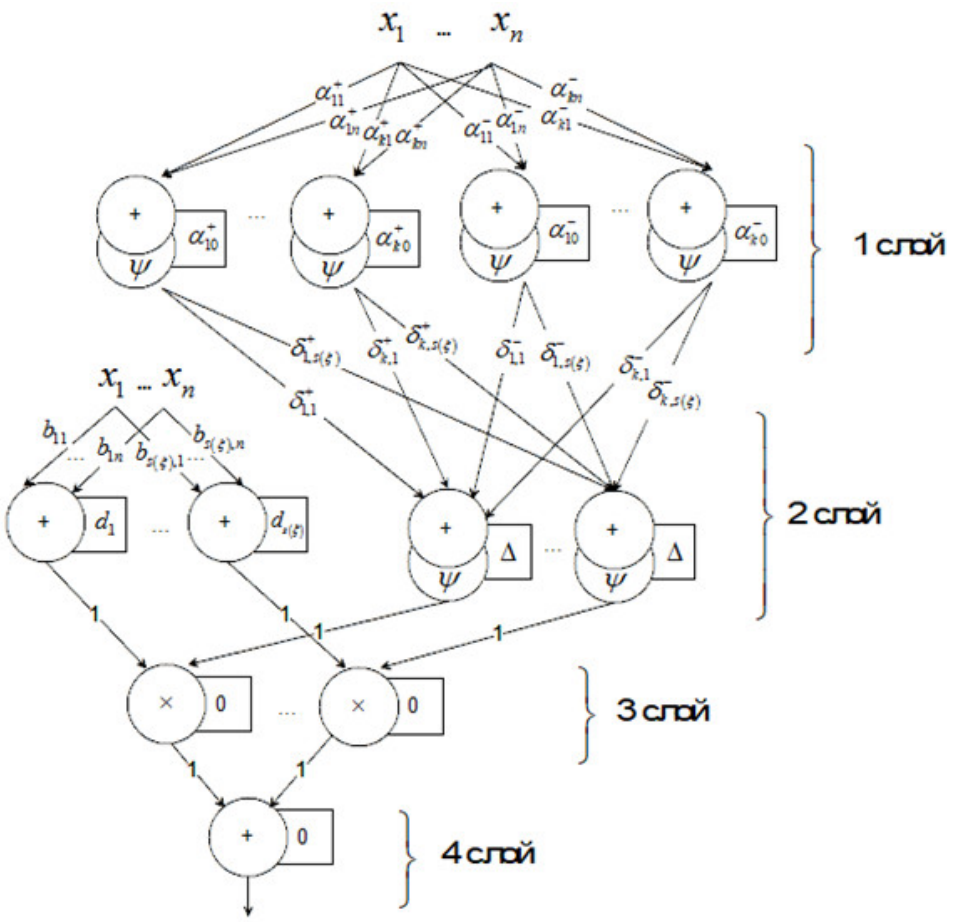


Рис. 4. Нейронная схема функции $G(\overline{x})$

пути встретится ровно 4 элемента такого вида, как на рисунках 8 и 9. Причем отметим, что:

1) На первом слое потребуется не более $2k$ нейронов-сумматоров, каждый из которых имеет функцию активации $\varphi(x) = \psi(x)$

2) На втором слое потребуется $2s(\xi) \leqslant 2s'$ нейронов-сумматоров, из которых $s(\xi)$ штук имеют функцию активации $\varphi(x) = \psi(x)$, а еще $s(\xi)$ штук – тождественную функцию активации, то есть $\varphi(x) = x$

3) На третьем слое потребуется $s(\xi) \leqslant s'$ нейронов-продукторов с тождественной функцией активации

4) На четвертом слое потребуется один нейрон-сумматор с тождественной функцией активации

Данная архитектура полностью соответствует архитектуре, заявленной в начале доказательства. □

Список литературы

[1] Колмогоров А.Н., "О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных", *Докл. АН СССР*, **108** (1956), 2.

[2] Колмогоров А.Н., "О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного и сложения", *Докл. АН СССР*, **114** (1957), 953–956.

[3] Kolmogorov A.N., "On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition", *American Math. Soc. Transl*, **28** (1963), 55–63.

[4] Арнольд В.И., "О представлении любой непрерывной функции трех переменных в виде суммы функций не более двух переменных", *Докл. АН СССР*, **114**:4 (1957).

[5] Арнольд В.И., "О представлении функций нескольких переменных в виде суперпозиции функций меньшего числа переменных", *Мат. Просвещение*, **3** (1958), 41–61.

[6] McCulloch W.S., Pitts W., "A logical calculus of the ideas immanent nervous activity", *Bull. Of math. Biophysics*, **5** (1943), 115–133.

[7] Мак-Каллок У., Питтс У., *Автоматы.* V. 5: *Логическое исчисление идей, относящихся к нервной активности*, ИЛ, М., 1956.

[8] Розенблатт Ф, *Принципы нейродинамики. Перцептрон и теория механизмов мозга*: *Логическое исчисление идей, относящихся к нервной активности*, Мир, М., 1965.

[9] Rosenblat F, *Principles of Neurodynamics: Perceptrons and the Theory of Brain mechanisms*, Spartan, Washington D.C., 1962.

[10] Минский М., Пейперт С., *Перцептроны*, Мир, М., 1971.

[11] Cybenko G., "Approximations by superpositions of sigmoidal functions", *Math. Control, Signals, Systems*, **2** (1989), 303–314.

[12] Funahashi K., "On the approximate realization of continuous mappings by neural networks", *Neural Networks*, **2**:3 (1989), 183–192.

[13] Hornick K., Stinchcombe M., White H., "Multilayer feedforward networks are universal approximations", *Neural Networks*, **2**:5 (1989), 359–366.

[14] Werbos P.J., *Beyond regression: New tools for prediction and analysis in the behavioral sciences*, Ph.D. Thesis, Harvard University, Cambridge, MA, 1974.

[15] Saarinen S., R.B. Bramley and G. Cybenko, "Neural networks, backpropagation and automatic differentiation", *Automatic Differentiation of Algorithms: Theory, Implementation and Application*, eds. A. Griewank and G.F. Corliss, SIAM, Philadelphia, 1992, 31–42.

[16] Werbos P.J., *Backpropagation through time: What it does and how to do it*. V. 78, Proceedings of the IEEE, 1990, 1550–1560.

[17] Алексеев Д.В., "Приближение функций нескольких переменных нейронными сетями", *Интеллектуальные системы*, **7**:1-4 (2003), 191–205.

[18] Половников В.С., *Об оптимизации структурной реализации нейронных сетей*, дисс. ... канд. физ.-матем. наук, МГУ, Москва, 2007.

[19] Яблонский С.В., *Введение в дискретную математику*, «Наука», Москва, 1986.

[20] Haykin S., *Neural Networks. A Comprehensive Foundation*, Prentice Hall International, Inc., Canada, 1999.

On the construction of an explicit neural network architecture that approximates particle-linear functions

Shishlyakov V.G.

This work considers the question of discovering an upper-bound estimation of parameters quantity of neural network architecture well-approximating particle-linear dependances. The main result of this article consists of the theorem asserting that any particle-linear function can be approximated with any degree of precision on the big part of space by neural network with sigmoidal activation functions. This theorem has a constructive proof, i.e. neural network architecture with mentioned features building explicitly.

Keywords: schemes of functional elements, neural networks, architecture, approximation, upper-bound estimation, particle-linear functions.

References

[1] Kolmogorov A.N., "Representation of continuous functions of several variables by superpositions of continuous functions of fewer variables", *Reports of the USSR Academy of Sciences*, **108** (1956), 2 (In Russian).

[2] Kolmogorov A.N., "Representation of continuous functions of several variables as a superposition of continuous functions of one variable and addition", *Reports of the USSR Academy of Sciences*, **114** (1957), 953–956 (In Russian).

[3] Kolmogorov A.N., "On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition", *American Math. Soc. Transl*, **28** (1963), 55–63 (In Russian).

[4] Arnold V.I., "Representation of any continuous function of three variables as a sum of functions of at most two variables", *Reports of the USSR Academy of Sciences*, **114**:4 (1957) (In Russian).

[5] Arnold V.I., "Representation of functions of several variables as a superposition of functions of a smaller number of variables", *Math Education*, **3** (1958), 41–61 (In Russian).

[6] McCulloch W.S., Pitts W., "A logical calculus of the ideas immanent nervous activity", *Bull. Of math. Biophysics*, **5** (1943), 115–133.

[7] McCulloch I., Pitts W., *Automatic machines*. V. 5: *Logical calculus of ideas related to nervous activity*, IL, Moscow, 1956 (In Russian).

[8] Rosenblatt F., *Principles of neurodynamics. Perceptron and the theory of brain mechanisms*: *Logical calculus of ideas related to nervous activity*, MIR, Moscow, 1965 (In Russian).

[9] Rosenblat F, *Principles of Neurodynamics: Perceptrons and the Theory of Brain mechanisms*, Spartan, Washington D.C., 1962.

[10] Minsky M., Peipert S., *Perceptrons*, MIR, Moscow, 1971 (In Russian).

[11] Cybenko G., "Approximations by superpositions of sigmoidal functions", *Math. Control, Signals, Systems*, **2** (1989), 303–314.

[12] Funahashi K., "On the approximate realization of continuous mappings by neural networks", *Neural Networks*, **2**:3 (1989), 183–192.

[13] Hornick K., Stinchcombe M., White H., "Multilayer feedforward networks are universal approximations", *Neural Networks*, **2**:5 (1989), 359–366.

[14] Werbos P.J., *Beyond regression: New tools for prediction and analysis in the behavioral sciences*, Ph.D. Thesis, Harvard University, Cambridge, MA, 1974.

[15] Saarinen S., R.B. Bramley and G. Cybenko, "Neural networks, backpropagation and automatic differentiation", *Automatic Differentiation of Algorithms: Theory, Implementation and Application*, eds. A. Griewank and G.F. Corliss, SIAM, Philadelphia, 1992, 31–42.

[16] Werbos P.J., *Backpropagation through time: What it does and how to do it.* V. 78, Proceedings of the IEEE, 1990, 1550–1560.

[17] Alekseev D.V., "Approximation of functions of several variables by neural networks", *Intelligent systems*, **7**:1-4 (2003), 191–205 (In Russian).

[18] Polovnikov V.S., *On optimization of the structural implementation of neural networks*, Ph.D. Thesis ... physical and mathematical sciences, MSU, Moscow, 2007 (In Russian).

[19] YAblonskij S.V., *Introduction to discrete mathematics*, eds. fiz.-mat.lit., «Science», Moscow, 1986 (In Russian).

[20] Haykin S., *Neural Networks. A Comprehensive Foundation*, Prentice Hall International, Inc., Canada, 1999.

Часть 3.
Математические модели

Кванторная выразимость в логике предикатов

Ю. С. Капустин[1]

В математике новые понятия часто вводятся путем кванторных определений. При наличии достаточно большого запаса таких понятий они могут позволить переформулировать новые кванторные определения бескванторным образом. Это делает заслуживающей рассмотрения задачу отыскания базисных понятий в заданной предметной области, которые делают избыточным дальнейшее кванторное определение.

В данной работе рассматривается кванторная выразимость небольшой глубины в 4 алгебраических системах. Были найдены базисы выразимости для небольшой глубины.

Ключевые слова: логика предикатов, кванторная выразимость, алгебраическая система.

1. Введение

Базисные операции и отношения алгебраической системы могут порождать при помощи формул алгебры логики новые операции и отношения. В данной работе исследуется возможность сведения кванторной выразимости таких операций и отношений к бескванторной. Новые понятия в математике определяются при помощи кванторных конструкций. При этом желательно обходиться минимальным количеством таких операций и отношений, выбрав системы понятий, через которые как можно большее число других понятий можно было бы выразить бескванторным образом.

В работе описываются системы отношений и операций, с помощью которых можно бескванторно выразить в различных алгебраических системах операции и отношения, заданные формулами фиксированной глубины.

Данная работа продолжает исследования, начатые в работе [1].

[1] *Капустин Юрий Сергеевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: kapustin.iu@yandex.ru

Kapustin Iurii Sergeevich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

2. Основные понятия и результаты

Напомним, что алгебраической системой называется множество M с определёнными на нём операциями и предикатами. Будем предполагать, что оно содержит $\varnothing$. Сигнатурой S над M будем называть отображение $\Sigma : S \to F$, сопоставляющее элементам множества символов S предикаты и операции, определенные на M. Формулы и термы в сигнатуре Σ определяются следующим образом:

1) x_i, где x_i – символ переменной из фиксированного счетного списка – терм.

2) Если $\Sigma(s) = f$ – n—местная операция на M, $t_1, \ldots, t_n$ – термы, то слово $s(t_1, \ldots, t_n)$ – терм.

3) Если $\Sigma(s) = p$ – n—местный предикат, определенный на M, $t_1, \ldots, t_n$– термы, то слово $s(t_1, \ldots, t_n)$ – формула.

4) Если $P_1, \ldots P_k$ — формулы, то слова $(P_1) \bigvee \cdots \bigvee (P_k)$,
$(P_1)\& \ldots \&(P_k), \neg(P_1), (P_1) \to (P_2)$ – формулы.

5) Если P – формула, $x_1, \ldots, x_n$ — символы переменных, то слова
$\forall x_1, \ldots, x_n(P), \exists x_1, \ldots, x_n$
(P) – формулы.

6) Если P – формула, x – переменная, то $set_x(P)$ — терм.

Каждая формула определяет естественным образом некоторый предикат, заданный на наборах элементов множества M. Каждый терм определяет естественным образом некоторую операцию, заданную на наборах элементов M, и принимающую значения в M.

Уточним понимание значения терма $set_x(P)$, где P – формула от свободной переменной x и свободных переменных $x_1, \ldots, x_n$. Будем считать, что этот терм при фиксированном значении свободных переменных $x_1, \ldots, x_n$ имеет значением множество значений x, для которых P принимает истинное значение. Если такое множество не определено корректно или не принадлежит M, значением $set_x(P)$ полагаем $\varnothing$.

Если предикат или операция f определяется какой-либо формулой или термом в сигнатуре Σ, то говорим, что f логически выразимо над F (через F). Если f определяется формулой или термом в Σ, не содержащим кванторов и описателей set, то говорим, что f бесквантворно (элементарно) выразимо над F.

Если формула или терм содержит цепочку вложенных кванторов и описателей длины n, но не содержит цепочку длины более n, то назовём n квантовой глубиной формулы или терма. Если предикат или операция f определяется какой-либо формулой или термом в сигнатуре Σ глубины n, назовём предикат или операцию f n-выразимым над F.

Для некоторых алгебраических систем удалось найти набор предикатов и операций, через которые бесквантворно выражаются все предикаты

и операции, n-выразимые в данной алгебраической системе для малых n. Были доказаны следующие теоремы:

Рассмотрим числовую прямую $\mathbb{R}$ с определённой на ней операцией $\leq$.

Обозначим через R множество $\mathbb{R} \bigcup 2^{\mathbb{R}}$, на котором определены предикаты $a \in b, a \leq b$.

Обозначим через major(a) операцию, значение которой — множество верхних граней множества a, а через minor(a) операцию, значение которой — множество его нижних граней.

Обозначения $a =_? b$ и $a \in_? b$ будем использовать для операций, значение которых равно $2^{\mathbb{R}}$, если истинен соответствующий предикат $a = b$ и $a \in b$. В противном случае значение предиката равно $\varnothing$.

Обозначим через $Crd_1(a)$ предикат, истинный в том и только в том случае, если a - одноэлементное множество.

Теорема

Все предикаты и операции, 2-выразимые в алгебраической системе $\mathbb{R} \bigcup 2^{\mathbb{R}}, \{\in, \leq\}$*, бескванторно выразимы над системой операций и предикатов* $\{a \in b, \mathbb{R} \backslash a, a \bigcup b, (-\infty, a]$,
$, [a, +\infty), a = \varnothing, a =_? \varnothing, major(a), minor(a), Crd_1(a)\}$*, причём выражение не содержит вложенных друг в друга операций вида* $major(a), minor(a)$ *и не содержит этих операций и предикат проверки на одноэлементность одновременно.*

Определим операции $lip(a, b), rip(a, b)$.

Значение операции lip(a, b) — это множество точек, для которых расстояние до ближайшей слева точки замыкания множества a больше, чем расстояние до ближайшей слева точки замыкания b

Значение операции rip(a, b) — это множество точек, для которых расстояние до ближайшей справа точки замыкания множества a больше, чем расстояние до ближайшей справа точки замыкания множества b.

Теорема

Все операции, 3-выразимые в алгебраической системе $\mathbb{R} \bigcup 2^{\mathbb{R}}$,
$\{\in, \leq\}$*, бескванторно выразимы над множеством операций* $\{a \in b, \mathbb{R} \backslash a, a \bigcup b, (-\infty, a], [a, +\infty), a =_? \varnothing, major(a), minor(a)$,
$lip(a, b), rip(a, b)\}$

Определим на натуральных числах и множествах натуральных числах операции div(x), mod (x), sidiv (x), simod (x), int (x). Значение операции div(x) — множество делителей x, mod (x) — множество кратных x, sidiv (x) — объединение собственных делителей элементов множества x, simod (x) — объединение собственных кратных элементов множества x, int (x) — множество чисел, являющихся для каждого элемента множества x или делителем, или кратным. Для тех x, для которых заданное таким образом значение одной из операций не определено корректно

(например, div(x), если x — множество чисел, а не число), значение этой операции равно $\varnothing$.

Теорема

Все предикаты и операции, 2-выразимые в алгебраической системе $\mathbb{N}\bigcup 2^{\mathbb{N}}, \{\in, |\}$ *бесквантворно выразимы над* $\{\in, \mathbb{N}\backslash, \bigcup, div(x), mod(x), = \varnothing, =_? \varnothing, int(x), sidiv(x), simod(x)\}$

Определим на точках булева куба и их множествах операции $^*a$, a^*, $^{**}a$, a^{**}, $int(a)$.

Здесь значение операции *a — множество точек, меньших a, операции a* — множество точек, больших a, $**a$ — множество точек, меньших хоть одного элемента a, a^{**} — множество точек, больших хоть одного элемента a, $int(a)$ — множество точек, сравнимых с каждым элементом множества a. Для тех a, для которых значение одной из операций не определено корректно, значение этой операции равно $\varnothing$.

Теорема

Все предикаты и операции, 2-выразимые в алгебраической системе $B^n\bigcup 2^{B^n}, \{\in, \leq\}$ *формулой, не зависящей от n, где n – произвольно, бесквантворно выразимы над* $\{\in, B^n\backslash, \bigcup, *a, a*, = \varnothing, =_? \varnothing, int(a), ^{**} a, a^{**}\}$ *формулой, не зависящей от n.*

Пусть $\mathbb{Z}$ — множество целых чисел. Будем использовать обозначения $U_1 = 2^{\mathbb{Z}}\bigcup\mathbb{Z}$, $U_2 = U_1\bigcup 2^{U_1}$. На множестве U_2 естественным образом задано отношение $a \in b$.

Определим элементах U_2 следующие операции:

—$a\uparrow$ — множество всех множеств, содержащих a.

— $\bigcap(a)$ — пересечение всех подмножеств a

— $\bigcup(a)$ — объединение всех подмножеств a

— trans(a,b,c) — множество элементов x из U_1, что $(a\bigcap x\bigcup b\backslash x) \in c$

— $\backslash(a, b)$ — множество попарных разностей элементов a и b

Для тех значений переменных, для которых значение одной из операций не определено корректно, значение этой операции равно $\varnothing$.

Теорема

Все операции, 2-выразимые в алгебраической системе $U_2 = U_1\bigcup 2^{U_1}, \{\in\}$, *бесквантворно выразимы над* $\{U_1\backslash, \bigcup, a*, a\uparrow, \bigcap(a), \bigcup(a), 2^a, trans(a, b, c), \{a\}, \backslash(a, b)\}$

3. Бесквантворная выразимость на множестве $\{\mathbb{R}\bigcup 2^{\mathbb{R}}\}$ с сигнатурой $\{\in, \leq\}$

Рассмотрим множество R=$\mathbb{R}\bigcup 2^{\mathbb{R}}$ с естественным образом определенной на нем сигнатурой $\{\in, \leq\}$. При этом считаем предикат $\leq$ принимающим

ложные значения всегда, когда он применен не к двум числам, а предикат $\in$ ложным всегда, когда он применён не к числу и множеству.

В этой главе числом называется элемент множества $\mathbb{R}$, множеством — элемент множества $\bigcup 2^{\mathbb{R}}$.

Так как принадлежать множествам-элементам R могут лишь элементы множества $\mathbb{R}$, для данного множества будем интерепретировать терм $set_x(P)$, где P – формула от свободной переменной x и свободных переменных $x_1, \ldots, x_n$, как $\{y \in \mathbb{R} : P(y) = \text{И}\}$ — операцию от переменных $x_1, \ldots, x_n$, значение которой равно множеству всех x, принадлежащих $\mathbb{R}$, на которых верен предикат, задаваемый формулой P.

В дальнейшем мы также будем рассматривать множества вида $\{M \bigcup 2^M\}$ и интерпретировать в них $set_x(P)$ как $\{y \in M : P(y) = \text{И}\}$.

Обозначения $a =_? b$ и $a \in_? b$ будем использовать для операций, значение которых равно $2^{\mathbb{R}}$, если истинен соответствующий предикат $a = b$ или $a \in b$. В противном случае значение предиката равно $\varnothing$.

Будем послойно изучать предикаты и операции, n — выразимые над R. Начнем с операций, выразимых на первом слое.

Теорема 1. $\{\mathbb{R}\backslash, \bigcup, (a \in_? b), (-\infty, a], [a, \infty)\}$ *образуют базис 1-выразимых операций в* R

Доказательство. Термы, которыми задаются эти операции, имеют вид:

$set_x(P(A_i(x, x_1, \ldots, x_n)))$, где:

P -формула, задающая некоторую булеву функцию.

A_i — бескванторные операции вида $x_k \leq x_j$ и $x_k \in x_j$

Булеву функцию, задающую P, можно выразить в базисе $\{\neg, \bigvee\}$. Используя равенства:

$set_x(\neg P) = \mathbb{R}\backslash(set_x(P))$

$set_x(P \bigvee Q) = (set_x(P)) \bigcup (set_x(Q))$

преобразуем терм к равносильному терму, выразимому над термами вида $set_x(A_i)$ и операциями $\bigcup$ и $\mathbb{R}\backslash$, где A_i — бескванторный терм.

Рассмотрим операции, выразимые термами $set_x(A_i)$:

$set_x(a \in b) = (a \in_? b)$ — условная операция, равная $\mathbb{R}$, если $a \in b$, и $\varnothing$ иначе.

$set_x(a \leq b) = (a \leq_? b)$ — условная операция, равная $\mathbb{R}$, если $a \leq b$, и $\varnothing$ иначе.

$set_x(x \leq a) = (-\infty, a]$. Результат данной операции считаем равным $\varnothing$, если a — множество

$set_x(a \leq x) = [a, \infty)$. Результат данной операции считаем равным $\varnothing$, если a — множество

$set_x(x \in a) = a$. Получается тождественная операция, суперпозицию которой можно не рассматривать.

$set_x(a \in x) = \varnothing$.

Следовательно, все операции, выразимые на первом уровне над R в данной сигнатуре бескванторно выражаются над $\{\mathbb{R}\backslash, \bigcup, (a \leq_? b), (a \in_? b), (-\infty, a], [a, \infty), \varnothing\}$. При этом все операции, бескванторно выразимые через данные, 1-выразимы:

- Если операция P выразима термом $set_x(P')$, операция Q — термом $set_x(Q')$, где P' и Q' не содержат кванторов и описателей, то операция $P \bigcup Q$ выражается термом $set_x(P' \bigvee Q')$, операция $\mathbb{R}\backslash P$ выражается формулой $set_x(\neg P')$

- Если операция P выразима термом $set_x(P')$, то результат операции P — некое множество, и значения $set_x(P \in b)$, $set_x(P \leq b)$, $set_x(a \leq P)$ равны $\varnothing$

- Если операция P выразима термом $set_x(P')$, то значения $set_x(P \in b)$, $set_x(P \leq b)$, $set_x(a \leq P)$ равны $\varnothing$

- Если операция P выразима термом $set_x(P')$, то $(a \in_? P)$ выражается термом $set_x(a \in P')$. При этом наружные для терма P' операции пересечения и дополнения до $\mathbb{R}$ можно вынести:

$set_x(a \in (\mathbb{R}\backslash Q')) = set_x((\neg a \in P')\&(x \leq a \bigvee a \leq x))$

$set_x(a \in (Q' \bigcup R')) = set_x(\neg a \in Q' \bigvee \neg a \in R')$

$set_x(a \in (set_y(P'')) = set_x(P''\&(x \leq a \bigvee a \leq x))$, если P'' задает условную операцию.

При этом $\varnothing$ при помощи фиктивной переменной выражается через остальные операции: $\varnothing = (x \leq_? (\mathbb{R}\backslash x))$

Операция $(a \leq_? b)$ также выразима через остальные:

$(a \leq_? b) = (a \in_? (-\infty, b])$

Остальные операции не выражаются друг через друга и, следовательно, образуют базис:

- Объединение — единственная двухместная операция, способная принимать значение — одноэлементное множество. Условные операции принимают только значения $\mathbb{R}$ и $\varnothing$, и примененный к их результату одноместные операции тоже могут давать только эти значения.

- Все операции, кроме $[a, \infty)$ сохраняют множество

$\{0, \varnothing, (-\infty, 0], (0, +\infty), \mathbb{R}\}$. Аналогично операция $(-\infty, a]$ также невыразима.

- Операция разности — единственная, не сохраняющая множество $\{\{0\}, \varnothing, \mathbb{R}\}$

- $(a \in_? b)$ также невыразима через остальные операции. Теорема доказана.

Теорема 2. *все 1-выразимые в R предикаты и операции выразимы через* $\{\mathbb{R}\backslash, \bigcup, (a \in_? b), (-\infty, a], [a, +\infty), = \varnothing\}$.

Рассмотрим 1-выразимые предикаты. Без ограничения общности можно рассмотреть предикаты, заданные формулой вида

$\forall x(A_1 \bigvee \cdots \bigvee A_n)$,

где A_i — бескванторные формулы или их отрицания. Заметим, что предикат $x \leq x$ истинен тогда и только тогда, когда x число. Поскольку формула $\forall x(A_1 \bigvee \cdots \bigvee A_n)$ равносильна формуле $\forall x(((x \leq x) \to (A_1 \bigvee \cdots \bigvee A_n))\&\forall x((\neg x \leq x) \to (A_1 \bigvee \cdots \bigvee A_n)))$, задача сводится к описанию предикатов, выражаемые формулами вида $\forall x((x \leq x) \to (A_1 \bigvee \cdots \bigvee A_n))$ и вида $\forall x((\neg x \leq x) \to (A_1 \bigvee \cdots \bigvee A_n))$

Все предикаты, выражаемые первой формулой, выразимы как $set_x(A_1 \bigvee \cdots \bigvee A_n) = \varnothing$. Следовательно, они выразимы над семейством $\{\mathbb{R}\backslash, \bigcup, (a \in_? b), (-\infty, a], [a, \infty), = \varnothing\}$.

Рассмотрим предикаты, выразимые формулой вида $\forall x((\neg x \leq x) \to (A_1 \bigvee \cdots \bigvee A_n))$. При этом можно считать, что все атомарные формулы A_i содержат x (иначе их можно вынести за квантор). Кроме того, зависящим от x является только значение атомарных формул вида $x_i \in x$ и их отрицаний. Следовательно, предикат, выразимый этой формулой, является проверкой существования множества, содержащего некий список элементов и не содержащего элементы из некоторого другого списка. Он выразим через проверку на принадлежность $\mathbb{R}$ ($x \in (\mathbb{R}\backslash\varnothing)$) и равенство элементов $\mathbb{R}$ ($(-\infty, a]\backslash(-\infty, b] = \varnothing\&(-\infty, b]\backslash(-\infty, a] = \varnothing)$)

Утверждение доказано.

Опишем операции, 2-выразимые над данным множеством.

Обозначим через major(a) операцию, значение которой — множество верхних граней множества a, а через minor(a) операцию, значение которой — множество его нижних граней. Определим операции $lip(a, b), rip(a, b)$, где где значение операции lip(a, b) — это множество точек, для которых расстояние до ближайшей слева точки замыкания множества a больше, чем расстояние до ближайшей слева точки замыкания b, а значение операции rip(a, b) — это множество точек, для которых расстояние до ближайшей справа точки замыкания множества a больше, чем расстояние до ближайшей справа точки замыкания множества b.

Теорема 3. *Все операции, 2-выразимые на множестве* $\{\mathbb{R} \bigcup 2^{\mathbb{R}}\}$ *над множеством предикатов* $\{\in, \leq\}$, *бескванторно выразимы над* $\{\mathbb{R}\backslash, \bigcup, (-\infty, a], [a, \infty), = \varnothing, =_? \varnothing, major(a), minor(a)\}$, *причём операции* $major(a), minor(a)$ *в бескванторном выражении не вложены друг в друга.*

Доказательство.

Все булевы функции выражаются над функциями $\{\neg, \bigvee\}$, которые можно вынести за *set*:

$set_x(\neg A) = \mathbb{R}\backslash A$

$set_x(A \bigvee B) = A \bigcup B$

Следовательно, достаточно рассматривать операции, выразимые термами типа $set_x(A = \varnothing)$, где A — 1-выразимый терм.

Если A не содержит x, то данный терм задаёт ту же операцию, что и терм $A =_? \varnothing$. При этом условная операция $a =_? \varnothing$, будучи добавленной к множеству $\{\mathbb{R}\backslash, \bigcup, (a \in_? b), (-\infty, a], [a, \infty), a = \varnothing\}$ позволяет бесквантворно выразить условную операцию $(a \in_? b)$:

$(a \in_? b) = (b\backslash\{a\}) = \varnothing$,

где $\{a\}$, определяемая как пустое множество, если $a \in 2^{\mathbb{R}}$ — выразимая операция:

$\{a\} = (-\infty, a] \bigcap [a, \infty)$ (операция $\bigcap$ выразима естественным образом).

Рассмотрим операции, выразимые термами вида $set_x(A = \varnothing)$, где терм A содержит x.

Так как $\bigcap$ — выразимая над $\{\mathbb{R}\backslash, \bigcup\}$ операция, терм $set_x(A = \varnothing)$ выражает ту же операцию, что и терм

$set_x(A \bigcap \{x\} = \varnothing \& A \bigcap (x, +\infty) = \varnothing \& A \bigcap (-\infty, x) = \varnothing) = set_x(A \bigcap \{x\} = \varnothing) \bigcup set_x(A \bigcap (x, +\infty) = \varnothing) \bigcup set_x(A \bigcap (-\infty, x) = \varnothing)$.

Следовательно, достаточно рассмотреть предикаты и операции, выразимые термами видов $set_x(A \bigcap \{x\} = \varnothing)$, $set_x(A \bigcap (x, +\infty) = \varnothing)$ и $set_x(A \bigcap (-\infty, x) = \varnothing)$.

Если выражение A содержит условную операцию $(a \in b_?)$, то терм $set_x(A \bigcap \{x\} = \varnothing)$ выражает ту же операцию, что и терм

$set_x(B \bigcap \{x\} = \varnothing) \bigcap set_x((a \in b)) \bigcup (set_x(C \bigcap \{x\} = \varnothing) \bigcap (\mathbb{R} \backslash set_x(a \in b)))$,

где терм B получен из терма A путём замены всех вхождений операции $(a \in b_?)$ на $\mathbb{R}$, а терм C получен из терма A путём замены всех вхождений операции $(a \in b_?)$ на $\varnothing$

Следовательно, устранив по индукции по вложенности все вхождения условной операции, без ограничения общности можем считать, что A не содержит вхождения условной операции.

Также если в выражении встречаются операции $(-\infty, a]$ и $[a, +\infty)$, где a — произвольный терм, не являющийся символом переменной, то, поскольку результаты всех операций из найденного базиса 1-выразимых предикатов являются множествами, такие операции можно заменить на $\varnothing$.

Все вхождения переменных, кроме вхождений в операции $(-\infty, a]$ и $[a, +\infty)$, можно заменить на выражение $set(x)$, которое определяется как $x \bigcup x$, что равно пустому множеству для x, принадлежащего $\mathbb{R}$, и равно самому x, если x — множество.

Следовательно, можно считать, что A представляет собой некоторый терм, выраженный над бесквантворными термами $set(a)$, $(-\infty, a]$, $[a, +\infty)$ при помощи операций $(a \bigcup b)$, $(\mathbb{R}\backslash a)$. Используя равенства:

$\mathbb{R}\backslash(\mathbb{R}\backslash a) = a$ (так как a — терм, значение которого — множество, а не число)

$\mathbb{R}\backslash(b \bigcup a) = \mathbb{R}\backslash b \bigcap \mathbb{R}\backslash a$

$\mathbb{R}\backslash(b \bigcap a) = \mathbb{R}\backslash b \bigcup \mathbb{R}\backslash a$

$a \bigcap (b \bigcup c) = (a \bigcap b) \bigcup (a \bigcap c)$

получим равносильный A терм, выраженный над $set(a)$, $(-\infty, a]$, $[a, +\infty)$, где a — переменная, при помощи операций $a \bigcup b$, $a \bigcap b$, $\mathbb{R}\backslash A$, где внешней операцией является первая, после неё идёт вторая, а третья является внутренней (форма, аналогичная ДНФ).

При этом поскольку $set_x((A_1 \bigcup \cdots \bigcup A_n) \bigcap p(x) = \varnothing)$ равно $set_x(A_1 \bigcap p(x) = \varnothing) \bigcap \cdots \bigcap set_x(A_n \bigcap p(x) = \varnothing)$, достаточно рассматривать операции, выразимые термами вида $set_x((A_1 \bigcap \cdots \bigcap A_n) \bigcap p(x) = \varnothing)$, где A_n есть одно из атомарных выражений — $set(a)$, $(-\infty, a]$, $[a, +\infty)$, где a — переменная, либо $\mathbb{R}\backslash B$, где B — атомарное выражение.

Обозначим через C объединение всех A_i, которые не содержат x. Получим выражение вида $set_x(C \bigcap (A_1 \bigcap \cdots \bigcap A_n) \bigcap p(x) = \varnothing)$, где все A_1 содержат x, а $p(x)$ имеет вид $\{x\}$, $(-\infty, x)$ или $(x, +\infty)$.

При этом результат этой операции не являтся пустым множеством в том и только в том случае, если все A_i включают $p(x)$ ($\{x\}$ включают $(-\infty, x]$ и $[x, -\infty)$, $(-\infty, x)$ включают $(-\infty, x]$ и $\mathbb{R}\backslash[x, +\infty)$, $(x, +\infty)$ включают $[x, +\infty)$ и $\mathbb{R}\backslash(-\infty, x]$ — то есть включение определено на формальных выражениях-термах, и не зависит от значения x). В этом случае терм равносилен терму $set_x(C \bigcap p(x) = \varnothing)$.

Терм $set_x(C \bigcap \{x\} = \varnothing)$ равносилен терму $\mathbb{R}\backslash$.

Терм $set_x(C \bigcap (x, +\infty) = \varnothing)$ равносилен терму $major(C)$

Терм $set_x(C \bigcap (-\infty, x) = \varnothing)$ равносилен терму $minor(C)$

Утверждение доказано.

Найдём все предикаты, 2-выразимые на множестве $\{\mathbb{R} \bigcup 2^{\mathbb{R}}\}$ над множеством операций $\{\in, \leq\}$. В дальнейшем для обозначения предиката "x n-элементно"будем использовать обозначение $Crd_n(x)$, для обозначения операции, выразимой термом "$set_y(x$ n-элементно)"как $Card_n(x)$. Эта операция принимает значение $\mathbb{R}$, если x n-элементно, и $\varnothing$ в проивном случае.

Теорема 4. *Все предикаты и операции, 2-выразимые на множестве* $\{\mathbb{R} \bigcup 2^{\mathbb{R}}\}$ *над множеством предикатов* $\{\in, \leq\}$, *бесквантворно выразимы над* $\{\mathbb{R}\backslash, \bigcup, (-\infty, a], [a, \infty), = \varnothing, =_? \varnothing, major(a), minor(a)\}$.

Все предикаты и операции, 2-выразимые на множестве $\{\mathbb{R} \bigcup 2^{\mathbb{R}}\}$ *над множеством предикатов* $\{\in, \leq\}$, *бесквантворно выразимы над* $\{\mathbb{R}\backslash, \bigcup, (-\infty, a], [a, \infty), = \varnothing, =_? \varnothing, major(a), minor(a),$
$Crd_n(a)\}$, *причём выражение не содержит вложенных друг в друга операций вида* $major(a), minor(a)$ *и не содержит этих операций и предикат проверки на одноэлементность одновременно.*

Доказательство.

Как и прежде, поскольку $\forall x((x \leq x) \to (A_1 \bigvee \cdots \bigvee A_n))$

$=set_x(A_1 \bigvee \cdots \bigvee A_n)$ достаточно рассмотреть выразимость предикатов, задаваемых формулой вида $\forall x((\neg x \leq x) \to (A_1 \bigvee \cdots \bigvee A_n))$, где A_n — 1-выразимый предикат, внешняя операция которого — $= \varnothing$ (поскольку это единственный предикат из базиса 1-выразимых предикатов и операций), и при этом все A_i содержат x.

Как и прежде, от вхождения условной операции $a \in_? b$ можно избавиться, добавив предикат $\in$. Действительно, $A = B\&(a \in b) \bigvee C\&(\neg a \in b)$, где формула B получается из A путём замены всех вхождений $a \in_? b$ на $\mathbb{R}$, а формула C получается из A путём замены всех вхождений $a \in_? b$ на $\varnothing$. По индукции по вложенности и числу предикатов можно доказать наличие равносильного A предиката, не содержащего $\in$, но содержащего $\in$.

Также как и ранее, без ограничения общности можно считать, что x не входит в качестве переменной ни в одну операцию $(-\infty, x)$ и $(x, +\infty)$ (так как x — множество). Более того, без ограничения общности можно считать, что в эти операции входят только переменные. Обозначив данные операции над свободными переменными за новые переменные ($y_i = (-\infty, x_i]$, $z_i = (x_i, +\infty)$), получим кванторный предикат над $\bigcap, \bigcup, \in, \mathbb{R}\backslash$.

По доказанному для U_1 в статье [1] утверждению, в этом случае данная формула выразима через операции $\bigcap, \bigcup, \in, \mathbb{R}\backslash$ и предикаты n-элементности (из доказательства утверждения для U_1 следует, что выразим лишь предикат (0+0+1)-элементности, то есть одноэлементности). При обратной замене новых переменных y_i и z_i на их значения к этому списку добавятся предикаты $y_i = (-\infty, x_i]$, $z_i = (x_i, +\infty)$.

Следовательно, к системе, выражающей все 2-выразимые предикаты и 1-выразимые предикаты и операции добавляется только предикат одноэлементности. Но он 1-выразим формулой $(\neg x = \varnothing \& x \backslash (major(x) \bigcap minor(major(x))) = \varnothing)$. Утверждение доказано.

Рассмотрим все 3-выразимые операции.

Теорема 5. *Все 3-выразимые в R над* $\{\leq, \in\}$ *операции выразимы над* $\{\mathbb{R}\backslash, \bigcup, (-\infty, a], [a, \infty), =_? \varnothing, major(a), minor(a), lip(A, C), rip(A, C)\}$

Как и прежде, достаточно рассматривать операции, выразимые термами типа $set_x(A = \varnothing)$, и $set_x(Crd_1(A))$, где A — 2-выразимый терм.

Если A не содержит x, то терм $set_x(A = \varnothing)$ задаёт ту же операцию, что и терм $A =_? \varnothing$, а терм $set_x(Crd_1(A))$ — ту же операцию, что и терм $Card_1(A) = \varnothing_?$ — выразимый над $\{\mathbb{R}\backslash, \bigcup, (-\infty, a], [a, \infty), =_? \varnothing, major(a), minor(a)\}$ терм.

Так же, как и при поиске системы операций, выражающих все 2-выразимые предикаты, без ограничения общности можно считать, что A не содержит вхождения условной операции, а также что операции $[a, +\infty)$ и $(-\infty, a]$ применены только к переменным.

Если 3-выразимая операция задаётся термом $set_x(Crd_1(A))$, то терм A не содержит операций minor и major. В этом случае A выражается через $\{\mathbb{R}\backslash, \bigcup, a =_? \varnothing, Crd_1(a)\}$ над x_i, $y_i = (-\infty, x_i]$, $z_i = [x_i, +\infty)$. Из теоремы для U_1 следует, что оператор $set_x(A = \varnothing)$ бесквантворно выразим над этими операциями и условными операциями n-элементности $Card_n$. При этом эти условные операции выразимы над операциями $=_? \varnothing$. Также они выразимы над операциями $minor$ и $major$ и $\backslash$. Это верно, поскольку значение $minor(major(a)) \bigcap major(a)$ есть точная верхняя грань a, и операция $Card_n(x)$ выражается рекурсивно:

$Card_n(x) = Card_{n-1}(x \backslash sup(x))$

Рассмотрим операции, выразимые термами вида $set_x(A = \varnothing)$, где терм A содержит x. Из утверждения 4, а также того, что $A = \varnothing$ 2-выразимая операция следует, что A не содержит вложенных друг в друга операций $minor$ и $major$.

Как и в случае 2-выразимости, достаточно рассмотреть предикаты и операции, выразимые термами видов $set_x(A \bigcap \{x\} = \varnothing)$, $set_x(A \bigcap (x, +\infty) = \varnothing)$ и $set_x(A \bigcap (-\infty, x) = \varnothing)$. Аналогично случаю 2-выразимости доказывается, что достаточно рассматривать предикаты и операции вид $set_x((A_1 \bigcap \cdots \bigcap A_n) \bigcap p(x) = \varnothing)$, где A_i — одно из атомарных выражений — $set(a)$, $(-\infty, a]$, $[a, +\infty)$, где a — переменная, либо $\mathbb{R}\backslash B$, где B — атомарное выражение или операция с внешней операцией $major$ или $minor$, и операндом, выразимым над $\{\mathbb{R}\backslash, \bigcup, (a \in_? b), (-\infty, a], [a, \infty), = \varnothing\}$.

Если $p(x)$ имеет вид $\{x\}$, то $set_x((A_1 \bigcap \cdots \bigcap A_n) \bigcap \{x\} = \varnothing) = set_x(A_1 \bigcap \{x\} = \varnothing) \bigcap \cdots \bigcap set_x(A_1 \bigcap \{x\} = \varnothing)$ При этом $set_x(A_1 \bigcap \{x\} = \varnothing) = \mathbb{R}\backslash$, следовательно, в этом случае операция 2-выразима.

Рассмотрим случай $p(x) = (x; +\infty)$. Аналогично доказательству случая 2-выразимости можно избавиться от всех A_i, содержащих x и не имеющих внешней операции $minor$ или $major$. Также обозначим $C = A_{i_1} \bigcap \cdots \bigcap A_{i_n}$, где $A_{i_1} \ldots A_{i_n}$ — все A_i, содержащие x.

Получим равносильный исходному терму терм $set_x((B_1 \bigcap \cdots \bigcap B_n) \bigcap C \bigcap \{x\} = \varnothing)$, где терм C не содержит x, а все термы B_i содержат его, имеют внешнюю операцию $minor$ или $major$ и выражаются через $\bigcap, \mathbb{R}\backslash, \bigcup$ над переменными и термами вида $[a, +\infty)$, $(a, -\infty]$

Заменим каждый терм B_i на равносильный терм, в котором внешняя операция — $\bigcup$, затем идёт $\bigcap$, а затем — $\mathbb{R}\backslash$. Пользуясь тем, что $minor(A \bigcup B) = minor(A) \bigcap minor(B)$, вынесем операцию $\bigcup$ за скобки

(и, если необходимо, снова объединим все операнды пересечения, не содержащие x).

После этого каждый терм B_i будет иметь или вид

$major(D_{i1} \bigcap \cdots \bigcap D_{in})$, или $minor(D_{i1} \bigcap \cdots \bigcap D_{in})$. Как и раньше, объединим операнды, не содержащие x. Тогда все B_i будут иметь вид $major(E_i \bigcap D_{i1} \bigcap \cdots \bigcap D_{in})$ или $minor(D_{i1} \bigcap \cdots \bigcap D_{in})$.

Далее, для каждого B_i $D_{i1} \bigcap \cdots \bigcap D_{in}$ включает(в указанном ранее смысле) или не включает $\{x\}, (x, +\infty), (-\infty, x)$. Кроме того, для каждого B_i $set_x(O)$ можно выразить как $(set_x(O) \bigcap E_i) \& (set_x(O) \bigcap (\mathbb{R} \backslash E_i))$. При работе с первой условной операцией можно считать, что $x \in E_i$, при работе со второй — что $\neg x \in E_i$ (так как $(set_x(O) \bigcap E_i) = set_x(O \& x \in E_i), (set_x(O) \bigcap (\mathbb{R} \backslash E_i)) = set_x(O \& \neg(x \in E_i))$. Аналогично каждый полученный set можно разложить дальше — $(set_x(O) \bigcap minor(E_i)) \& (set_x(O) \bigcap (\mathbb{R} \backslash minor(E_i)))$, и при работе с первым термом можно считать, что $(-\infty, x)$ не пересекается с E_i, а при работе со вторым — что пересекается. Аналогичное разложение проведём и для $major(x)$).

В таком случае про выражение, входящее во внешнюю операцию в каждом B_i при соответствующих условиях можно будет сказать, содержит ли он x и хоть один элемент $(x, +\infty)$ и $(-\infty, x)$.

Для каждого B_i рассмотрим его внешнюю операцию. Если она $major$, то:

- В случае, если операнд внешней операции B_i не содержит элементов, больших x, то он может быть исключён из дизъюнкции, так как в этом случае пересечение B_i с $(x, +\infty)$ равно $(x, +\infty)$

- В случае, если операнд внешней операции B_i содержит элементы, большие x, то его можно заменить на соответствующее выражение $major(E_i)$, не содержащее x, и добавить к C.

Если внешняя операция B_i — minor, то:

- В случае, если операнд внешней операции B_i не содержит элементы, меньшие x, то вся дизъюнкция равна пустому множеству, так как в этом случае результат B_i включается в $(-\infty; x]$ и не пересекается с $(x; +\infty)$.

- В случае, если операнд внешней операции B_i содержит элементы, меньшие x, то его можно заменить на соответствующее выражение $minor(E_i)$, не содержащее x, и добавить к C.

Осталось выразить операции, задаваемые только теми термами B_i, которые имеют внешнюю операцию minor и содержат только элементы, большие x. Воспользовавшись равенством

$minor(a) \bigcap minor(b) = minor(a \bigcap b)$, приведём терм к виду:

$set_x((minor(A \bigcap (x, +\infty)) \bigcap C \bigcap (x, +\infty)) = \varnothing)$

Данная операция была обозначена как $lip(A, C)$ — это множество точек, для которых расстояние до ближайшей справа точки замыкания

множества A выше, чем расстояние до ближайшей справа точки замыкания C. Например, в случае $C = \mathbb{R}$ это множество правых предельных точек.

Аналогично доказывается, что в случае $p = (-\infty, x)$ операция, выразимая термом $set_x(A \bigcap p = \varnothing)$, бесквантворно выразима через $\{\mathbb{R}\backslash, \bigcup, (-\infty, a], [a, \infty), =_? \varnothing, major(a), minor(a),$
$lip(A, C), rip(A, C)\}$ Теорема доказана.

4. Бесквантворная выразимость на множестве $\mathbb{N} \bigcup 2^{\mathbb{N}}$ с сигнатурой $\{\in, \leq\}$

Рассмотрим множество N=$\mathbb{N} \bigcup 2^{\mathbb{N}}$ с естественным образом определенной на нем сигнатурой $\{\in, |\}$. При этом считаем предикат | принимающим ложные значения всегда, когда он применен не к двум числам, а предикат $\in$ ложным всегда, когда первый операнд — не число или второй операнд — не множество. Для данного множества будем интерепретировать $set_x(P)$, как $\{y \in \mathbb{N} : P(y) =$И$\}$.

Будем послойно изучать предикаты и операции, n — выразимые над $\mathbb{N}$.

Определим на натуральных числах и множествах натуральных числах операции div(x), mod (x), sidiv (x), simod (x), int (x) , где значение операции div(x) — множество делителей x, mod (x) — множество кратных x, sidiv (x) — объединение собственных делителей элементов множества x, simod (x) — объединение собственных кратных элементов множества x, int (x) — множество чисел, являющихся для каждого элемента множества x или делителем, или кратным. Для тех x, для которых значение операций не определено корректно (например, div(x), если x — множество чисел, а не число), значение операции равно $\varnothing$.

Теорема 6. *все 1-выразимые в N предикаты и операции выразимы через* $\{\mathbb{N}\backslash, \bigcup, (a \in_? b), div(a), mod(a), = \varnothing, \}$.

Начнем с операций, выразимых на первом слое. Термы, которыми они задаются, имеют вид:

$set_x(P(A_i(x, x_1, \ldots, x_n)))$, где:

P -формула, задающая некоторую булеву функцию.

A_i — бесквантворные операции вида $x_i|x_j$ и $x_i \in x_j$

Булеву функцию, задающую P, можно выразить в базисе $\{\neg, \bigvee\}$. Используя равенства:

$set_x(\neg P) = \mathbb{N}\backslash(set_x(P))$

$set_x(P \bigvee Q) = (set_x(P)) \bigcup (set_x(Q))$

преобразуем терм к равносильному терму, выразимому над термами вида $set_x(A_i)$ и операциями $\bigcup$ и $\mathbb{N}\backslash$, где A_i — бесквантворный терм.

Рассмотрим операции, выразимые термами $set_x(A_i)$:

$set_x(a \in b) = (a \in_? b)$ — условная операция, равная $\mathbb{N}$, если $a \in b$, и $\varnothing$ иначе.

$set_x(a|b) = (a|_?b)$ — условная операция, равная $\mathbb{N}$, если $a|b$, и $\varnothing$ иначе.

$set_x(x|a)$ — множество кратных a. Обозначим данную операцию как $mul(a)$ (её результат считаем равным $\varnothing$, если a — множество)

$set_x(a|x)$ — множество делителей a. Обозначим данную операцию как $div(a)$ (её результат считаем равным $\varnothing$, если a — множество)

$set_x(x \in a) = a$. Получается тождественная операция, суперпозиция которой ничего не добавляет.

$set_x(a \in x) = \varnothing$.

При этом $(a|_?b) = (a \in_? mul(b))$, $\varnothing = (\mathbb{N}\backslash(x \bigcup(\mathbb{N}\backslash x)))$

Рассмотрим 1-выразимые предикаты. Без ограничения общности можно рассмотреть предикаты, заданные формулой вида $\forall x(A_1 \bigvee \cdots \bigvee A_n)$, где A_i — бесквантворные формулы или их отрицания. Так как предикат $x|x$ проверяет, является ли x числом, и формула $\forall x(A_1 \bigvee \cdots \bigvee A_n)$ равносильна формуле $\forall x(((x|x) \to (A_1 \bigvee \cdots \bigvee A_n))\&\forall x((\neg x|x) \to (A_1 \bigvee \cdots \bigvee A_n)))$, можно рассматривать отдельно предикаты, выражаемые формулами вида $\forall x((x|x) \to (A_1 \bigvee \cdots \bigvee A_n))$ и вида $\forall x((\neg x|x) \to (A_1 \bigvee \cdots \bigvee A_n))$

Все предикаты, выражаемые первой формулой, выразимы как $set_x(A_1 \bigvee \cdots \bigvee A_n) = \varnothing$. Следовательно, они выразимы над семейством $\{\mathbb{N}\backslash, \bigcup, (a \in_? b), div(a), mul(a), = \varnothing\}$.

Рассмотрим предикаты, выразимые формулой вида $\forall x((\neg x|x) \to (A_1 \bigvee \cdots \bigvee A_n))$. При этом можно считать, что все атомарные формулы содержат x (иначе их можно вынести за квантор). Кроме того, нетривиальным является только значение атомарных формул вида $x_i \in x$ и их отрицаний. Следовательно, этот предикат является проверкой существования множества, содержащего некий список элементов и не содержащего элементы из некоторого другого списка, и выразим через проверку на принадлежность $\mathbb{N}$ $(x \in (\mathbb{N}\backslash\varnothing))$ и равенство элементов $\mathbb{N}$ $(div(a)\backslash div(b) = \varnothing \& div(b)\backslash div(a) = \varnothing))$

Утверждение доказано.

Теорема 7. *Все 2-выразимые в N операции выразимы над множеством операций*

$\{\mathbb{N}\backslash, \bigcup, div(a), mod(a), a =_? \varnothing, int(B), sidiv(B), simod(B)\}$,

причём три последних операции в выражающем терме не вложены друг в друга.

Доказательство.

Рассмотрим операции, 2-выразимые над обозначенным набором операций. Так как все булевы функции выражаются над $\{\neg, \bigvee\}$, и верны равенства:

$set_x(\neg a) = \mathbb{N}\backslash set_x(a)$ и

$set_x(a \bigvee b) = set_x(b) \bigcup set_x(a)$, тотакие операции бесквантворно выражаются через операции вида:

$set_x(A)$, где A — формула, выраженная над $\{\mathbb{N}\backslash, \bigcup, (a \in_? b), div(a), mod(a), = \varnothing\}$, не содержащая булевых операций. Так как $= \varnothing$ — единственный предикат в указанном множестве, то достаточно рассматривать предикаты, выразимые формулой вида $set_x(A = \varnothing)$

Если A не содержит x, то данный терм задаёт ту же операцию, что и терм $A =_? \varnothing$

Также аналогично выразимости в $\mathbb{R}$ можно считать, что A не содержит условных операций. Действительно, $set_x(A) = set_x(B\&(a \in b)) \bigcup set_x(C\&\neg(a \in b))$, где терм B получен из A путём замены терма $(a \in b)$ на $\mathbb{N}$ (выразимая операция, равная $\mathbb{N}\backslash\varnothing$, где $\varnothing$ бесквантворно выразимо без использования условной операции), терм C получен из терма A путём замены терма $(a \in b)$ на $\varnothing$.

Аналогично случаю выразимости в $\mathbb{R}$ можем считать, что в формуле A операции mul() и div() применяются только к переменным, так как вхождения тех из них, которые применены не к переменным можно заменить на $\varnothing$, не меняя значения формулы.

Следовательно, можно рассматривать предикаты и операции, выразимые с помощью термов вида $set_x(A = \varnothing)$, где A — терм над $\mathbb{N}\backslash, \bigcup, x, x_i, div(x), div(x_i)$,
$mul(x), mul(x_i)$. Как было показано в главе про $\mathbb{R}$, его можно выразить над $\mathbb{N}\backslash, \bigcup, \bigcap, x, x_i$,
$, div(x), div(x_i), mul(x), mul(x_i)$ таким образом, что наружной операцией будет $\bigcup$, затем будет идти $\bigcap$, затем — $\mathbb{N}\backslash$, и применяться эти операции будут к $x, x_i, div(x), div(x_i), mul(x), mul(x_i)$ (некоторые операции в цепочке вложенностей могут отсутствовать). Так как $set_x(A_1 \bigcup \cdots \bigcup A_n = \varnothing) = set_x(A_1 = \varnothing) \bigcap \cdots \bigcap set_x(A_n = \varnothing)$, можно рассматривать те термы, для которых внешняя операция $\bigcup$ отсутствует.

Так как $\mathbb{N} = (mul(x)\backslash x) \bigcup x \bigcup (div(x)\backslash x) \bigcup (\mathbb{N}\backslash div(x)\backslash mul(x))$, то $set_x(B = \varnothing)$ можно заменить на $set_x(B \bigcap (\mathbb{N}\backslash div(x)\backslash mul(x)) = \varnothing) \bigcap set_x(B \bigcap (x) = \varnothing) \bigcap (B \bigcap div(x)\backslash x) \bigcap (B \bigcap mod(x)\backslash x)$, где B — терм над $x, x_i, div(x), div(x_i), mul(x), mul(x_i)$ как переменными с внешней операцией $\bigcap$ и внутренней $\mathbb{N}\backslash$. Пусть она имеет вид $C_1 \bigcap \cdots \bigcap C_n$

Аналогично случаю $\mathbb{R}$, про все C_i, содержащие x, можно сказать, включают ли они в себя результат соответствующей операции над x $((mul(x)\backslash x), x, (div(x)\backslash x), (\mathbb{N}\backslash div(x)\backslash mul(x)))$, а объединение C_i, не содержащих x, обозначить как D. Следовательно, исходный терм бес-

кванторно выражается над термами вида $set_x(B \bigcap p(x) = \varnothing)$, где $p(x)$ — один из термов $(mul(x)\backslash x), x, (div(x)\backslash x), (\mathbb{N}\backslash div(x)\backslash mul(x))$, а B — 1-выразимый терм над свободными переменными.

Имеем: $set_x(B \bigcap x = \varnothing) = B$ $set_x(B \bigcap (mul(x)\backslash x) = \varnothing) = \mathbb{N}\backslash sidiv(B)$ $set_x(B \bigcap (div(x)\backslash x) = \varnothing) = \mathbb{N}\backslash simod(B)$ $set_x(B \bigcap (\mathbb{N}\backslash div(x)\backslash mul(x)) = \varnothing) = int(B)$

Следовательно, все 2-выразимые в N операции выразимы над $\{\mathbb{N}\backslash, \bigcup, div(a), mod(a), a =_? \varnothing, int(B), sidiv(B), simod(B)\}$.

Также заметим, что $(a \in_? b)$ выразима над $\{\mathbb{N}\backslash, \bigcup, (a \in_? b), div(a), mod(a), a =_? \varnothing, int(B), sidiv(B), sidiv(B)\}$:

$(a \in_? b) = (b\backslash(\mathbb{N}\backslash((\mathbb{N}\backslash div(a)) \bigcup (\mathbb{N}\backslash mul(a)))) = \varnothing,$

следовательно, её можно исключить из множества операций, необходимых для бесквантороного выражения всех 2-выразимых операций.

Утверждение доказано.

Например, к 2-выразимым операциям относятся $cd(A)$ — множество общих делителей A ($cd(A) = int(A)\backslash simod(A)$), $cm(A)$ — множество общих кратных A ($cm(A) = int(A)\backslash sidiv(A)$), $ud(A)$ — объединение множеств делителей элементов A = $ud(A) \bigcup A$, $um(A)$ — объединение множеств кратных элементов A = $um(A) \bigcup A$. Однако в утверждении теоремы заменить операции $int(B), sidiv(B), simod(B)$ на более естественные нельзя — ни одна из них не выразима над множеством $\{\mathbb{N}\backslash, \bigcup, a =_? \varnothing, int(B), sidiv(B), simod(B), cd(B), cm(B), ud(B), um(B)\}$ с исключённой этой операцией бесквантороной формулой, в которой последние семь операций не вложены друг в друга.

Заметим что, если первые три операции применить к множеству которое или содержит некоторые два заданных элемента, или их не содержат, то результат операции не может содержать ровно один из этих элементов. Приведём примеры множеств и двух чисел, шесть из последних семи последних операций, применённых к которому (а также к его дополнению) будут содержать или оба этих числа, или ни одного, а седьмая будет содержать ровно одно это число.

Пример 1 Рассмотрим $B = \{2, 4, 8, 9\}$ и числа 2,4. Тогда $int(B), sidiv(B), ud(B),$
$um(B)$ содержат оба этих числа, $cd(B), cm(B)$ — ни одного из этих чисел, а $simod(B)$ содержит 2, но не 4. Также $sidiv(\mathbb{N}\backslash B), ud(\mathbb{N}\backslash B), um(\mathbb{N}\backslash B)$ содержат оба этих числа, а $cd(\mathbb{N}\backslash B), cm(\mathbb{N}\backslash B), int(\mathbb{N}\backslash B)$ — нет. При применении этих операций к $\mathbb{N}$ и $\varnothing$ также результат или будет содержать и 2, и 4, или не будет содержать ни одного из чисел. Следовательно, $simod(B)$ не выражается над $\{\mathbb{N}\backslash, \bigcup, a =_? \varnothing, int(B), sidiv(B), cd(B), cm(B), ud(B), um(B)\}$ термом, в котором шесть последних операций не вложены друг в друга.

Пример 2 Аналогичный пример с $B = \{2, 4, 8, 9\}$ и выбранными числами 8, 4 показывает, что операция $sidiv(B)$ не выражается над

$\{\mathbb{N}\backslash, \bigcup, a =_? \varnothing, int(B), simod(B), cd(B), cm(B), ud(B), um(B)\}$

термом, в котором шесть последних операций не вложены друг в друга.

Пример 3 Рассмотрим $B = \{2, 3, 12\}$ и числа 4, 6. Тогда $sidiv(B), simod(B)$,
$ud(B), um(B)$ содержат оба этих числа, $cd(B), cm(B)$ — ни одного из этих чисел, а $int(B)$ содержит 6, но не 4. Также $sidiv(\mathbb{N}\backslash B), simod(\mathbb{N}\backslash B), ud(\mathbb{N}\backslash B)$,
$um(\mathbb{N}\backslash B)$ содержат оба этих числа, а $cd(\mathbb{N}\backslash B), cm(\mathbb{N}\backslash B)$ — нет. При применении этих операций к $\mathbb{N}$ и $\varnothing$ также результат или будет содержать и 6, и 4, или не будет содержать ни одного из чисел. Следовательно, $int(B)$ не выражается над $\{\mathbb{N}\backslash, \bigcup, a =_? \varnothing, simod(B), sidiv(B)$,
$cd(B), cm(B), ud(B), um(B)\}$ термом, в котором шесть последних операций не вложены друг в друга.

Найдём все предикаты, 2-выразимые на множестве $\{\mathbb{N} \bigcup 2^{\mathbb{N}}\}$ над множеством предикатов $\{\in, |\}$.

Теорема 8. *Все 2-выразимые в N предикаты выразимы над* $\{\mathbb{N}\backslash, \bigcup,$
$div(a), mod(a), a =_? \varnothing, int(B), sidiv(B), simod(B), a = \varnothing\}$, *причём три последних операции в выражающем терме не вложены друг в друга.*

Доказательство: Аналогично случаю с алгебраической системой $\mathbb{R}$, поскольку $\forall x((x \leq x) \to (A_1 \bigvee \cdots \bigvee A_n)) = set_x(A_1 \bigvee \cdots \bigvee A_n)$, достаточно рассмотреть выразимость предикатов, задающих существование удовлетворяющего условию множества x (не числа). То есть задаваемых формулой вида

$\forall x((\neg x \leq x) \to (A_1 \bigvee \cdots \bigvee A_n))$, где A_n — 1-выразимый предикат, внешняя операция которого — $= \varnothing$ (поскольку это единственный предикат из базиса 1-выразимых предикатов и операций), и при этом все A_i содержат x. Кроме того, можно считать, что x не входит в операции mul(x) и div(x), а также что в них входят только переменные (то есть mul и div не имеют вложенных операций). Обозначив данные операции над свободными переменными за новые переменные ($y_i = mul(x_i)$, $z_i = div(x_i)$), получим кванторный предикат над $\bigcap, \bigcup, \in, \mathbb{N}\backslash$.

По доказанному в статье [1] для U_1 утверждению в этом случае данная формула выразима через операции $\bigcap, \bigcup, \in, \mathbb{N}\backslash$ и предикат 1-элементности. При обратной замене новых переменных y_i и z_i на их значения к этому списку добавятся предикаты $y_i = mul(x_i)$, $z_i = div(x_i)$.

Следовательно, к системе, выражающей все 2-выразимые предикаты и 1-выразимые предикаты и операции добавляется только предикат од-

ноэлементности. Но он 1-выразим формулой ($x \backslash int(x) = \varnothing$ $\& int(x) \backslash x = \varnothing \& x \bigcap sidiv(x) = \varnothing \& x \bigcap simul(x) = \varnothing$), не содержащей вложенных друг в друга операций $simul, sidiv, int$. Следовательно, утверждение теоремы доказано.

5. Бесквантерная выразимость на множестве $B^n \bigcup 2^{B^n}$ с сигнатурой $\{\in, \leq\}$

Пусть B^n — n-мерный булев куб, на элементах которого естественным образом задан порядок <=. Рассмотрим множество $\mathbb{B}{=}B^n \bigcup 2^{B^n}$. Как и ранее, будем определять $set_n(P)$ как множество элементов булева куба, удовлетворяющих условию P.

Если считать число n известным, то ввиду конечности множества задача выразимости тривиальна — формульно выразимы те и только те предикаты и операции, принимающие значения-множества, которые сохранаются при любом автоморфизме куба. (Действительно, рассмотрим формулу вида $\exists a_1, \ldots, a_{n+2^n}$

(Все соотношения принадлежности и неравенства, верные для булева куба & Все возможные значения свободных переменных,

удовлетворяющие формуле). Аналогичное построение можно совершить для предикатов, используя выразимость для операции $p(a_1, \ldots a_n)$ формулы $p'(a_1, \ldots, a_n, b)$, истинной в том и только том случае, если

$b \in p(a_1, \ldots a_n))$

Однако заслуживающей рассмотрения является задача поиска предикатов и операций, определённых и выразимых одной формулой или одним термом для булева куба любой размерности.

Поскольку булев куб с отношением порядка изоморфен подмножеству натуральных чисел (а именно, если рассмотреть n простых чисел $p_1 \ldots p_n$, то n-мерному булеву кубу изоморфно подмножество всех чисел, каждый в степени не более 1), предикаты и операции, n-выразимые над множеством натуральных чисел, будут также выразимы и над булевым кубом.

Определим на точках булева куба и их множествах операции ${}^*a$, a^*, ${}^{**}a$, a^{**}, $int(a)$. Здесь значение операции *a — множество точек, меньших a, операции a* — множество точек, больших a, $**a$ — множество точек, меньших хоть одного элемента a, a^{**} — множество точек, больших хоть одного элемента a, $int(a)$ — множество точек, сравнимых с каждым элементом множества a. Для тех a, для которых значение операций не определено корректно, значение операции равно $\varnothing$.

Полностью аналогично случаю натуральных чисел доказываются следующие утверждения:

Теорема 9. *все 1-выразимые в* $\mathbb{B}$ *операции выразимы через* $\{B^n\backslash, \bigcup, (a \in_? b),^* a, a^*, = \varnothing, \}$.

, где $^*a := set_x(x \leq a), a^* := set_x(x \geq a)$

Теорема 10. *Все 2-выразимые в* $\mathbb{B}$ *предикаты и операции выразимы над* $\{B^n\backslash, \bigcup,$

$*x, x*, x =_? \varnothing, x = \varnothing, int(x),^{**} x, x^{**}\}$, *причём три последних операции в выражающем терме или формуле не вложены друг в друга.*

Теорема 11. *все 2-выразимые в* $\mathbb{B}$ *операции выразимы над* $\{B^n\backslash, \bigcup,$ $^*a, a^*, a =_? \varnothing, int(x),^{**} x, x^{**}\}$.

Докажем второе утверждение. Рассмотрим операции, 2-выразимые над обозначенным набором операций. Так как все булевы функции выражаются над $\{\neg, \bigvee\}$, и верны равенства:

$set_x(\neg a) = \mathbb{B}\backslash set_x(a)$ и

$set_x(a \bigvee b) = set_x(b) \bigcup set_x(a)$, то такие операции бесквантorно выражаются через $B^n\backslash, \bigcup$ и операции вида:

$set_x(A)$, где A — формула, выраженная над $\{\mathbb{B}\backslash, \bigcup, (a \in_? b),^* a, a^*, = \varnothing, \}$, не содержащая булевых операций. Так как $= \varnothing$ — единственный предикат в указанном множестве, то достаточно рассматривать предикаты, выразимые формулой вида $set_x(A = \varnothing)$

Если A не содержит x, то данный терм задаёт ту же операцию, что и терм $A =_? \varnothing$

Также аналогично выразимости в $\mathbb{R}$ можно считать, что A не содержит условных операций и в формуле A операции $^*a$ и a^* применяются только к переменным, так как вхождения тех из них, которые применены не к переменным можно заменить на $\varnothing$, не меняя значения формулы.

Следовательно, можно рассматривать предикаты и операции, выразимые с помощью термов вида $set_x(A = \varnothing)$, где A — терм над $\mathbb{B}\backslash, \bigcup, x, x_i,^* x,^* x_i,^* x,^* x_i$. Как было показано в разделе про $\mathbb{R}$, его можно выразить над $\mathbb{B}\backslash, \bigcup, \bigcap, x, x_i, div(x), div(x_i), mul(x), mul(x_i)$ таким образом, что наружной операцией будет $\bigcup$, затем будет идти $\bigcap$, затем — $\mathbb{B}\backslash$, и применяться эти операции будут к $x, x_i,^* x,^* x_i, x^*, x_i^*$ (некоторые операции в цепочке вложенностей могут отсутствовать). Так как $set_x(A_1 \bigcup \cdots \bigcup A_n = \varnothing) = set_x(A_1 = \varnothing) \bigcap \cdots \bigcap set_x(A_n = \varnothing)$, можно рассматривать те термы, для которых внешняя операция $\bigcup$ отсутствует.

Так как $B^n = (x^*\backslash x) \bigcup x \bigcup (^*x\backslash x) \bigcup (B^n\backslash ^*x\backslash x^*)$, то $set_x(A = \varnothing)$ можно заменить на $set_x(A \bigcap (\mathbb{B}\backslash ^*x\backslash x^*) = \varnothing) \bigcap set_x(A \bigcap (x) = \varnothing) \bigcap (A \bigcap ^* x\backslash x) \bigcap (A \bigcap x^*\backslash x)$, где A — терм над $x, x_i,^* x,^* x_i, x^*, x_i^*$ как переменными с внешней операцией $\bigcap$ и внутренней $\mathbb{B}N\backslash$. Пусть она имеет вид $C_1 \bigcap \cdots \bigcap C_n$

Аналогично случаю $\mathbb{N}$, про все C_i, содержащие x, можно сказать, включают ли они в себя результат соответствующей операции над x: $(((x^{**})\backslash x), x, ((^{**}x)\backslash x), (\mathbb{B}\backslash(^{**}x)\backslash(x^{**})))$, а объединение C_i, не содержащих x, обозначить как D. Следовательно, исходный терм бесквантorно выражается над термами вида $set_x(A \bigcap p(x) = \varnothing)$, где $p(x)$ — один из термов $((x^{**})\backslash x), x, ((^{**}x)\backslash x), (\mathbb{B}\backslash(^{**}x)\backslash(x^{**}))$, а A — 1-выразимый терм над свободными переменными.

Имеем: $set_x(A \bigcap x = \varnothing) = A$ $set_x(A \bigcap (x^*)\backslash x) = \varnothing) = B^n\backslash^{**}A$. $set_x(A \bigcap (^*x\backslash x) = \varnothing) = B^n\backslash A$. $set_x(A \bigcap (B^n\backslash^*x\backslash x^*) = \varnothing) = int(A)$.

Аналогично случаю $\mathbb{N}$ ($a \in_? b$) выразима над остальными операциями, и её можно исключить из множества операций, необходимых для бесквантorного выражения всех 2-выразимых операций.

Следовательно, все 2-выразимые операции выразимы над $\{B^n\backslash, \bigcup, {}^*a, a^*, a =_? \varnothing, int(x), {}^{**}x, x^{**}\}$. Теорема доказана.

6. Бесквантorная выразимость на множестве U_2 с сигнатурой $\{\in\}$

Рассмотрим множества $U_0 = \mathbb{Z}$, $U_1{=}U_0 \bigcup 2^{U_0}$, $U_2{=}U_1 \bigcup 2^{U_1}$. На элементах множества U_2 естественным образом определена операция $a \in b$. Для данного множества будем интерепретировать $set_x(P)$, как $\{y \in \mathbb{U}_{И} : P(y) = И\}$.

Будем послойно изучать предикаты и операции, n — выразимые над U_2.

Рассмотрим операции, выразимые на первом слое, то есть термами вида $set_x(P)$, где P — бесквантorно выразим над $\in$. Аналогично разобранным случаям, операция, заданная данным термом, бесквантorно выразима над термами вида $set_x(A_i)$ и операциями $\bigcup$ и $U_1\backslash$, где A_i — бесквантorный терм.

Рассмотрим операции, выразимые термами $set_x(A_i)$:

$set_x(a \in b) = (a \in_? b)$ — условная операция, равная U_1, если $a \in b$, и $\varnothing$ иначе.

$set_x(x \in a) = a$. Получается тождественная операция, суперпозиция которой ничего не добавляет.

$set_x(a \in x)$ — это множество всех множеств из U_1, содержащих a. Обозначим эту операцию через a^*. В случае, если $\neg(a \in \mathbb{Z})$, её значение примем равным $\varnothing$.

Следовательно, все операции, выразимые на первом уровне над U_2 над операцией $\in$, бесквантorно выражаются над $\{U_1\backslash, \bigcup, (a \in_? b), a^*\}$. При этом в данном случае не все операции, бесквантorно выразимые через данные, 1-выразимы.

Например, рассмотрим операцию $((U_1 \backslash b) \in_? c)$. Она является 1-выразимой, но не равна тождественно $\varnothing$, так как множества могут быть элементами множеств. Например, результат этой операции равен U_2 в случае, если $b = U_1$, $c = \{\varnothing\}$.

Следовательно, доказано утверждение:

Теорема 12. *все 1-выразимые в U_2 операции выразимы через* $\{U_1\backslash, \bigcup, (a \in_? b), a^*\}$.

Рассмотрим 1-выразимые предикаты. Аналогично разобранным выше случаям, можно рассматривать предикаты вида $\forall x(((x \in U_1) \to (A_1 \bigvee \cdots \bigvee A_n))\&\forall x((\neg x \in U_1) \to (A_1 \bigvee \cdots \bigvee A_n)))$. Предикат, задаваемый формулой $\forall x(((x \in U_1) \to (A_1 \bigvee \cdots \bigvee A_n))$ равносилен предикату, задаваемому формулой $\neg(set_x(A_1 \bigvee \cdots \bigvee A_n) = \varnothing)$. Второй предикат есть предикат существования множеств, содержащих некоторый набор элементов, и не содержащих некий другой набор элементов, и выражается через $=$ и $\in U_1$. Поскольку $x = y \equiv ((x^*)\backslash(y^*) = \varnothing\&(y^*)\backslash(x^*) = \varnothing\&y\backslash x = \varnothing\&y\backslash x = \varnothing$, доказано утверждение:

Теорема 13. *все 1-выразимые в U_2 предикаты и операции выразимы через* $\{U_1\backslash, \bigcup,$
$(a \in_? b), a^*, = \varnothing\}$.

Опишем все 2-выразимые в U_2 предикаты. Аналогично разобранным ранее случаям, можно считать, что * применена только к переменным. Как и ранее, от вхождений операции $(a \in b_?)$ можно избавиться, но в этом случае необходимо добавить операцию $\in$:

$set_x(P)$, содержащий $a \in_? b$ можно заменить на $set_x(Q)\&a \in b \bigvee set_x(R)\&\neg(a \in b)$, где Q получено путём замены всех вхождений $a \in b$ на U_1, R получено путём замены всех вхождений $a \in b$ на $\varnothing$. По индукции таким образом можно устранить все условные предикаты. Таким образом, исходный терм бесквантorно выразим над множеством 1-выразимости, термами вида $set_x(a \in b_?)$ и вида $set_x(a = \varnothing)$, где a и b — 1-выразимые операции. Рассмотрим операции. выразимые термами первого вида.

Для этих термов рассмотрим следующее разложение: $set_x(A) = set_x(x^* \in U_1\&A) \bigcup set_x(\neg(x^* \in U_1)\&A)$, где предикат $x^* \in b$ равен И тогда и только тогда, когда $\neg x \in U_1$.

Рассмотрим операцию, выразимую термом $set_x(\neg(x^* \in U_1)\&A)$. Все вхождения x в данный терм, кроме вхождений в виде x^*, можно заменить на $\varnothing$, сохраняя значение операции. Следовательно, предикат A выразим над $\bigcup, U_1 \bigcap, = \varnothing$ и x^*, x_i, x_i^*. Аналогично предыдущему, A имеет вид $a = \varnothing$, где a можно привести к форме, где снаружи идут операции объединения, затем пересечения, и затем дополнения до U_1, и затем вынести

операцию объединения за предикат равенства, сведя предикат к конъюнкции предикатов вида $B \bigcap (U_1 \backslash (x^*))$, B, $B \bigcap (x^*)$, $B \bigcap (x^*) \bigcap (U_1 \backslash (x^*))$, где B не содержит x.

$set_x(B \bigcap (U_1 \backslash (x^*)) = \varnothing) = \bigcap (B)$ — пересечение всех элементов множества B.

$set_x(B = \varnothing) = (B =_? \varnothing)$ (через этот предикат и предикат $\bigcap$ выражается предикат $=_?$)

$set_x(B \bigcap (x^*) \bigcap (U_1 \backslash (x^*)) = \varnothing) = U_1 = U_1$

$set_x(B \bigcap (x^*) = \varnothing) = U_1 \backslash \bigcup (B)$, где $\bigcup (B)$ — объединение всех множеств — элементов B.

Следовательно, к набору, выражающему 1-выразимые предикаты и операции, добавляются операции $\bigcap (a)$, $\bigcup (a)$ и $\mathbb{Z}$.

Рассмотрим операции, выразимые термами вида $set_x(x^* \in U_1 \& A)$. Аналогично предыдущему получим, что все они выражаются через 1-выразимые операции и операции двух видов:

$set_x(B \bigcap (U_1 \backslash (x)) = \varnothing)$ и

$set_x(B \bigcap (x) = \varnothing) = U_1 \backslash \bigcup (B)$.

$set_x(B \bigcap (x) = \varnothing) = U_1 \backslash \bigcup (B)$ есть множество подмножеств $U_1 \backslash B$, то есть $2^{U_1 \backslash B}$. $set_x(B \bigcap (U_1 \backslash (x)) = \varnothing)$ — множество всех множеств из U_1, которые включают B. Обозначим эту операцию через $B \uparrow$.

Рассмотрим операции, выразимые термами вида $set_x(a \in b)$, где a и b выразимы над $U_1 \backslash, \bigcup^*$. Аналогично предыдущему можно отдельно рассматривать случаи, когда x принадлежит и не принадлежит $\mathbb{Z}$. Поскольку $a \in (b \bigcup c) = a \in b \bigvee a \in c$, $a \in (b \bigcap c) = a \in b \bigvee a \in c$, $a \in (U_1 \backslash b) = a \in U_1 \& \neg a \in b$, то достаточно рассматривать случаи,когда выражение b имеет вид x^*, x или a, где a не содержит x.

Первый же операнд путём приведения его к нормальной форме и используя следующие упрощения:

$x \bigcap U_1 \backslash x = \varnothing$

$a \bigcap U_1 \backslash x = a \backslash x$

$c = c \bigcap x \bigcup c \backslash x$

$a \backslash x \bigcup b \backslash x = (a \bigcup b) \backslash x$

$a \bigcap x \bigcup b \bigcap x = (a \bigcup b) \bigcap x$

можно привести к виду $a \bigcap x \bigcup b \backslash x$ (в случае, когда x принадлежит $\mathbb{Z}$, к виду $a \bigcap x^* \bigcup b \backslash x^*$).

Следовательно, достаточно рассматривать термы следующих видов:

$set_x(a \in x), set_x(a \in b), set_x(a \in x^*) = set_x(a \in U_1 \& x \in a)$ — 1-выразимые.

$set_x(a \bigcap x \bigcup b \backslash x) \in c$ — новая операция. Обозначим её через $trans(a, b, c)$

$set_x(a \bigcap x \bigcup b \backslash x) \in x = \varnothing$, так как $x \in U_1$ и не может содержать элементов U_1.

$set_x(a \bigcap x \bigcup b \backslash x) \in U_1 \backslash x = U_1$ аналогично.

$set_x(a \bigcap x^* \bigcup b \backslash x^*) \in x^* = set_x(a \bigcap x^* = \varnothing) \bigcap set_x(b \backslash x^*) \in x^* = set_x(a \bigcap x^* = \varnothing) \bigcap b \bigcap set_x(b \backslash \mathbb{Z} \backslash x^* = \varnothing)$ — выразим через уже выбранные для выражающей системы предикаты.

$set_x(a \bigcap x^* \bigcup b \backslash x^*) \in U_1 \backslash x^* = set_x(a \bigcap x^* = \varnothing) \bigcap set_x(b \backslash x^*) \in U_1 \backslash x^* = set_x(a \bigcap x^* = \varnothing) \bigcap b \bigcap set_x(b \backslash \mathbb{Z} \bigcap x^* = \varnothing)$. — также выразим через уже имеющиеся предикаты.

Теорема 14. *все 2-выразимые в U_2 операции выразимы через* $\{U_1 \backslash a, a \bigcup b, a^*, a =_? \varnothing, a \uparrow, \bigcup(a), \bigcap(a), 2^a, trans(a,b,c)\}$, *где последние пять операций не вложены друг в друга.*

Список литературы

[1] Ю.С. Капустин, Об элементарной выразимости в логике предикатов, журнал "Интеллектуальные системы. Теории и приложения т. 23, выпуск 2 2019, Москва, с.135-166

[2] Клини С.К. Введение в метаматематику. Пер. с анг. — М., 1957. — 528 с.

Quantifier expressibility in predicate logic

I.S. Kapustin

New mathematical concepts are often introduced with some quantifier definitions. If we have a sufficiently large stock of such notions, it can allow to reformulate the new quantifier definitions in a quantifier-free form. This makes the problem of finding basic concepts, which make further quantifiable definition redundant, worth considering. Creating computer programs that automatically introduce such bases is also worth considering.

The present paper considers the quantifier expressibility in 4 algebraic systems. This paper provides bases of quantifier expressibility of small depth.

Keywords: predicate logic, quantifier expressibility, algebraic system

Список литературы

[1] Kapustin I. S., “On the elementary expressibility in predicate logic”, *Intelligent systems*, **23**:2 (2019).

[2] S. Kleene, *Introduction to Metamathematics*, Foreign Languages Publishing House, M., 1957.

Математическая модель и методы верификации криптографических протоколов

А. М. Миронов[1]

В настоящей работе излагается новая математическая модель криптографических протоколов, и приводятся примеры применения этой модели для решения задач верификации криптографических протоколов. Криптографические протоколы – это распределенные алгоритмы, предназначенные для обеспечения передачи конфиденциальной информации в небезопасной среде. Они используются, например, в электронных платежах, электронных процедурах голосования, системах доступа к конфиденциальным данным, и т.д. Ошибки в криптографических протоколах могут привести к большому ущербу, поэтому необходимо использовать математические методы для обоснования различных свойств корректности и безопасности криптографических протоколов. В работе излагаются новые методы формальной верификации криптографических протоколов.

Ключевые слова: криптографические протоколы, последовательные процессы, распределенные процессы, верификация.

1. Введение

1.1. Понятие криптографического протокола

Криптографический протокол (КП) представляет собой распределенный алгоритм, описывающий порядок обмена сообщениями между несколькими агентами. Примеры таких агентов – компьютерные системы, банковские карточки, люди, и т.д.

Для обеспечения свойств безопасности КП (таких например как конфиденциальность передаваемых данных) в КП могут использоваться криптографические преобразования (шифрование, электронная подпись, хэш-функции, и т.п.). Мы предполагаем, что криптографические преобразования, используемые в КП, являются идеальными, т.е. удовлетворяют некоторым аксиомам, выражающим, например, невозможность

[1] *Миронов Андрей Михайлович* — доцент каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: amironov66@gmail.com.

Mironov Andrew Mikhailovich — associate professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

извлечения открытых текстов из шифртекстов без знания соответствующих криптографических ключей.

1.2. Уязвимости в криптографических протоколах

Многие уязвимости в КП связаны не с плохими криптографическими качествами используемых в них криптографических примитивов, а с логическими ошибками в КП. Наиболее ярким примером уязвимости в КП является уязвимость в КП аутентификации Нидхэма-Шредера [78NS], который был опубликован в 1978 г., и использовался в критических по безопасности информационных системах. Спустя более 16 лет после начала использования этого КП в нем обнаружилась логическая ошибка [95L], связанная с возможностью непредусмотренного нечестного поведения одного из участников этого КП и подрывающая безопасность этого КП. Особенность этой ошибки заключается в том, что данный КП является предельно простым распределенным алгоритмом, состоящим всего из трех действий, и при визуальном анализе этого КП отсутствие в нем ошибок не вызывало никаких сомнений. Ошибка была обнаружена лишь при помощи инструмента автоматизированной верификации КП.

Другой пример логической ошибки в КП (взят из статьи [14CK]): в КП входа в портал Google, позволяющем пользователю идентифицировать себя только один раз, а затем обращаться к различным приложениям (таким, например, как Gmail или календарь Google), обнаружена логическая ошибка, позволяющая нечестному поставщику услуг выдавать себя за любого из своих пользователей для другого поставщика услуг.

Существует много других примеров КП (см. например [81DS], [87NS], [95AN] [08CJSTW]), в которых обнаружились уязвимости следующего вида:

- участники этих КП могут получать искаженные сообщения (или вообще терять их) в результате перехвата, удаления или искажения противником передаваемых сообщений, что нарушает свойство целостности передаваемых сообщений,
- противник может узнать секретную информацию, содержащуюся в перехваченных сообщениях, в результате чего нарушается свойство конфиденциальности передаваемых сообщений.

Также есть примеры уязвимостей в КП, используемых для аутентификации перед провайдерами мобильной телефонной связи, для снятия денег в банкомате, для работы с электронными паспортами, проведения электронных выборов, и т.д.

Все эти примеры являются обоснованием того, что в критических по безопасности системах недостаточно неформального анализа требуемых свойств безопасности используемых в них КП, необходимо

- построение **математических моделей** анализируемых КП,
- описание свойств анализируемых КП в виде математических объектов, называемых **спецификациями** свойств этих КП, и
- построение формальных доказательств утверждений о том, что анализируемые КП удовлетворяют (или не удовлетворяют) своим спецификациям, процедура построения таких доказательств называется **верификацией** анализируемых КП.

В настоящей работе строится новая математическая модель КП, в терминах которой можно выражать такие свойства корректности КП, как например целостность и конфиденциальность передаваемых сообщений (т.е. обоснование следующих свойств анализируемого КП: сообщения, посланные одним участником этого КП другому участнику этого КП, доходят до получателя в неискаженном виде, и содержание этих сообщений не будет известно противнику), или аутентификация (т.е. доказательство подлинности) участников КП.

1.3. Основные методы моделирования и верификации криптографических протоколов

Обзоры наиболее широко используемых методов моделирования и верификации КП содержатся в книгах [11СК] и [12СМ]. Основные классы моделей КП и подходов к верификации КП имеют следующий вид.

1) **Логические модели**.

   Данный класс моделей был самым первым подходом к моделированию и верификации КП. На основе данного класса моделей проблема верификации КП сводится к проблеме построения в некотором логическом исчислении доказательства теоремы о том, что анализируемый КП обладает заданными свойствами. В работе [90BAN] была изложена первая математическая модель КП, называемая **логикой BAN** (название этой логики соответствует фамилиям ее создателей – Бэрроуза, Абади и Нидхэма). Данная модель имеет большие ограничения: в ней предполагается, что участники анализируемого КП являются честными, т.е. точно выполняют предписания КП. Такое ограничение не позволило обнаружить упомянутую выше уязвимость в КП Нидхэма-Шредера. Кроме того, данная модель не позволяет анализировать КП с неограниченным порождением сеансов. Аппарат логики BAN был развит в работах [90GNY],

[91AT], [93vO] [93SM], [94KMM], [96SvO], [02SW]. Важным классом логических исчислений для моделирования и анализа КП является композиционная логика протоколов (Protocol Composition Logic), которой посвящены работы [01DMP], [07DDMR], [08C], [11DMRS].

Одним из классов логических моделей КП связан с логическим программированием. В данных моделях шаги протокола представляются в виде правил переписывания термов. Для моделирования КП используются клаузы Хорна и системы уравнений с ограничениями (constraint systems). Данный подход излагается в работах [01B], [05AB], [14CK] и др.

Важным классом логических методов моделирования и анализа КП является индуктивный метод Паульсона: [97P], [98P], [99P], [00B].

2) Модели, основанные на **алгебре процессов**.

Источником данного класса моделей является основополагающая работа Р.Милнера [80M]. В данной работе строится модель взаимодействующих процессов, в которой процессы представляются термами. На этих термах вводится отношение наблюдаемой эквивалентности, которое позволяет эффективно выражать различные свойства процессов, связанные с безопасностью (в частности свойства секретности и анонимности). Первой работой, в которой излагается модель КП на базе подхода Р.Милнера, является статья М.Абади и А.Гордона [99AG]. Среди других работ, относящихся к этому направлению, можно отметить работы [00RS], [01AF], [05KR], [07ABF], [11RS], [16ABF], [16B], [17CW], [21CDS].

3) Модели, основанные на **CSP**.

CSP (**Communicating Sequential Processes**) – это математический аппарат, разработанный А.Хоаром [85H] и предназначенный для моделирования и анализа распределенных вычислительных процессов. На базе этого аппарата построен метод моделирования и верификации КП, наиболее полно изложенный в книге [00RSGLR]. Дедуктивная верификация КП на основе данного подхода использует понятие **ранг-функции**. Среди работ, относящихся к данному направлению, можно отметить работы [96SS], [96S], [97LR], [97DS], [98S], [21RCSSS].

4) Модели, основанные на **пространствах нитей** (**strand spaces**).

Пространства нитей позволяют представлять процессы, входящие в КП, в виде графических объектов (называемых нитями), в которых указаны зависимости между действиями, относящимися к различным процессам. Среди работ, относящихся к методам модели-

рования и верификации КП на основе понятия пространства нитей, можно отметить работы [98THG1], [98THG2], [99THG1], [99THG2], [00GT2], [02GT], [05CDLMS], [07DGT1], [07DGT2], [07DGT3], [12G], [13LP], [16YEMM].

1.4. Сравнение предлагаемой модели криптографических протоколов с другими моделями

Модель КП, излагаемая в настоящей работе, унаследовала наиболее существенные качества моделей каждого из перечисленных выше четырех классов. В этой модели КП представляются в виде распределенных процессов (РП), взаимодействующих путем асинхронной передачи сообщений через каналы. Каждый РП, соответствующий какому-либо КП, представляет собой совокупность последовательных процессов (ПП), моделирующих работу участников этого КП. Как правило,

- эти ПП представляют собой последовательности действий, которые графически можно изобразить в виде нитей, и
- выполнение всего КП можно представить в виде пространства нитей, точки на которых связаны ребрами, изображающими передачу и прием сообщений, см. например (68) и (95).

Свойства КП могут представляться в виде логических формул, для обоснования которых могут использоваться стандартные алгоритмы логического вывода. Кроме того, некоторые свойства КП (например анонимность) м.б. выражены в виде отношения наблюдаемой эквивалентности между соответствующими РП, аналогично тому, как это делается в моделях КП основанных на процессной алгебре.

Перечислим основные достоинства предложенной модели.

1) Доказательства свойств корректности КП на базе данной модели, существенно короче, чем доказательства этих свойств на базе других моделей КП. Для обоснования этого утверждения мы приводим примеры верификации двух КП: Yahalom [00RSGLR] и КП передачи сообщений, взятый из [99AG]. Верификация этих КП в вышеприведенных источниках занимает несколько десятков страниц, в то время как верификация КП Yahalom на базе предложенной модели (пункты 3.2.3, 3.2.4, 3.2.5) занимает менее 4 страниц, а верификация второго КП (пункт 3.3.2) – менее 3 страниц. Кроме того, анализ доказательств корректности данных КП показывает, что эти доказательства производятся по шаблонной методике и м.б. порождены автоматически.

2) Если анализируемые КП состоят из конечного числа компонентов без циклов, то верификация таких КП может быть проведена полностью автоматически, на основе введенного в настоящей работе понятия графа переходов, что показано на четырех примерах КП (пункты 3.1.4, 3.1.5, 3.1.7, 3.1.8), взятых из работы [99AG]. В этой работе верификация данных КП представляет собой нетривиальные математические рассуждения, в то время как в настоящей работе данные КП верифицируются путем вычисления формул, истинных в вершинах графов переходов.

3) Введенный в настоящей работе язык описания РП позволяет строить такие модели КП, которые имеют существенное сходство с исходными описаниями КП. Это обстоятельство является существенным в том случае, когда в анализируемом КП в результате верификации обнаружена ошибка, и необходимо так модифицировать анализируемый КП, чтобы устранить эту ошибку. Если модель КП схожа с описанием этого КП на исходном языке, то для устранения обнаруженной ошибки в КП сначала может быть выполнена коррекция модели этого КП, которая затем несложно преобразуется в коррекцию анализируемого КП на исходном языке.

   Отметим, что изложенный в настоящей работе язык описания РП имеет самостоятельную ценность, и может рассматриваться как новый язык описания распределенных алгоритмов.

2. Последовательные и распределенные процессы

В этом параграфе мы излагаем понятия последовательного и распределенного процессов. Последовательный процесс является моделью участника КП, а распределенный процесс является моделью всего КП. Предложенная модель является теоретической основой для методов верификации КП, излагаемых в параграфе 3.

2.1. Вспомогательные понятия

2.1.1Типы, константы, переменные, функциональные символы

Предполагаем, что заданы множества $Types$, Con, Var и Fun, элементы которых называются **типами**, **константами**, **переменными**, и **функциональными символами (ФС)**, соответственно.

Каждому элементу x множеств Con, Var и Fun сопоставлен некоторый тип $\tau(x) \in Types$, причем если $x \in Fun$, то $\tau(x)$ имеет вид

$$(\tau_1, \ldots, \tau_n) \to \tau, \quad \text{где } \tau_1, \ldots, \tau_n, \tau \in Types.$$

2.1.2. Термы

В этом пункте определяется множество Tm **термов**, которые предназначены для описания сообщений, пересылаемых во время выполнения КП. Множество Tm определяется индуктивно. Каждому терму e сопоставлен некоторый тип $\tau(e) \in Types$. Определение терма имеет следующий вид:

- $\forall\, x \in Con \cup Var \;\; x$ является термом типа $\tau(x)$,
- если $f \in Fun$, $e_1, \ldots, e_n$ – термы, и $\tau(f) = (\tau(e_1), \ldots, \tau(e_n)) \to \tau$, то запись $f(e_1, \ldots, e_n)$ является термом типа τ.

Будем использовать следующие обозначения:

- $\forall\, e \in Tm \;\; Var(e) = \{x \in Var \mid x \text{ входит в } e\}$,
- $\forall\, X \subseteq Var \;\; Tm(X) = \{e \in Tm \mid Var(e) \subseteq X\}$,
- $\forall\, E \subseteq Tm \;\; E_{\mathbf{X}} = E \cap Var$, и $\forall\, \tau \in Types \;\; E_\tau = \{e \in E \mid \tau(e) = \tau\}$.

Пусть $e, e' \in Tm$. Терм e называется **подтермом** терма e', если $e = e'$, или e' имеет вид $f(e_1, \ldots, e_n)$, где $f \in Fun$, и $\exists\, i \in \{1, \ldots, n\}$: e – подтерм терма e_i. Запись $e \subseteq e'$, где $e, e' \in Tm$, означает, что e – подтерм e'. Запись $e \subset e'$, где $e, e' \in Tm$, означает, что $e \subseteq e'$ и $e \neq e'$.

Индукцией по структуре терма $e \in Tm$ нетрудно доказать, что

$$\begin{array}{l}\text{если } e_1 \text{ и } e_2 \text{ – различные подтермы терма } e, \text{ то либо } e_1 \subset e_2,\\ \text{либо } e_2 \subset e_1, \text{ либо } e_1 \text{ и } e_2 \text{ не имеют общих компонентов.}\end{array} \qquad (1)$$

Запись $x \in e$, где $x \in Var, e \in Tm$ обозначает утверждение $x \subseteq e$.

Ниже для каждой рассматриваемой функции вида $f : E \to E'$, где $E, E' \subseteq Tm$, будем предполагать, что $\forall\, e \in E \;\; \tau(f(e)) = \tau(e)$.

2.1.3. Примеры типов

Будем считать, что $Types$ содержит следующие типы:

- тип **A**, термы этого типа называются **агентами**,
- тип **C**, термы этого типа называются **каналами**, они обозначают каналы связи, при помощи которых агенты взаимодействуют друг с другом путем передачи сообщений,

- тип $\mathbf{K}$, термы этого типа называются **ключами**, они обозначают криптографические ключи, которые агенты могут использовать для шифрования или дешифрования сообщений,
- тип $\mathbf{M}$, термы этого типа называются **сообщениями**, они обозначают сообщения, которые агенты могут пересылать друг другу во время своей работы,
- тип $\mathbf{N}$, термы этого типа называются **нонсами**, они обозначают переменные с уникальными значениями,
- тип $\mathbf{P}$, термы этого типа называются **процессами**.

Записи *Agents*, *Channels*, *Keys*, *Messages*, *Nonces* и *Processes* обозначают множества всех агентов, каналов, ключей, сообщений, нонсов и процессов, соответственно.

Будем использовать следующие соглашения и обозначения:

- множество *Channels* содержит переменную, обозначаемую символом $\circ$, и называемую **открытым каналом**,
- тип $\mathbf{M}$ включает все другие типы из $Types$, т.е. терм любого типа является также термом типа $\mathbf{M}$,
- $\forall\, n \geq 1, \forall\, \tau \in Types$ множество $Types$ содержит тип τ_n, значения которого – кортежи длины n, состоящие из значений типа τ.

2.1.4. Примеры функциональных символов

Будем предполагать, что Fun содержит следующие ФС.

- ФС $tuple_n$, где $n \geq 1$ и $\tau(tuple_n) = (\underbrace{\mathbf{M}, \ldots, \mathbf{M}}_{n}) \to \mathbf{M}_n$.

  Для каждого списка $(e_1, \ldots, e_n)$ термов терм $tuple_n(e_1, \ldots, e_n)$ будет обозначаться более короткой записью $(e_1, \ldots, e_n)$.

- ФС $pr_{n,i}$, где $n \geq 1$, $i \in \{1, \ldots, n\}$, и $\tau(pr_{n,i}) = \mathbf{M}_n \to \mathbf{M}$.

  $\forall\, e \in Tm_{\mathbf{M}_n}$ терм $pr_{n,i}(e)$ является i–й компонентой кортежа e, и будет обозначаться записью $(e)_i$.

- ФС $hash\_function$ (возможно с индексами) типа $\mathbf{M} \to \mathbf{M}$.

  Терм вида $hash\_function(e)$ обозначает значение **хэш-функции** сообщения e.

- ФС *encrypt* и *decrypt* типа $(\mathbf{K}, \mathbf{M}) \to \mathbf{M}$.

  Термы вида $encrypt(k, e)$ и $decrypt(k, e)$ обозначают сообщения, получаемые шифрованием (и дешифрованием, соответственно) сообщения e на ключе k.

- ФС $public\_key$ типа $\mathbf{A} \to \mathbf{K}$.

  Терм $public\_key(A)$ называется **открытым ключом** агента A.

  Термы вида $encrypt(k, e)$ и $encrypt(public\_key(A), e)$ будет обозначаться записями $k(e)$ и $A(e)$ соответственно, данные термы называются **шифрованными сообщениями (ШС)**.

- ФС $shared\_key$ типа $\mathbf{A}_n \to \mathbf{K}$, где $n \geq 2$ (т.е. одно и то же обозначение $shared\_key$ используется для семейства ФС).

  Терм вида $shared\_key(A_1, \ldots, A_n)$ называется **разделяемым ключом** агентов $A_1, \ldots, A_n$ и будет обозначаться записью $k_{A_1 \ldots A_n}$.

- ФС $shared\_channel$ типа $\mathbf{A}_n \to \mathbf{C}$, где $n \geq 2$ (т.е. одно и то же обозначение $shared\_channel$ используется для семейства ФС).

  Терм вида $shared\_channel(A_1, \ldots, A_n)$ называется **разделяемым каналом** агентов $A_1, \ldots, A_n$ и будет обозначаться записью $c_{A_1 \ldots A_n}$.

- ФС $digital\_signature$ типа $(\mathbf{M}, \mathbf{A}) \to \mathbf{M}$.

  Терм вида $digital\_signature(e, A)$ обозначает **цифровую подпись** сообщения e, сделанную агентом A.

  Тройка $(e, A, digital\_signature(e, A))$ будет обозначаться $(e)_A$.

Будем использовать следующие обозначения: $\forall\, e \in Tm$

$$\begin{aligned} VarEncKeys(e) &= \{k \in Var_{\mathbf{K}} \mid \exists\, e' \in Tm : k(e') \subseteq e\}, \\ OpenEncKeys(e) &= \{A \in Var_{\mathbf{A}} \mid \exists\, e' \in Tm : A(e') \subseteq e\}. \end{aligned}$$

2.1.5. Выражения

В этом пункте определяется множество $Expr$ **выражений**, которые предназначены для описания множеств термов. Например, в качестве такого множества может выступать совокупность термов, доступных в текущий момент какому-либо процессу, или совокупность сообщений, находящихся в текущий момент в каком-либо канале.

Выражением называется запись одного из следующих видов:

- E, где $E \subseteq Tm$,

- $[P]$ и $[c]$, где $P \in Processes$, $c \in Channels$,

- $k^{-1}(E)$, где $k \in Keys$, и $E \in Expr$,
- $E \cap E'$, $E \cup E'$, $\neg E$, где $E, E' \in Expr$.

$\forall\, E \in Expr \;\; Var(E) = \{x \in Var \mid x \text{ входит в } E\}$.

Выражения вида $k^{-1}([P])$ и $k^{-1}([c])$ обозначаются $k^{-1}[P]$ и $k^{-1}[c]$ соответственно. Выражения вида $\{e\}$, где $e \in Tm$, обозначаются без фигурных скобок.

Ниже каждому выражению сопоставляется **значение** этого выражения в текущий момент времени, которое является множеством термов.

2.1.6. Формулы

В этом пункте определяется понятие **формулы**, которое предназначено для описания свойств множеств термов. В определении данного понятия используется понятие **элементарной формулы (ЭФ)**, которая представляет собой запись одного из следущих видов:

1) $e \in E$, $E = E'$, $E \subseteq E'$, $E \supseteq E'$, где $e \in Tm$, $E, E' \in Expr$,
2) $E \perp_{\mathbf{C}} P$ и $E \perp_{\mathbf{K}} P$, где $E \subseteq Tm, P \in Processes$,
3) $at_P = i$, где $P \in Processes$.

ЭФ выражают свойства значений входящих в них выражений в текущий момент времени. ЭФ из первого пункта выражают свойства, соответствующие входящим в них теоретико-множественным символам. ЭФ из второго пункта выражают свойства, изложенные в пункте 2.2.4, ЭФ из третьего пункта выражают свойства текущего состояния последовательного процесса, подробнее см. в пункте 2.2.3.

Примеры ЭФ:

$$\left.\begin{array}{ll} decrypt(k, k(e)) = e, & \text{где } k \in Var_{\mathbf{K}}, e \in Tm \\ pr_{n,i}(e_1, \ldots, e_n) = e_i, & \text{где } n > 0, i \in \{1, \ldots, n\}, e_1, \ldots, e_n \in Tm. \end{array}\right\} \quad (2)$$

Формулой называется произвольная совокупность ЭФ. Каждая формула $\varphi = \{\varphi_i \mid i \in I\}$ выражает утверждение, представляющее собой конъюнкцию утверждений, выражаемых ЭФ $\varphi_i \;\; (i \in I)$.

Множество всех формул обозначается записью Fm. $\forall\, \varphi \in Fm$ запись $Var(\varphi)$ обозначает множество всех переменных, входящих в φ.

Для каждого списка формул φ_1, …, $\varphi_n \in Fm$ формула $\varphi_1 \cup \ldots \cup \varphi_n$ будет обозначаться записью $\{\varphi_1, \ldots, \varphi_n\}$.

Формулы вида $\neg(e \in E)$ будут обозначаться записями вида $e \notin E$.

Запись вида $E_0 \rho_1 E_1 \ldots \rho_n E_n$, где $E_0, \ldots, E_n$ – выражения, и ρ_1, …, ρ_n – символы из $\{=, \subseteq\}$, обозначает формулу $\{E_0 \rho_1 E_1, \ldots, E_{n-1} \rho_n E_n\}$.

2.1.7. Связывания

Связыванием называется произвольная функция $\theta : Var \to Tm$. Будем говорить, что связывание θ связывает переменную $x \in Var$ с термом $\theta(x)$.

Будем использовать следующие обозначения:

- множество всех связываний обозначается символом Θ,
- id обозначает тождественное связывание: $\forall\, x \in Var \;\; id(x) = x$,
- $\forall\, X \subseteq Var \quad \Theta(X) = \{\theta \in \Theta \mid \forall\, x \in Var \setminus X \;\; \theta(x) = x\}$,
- связывание $\theta \in \Theta$ может обозначаться записями

$$x \mapsto \theta(x) \quad \text{или} \quad (\theta(x_1)/x_1, \ldots, \theta(x_n)/x_n), \tag{3}$$

  вторая запись в (3) используется, когда $\theta \in \Theta(\{x_1, \ldots, x_n\})$,
- $\forall\, \theta \in \Theta,\ \forall\, e \in Tm$ запись e^θ обозначает терм, получаемый из e заменой $\forall\, x \in Var(e)$ каждого вхождения x в e на терм $\theta(x)$, терм e называется **шаблоном** терма e^θ относительно связывания θ,
- $\forall\, \theta \in \Theta,\ \forall\, E \subseteq Tm$ запись E^θ обозначает множество $\{e^\theta \mid e \in E\}$,
- $\forall\, \theta, \theta' \in \Theta$ запись $\theta\theta'$ обозначает связывание $x \mapsto (x^\theta)^{\theta'}$.

Пусть $X \subseteq X' \subseteq Var$, $\theta \in \Theta(X)$, $\theta' \in \Theta(X')$. Связывание θ' называется **продолжением** связывания θ, если $\forall\, x \in X \;\; \theta(x) = \theta'(x)$.

2.2. Последовательные процессы

2.2.1. Действия

Действие – это запись одного из следующих видов:

$$c!e, \quad c?e, \quad e := e', \quad \text{где } c \in Channels,\ e, e' \in Tm,$$

которые называются **посылкой** сообщения e в канал c, **приемом** сообщения e из канала c, и **присваиванием**, соответственно.

Множество всех действий обозначается записью Act. $\forall\, \alpha \in Act$ множество всех переменных, входящих в α, обозначается записью $Var(\alpha)$.

Если $\theta \in \Theta$ и $\alpha \in Act$, то запись α^θ обозначает действие $c^\theta!e^\theta$, $c^\theta?e^\theta$ и $e^\theta := (e')^\theta$, если $\alpha = c!e$, $c?e$ и $e := e'$, соответственно.

В некоторых случаях, для облегчения визуального восприятия, действия могут записываться в круглых скобках, т.е., например, вместо записи $c!e$ может использоваться запись $(c!e)$, и т.д.

2.2.2. Понятие последовательного процесса

Последовательный процесс (ПП) – это четверка $(P, A, X, \bar{X})$, компоненты которой имеют следующий смысл:

- P – граф с выделенной вершиной (называемой **начальной вершиной**, и обозначаемой записью $Init(P)$), каждому ребру которого сопоставлена метка $\alpha \in Act$,
- A – **агент**, связанный с этим ПП,
- $X \subseteq Var$ – **инициализированные переменные**,
- $\bar{X} \subseteq X$ – **скрытые переменные**, они обозначают секретные ключи, скрытые каналы, или нонсы, эти переменные инициализированы уникальными значениями.

ПП является формальным описанием поведения динамической системы, работа которой заключается в последовательном выполнении действий, связанных с посылкой или приемом сообщений, а также с инициализацией неинициализированных переменных.

Для каждого ПП $(P, A, X, \bar{X})$

- данный ПП может сокращенно обозначаться тем же символом P, что и соответствующий ему граф, множество вершин графа P также обозначается символом P,
- начальная вершина P обозначается символом $\odot$; те вершины P, из которых не выходит ни одного ребра, обозначаются символом $\otimes$,
- $Agent(P)$, $X(P)$, $\bar{X}(P)$ обозначают соответствующие компоненты P, $Var(P)$ обозначает множество всех переменных, входящих в P,
- $\tilde{X}(P)$ обозначает множество $X(P) \setminus \bar{X}(P)$ инициализированных нескрытых переменных процесса P,
- $\hat{X}(P)$ обозначает множество $Var(P) \setminus X(P)$ неинициализированных переменных процесса P.

С каждым ПП связана переменная из множества $Processes$, называемая **именем** этого ПП. Будем обозначать имена ПП теми же записями, которыми обозначаются сами ПП.

Если P не имеет ребер и $X(P) = \emptyset$, то P обозначается символом $\mathbf{0}$.

Действия вида $\circ!e$ и $\circ?e$ будут более коротко обозначаться записями $!e$ и $?e$ соответственно.

2.2.3. Состояние последовательного процесса

Состояние ПП P – это пятерка $s = (at, \alpha, [P], \theta, \{[c] \mid c \in \mathit{Channels}\})$, где

- $at \in P$ – вершина графа P в состоянии s,
- $\alpha \in \{\mathit{init}\} \sqcup \mathit{Act}$ – действие перед переходом в s,
- $[P] \subseteq \mathit{Var}$ – множество инициализированных переменных в s,
- $\theta \in \Theta([P])$ – связывание в s,
- $\forall\, c \in \mathit{Channels}\ \ [c] \subseteq Tm$ – содержимое канала c в s.

Компоненты состояния s обозначаются записями at_s, α_s, $[P]_s$, θ_s, $[c]_s$ соответственно. Будем обозначать записью $\langle P \rangle_s$ множество $Tm([P]_s)$.

Состояние ПП P называется **начальным** (и обозначается 0_P), если оно имеет вид $(\mathit{Init}(P), \mathit{init}, X(P), \mathit{id}, \{\emptyset \mid c \in \mathit{Channels}\})$.

2.2.4. Значения выражений и формул в состояниях последовательных процессов

Пусть заданы ПП P, состояние s ПП P, выражение E, и формула φ.

Запись E^s обозначает множество термов, называемое **значением** E в s, и определяемое следующим образом:

- $\forall\, E \subseteq Tm \quad E^s = \{e^{\theta_s} \mid e \in E\}$,
  $\forall\, e \in Tm$ множество вида $\{e\}^s$, а также единственный элемент этого множества, будем обозначать записью e^s,
- $[P]^s = ([P]_s)^s$, $\langle P \rangle^s = (\langle P \rangle_s)^s$, $[c]^s = [c^s]_s$, где $P \in \mathit{Processes}$, $c \in \mathit{Channels}$,
- $k^{-1}(E)^s = \{e \in Tm \mid \exists\, e' \in E^s : k^s(e) \subseteq e'\}$,
- $(E \cap E')^s = E^s \cap (E')^s$, $(E \cup E')^s = E^s \cup (E')^s$, $(\neg E)^s = Tm \setminus E^s$.

Запись $s \models \varphi$ обозначает утверждение "φ **истинна в** s". Это утверждение верно, если $\mathit{Var}(\varphi)_{\mathbf{P}} \subseteq \{P\}$, и выполнено одно из условий:

- $\varphi = (e \in E)$, $(E = E')$, $(E \subseteq E')$, или $(E \supseteq E')$, где $e \in Tm$, $E, E' \in \mathit{Expr}$, и

  $$e^s \in E^s,\ E^s = (E')^s,\ E^s \subseteq (E')^s,\ E^s \supseteq (E')^s,\ \text{соответственно}$$

- $\varphi = (E \perp_{\mathbf{C}} P)$, $\forall\, e \in E^s \;\; Agent(P) \notin e$, и

$$\left.\begin{array}{l} \forall\, x \in E^s_{\mathbf{X}}, \forall\, y \in [P]_s \;\; x \notin y^s \\ \forall\, x \in E^s_{\mathbf{X}}, \forall\, c \in Channels \;\; \text{если } \exists\, e \in [c]_s : x \in e, \;\text{то } c \in E^s \end{array}\right\} \quad (4)$$

(4) можно интерпретировать как следующее утверждение: каждая переменная из $E^s_{\mathbf{X}}$ не входит в термы, доступные процессу P в состоянии s, и входит в термы из содержимого только таких каналов, которые недоступны для P,

- $\varphi = (E \perp_{\mathbf{K}} P)$, $\forall\, e \in E^s \;\; Agent(P) \notin e$, и

$$\left.\begin{array}{l} \forall\, x \in E^s_{\mathbf{X}},\; \forall\, y \in [P]_s \;\; x \perp_{\mathbf{K},E} y^s \\ \forall\, x \in E^s_{\mathbf{X}},\; \forall\, c \in Channels, \forall\, e \in [c]_s \;\; x \perp_{\mathbf{K},E} e \end{array}\right\} \quad (5)$$

где $x \perp_{\mathbf{K},E} e$, означает, что

$$\begin{array}{l} \text{каждое вхождение } x \text{ в } e \text{ содержится} \\ \text{в подтерме } k(\ldots) \subseteq e, \text{ где } k \in E^s_{\mathbf{K}} \end{array} \quad (6)$$

(5) можно интерпретировать как следующее утверждение: переменные из $E^s_{\mathbf{X}}$ входят в термы, доступные процессу P в состоянии s, а также в термы из содержимого произвольного канала, в "защищённом" виде, т.е. входят в подтермы вида $k(\ldots)$, где $k \in E^s_{\mathbf{K}}$,

- $\varphi = (at_P = i)$, и $at_s = i$,
- $\varphi = \{\varphi_i \mid i \in I\}$ – совокупность ЭФ, и $\forall\, i \in I \;\; s \models \varphi_i$.

2.2.5. Выполнение последовательного процесса

Выполнение ПП P можно понимать как обход вершин P, начиная с $Init(P)$, с выполнением действий, являющихся метками проходимых рёбер. С каждым шагом выполнения ПП P связано некоторое состояние s ПП P, называемое **текущим состоянием** ПП P на этом шаге (на первом шаге текущим состоянием является 0_P).

Если текущий шаг выполнения ПП P не является заключительным, то на этом шаге происходит замена текущего состояния s на состояние s', которое будет текущим состоянием на следующем шаге, для этого

1) либо выбирается выходящее из at_s ребро графа P, метка α которого обладает следующими свойствами:

   - если α^{θ_s} содержит вхождение терма вида $shared\_key(\ldots)$, или $shared\_channel(\ldots)$, то $Agent(P)$ присутствует в этом вхождении,

- выполнено одно из условий:

$$\left.\begin{array}{ll}
\text{(a)} & \alpha = c!e,\ c, e \in \langle P\rangle_s \\
\text{(b)} & \alpha = c?e,\ c \in \langle P\rangle_s, \\
 & \quad VarEncKeys(e^s) \subseteq [P]_s, \\
 & \quad OpenEncKeys(e^s) \subseteq \{Agent(P)\}, \\
 & \quad \exists\, \theta \in \Theta(\, Var(e) \setminus [P]_s) : (e^\theta)^s \in [c]^s \\
\text{(c)} & \alpha = (e := e'),\ e' \in \langle P\rangle_s, \\
 & \quad VarEncKeys(e^s) \subseteq [P]_s, \\
 & \quad OpenEncKeys(e^s) \subseteq \{Agent(P)\}, \\
 & \quad \exists\, \theta \in \Theta(\, Var(e) \setminus [P]_s) : e^\theta = e'
\end{array}\right\} \qquad (7)$$

и компоненты состояния s' имеют следующий вид: $at_{s'}$ – конец выбранного ребра, $\alpha_{s'} = \alpha$, и

- если верно (a) в (7), то $[P]_{s'} = [P]_s$, $\theta_{s'} = \theta_s$, $[c^s]_{s'} = [c^s]_s \cup \{e^s\}$, $\forall\, c' \in Channels \setminus \{c^s\}\ \ [c']_{s'} = [c']_s$,
- если верно (b) или (c) в (7), то $[P]_{s'} = [P]_s \cup Var(e)$, $\theta_{s'} = \theta\theta_s$, $\forall\, c' \in Channels\ \ [c']_{s'} = [c']_s$,
  (будем говорить, что при переходе от s к s' каждая переменная $x \in Var(e) \setminus [P]_s$ инициализируется значением $x^{\theta_{s'}}$, которое становится доступным P),

2) либо все компоненты состояния s', кроме последней, совпадают с соответствующими компонентами состояния s, и $\forall\, c \in Channels$ множество $[c]_{s'}$ либо совпадает с $[c]_s$, либо получается путем добавления терма к множеству $[c]_s$ в результате выполнения текущего шага другим ПП.

Если имеет место первая (вторая) из указанных выше ситуаций, то будем говорить, что s' получается **активным** (соответственно, **пассивным**) переходом из s. Запись $s \overset{P}{\rightarrow} s'$ ($s \rightarrow s'$) обозначает, что s' получается активным (соответственно, пассивным) переходом из s.

Во время каждого выполнения каждого ПП P переменные из $Var(P)$ имеют следующие особенности: $\forall\, x \in Var(P)$

1) если $x \in \hat{X}(P)$, то в начальный момент каждого выполнения ПП P переменная x не инициализирована, т.е. ей не сопоставлено никакого значения,

2) если $x \in \bar{X}(P)$, то это означает, что в начальный момент каждого выполнения $Exec$ ПП P данная переменная инициализирована **уникальным значением**, т.е. значением, которое отличается

- от значений, сопоставленных другим инициализированным переменным при выполнении $Exec$, и
- от значений, сопоставленных инициализированным переменным при любом выполнении $Exec' \neq Exec$ любого ПП.

Интерпретация условий, описанных в (7), имеет следующий вид.

- Условие в пункте (a) связано с выполнением посылки сообщения:
  - имя c^s канала, в который посылается сообщение, должно быть доступно процессу P в состоянии s, и
  - посылаемое сообщение e^s должно быть термом, компоненты которого также доступны процессу P в состоянии s.
- Условие в пункте (b) связано с выполнением приёма сообщения:
  - имя c^s канала, из которого принимается сообщение, должно быть доступно процессу P в состоянии s,
  - все ШС в принимаемом сообщении, которые
    - дешифруются во время приёма этого сообщения, и
    - зашифрованы не на открытом или разделяемом ключе,

    имеют вид $k(\ldots)$, где значение ключа k должно быть доступно процессу P в состоянии s, это свойство выражается во второй строке пункта (b) в (7),
  - все ШС в принимаемом сообщении, которые
    - дешифруются во время приёма этого сообщения, и
    - зашифрованы на открытом ключе,

    имеют вид $Agent(P)(e)$, это свойство выражается в третьей строке пункта (b) в (7),
  - терм e является шаблоном некоторого терма из $[c]^s$ относительно некоторого продолжения связывания θ_s, данное свойство выражается в последней строке пункта (b) в (7).
- Условие в пункте (c) связано с выполнением присваивания:
  - каждая компонента терма $(e')^s$ должна быть доступна P в s,
  - смысл свойств во второй и третьей строках пункта (c) в (7) совпадает со смыслом соответствующих свойств в пункте (b): каждое ШС в $(e')^s$, которое должно быть дешифровано во время выполнения этого присваивания, должно иметь вид $k(\ldots)$ или $Agent(P)(\ldots)$, причем

  - либо k разделяемый ключ,
  - либо $k \in Var_{\mathbf{K}}$ и значение ключа k должно быть доступно P в состоянии s,
- терм e является шаблоном терма e' относительно некоторого связывания $\theta \in \Theta(\, Var(e) \setminus [P]_s)$.

2.2.6. Процесс противника

Процесс противника – это ПП, обозначаемый записью $P_\dagger$, и обладающий следующими свойствами:

- граф ПП $P_\dagger$ состоит из единственной вершины,
- $\forall\, \tau \in Types$ множества $\bar{X}(P_\dagger)_\tau$ и $\hat{X}(P_\dagger)_\tau$ счетны,
- $\forall\, \alpha \in Act$ граф $P_\dagger$ содержит ребро с меткой α.

Ниже будем предполагать, что $P_\dagger$ – единственный из всех рассматриваемых ПП, граф которого имеет циклы.

2.2.7. Переименование переменных

Переименование переменных (называемое также просто **переименованием**) – это инъективная функция $\eta : X \to X'$, где $X, X' \subseteq Var$.

Для каждого переименования $\eta : X \to X'$, каждого $e \in Tm$ и каждого ПП P записи e^η и P^η обозначают терм или ПП соответственно, получаемые из e или P заменой $\forall\, x \in X$ каждого вхождения x на $\eta(x)$.

Если переименование η имеет вид $\eta : \bar{X}(P) \cup \hat{X}(P) \to Var \setminus \tilde{X}(P)$, то ПП P и P^η будем рассматривать как равные.

2.3. Распределенные процессы

В этом пункте вводится понятие распределенного процесса, которое является моделью КП. Все КП, рассматриваемые в этом тексте, мы будем отождествлять с соответствующими им распределенными процессами.

2.3.1. Понятие распределенного процесса

Распределенный процесс (РП) – это семейство ПП:

$$\mathcal{P} = \{P_i \mid i \in I\}$$

(некоторые из которых могут совпадать). С каждым РП связана переменная типа **P**, называемая **именем** этого РП.

РП $\mathcal{P}$ является моделью распределенного алгоритма, компонентами которого являются входящие в него ПП, взаимодействующие друг с другом путем передачи сообщений через каналы.

Пусть задан РП $\mathcal{P}$. Будем использовать следующие обозначения и предположения:

- $Var(\mathcal{P}) = \bigcup_{P \in \mathcal{P}} Var(P)$, множества $X(\mathcal{P})$, $\bar{X}(\mathcal{P})$, $\tilde{X}(\mathcal{P})$, $\hat{X}(\mathcal{P})$ определяются аналогично,

- будем предполагать, что

$$\begin{array}{l}\text{компоненты семейства } \{\bar{X}(P) \cup \hat{X}(P) \mid P \in \mathcal{P}\} \\ \text{дизъюнктны и не пересекаются с } \tilde{X}(\mathcal{P})\end{array} \tag{8}$$

  (если это не так, то заменим каждый из компонентов P семейства $\mathcal{P}$ на равный ему в том смысле, который указан в конце пункта 2.2.7, так, чтобы свойство (8) выполнялось),

- РП $\mathcal{P}$ может обозначаться записью

  - $\{P_1, \ldots, P_n\}$, если $I = \{1, \ldots, n\}$ (в случае $n = 1$ скобки м.б. опущены, т.е. вместо $\{P_1\}$ пишется P_1), или
  - P^*, если I – множество натуральных чисел, и все ПП, входящие в $\mathcal{P}$, совпадают с P,

- запись $\mathcal{P}_\dagger$ обозначает РП $\{\mathcal{P}, P_\dagger\}$,

- если $\{\mathcal{P}_i \mid i \in I\}$ – семейство РП, и $\forall\, i \in I$ РП $\mathcal{P}_i$ является семейством ПП вида $\{P_{i'} \mid i' \in I_i\}$, где множества индексов I_i $(i \in I)$ дизъюнктны (если это не так, то заменим их на соответствующие дизъюнктные копии), то запись $\{\mathcal{P}_i \mid i \in I\}$ обозначает также РП $\{P_{i'} \mid i' \in \bigsqcup_{i \in I} I_i\}$.

Будем использовать следующее соглашение:

- если в каком-либо рассуждении, связанном с РП вида P^*, некоторый ПП является первым из рассматриваемых ПП, входящих в P^*, то этот ПП и все его переменные обозначаются теми же записями, которые используются в ПП P,

- если кроме этого ПП рассматривается другой ПП, входящий в P^*, то он уже обозначается записью $\grave{P}$, и в обозначениях тех его переменных, которые соответствуют переменным из $\bar{X}(P) \cup \hat{X}(P)$, используются обратные штрихи, и т.д.

2.3.2. Понятие состояния распределенного процесса

Пусть задан РП $\mathcal{P}$.

Состоянием РП $\mathcal{P}$ называется семейство $s = \{s_P \mid P \in \mathcal{P}\}$ состояний ПП, входящих в $\mathcal{P}$, такое, что $\forall\, c \in \mathit{Channels}$ все множества в семействе $\{[c]_{s_P} \mid P \in \mathcal{P}\}$ одинаковы (будем обозначать их записью $[c]_s$).

Пусть $s = \{s_P \mid P \in \mathcal{P}\}$ – состояние РП $\mathcal{P}$. Тогда

- s называется **начальным** состоянием РП $\mathcal{P}$, и обозначается $0_{\mathcal{P}}$, если $\forall\, P \in \mathcal{P}\ \ s_P = 0_P$,
- $at_s = \{at_{s_P} \mid P \in \mathcal{P}\}$, $[\mathcal{P}]_s = \bigcup_{P \in \mathcal{P}} [P]_s$, $\langle \mathcal{P} \rangle_s = Tm([\mathcal{P}]_s)$,
- θ_s обозначает связывание из $\Theta([\mathcal{P}]_s)$, такое, что

$$\forall\, P \in \mathcal{P},\ \forall\, x \in [P]_s \quad \theta_{s_P}(x) = \theta_s(x)$$

  (существование такого связывания следует из предположения (8)).

Понятия значения выражения и значения формулы в состоянии РП определяются аналогично соответствующим понятиям для ПП.

$\forall\, \varphi, \psi \in Fm$ запись $\varphi \leq \psi$ означает, что для каждого РП $\mathcal{P}$ и каждого состояния s РП $\mathcal{P}$ верна импликация $s \models \varphi \Rightarrow s \models \psi$.

Если формулы $\varphi, \psi \in Fm$ таковы, что $\varphi \leq \psi$ и $\psi \leq \varphi$, то будем рассматривать такие формулы как одинаковые. Если формулы φ и ψ одинаковы, то будем обозначать этот факт записью $\varphi = \psi$.

Примеры одинаковых формул:

- $\{f(e_1, \ldots, e_n) = f(e'_1, \ldots, e'_n)\}$, где $f \in Fun$, и $\{e_1 = e'_1, \ldots, e_n = e'_n\}$,
- $\{[c] = \{e\}, e' \in [c]\}$ и $\{[c] = \{e\}, e = e'\}$.

2.3.3. Выполнение распределённого процесса

Пусть задан РП $\mathcal{P}$. **Выполнение** РП $\mathcal{P}$ представляет собой недетерминированное чередование выполнений ПП, входящих в $\mathcal{P}$. На каждом шаге выполнения РП $\mathcal{P}$

- только один ПП из $\mathcal{P}$ выполняет активный переход, и
- остальные ПП из $\mathcal{P}$ выполняют пассивные переходы.

Выполнение РП $\mathcal{P}$ можно определить как порождение последовательности состояний этого РП (начиная с начального состояния $0_{\mathcal{P}}$), в которой каждое состояние s, не являющееся последним в этой последовательности, связано со следующим состоянием s' **отношением перехода**, что

означает следующее: $\exists\, P \in \mathcal{P}$:

$$\begin{array}{l} \overset{P}{s_P} \to\ s'_P,\ \ \forall\, P' \in \mathcal{P} \setminus \{P\}\ \ s_{P'} \to s'_{P'} \\ \text{где } s = \{s_P \mid P \in \mathcal{P}\},\ s' = \{s'_P \mid P \in \mathcal{P}\}. \end{array} \tag{9}$$

Свойство (9) обозначается записью $s \xrightarrow{\alpha_P} s'$, где $\alpha = \alpha_{s'_P}$.

Множество всех состояний РП $\mathcal{P}$ можно рассматривать как граф, в котором существует ребро из s в s' с меткой α_P тогда и только тогда когда $s \xrightarrow{\alpha_P} s'$. Обозначение ПП P в метке α_P можно опускать.

Для каждой пары состояний s, s' РП $\mathcal{P}$ запись $s \to s'$ означает, что s связано с s' отношением перехода, и запись $s \Rightarrow s'$ означает, что существует последовательность $s_1, \ldots, s_n$ состояний, такая, что $s_1 = s$, $s_n = s'$, и $\forall\, i = 1, \ldots, n-1\ \ s_i \to s_{i+1}$.

Состояние s РП $\mathcal{P}$ называется **достижимым**, если $0_{\mathcal{P}} \Rightarrow s$. Множество достижимых состояний РП $\mathcal{P}$ обозначается записью $\Sigma_{\mathcal{P}}$.

Если задан путь π из $0_{\mathcal{P}}$ в s, и s' – какое-либо состояние, входящее в π, то мы будем обозначать этот факт записью $s' \leq_\pi s$. Запись $s' <_\pi s$ обозначает, что $s' \leq_\pi s$ и $s' \neq s$. Если путь π ясен из контекста, то обозначение этого пути в записях $\leq_\pi$ и $<_\pi$ м.б. опущено.

2.3.4. Наблюдаемая эквивалентность

Понятие наблюдаемой эквивалентности РП имеет следующий неформальный смысл: РП $\mathcal{P}$ и $\mathcal{P}'$ наблюдаемо эквивалентны, если для каждого наблюдателя, который анализирует выполнение РП $\mathcal{P}_\dagger$ и $\mathcal{P}'_\dagger$ путем наблюдения за содержимым канала $\circ$, эти РП будут неразличимы.

Пусть $\mathcal{P}$, $\mathcal{P}'$ – РП, $s \in \Sigma_{\mathcal{P}_\dagger}$, $s' \in \Sigma_{\mathcal{P}'_\dagger}$, и $\eta : X \to X'$ – переименование. Запись $s \underset{\eta}{\sim} s'$ означает, что $[\circ]_s \subseteq Tm(X)$ и $[\circ]_{s'} = \{e^\eta \mid e \in [\circ]_s\}$.

Будем говорить, что РП $\mathcal{P}$ и $\mathcal{P}'$ **наблюдаемо эквивалентны**, если существует множество μ троек вида (s, s', η), где $s \in \Sigma_{\mathcal{P}_\dagger}$, $s' \in \Sigma_{\mathcal{P}'_\dagger}$, $s \underset{\eta}{\sim} s'$, такое, что выполнены следующие условия:

- $(0_{\mathcal{P}_\dagger}, 0_{\mathcal{P}'_\dagger}, \emptyset) \in \mu$ (где $\emptyset$ - функция с пустой областью определения),
- $\forall\, (s, s', \eta) \in \mu$ если $s \to \tilde{s}$ или $s' \to \tilde{s}'$, то $\exists\, (\tilde{s}, \tilde{s}', \tilde{\eta}) \in \mu$: $\tilde{\eta}$ – продолжение η, и $s' \Rightarrow \tilde{s}'$ или $s \Rightarrow \tilde{s}$, соответственно.

Отметим, что данное определение наблюдаемой эквивалентности не является единственно возможным, и м.б. модифицировано в зависимости от решаемой задачи. В некоторых задачах более подходящим определением наблюдаемой эквивалентности является такое огрубление опреде-

ленного выше отношения, относительного которого будут эквивалентными РП $\mathcal{P} = \{P\}$, $\mathcal{P}' = \{P'\}$, где

$$P = \odot \xrightarrow{!k(e)} \otimes,\ P' = \odot \xrightarrow{!k'(e')} \otimes,\ \ k \in \bar{X}(P)_{\mathbf{K}},\ k' \in \bar{X}(P')_{\mathbf{K}}.$$

2.4. Теоремы о сохранении значений формул при переходах распределенных процессов

В этом параграфе формулируются и доказываются теоремы о сохранении значений некоторых формул при переходах РП. Данные теоремы используются при решении задач верификации РП. В излагаемых ниже примерах применения этих теорем

- ПП P, упоминаемый в этих теоремах – это ПП противника $P_\dagger$, и
- говоря неформально, данные теоремы утверждают что
  - если имена каких-либо каналов защищены от противника, то содержимое этих каналов не м.б. изменено противником, и
  - если какие-либо криптографические ключи защищены от противника, то содержимое ШС, зашифрованных на этих ключах, недоступно противнику.

2.4.1. Теоремы о защищенных каналах

Первая теорема связана с сохранением значений формул вида

$$E \perp_{\mathbf{C}} P, \quad \text{где } E \subseteq Tm, \text{ и } P - \text{ПП} \tag{10}$$

при переходах между состояниями. Данная теорема м.б. интерпретирована как следующее утверждение: если

- какому-либо из ПП, входящих в некоторый РП, недоступны сообщения, выражаемые термами из некоторого множества E, и
- в текущем состоянии этого РП ни в каком из каналов, доступных этому ПП, нет сообщений, соответствующих термам из E,

то никакая собственная активность этого ПП не приведет к тому, что сообщения, выражаемых термами из E, станут доступны этому ПП.

Каналы, имена которых входят в E, могут интерпретироваться как **защищённые** каналы относительно действий ПП P.

Теорема 1.
Пусть РП $\mathcal{P}$, ПП $P \in \mathcal{P}$, и состояния $s, s' \in \Sigma_{\mathcal{P}}$, таковы, что $s \xrightarrow{\alpha_P} s'$.

Тогда $\forall\, E \subseteq \langle \mathcal{P} \rangle_0$ верна импликация

$$s \models E \perp_{\mathbf{C}} P \;\; \Rightarrow \;\; s' \models E \perp_{\mathbf{C}} P.$$

Доказательство.
Напомним, что $s \models E \perp_{\mathbf{C}} P$ означает, что $\forall\, e \in E \;\; Agent(P) \notin e$, и

$$\left.\begin{array}{l} \forall\, x \in E_{\mathbf{X}}, \forall\, y \in [P]_s \;\; x \notin y^s \\ \forall\, x \in E_{\mathbf{X}}, \forall\, c \in Channels \;\; \text{если } \exists\, e \in [c]_s : x \in e, \text{ то } c \in E \end{array}\right\} \qquad (11)$$

Докажем, что из (11) следует $s' \models E \perp_{\mathbf{C}} P$, т.е.

$$\left.\begin{array}{l} \forall\, x \in E_{\mathbf{X}}, \forall\, y \in [P]_{s'} \;\; x \notin y^{s'} \\ \forall\, x \in E_{\mathbf{X}}, \forall\, c \in Channels \;\; \text{если } \exists\, e \in [c]_{s'} : x \in e, \text{ то } c \in E \end{array}\right\} \qquad (12)$$

1) Пусть неверно первое утверждение в (12). Тогда из первого утверждения в (11) следует, что $[P]_s \neq [P]_{s'}$, и имеет место один из двух перечисленных ниже случаев.

   - Первый случай:

     $$\left.\begin{array}{l} \alpha = c?e, \text{ где } c \in \langle P \rangle_s, e^{s'} \in [c^s]_s \\ [P]_{s'} = [P]_s \cup Var(e), \exists\, x \in E_{\mathbf{X}}, \exists\, y \in Var(e) : x \in y^{s'}. \end{array}\right\} \qquad (13)$$

     Из соотношений $x \in y^{s'} \subseteq e^{s'} \in [c^s]_s$ и второго утверждения в (11) следует, что $c^s \in E$.

     Если $c^s \notin E_{\mathbf{X}}$, то c имеет вид $shared\_channel(\ldots)$, и в этом случае из определения понятия выполнения ПП в пункте 2.2.5 следует, что $Agent(P) \in c^s$. Однако это и утверждение $c^s \in E$ противоречат предположению $\forall\, e \in E \;\; Agent(P) \notin e$.

     Таким образом, $c^s \in E_{\mathbf{X}}$. Отсюда следует, что $c \in [P]_s$.

     Согласно первому утверждению в (11) (в котором в качестве x и y берем c^s и c соответственно), должно быть верно утверждение $c^s \notin c^s$, которое является ложным.

   - Второй случай:

     $$\left.\begin{array}{l} \alpha \text{ имеет вид } e := e', \text{ где } e' \in \langle P \rangle_s, e^{s'} = (e')^s, \\ [P]_{s'} = [P]_s \cup Var(e), \exists\, x \in E_{\mathbf{X}}, \exists\, y \in Var(e) : x \in y^{s'} \end{array}\right\} \qquad (14)$$

     Из соотношений $x \in y^{s'} \subseteq e^{s'} = (e')^s \in \langle P \rangle^s$ следует, что

     $$\exists\, z \in [P]_s : x \in z^s. \qquad (15)$$

     Однако (15) противоречит первому утверждению в (11).

2) Пусть неверно второе утверждение в (12), т.е.

$$\exists\, x \in E_{\mathbf{X}}, \exists\, c' \in Channels, \exists\, e' \in [c']_{s'} : x \in e',\ \text{но}\ c' \notin E.$$

Тогда из второго утверждения в (11) следует, что $[c']_{s'} \neq [c']_s$, и

$$\alpha \text{ имеет вид } c!e, \text{ где } c, e \in \langle P \rangle_s, x \in e' = e^s.$$

Из $x \in e^s$ следует, что $\exists\, y \in [P]_s : x \in y^s$, что противоречит первому утверждению в (11). ■

Следующая теорема является усилением теоремы 1. В ней утверждается, что в условиях теоремы 1 нижняя и верхняя границы на содержимое защищенных каналов не изменяются при выполнении действий P.

Теорема 2.
Пусть РП $\mathcal{P}$, ПП $P \in \mathcal{P}$, и состояния $s, s' \in \Sigma_{\mathcal{P}}$, таковы, что $s \xrightarrow{\alpha_P} s'$. Тогда $\forall\, E \subseteq \langle \mathcal{P} \rangle_0$, $\forall\, E', E'' \subseteq Tm$, $\forall\, c \in E_{\mathbf{C}}$ верна импликация

$$s \models \varphi \Rightarrow s' \models \varphi, \quad \text{где } \varphi = \{E \perp_{\mathbf{C}} P, E' \subseteq [c] \subseteq E''\}.$$

Доказательство.
Согласно теореме 1, из $s \models E \perp_{\mathbf{C}} P$ следует $s' \models E \perp_{\mathbf{C}} P$. Кроме того, $[c]_s \subseteq [c]_{s'}$. Таким образом, для доказательства теоремы достаточно доказать импликацию

$$s \models \varphi \quad \Rightarrow \quad s' \models [c] \subseteq E''. \tag{16}$$

Если заключение импликации (16) неверно, то $[c]_s \neq [c]_{s'}$.

Т.к. по предположению $c \in E_{\mathbf{C}} \subseteq \langle \mathcal{P} \rangle_0$, то неравенство $[c]_s \neq [c]_{s'}$ возможно только если α имеет вид $c'!e$, где $(c')^s = c$.

Если c не является переменной, то c является разделяемым каналом, и по определению выполнения действия вида $c'!e$, в этом случае должно быть выполнено условие $Agent(P) \in c$, что противоречит предположению $s \models E \perp_{\mathbf{C}} P$ (т.к. в частности должно быть верно, что $Agent(P)$ не входит в термы из E). Следовательно, $c \in Var$, поэтому $c' \in Var$, и $c' \in [P]_s$.

Т.к. $c \in E_{\mathbf{X}}$ и $c' \in [P]_s$, то согласно предположению $s \models E \perp_{\mathbf{C}} P$, должно быть верно утверждение $c \notin c$, которое является ложным. ■

2.4.2. Теоремы о защищённых ключах

В этом пункте доказываются теоремы, аналогичные теоремам 1 и 2. В них вместо защищенных каналов рассматриваются защищенные ключи.

Первая теорема связана с сохранением значений формул вида

$$E \perp_{\mathbf{K}} P, \quad \text{где } E \subseteq Tm, \text{ и } P - \text{ПП} \tag{17}$$

при переходах между состояниями. Данная теорема м.б. интерпретирована как следующее утверждение: если

- какому-либо из ПП, входящих в некоторый РП, недоступны сообщения, выражаемые термами из некоторого множества E, и
- в текущем состоянии этого РП выполнено условие, выражаемое формулой (17),

то никакая собственная активность этого ПП не приведет к тому, что ключи, имена которых входят в E, когда-нибудь станут доступны этому ПП. Данные ключи могут интерпретироваться как **защищённые** ключи относительно действий ПП P.

Теорема 3.
Пусть РП $\mathcal{P}$, ПП $P \in \mathcal{P}$, и состояния $s, s' \in \Sigma_{\mathcal{P}}$, таковы, что $s \xrightarrow{\alpha_P} s'$. Тогда $\forall\, E \subseteq \langle \mathcal{P} \rangle_0$ верна импликация

$$s \models E \perp_{\mathbf{K}} P \Rightarrow s' \models E \perp_{\mathbf{K}} P.$$

Доказательство.
Напомним, что $s \models E \perp_{\mathbf{K}} P$ означает, что $\forall\, e \in E \;\; Agent(P) \notin e$, и

$$\left.\begin{array}{l} \forall\, x \in E_{\mathbf{X}},\, \forall\, y \in [P]_s \;\; x \perp_{\mathbf{K},E} y^s \\ \forall\, x \in E_{\mathbf{X}},\, \forall\, c \in Channels, \forall\, e \in [c]_s \;\; x \perp_{\mathbf{K},E} e \end{array}\right\} \tag{18}$$

Докажем, что из (18) следует $s' \models E \perp_{\mathbf{K}} e$, т.е. $\forall\, x \in E_{\mathbf{X}}$

$$\left.\begin{array}{l} \forall\, y \in [P]_{s'} \;\; s' \models x \perp_{\mathbf{K},E} y^{s'} \\ \forall\, c \in Channels, \forall\, e \in [c]_{s'} \;\; s' \models x \perp_{\mathbf{K},E} e \end{array}\right\} \tag{19}$$

1) Если первое утверждение в (19) неверно, то из первого утверждения в (18) следует, что $[P]_s \neq [P]_{s'}$, и имеет место один из двух случаев:

   - Первый случай: верно утверждение

$$\left.\begin{array}{l} \alpha \text{ имеет вид } c?e, c \in \langle P \rangle_s, e^{s'} \in [c^s]_s, \\ [P]_{s'} = [P]_s \cup Var(e), \exists\, y \in Var(e) : \\ \exists \text{ вхождение } x \text{ в } y^{s'}, \text{ которое не содержится} \\ \text{ни в каком подтерме вида } k(\ldots) \subseteq y^{s'}, \text{ где } k \in E_{\mathbf{K}}. \end{array}\right\} \tag{20}$$

Поскольку упомянутое в (20) вхождение x содержится в терме $y^{s'} \subseteq e^{s'} \in [c^s]_s$, то из второго утверждения в (18) следует, что это вхождение x содержится в подтерме $k(\tilde{e}) \subseteq e^{s'}$, где $k \in E_{\mathbf{K}}$. Из (20) следует, что $k(\tilde{e})$ не м.б. подтермом терма $y^{s'}$. Поскольку термы $k(\tilde{e})$ и $y^{s'}$ имеют непустое пересечение (оба содержат вышеупомянутое вхождение x), то из (1) следует, что $y^{s'} \subset k(\tilde{e})$. Таким образом,

$$y^{s'} \subset k(\tilde{e}) \subseteq e^{s'}. \tag{21}$$

Докажем индукцией по структуре терма e, что из (21) следует утверждение

$$\exists\, z \in Var(e) : \;\; k(\tilde{e}) \subseteq z^{s'} \subseteq e^{s'}. \tag{22}$$

Если $e \in Con \cup Var$, то утверждение (22) верно.

Если e имеет вид $f(e_1, \ldots, e_n)$, где $f \in Fun$, то

- если $f = encrypt$, т.е. e имеет вид $k_1(e_1)$, то $k_1 \in Keys(e)$, согласно (7)(b) имеется включение $Keys(e) \subseteq [P]_s$, откуда следует, что $k_1 \in [P]_s$, и возможны следующие случаи:
  - $k(\tilde{e}) = e^{s'} = k_1^{s'}(e_1^{s'})$, в этом случае $k = k_1^{s'} = k_1^s \in [P]^s$, однако поскольку $k \in E_{\mathbf{K}}$, то по первому утверждению в (18) вхождение k в k должно содержаться в подтерме вида $k'(\ldots) \subseteq k$, что невозможно,
  - $k(\tilde{e}) \subseteq k_1^{s'}$, данный случай невозможен по определению термов типа $\mathbf{K}$,
  - $k(\tilde{e}) \subseteq e_1^{s'}$, в данном случае утверждение (22) следует из индуктивного предположения,
- если $f \neq encrypt$, то $\exists\, i \in \{1, \ldots, n\} : k(\tilde{e}) \subseteq e_i^{s'}$, и утверждение (22) следует из индуктивного предположения.

Из (21) и (22) следует, что

$$y^{s'} \subset k(\tilde{e}) \subseteq z^{s'} \subseteq e^{s'}. \tag{23}$$

Таким образом, терм e содержит вхождения переменных y и z, обладающие следующим свойством: $y^{s'} \subset z^{s'}$, откуда для данных вхождений следует включение $y \subset z$, что невозможно.

- Второй случай: верно утверждение

$$\left.\begin{array}{l} \alpha \text{ имеет вид } e := e', \text{ где } e' \in \langle P \rangle_s, e^{s'} = (e')^s, \\ [P]_{s'} = [P]_s \cup Var(e), \exists\, y \in Var(e) : \\ \exists \text{ вхождение } x \text{ в } y^{s'}, \text{ которое не содержится} \\ \text{ни в каком подтерме вида } k(\ldots) \subseteq y^{s'}, \text{ где } k \in E_{\mathbf{K}}. \end{array}\right\} \tag{24}$$

Поскольку упомянутое в (24) вхождение x содержится в терме $y^{s'} \subseteq e^{s'} = (e')^s$, то это вхождение x содержится в подтерме $(z')^s \subseteq (e')^s$, где $z' \in Var(e')$.

По предположению, $e' \in \langle P \rangle_s$, поэтому $Var(e') \subseteq [P]_s$, и, следовательно, $z' \in [P]_s$. Из первого утверждения в (18) следует, что упомянутое в (24) вхождение x в $(z')^s$ содержится в некотором подтерме $k(\tilde{e}) \subseteq (z')^s$, где $k \in E_{\mathbf{K}}$.

Из (24) следует, что $k(\tilde{e})$ не м.б. подтермом терма $y^{s'}$.

Поскольку термы $k(\tilde{e})$ и $y^{s'}$ имеют непустое пересечение (оба содержат упомянутое в (24) вхождение x), то из (1) следует, что $y^{s'} \subset k(\tilde{e})$.

Из равенства $e' = e^\theta$ следует, что $\exists\, z \in Var(e)$: вышеупомянутое вхождение z' в e' входит в подтерм $z^\theta \subseteq e^\theta = e'$. Следовательно, $(z')^s \subseteq (z^\theta)^s = z^{s'} \subseteq e^{s'}$.

Таким образом, получаем:

$$y^{s'} \subset k(\tilde{e}) \subseteq (z')^s \subseteq z^{s'} \subseteq e^{s'}. \tag{25}$$

Как и в предыдущем пункте, на основании (25) заключаем, что терм e содержит вхождения переменных y и z, обладающие следующим свойством: $y^{s'} \subset z^{s'}$, откуда для данных вхождений следует включение $y \subset z$, что невозможно.

2) Если второе утверждение в (19) неверно, то из второго утверждения в (18) следует, что

$$\left.\begin{array}{l} \alpha \text{ имеет вид } c!e, \text{ где } e \in \langle P \rangle_s, \\ \exists \text{ вхождение } x \text{ в } e^s, \text{ которое не содержится} \\ \text{ни в каком подтерме вида } k(\ldots) \subseteq e^s, \text{ где } k \in E_{\mathbf{K}}. \end{array}\right\} \tag{26}$$

Поскольку $e \in \langle P \rangle_s$, то упомянутое в (26) вхождение x в e^s содержится в подтерме вида y^s терма e^s, где y – некоторая переменная из $[P]_s$. Согласно первому утверждению в (18), это вхождение x в y^s содержится в подтерме вида $k(\ldots) \subseteq y^s \subseteq e^s$, где $k \in E_{\mathbf{K}}$. Но это противоречит (26). ■

Теорема 4.

Пусть заданы РП $\mathcal{P}$ такой, что $Var(\mathcal{P})_{\mathbf{C}} = \{\circ\}$, ПП $P \in \mathcal{P}$, состояния $s, s' \in \Sigma_{\mathcal{P}}$, такие, что $s \xrightarrow{\alpha_P} s'$, и подмножество $E \subseteq \langle \mathcal{P} \rangle_0$.

Тогда $\forall\, k \in E_{\mathbf{K}} : k \neq public\_key(\ldots)$ верна импликация

$$s \models \varphi \Rightarrow s' \models \varphi, \quad \text{где } \varphi = \{E \perp_{\mathbf{K}} P, k^{-1}[P] \subseteq k^{-1}[\circ]\}.$$

Доказательство.

По теореме 3, из $s \models E \perp_{\mathbf{K}} P$ следует $s' \models E \perp_{\mathbf{K}} P$. Таким образом, для доказательства теоремы 4 достаточно доказать импликацию

$$s \models \varphi \;\; \Rightarrow \;\; s' \models k^{-1}[P] \subseteq k^{-1}[\circ]. \tag{27}$$

Если (27) неверно, то α – не посылка, $[P]_{s'}$ имеет вид $[P]_s \cup Var(e')$, и для некоторого терма $e \in k^{-1}[P]^{s'}$ выполняется свойство

$$e \notin k^{-1}[\circ]_s, \text{ т.е. } \exists\, x \in [P]_{s'} : k(e) \subseteq x^{s'}, \forall\, \dot{e} \in [\circ]_s \;\; k(e) \not\subseteq \dot{e}. \tag{28}$$

Из предположения $s \models \varphi$ и из (28) следует, что $x \in Var(e')$, откуда получаем: $k(e) \subseteq x^{s'} \subseteq (e')^{s'}$.

Рассмотрим по отдельности каждый из двух возможных видов α.

1) $\alpha = ?e'$, в этом случае $k(e) \subseteq (e')^{s'} \in [\circ]_s$. Полагая в (28) терм $\dot{e}$ равным $(e')^{s'}$, получаем противоречие.

2) $\alpha = (e' := e'')$, в этом случае $e'' \in \langle P \rangle_s$, $(e')^{s'} = (e'')^s$, поэтому

$$k(e) \subseteq (e'')^s. \tag{29}$$

Докажем, что $k \notin e''$ и $k \in E_{\mathbf{X}}$.

Предположим, что $k \in e''$. По условию теоремы, $k \neq public\_key(\ldots)$. Если $k = shared\_key(\ldots)$, то, по определению выполнения ПП в пункте 2.2.5, должно быть верно соотношение $Agent(P) \in k \in E_{\mathbf{K}}$, которое противоречит первому условию свойства $s \models E \perp_{\mathbf{K}} P$. Напомним, что данное свойство имеет вид: $\forall\, \tilde{e} \in E \;\; Agent(P) \notin \tilde{e}$, и

$$\left.\begin{array}{l} \forall\, x \in E_{\mathbf{X}}, \forall\, y \in [P]_s \;\; x \perp_{\mathbf{K},E} y^s \\ \forall\, x \in E_{\mathbf{X}}, \forall\, \tilde{e} \in [\circ]_s \;\; x \perp_{\mathbf{K},E} \tilde{e} \end{array}\right\} \tag{30}$$

Следовательно, $k \in E_{\mathbf{X}}$, поэтому из $k \in e'' \in \langle P \rangle_s$ следует $k \in [P]_s$. Однако, полагая в первом соотношении в (30) x и y равными k, получаем утверждение $k \perp_{\mathbf{K},E} k$, которое является ложным, согласно определению (6).

Аналогично доказательству импликации (21) $\Rightarrow$ (22) в теореме 3, можно доказать, что из свойств $k \notin e''$ и (29) следует, что

$$\exists\, y \in Var(e'') \subseteq [P]_s : k(e) \subseteq y^s. \tag{31}$$

По предположению, $s \models k^{-1}[P] \subseteq k^{-1}[\circ]$, т.е. $k^{-1}[P]^s \subseteq k^{-1}[\circ]_s$. Из (31) следует, что $e \in k^{-1}[P]^s$. Следовательно, $e \in k^{-1}[\circ]_s$, что противоречит предположению (28). ■

Следующая теорема является усилением теоремы 4. В ней утверждается, что в условиях теоремы 4 нижняя и верхняя границы на множество ШС, содержащихся в открытом канале, и зашифрованных на защищённых ключах, не изменяются при выполнении действий ПП P.

Теорема 5.

Пусть заданы РП $\mathcal{P}$ такой, что $Var(\mathcal{P})_{\mathbf{C}} = \{\circ\}$, ПП $P \in \mathcal{P}$, состояния $s, s' \in \Sigma_{\mathcal{P}}$, такие, что $s \xrightarrow{\alpha_P} s'$, и подмножества $E \subseteq \langle\mathcal{P}\rangle_0$, $E', E'' \subseteq Tm$.

Тогда $\forall\, k \in E_{\mathbf{K}} : k \neq public\_key(\ldots)$ верна импликация

$$s \models \varphi \Rightarrow s' \models \varphi, \text{ где } \varphi = \left\{ \begin{array}{l} E \perp_{\mathbf{K}} P, k^{-1}[P] \subseteq k^{-1}[\circ] \\ E' \subseteq k^{-1}[\circ] \subseteq E'' \end{array} \right\}. \qquad (32)$$

Доказательство.

По теореме 4, для доказательства (32) достаточно доказать, что

$$s \models \varphi \;\; \Rightarrow \;\; s' \models \{E' \subseteq k^{-1}[\circ], k^{-1}[\circ] \subseteq E''\}. \qquad (33)$$

- Утверждение $s' \models E' \subseteq k^{-1}[\circ]$ следует из того, что $[\circ]_s \subseteq [\circ]_{s'}$.
- Докажем утверждение $s' \models k^{-1}[\circ] \subseteq E''$. Если бы оно было неверным, то было бы верно неравенство $[\circ]_s \neq [\circ]_{s'}$. Это возможно только если α имеет вид $!e$, где $e \in \langle P\rangle_s$, и

  $$[\circ]_{s'} = [\circ]_s \cup \{e^s\}, \text{ причем } \exists\, e' \notin (E'')^s \supseteq k^{-1}[\circ]_s : k(e') \subseteq e^s. \qquad (34)$$

  Так же, как и в теореме 4, доказываем, что $k \in E_{\mathbf{X}}$, и что из $e \in \langle P\rangle_s$ следует соотношение $k \notin e$.

  Аналогично доказательству импликации (21) $\Rightarrow$ (22) в теореме 3, можно доказать, что из $k \notin e$ и $k(e') \subseteq e^s$ следует, что

  $$\exists\, x \in Var(e) \subseteq [P]_s : \; k(e') \subseteq x^s,$$

  поэтому $e' \in k^{-1}[P]^s$. Отсюда, используя предположение $s \models \varphi$, следствием которого является включение $k^{-1}[P]^s \subseteq k^{-1}[\circ]_s$, получаем: $e' \in k^{-1}[\circ]_s$, что противоречит (34). ■

2.5. Теорема для доказательства свойства соответствия

Теорема, излагаемая в этом параграфе, может использоваться для доказательства **свойства соответствия** протоколов аутентификации, которое имеет следующий смысл: если один из участников протокола аутентификации после выполнения этого протокола пришел к выводу, что другой участник этого протокола является подлинным (т.е. объявленные им свое имя и параметры совпадают с его реальными именем и параметрами), то это действительно так. Доказываемая ниже теорема применяется для обоснования того, что если

- РП $\mathcal{P}$ использует для взаимодействия только открытый канал $\circ$, и
- в некотором состоянии $s \in \Sigma_{\mathcal{P}}$ в этом канале содержится сообщение, содержащее подтерм вида $k(e)$, где ключ k недоступен в этом состоянии для некоторого ПП P, входящего в $\mathcal{P}$,

то в некотором состоянии $s' <_\pi s$ другой ПП $P' \neq P$ из $\mathcal{P}$ послал в открытый канал $\circ$ сообщение, содержащее тот же самый подтерм $k(e)$.

В параграфах 3.2 и 3.3 мы рассматриваем примеры применения данной теоремы для верификации КП Yahalom и КП передачи ШС между несколькими агентами.

Теорема 6.

Пусть заданы РП $\mathcal{P}$, такой, что $Var(\mathcal{P})_{\mathbf{C}} = \{\circ\}$, ПП $P \in \mathcal{P}$, множество $E \subseteq \langle\mathcal{P}\rangle_0$, не содержащее открытых ключей, и состояние $s \in \Sigma_{\mathcal{P}}$, причем

- $s \models E \perp_{\mathbf{K}} P$, и
- $[\circ]_s$ содержит терм с подтермом $k(e)$, где $k \in E_{\mathbf{K}}$.

Тогда для каждого пути π из начального состояния 0 РП $\mathcal{P}$ в состояние s существует ПП $P' \in \mathcal{P} \setminus \{P\}$, такой, что π содержит ребро вида

$$\dot{s} \xrightarrow{(!\dot{e})_{P'}} s', \quad \text{где } k(e) \subseteq \dot{e}^{\dot{s}}. \tag{35}$$

Доказательство.

Обозначим записью s' первое состояние на пути π, такое, что $[\circ]_{s'}$ содержит терм e' с подтермом $k(e)$. Т.к. $[\circ]_0 = \emptyset$, то $s' \neq 0$.

Пусть ребро на пути π с концом в s' имеет вид $\dot{s} \overset{\alpha_{P'}}{\rightarrow} s'$. Т.к. $e' \notin [\circ]_{\dot{s}}$, то $\alpha = !\dot{e}$, где $\dot{e}^{\dot{s}} = e'$. Если $P' \neq P$, то теорема доказана.

Докажем, что другой возможный случай ($P' = P$) невозможен.

Предположим, что $P' = P$, т.е. $\dot{s} \xrightarrow{(!\dot{e})_P} s'$.

Докажем, что $k \in E_{\mathbf{X}}$.

Если это неверно, т.е. k – разделяемый ключ, то, согласно определению выполнения ПП в пункте 2.2.5, должно быть $Agent(P) \in k \in E_{\mathbf{K}}$, что противоречит предположению $\forall\, \tilde{e} \in E \;\; Agent(P) \notin \tilde{e}$.

Из $s \models E \perp_{\mathbf{K}} P$ следует, что $\dot{s} \models E \perp_{\mathbf{K}} P$, откуда получаем $k \notin [P]_{\dot{s}}$, т.к. если $k \in [P]_{\dot{s}}$, то, согласно (5), вхождение k в $k^{\dot{s}} = k$ должно содержаться в подтерме вида $k'(\ldots)$, что невозможно.

Из $k \notin [P]_{\dot{s}}$ и из условия $\dot{e} \in \langle P\rangle_{\dot{s}}$, которое верно согласно (7)(a), следует, что $k \notin \dot{e}$.

Аналогично доказательству импликации (21) $\Rightarrow$ (22) в теореме 3, можно доказать, что из свойств $k(e) \subseteq e' = \dot{e}^{\dot{s}}$ и $k \notin \dot{e}$ следует, что

$$\exists\, x \in Var(\dot{e}) \subseteq [P]_{\dot{s}} : k(e) \subseteq x^{\dot{s}} \in [P]^{\dot{s}}. \tag{36}$$

Обозначим записью s'' первое состояние на пути π, такое, что $[P]^{s''}$ содержит терм с подтермом $k(e)$, т.е.

$$\exists\, x \in [P]_{s''} : k(e) \subseteq x^{s''}. \tag{37}$$

Из (36) следует, что s'' находится на пути π левее s'. Нетрудно видеть, что $s'' \neq 0$, поэтому на пути π существует ребро вида $\ddot{s} \xrightarrow{\alpha_{P''}} s''$. Из выбора s'' следует, что $x \notin [P]_{\ddot{s}}$, поэтому $P'' = P$, и возможны два случая:

1) $\alpha = ?\ddot{e}, x \in Var(\ddot{e}), \ddot{e}^{s''} \in [\circ]_{\ddot{s}}$,

   т.к. $k(e) \subseteq x^{s''} \subseteq \ddot{e}^{s''} \in [\circ]_{\ddot{s}}$, то получаем противоречие с выбором s' как самого первого состояния на пути π, такого, что $[\circ]_{s'}$ содержит терм e' с подтермом $k(e)$: состояние $\ddot{s}$ имеет аналогичное свойство, и находится левее s',

2) $\alpha = (\ddot{e} := \tilde{e}), x \in Var(\ddot{e}), \tilde{e} \in \langle P \rangle_{\ddot{s}}, \ddot{e}^{s''} = \tilde{e}^{\ddot{s}}$,

   поскольку

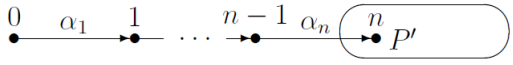
   - $k(e) \subseteq x^{s''} \subseteq \ddot{e}^{s''} = \tilde{e}^{\ddot{s}}$ и
   - $\tilde{e}$ не содержит k, т.к. выше было установлено, что $k \notin [P]_{\dot{s}}$, поэтому, учитывая свойство $\ddot{s} \leq \dot{s}$, из которого следует включение $[P]_{\ddot{s}} \subseteq [P]_{\dot{s}}$, получаем: $k \notin [P]_{\ddot{s}}$, и, следовательно, терм $\tilde{e} \in \langle P \rangle_{\ddot{s}}$ тоже не содержит k,

   то, аналогично доказательству импликации (21) $\Rightarrow$ (22) в теореме 3, можно доказать, что $\exists\, y \in [P]_{\ddot{s}} : k(e) \subseteq y^{\ddot{s}}$, что противоречит выбору s'' как самого первого состояния на пути π со свойством (37): $\ddot{s}$ имеет аналогичное свойство, и находится левее s''. ■

2.6. Схемы распределенных процессов

2.6.1. Префиксные последовательные процессы

Будем говорить, что ПП P является **префиксным**, если он имеет вид

$P =$ 

(38)

т.е. в P имеется совокупность вершин, занумерованных натуральными числами $0, 1, \ldots, n$ $(n \geq 1)$, причем $Init(P) = 0$, $\forall\, i = 0, \ldots, n-1$ из

вершины i выходит ровно одно ребро с концом $i+1$ и меткой α_i. Запись P' в (38) обозначает подграф графа P, состоящий из вершин и ребер графа P, за исключением вершин $0, \ldots, n-1$ и связанных с ними ребер.

Подграфы $0 \overset{\alpha_1}{\to} 1 \overset{\alpha_2}{\to} \ldots \overset{\alpha_n}{\to} n$ и P' графа (38) будем называть **префиксом** и **постфиксом** ПП P соответственно, и обозначаются записями $Pref(P)$ и $Post(P)$ соответственно. Последняя вершина в $Pref(P)$ называется **конечной** вершиной этого префикса. Если $Post(P)$ состоит из одной вершины, то он обозначается символом $\mathbf{0}$.

Если ПП P имеет вид (38), то будем обозначать этот факт записью

$$P = \alpha_1; \ldots; \alpha_n; P'. \tag{39}$$

2.6.2. Понятие схемы распределенного процесса

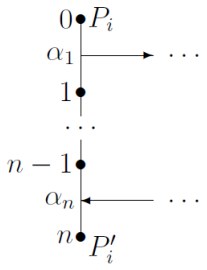
Пусть задан РП $\mathcal{P}$, и $\forall\, P \in \mathcal{P}$ ПП P – префиксный, причем для каждой посылки (каждого приема) в $Pref(P)$

- предполагаемым получателем (отправителем) того сообщения, которое посылается (принимается) при выполнении этого действия, является определенный ПП P' из $\mathcal{P}$, и
- действие ПП P', соответствующее приему (посылке) этого сообщения, находится в $Pref(P')$.

Эти зависимости между действиями можно выразить в виде **схемы** РП $\mathcal{P}$, которая имеет следующий вид:

- каждый ПП $P \in \mathcal{P}$ представляется в этой схеме **нитью**, т.е. вертикальной линией, на которой выделены точки, соответствующие вершинам из $Pref(P)$, верхняя точка соответствует $Init(P)$, причем
  - у каждой точки указан номер соответствующей вершины,
  - рядом с верхней точкой указывается имя ПП P,
  - если $Post(P) = P' \neq \mathbf{0}$, то у нижней точки указано P',
  - рядом с каждым отрезком l, соединяющим соседние точки нити, указана метка α_l ребра из $Pref(P_i)$, соответствующего l,
- для каждого отрезка l, соединяющего соседние точки нити, если α_l – посылка сообщения, то в схеме присутствует стрелка, начало которой лежит на l, а конец – на отрезке l', таком, что $\alpha_{l'}$ – действие соответствующего ПП $P_{i'}$ по приему этого сообщения.

Например если $P_i = \alpha_1;\ \ldots\ \alpha_n; P_i'$, где α_1 – посылка, а α_n – прием, то процессу P_i соответствует нить



(40)

Отметим, что стрелки от посылок к приемам сообщений изображают лишь желаемую связь между этими действиями, но отнюдь не реальную связь: возможно что посланное сообщение будет принято из канала совсем не тем процессом, для которого оно было предназначено.

В целях большей наглядности будем использовать следующее соглашение в обозначениях переменных.

- Будем указывать горизонтальную черту над любым обозначением какой-либо переменной x, если она рассматривается как элемент множества вида $\bar{X}(P)$ (т.е. эта переменная обозначается $\bar{x}$).
- Если P – ПП вида (39), и переменная $x \in \hat{X}(P)$ входит в метку α_i i–го отрезка нити ПП P, причем i – первая метка в (39), в которую входит x (т.е. $\forall\, i' = 1, \ldots, i-1$ x не входит в $\alpha_{i'}$), то над этим вхождением x указывается уголок (т.е. это вхождение обозначается записью $\hat{x}$). Данные обозначения переменных используются также в записях вида (39).

2.6.3. Примеры схем распределенных процессов

1) Первый пример – это РП $\mathcal{P}_1 = \{A, B\}$, являющийся моделью передачи от A к B сообщения x по каналу c_{AB}, где только A и B знают имя этого канала, т.е. $c_{AB} = shared\_channel(A, B)$.

   Данный РП работает следующим образом:

   - A посылает B сообщение x по каналу c_{AB},
   - B принимает из канала c_{AB} сообщение и записывает его в переменную y, после чего ведет себя так же, как процесс P.

   ПП A и B имеют следующий вид: $A = c_{AB}!x; \mathbf{0}$, $B = c_{AB}?\hat{y}; P$.

Схема РП $\mathcal{P}_1$ имеет следующий вид:

(41)

2) Второй пример – это РП $\mathcal{P}_2 = \{A, B\}$, являющийся моделью передачи от A к B ШС $k_{AB}(x)$ по открытому каналу $\circ$. Предполагается, что A и B имеют общий секретный ключ k_{AB}, на котором они могут шифровать и дешифровать сообщения, используя симметричную систему шифрования, причем только A и B знают k_{AB}, т.е. $k_{AB} = shared\_key(A, B)$.

   Данный РП работает следующим образом:

   - A посылает B ШС $k_{AB}(x)$ по каналу $\circ$,
   - B принимает из канала $\circ$ сообщение $k_{AB}(x)$, дешифрует его, записывает извлеченное сообщение x в переменную y, после чего ведет себя так же, как процесс P.

   ПП A и B имеют следующий вид: $A = !k_{AB}(x); \mathbf{0}$, $B = ?k_{AB}(\hat{y}); P$.

   Схема РП $\mathcal{P}_2$ имеет следующий вид:

(42)

3) Третий пример – это РП $\mathcal{P}_3 = \{A, B, J\}$, являющийся моделью передачи от A к B одного сообщения x по скрытому каналу $\bar{c}$ при помощи **доверенного посредника** J, где A и J (B и J) взаимодействуют по скрытому каналу c_{AJ} (c_{BJ}), причем только A и J (B и J) знают имя c_{AJ} (c_{BJ}), т.е. $c_{AJ} = shared\_channel(A, J)$, $c_{BJ} = shared\_channel(B, J)$.

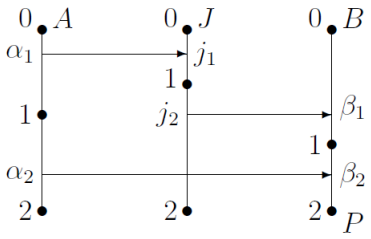
   Данный РП работает следующим образом:

   - A посылает J имя скрытого канала $\bar{c}$ (которое сначала знает только он один) по каналу c_{AJ},
   - J посылает B полученное имя канала $\bar{c}$ по каналу c_{BJ},
   - A посылает B сообщение x по каналу $\bar{c}$,
   - B принимает из канала $\bar{c}$ сообщение и записывает его в переменную y, после чего ведет себя так же, как процесс P.

ПП A, B и J определяются следующим образом:

$$\begin{array}{llll} A = \alpha_1; \alpha_2; \mathbf{0}, & \text{где} & \alpha_1 = c_{AJ}!\bar{c}, & \alpha_2 = \bar{c}!x, \\ J = j_1; j_2; \mathbf{0}, & \text{где} & j_1 = c_{AJ}?\hat{u}, & j_2 = c_{BJ}!u, \\ B = \beta_1; \beta_2; P, & \text{где} & \beta_1 = c_{BJ}?\hat{v}, & \beta_2 = v?\hat{y}. \end{array} \tag{43}$$

Схема РП $\mathcal{P}_3$ имеет следующий вид:



(44)

4) Четвертый пример – это РП $\mathcal{P}_4 = \{A, B, J\}$ (называемый **протоколом Wide-Mouth Frog (WMF)**), являющийся моделью передачи от A к B ШС $\bar{k}(x)$ по открытому каналу $\circ$ при помощи доверенного посредника J, с которым A и B взаимодействуют по открытому каналу. A создает секретный ключ $\bar{k}$, посылает B этот ключ в зашифрованном виде черед доверенного посредника J, и затем посылает B ШС $\bar{k}(x)$.

Предполагается, что A и J (B и J) имеют общий секретный ключ k_{AJ} (k_{BJ}), на котором они могут шифровать и дешифровать сообщения, используя симметричную систему шифрования, причем только A и J (B и J) знают k_{AJ} (k_{BJ}), т.е. $k_{AJ} = shared\_key(A, J)$, $k_{BJ} = shared\_key(B, J)$.

Данный РП работает следующим образом.

- A создает секретный ключ $\bar{k}$ (сначала только A знает этот ключ) и посылает доверенному посреднику J ШС $k_{AJ}(\bar{k})$ по каналу $\circ$, после чего A посылает B ШС $\bar{k}(x)$ по каналу $\circ$,
- J принимает сообщение от A, дешифрует его, затем шифрует извлеченный ключ $\bar{k}$ на ключе k_{BJ}, и посылает B ШС $k_{BJ}(\bar{k})$ по каналу $\circ$,
- B извлекает из полученного сообщения от J ключ $\bar{k}$, и затем использует этот ключ для извлечения из полученного сообщения от A сообщения x, записывает его в переменную y, после чего ведет себя так же, как процесс P.

ПП A, B и J определяются следующим образом:

$$\begin{array}{llll} A = \alpha_1; \alpha_2; \mathbf{0}, & \text{где} & \alpha_1 = !k_{AJ}(\bar{k}), & \alpha_2 = !\bar{k}(x), \\ J = j_1; j_2; \mathbf{0}, & \text{где} & j_1 = ?k_{AJ}(\hat{u}), & j_2 = !k_{BJ}(u), \\ B = \beta_1; \beta_2; P, & \text{где} & \beta_1 = ?k_{BJ}(\hat{v}), & \beta_2 = ?v(\hat{y}). \end{array} \tag{45}$$

Схема РП $\mathcal{P}_4$ имеет вид (44).

2.7. Графы переходов распределенных процессов

В этом параграфе рассматриваются РП, состоящие из конечного числа ПП. Для наглядного представления выполнения таких РП вводится понятие **графа переходов** РП. Выполнение РП м.б. представлено как обход вершин ГП, соответствующего этому РП.

Ниже в этом параграфе символ $\mathcal{P}$ обозначает РП, состоящий из конечного числа ПП, каждый из которых отличен от $P_\dagger$.

2.7.1. Понятие графа переходов распределенного процесса

Графом переходов (ГП) РП $\mathcal{P} = \{P_1, \ldots, P_n\}$ называется граф $G_\mathcal{P}$,

- каждая вершина которого представляет собой список

$$at = (at_1, \ldots, at_n), \quad \text{где } \forall\, i = 1, \ldots, n \;\; at_i \in P_i,$$

- каждое ребро которого имеет вид

$$(at_1, \ldots, at_n) \xrightarrow{\alpha_{P_i}} (at'_1, \ldots, at'_n) \quad (i \in \{1, \ldots, n\}), \tag{46}$$

где P_i содержит ребро $at_i \overset{\alpha}{\to} at'_i$, и $at_{i'} = at'_{i'}$ при $i' \neq i$.

Вершина $(Init(P_1), \ldots, Init(P_n))$ ГП $G_\mathcal{P}$ называется **начальной** вершиной ГП $G_\mathcal{P}$, и обозначается записью $Init(G_\mathcal{P})$.

Нетрудно доказать, что если графы всех ПП, входящих в $\mathcal{P}$, ациклични, то $G_\mathcal{P}$ тоже ацикличен.

Для каждого состояния $s \in \Sigma_\mathcal{P}$ компонента at_s состояния s может рассматриваться как вершина ГП $G_\mathcal{P}$.

Пусть задано некоторое выполнение РП $\mathcal{P}$. Если при этом выполнении порождается последовательность состояний $s_0, s_1, \ldots, s_n$, то из определения отношения перехода (9) следует, что этой последовательности соответствует путь в ГП $G_\mathcal{P}$, который можно рассматривать как наглядное представление выполнения РП $\mathcal{P}$:

$$Init(G_\mathcal{P}) = at_{s_0} \xrightarrow{(\alpha_1)_{P_{i_1}}} at_{s_1} \xrightarrow{(\alpha_2)_{P_{i_2}}} \ldots \xrightarrow{(\alpha_n)_{P_{i_n}}} at_{s_n}$$

Напомним, что $\mathcal{P}_\dagger = \{\mathcal{P}, P_\dagger\}$. ГП $G_{\mathcal{P}_\dagger}$ можно рассматривать как граф, получаемый добавлением к $G_\mathcal{P}$ рёбер $at \xrightarrow{\alpha_{P_\dagger}} at$, где $at \in G_\mathcal{P}$, $\alpha \in Act$.

Вершина $at \in G_\mathcal{P}$ называется **достижимой**, если $\exists\, s \in \Sigma_{\mathcal{P}_\dagger}\colon at = at_s$.

Ребро ГП $G_\mathcal{P}$ называется **реализуемым**, если оно лежит на пути, соответствующем какому-либо выполнению РП $\mathcal{P}_\dagger$.

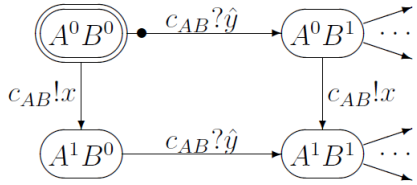
В изображении ГП $G_\mathcal{P}$ будут использоваться следующие соглашения:

- каждая вершина $at = (at_1, \ldots, at_n)$ ГП $G_\mathcal{P}$ обозначается овалом, внутри которого указан список $at_1 \ldots at_n$ компонентов at,
- начальная вершина $Init(G_\mathcal{P})$ изображается двойным овалом,
- черный кружочек на каком–либо ребре ГП $G_\mathcal{P}$ означает нереализуемость этого ребра (обоснование нереализуемости рёбер должно проводиться специальными рассуждениями),
- в целях сокращения обозначений, метка ребра $at \xrightarrow{\alpha_P} at'$ ГП $G_\mathcal{P}$ может обозначаться просто действием α в этой метке (без указания ПП P, выполняющего действие α при данном переходе).

2.7.2. Примеры графов переходов распределенных процессов

В этом пункте приводятся примеры ГП для РП, представленных схемами из пункта 2.6.3. Будем использовать следующее соглашение: если A – имя ПП, входящего в какой-либо из этих РП, и i – номер точки на нити, соответствующей этому ПП, то вершина графа A, соответствующая этой точке, обозначается записью A^i.

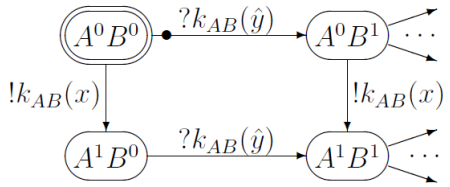
1) ГП для РП $\mathcal{P}_1$, описываемого схемой (41):



(47)

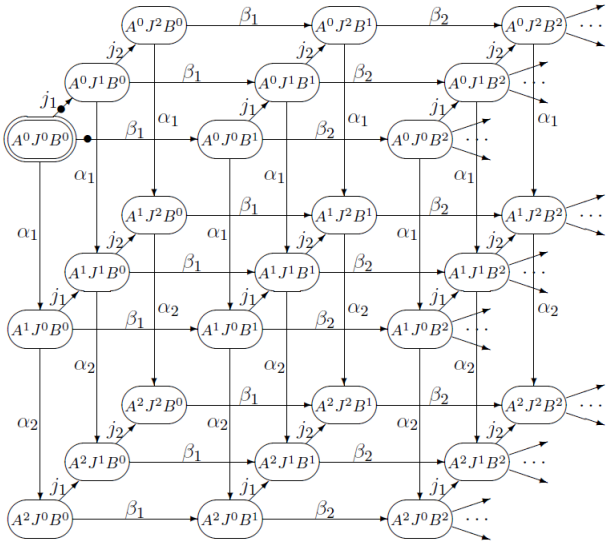
где наклонные стрелки обозначают ребра ГП, выходящие из соответствующих вершин, а также части ГП, достижимые после прохода по этим ребрам, которые не представлены в данном изображении ГП, это соглашение будет использоваться также и в нижеследующих примерах ГП.

2) ГП для РП $\mathcal{P}_2$, описываемого схемой (42):



(48)

3) ГП для РП $\mathcal{P}_3$ и $\mathcal{P}_4$, описываемых схемой (44):



(49)

2.7.3. Истинность формул в вершинах графов переходов распределенных процессов

Пусть задан РП $\mathcal{P}$. Для каждой вершины $at \in G_{\mathcal{P}}$ и каждой формулы $\varphi \in Fm$ запись $at \models \varphi$ обозначает утверждение

$$\varphi \text{ истинна в вершине } at$$

которое считается верным, если $\forall\, s \in \Sigma_{\mathcal{P}_\dagger}\ \ at_s = at \Rightarrow s \models \varphi$.

Нетрудно доказать, что ГП $G_{\mathcal{P}_i}$, где $\mathcal{P}_i$ $(i = 1, \ldots, 4)$ – РП, определенные в пункте 2.6.3, обладают следующими свойствами:

$$\left.\begin{array}{l} Init(G_{\mathcal{P}_1}) \models \{\varphi_1, [c_{AB}] = \emptyset\} \\ Init(G_{\mathcal{P}_2}) \models \{\varphi_2, k_{AB}^{-1}[\circ] = \emptyset\} \\ Init(G_{\mathcal{P}_3}) \models \{\varphi_3, [c_{AJ}] = \emptyset, [c_{BJ}] = \emptyset, [\bar{c}] = \emptyset\} \\ Init(G_{\mathcal{P}_4}) \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \emptyset, k_{BJ}^{-1}[\circ] = \emptyset, \bar{k}^{-1}[\circ] = \emptyset\} \end{array}\right\} \quad (50)$$

где

$$\varphi_1 = \{c_{AB}\} \perp_{\mathbf{C}} P_\dagger$$
$$\varphi_2 = \{\{k_{AB}\} \perp_{\mathbf{K}} P_\dagger, k_{AB}^{-1}[P_\dagger] \subseteq k_{AB}^{-1}[\circ]\}$$
$$\varphi_3 = \{c_{AJ}, c_{BJ}, \bar{c}\} \perp_{\mathbf{C}} P_\dagger$$
$$\varphi_4 = \left\{ \begin{array}{l} \{k_{AJ}, k_{BJ}, \bar{k}\} \perp_{\mathbf{K}} P_\dagger \\ k_{AJ}^{-1}[P_\dagger] \subseteq k_{AJ}^{-1}[\circ], k_{BJ}^{-1}[P_\dagger] \subseteq k_{BJ}^{-1}[\circ], \bar{k}^{-1}[P_\dagger] \subseteq \bar{k}^{-1}[\circ] \end{array} \right\}$$

Действительно, для любого РП $\mathcal{P}$ состояние $s \in \Sigma_{\mathcal{P}_\dagger}$ обладает свойством $at_s = Init(G_{\mathcal{P}})$, если $s = 0$, или существует путь из 0 в s с метками ребер вида $\alpha_{P_\dagger}$, и для каждой из формул $\psi_i = \{\varphi_i, \ldots\}$ в (50)

- истинность ψ_i в начальном состоянии РП $(\mathcal{P}_i)_\dagger$ следует из определений понятия начального состояния и ПП $P_\dagger$, и
- истинность ψ_i в состоянии s, в которое существует путь из 0 с метками ребер вида $\alpha_{P_\dagger}$, обосновывается утверждением

$$\forall\, s', s'' \in \Sigma_{\mathcal{P}_\dagger} : s' \xrightarrow{\alpha_{P_\dagger}} s'' \quad (s' \models \psi_i \;\; \Rightarrow \;\; s'' \models \psi_i) \tag{51}$$

  которое следует из теоремы 2 ($i = 1, 3$), или теоремы 5 ($i = 2, 4$).

Мы будем использовать соотношения (50) в излагаемом ниже решении задачи верификации некоторых свойств РП $\mathcal{P}_i$ $(i = 1, \ldots, 4)$.

3. Методы верификации криптографических протоколов

Излагаемые в этом параграфе методы верификации КП основаны на представлении КП в виде РП. Для доказательства свойств РП используются теоремы из предыдущего параграфа. Первый из излагаемых ниже методов использует понятие ГП, а второй основан на теореме 6 и наиболее подходит для верификации КП аутентификации.

В этом параграфе предполагается что символ $\mathcal{P}$ обозначает РП, не содержащий ПП противника $P_\dagger$.

3.1. Верификация на основе графов переходов

3.1.1. Описание метода

Некоторые свойства РП могут выражаться формулами, связаными с достижимыми вершинами соответствующих ГП. Например, одно из

свойств РП $\mathcal{P}_i$ $(i = 1, \ldots, 4)$, определенных в пункте 2.6.3, имеет такой вид:

$$\begin{array}{l}\text{если достижимая вершина } at = (at_A, at_B) \text{ или } (at_A, at_J, at_B) \\ \text{ГП } G_{\mathcal{P}_i} \text{ такова, что } at_B = 1 \ (i = 1, 2) \text{ или } at_B = 2 \ (i = 3, 4), \\ \text{то } at \models x = y.\end{array} \qquad (52)$$

Данное свойство называется **целостностью** и имеет следующий смысл:

- если выполнение РП $(\mathcal{P}_i)_\dagger = \{A, B, P_\dagger\}$ или $\{A, J, B, P_\dagger\}$ достигло состояния, где принимающий ПП B заканчивает часть своего выполнения, связанную с приемом сообщения от передающего ПП A,
- то при любом противодействии противника $P_\dagger$ передаваемое сообщение x в передающем ПП A совпадает с тем значением, которое будет иметь переменная y в принимающем ПП B.

Если свойство РП $\mathcal{P}$ имеет вид $at \models \varphi$, где at – некоторая достижимая вершина ГП $G_{\mathcal{P}}$, то один из методов верификации этого свойства заключается в следующем:

- для каждой достижимой вершины at', находящейся на каком-либо пути из $Init(G_{\mathcal{P}})$ в at, вычисляется формула $\varphi_{at'}$, истинная в at', и
- проверяется свойство $\varphi_{at} \leq \varphi$.

Необходимость вычисления вышеупомянутых формул для всех вершин на путях из $Init(G_{\mathcal{P}})$ в at связана с тем, что для вычисления φ_{at} необходимо знать формулы $\varphi_{at'}$ для каждой достижимой вершины at', из которой существует реализуемое ребро в at, и т.д.

Метод вычисления формулы φ_{at} имеет следующий вид:

- если вычислена формула $\varphi_{at'}$ для какой-либо достижимой вершины at', такой, что существует ребро ГП $G_{\mathcal{P}}$ вида $at' \overset{\alpha}{\to} at$, то вычисляется формула $\alpha(\varphi_{at'})$, смысл которой заключается в следующем: если в текущем состоянии s была верна формула $\varphi_{at'}$, и $s \overset{\alpha}{\to} s'$, то формула $\alpha(\varphi_{at'})$ верна в состоянии s', а также в каждом состоянии, в которое существует путь из s' с метками ребер вида $\alpha_{P_\dagger}$,
- искомая формула φ_{at} определяется как аналог дизъюнкции всех формул вида $\alpha(\varphi_{at'})$.

Для начальной вершины $at^0 = Init(G_{\mathcal{P}})$ соответствующая формула φ_{at^0} предполагается заданной. Например, для каждого из РП $\mathcal{P}_i$ ($i = 1, \ldots, 4$), определенных в пункте 2.6.3, в качестве такой формулы можно взять формулу из соответствующего соотношения в (50).

Ниже излагаются примеры применения данного метода для решения задач верификации РП, в которых используются защищенные каналы ($\mathcal{P}_1$ и $\mathcal{P}_3$), или защищенные ключи ($\mathcal{P}_2$ и $\mathcal{P}_4$). Перед решением задач верификации данных РП мы изложим теоремы, которые будут использоваться для решения этих задач.

3.1.2. Теоремы, используемые для верификации распределенных процессов с защищенными каналами

В этом пункте будем использовать следующее обозначение:

$$\begin{array}{l} \forall\, E \subseteq Tm, \forall\, \alpha \in Act, \forall\, c \in Channels \\ E_{c,\alpha} = E \cup \{e\}, \text{ если } \alpha = c!e, \text{ и } E_{c,\alpha} = E, \text{ иначе.} \end{array} \tag{53}$$

Теорема 7.

Пусть заданы РП $\mathcal{P}$, подмножество $E \subseteq \langle \mathcal{P} \rangle_0$, состояния $s, s' \in \Sigma_{\mathcal{P}_\dagger}$ такие, что $s \xrightarrow{\alpha_P} s'$, где $P \in \mathcal{P}$, и если $\alpha = c!e$, то верна импликация

$$c^s \notin E \;\; \Rightarrow \;\; Var(e^s) \cap E = \emptyset. \tag{54}$$

Тогда $\forall\, E', E'' \subseteq Tm$, $\forall\, c \in E_{\mathbf{C}}$ верна импликация

$$s \models \left\{ \begin{array}{l} E \perp_{\mathbf{C}} P_\dagger \\ E' \subseteq [c] \subseteq E'' \end{array} \right\} \Rightarrow s' \models \left\{ \begin{array}{l} E \perp_{\mathbf{C}} P_\dagger \\ E'_{c,\alpha} \subseteq [c] \subseteq E''_{c,\alpha} \end{array} \right\}$$

Доказательство.

Согласно (4), значение формулы $E \perp_{\mathbf{C}} P_\dagger$ в s зависит только от множеств $[P_\dagger]^s$ и $[c]^s$ ($\forall\, c \in Channels$), и

- если α имеет вид $c?e$ или $e := e'$, то при переходе от s к s' данные множества не изменяются, и
- если α имеет вид $c!e$, то при переходе от s к s' может измениться лишь множество $[c]^s$ путем добавления к нему терма e^s,

поэтому из (54) следует импликация $s \models E \perp_{\mathbf{C}} P_\dagger \Rightarrow s' \models E \perp_{\mathbf{C}} P_\dagger$.

Импликация $s \models E' \subseteq [c] \subseteq E'' \Rightarrow s' \models E'_{c,\alpha} \subseteq [c] \subseteq E''_{c,\alpha}$ следует из определения (53). ■

Теорема 8.

Пусть заданы

- РП $\mathcal{P}$, подмножество $E \subseteq \langle\mathcal{P}\rangle_0$, вершина $at \in G_{\mathcal{P}}$,
- множество $\{at_i \overset{\alpha_i}{\to} at \mid i \in I\}$ рёбер ГП $G_{\mathcal{P}}$ (с общим концом at), причем если $G_{\mathcal{P}}$ содержит ребро вида $at' \overset{\alpha}{\to} at$, не входящее в это множество, то вершина at' недостижима,
- множество $\{\varphi_i \mid i \in I\}$ формул, соответствующих вышеупомянутым ребрам, где $\forall\, i \in I \;\; at_i \models \varphi_i$, и φ_i состоит из следующих ЭФ:

$$\begin{cases} E \perp_{\mathbf{C}} P_\dagger \\ E'_{i,c} \subseteq [c] \subseteq E''_{i,c} \; (\forall\, c \in E_{\mathbf{C}}), \text{ где } E'_{i,c}, E''_{i,c} \subseteq Tm, \text{ и} \\ \text{равенства вида } e = e', \text{ где } e, e' \in Tm. \end{cases}$$

Тогда $at \models \varphi$, где φ состоит из следующих ЭФ:

$$\begin{cases} E \perp_{\mathbf{C}} P_\dagger \\ \bigcap_{i\in I}(E'_{i,c})_{c,\alpha_i} \subseteq [c] \subseteq \bigcup_{i\in I}(E''_{i,c})_{c,\alpha_i} \; (\forall\, c \in E_{\mathbf{C}}), \text{ и} \\ \text{равенства вида } e = e', \text{ где } e, e' \in Tm, \\ \qquad\qquad \text{входящие в каждую из формул } \varphi_i \; (i \in I). \end{cases}$$

Доказательство.
Данная теорема является следствием теорем 7 и 2. ■

Теорема 9.
Пусть заданы РП $\mathcal{P}$, и состояния $s, s' \in \Sigma_{\mathcal{P}_\dagger}$, такие, что $s \xrightarrow{c?\hat{x}} s'$.
Тогда верна импликация $s \models \{[c'] = \{e\}, c = c'\} \Rightarrow s' \models \{x = e\}$.

Доказательство.
Данная теорема непосредственно вытекает из определения выполнения действия вида $c?\hat{x}$ (см. пункт 2.2.5). ■

3.1.3. Редукция графов переходов

Если анализируемые свойства ГП $G_{\mathcal{P}}$ связаны только с его достижимыми вершинами, то при решении задач анализа таких свойств недостижимые вершины и связанные с ними ребра м.б. удалены из этого ГП. Будем называть такую операцию удаления **редукцией** ГП. Получившийся после такого удаления граф будем называть **редуцированным** ГП, и обозначать его той же записью $G_{\mathcal{P}}$. Если в редуцированном ГП обнаружатся неудаленные недостижимые вершины, то этот ГП можно еще раз редуцировать, и т.д.

Нахождение нереализуемых ребер и недостижимых вершин ГП можно производить с помощью следующих теорем.

Теорема 10.
Пусть задан РП $\mathcal{P}$. Если вершина $at \in G_{\mathcal{P}}$ такова, что

- $at \models \{[c] = \emptyset, c = c'\}$, где $c, c' \in Channels$, или
- $at \models \{k^{-1}[\circ] = \emptyset, k = k'\}$, где $k, k' \in Keys$,

и из at выходит ребро с меткой $c'?e$ или $?k'(e)$ соответственно, то это ребро нереализуемо. ■

Теорема 11.
Пусть задан РП $\mathcal{P}$. $\forall\, at \in G_{\mathcal{P}}$ верны следующие утверждения:

- если все рёбра с концом at нереализуемы, то at недостижима, и
- если at недостижима, то все рёбра с началом at нереализуемы. ■

3.1.4. Верификация распределенного процесса $\mathcal{P}_1$

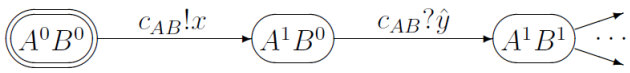
Применим доказанные выше теоремы для доказательства свойства (52) РП $\mathcal{P}_1$, описываемого схемой (41). ГП $G_{\mathcal{P}_1}$ имеет вид (47).

Теорема 10 и первое соотношение в (50), которое имеет вид

$$A^0B^0 \models \{\varphi_1, [c_{AB}] = \emptyset\} \tag{55}$$

обосновывают нереализуемость отмеченного черным кружочком ребра в ГП (47). По теореме 11, отсюда следует недостижимость вершины A^0B^1.

После редукции ГП (47) путем удаления недостижимых вершин и связанных с ними ребер получаем граф



(56)

В (56) существует единственная вершина (A^1B^1), удовлетворяющая условию в (52). Таким образом, требуется доказать, что $A^1B^1 \models x = y$.

Из (55) по теореме 8 следует, что $A^1B^0 \models \{\varphi_1, [c_{AB}] = \{x\}\}$, откуда по теоремам 8 и 9 следует желаемое свойство $A^1B^1 \models x = y$.

3.1.5. Верификация распределенного процесса $\mathcal{P}_3$

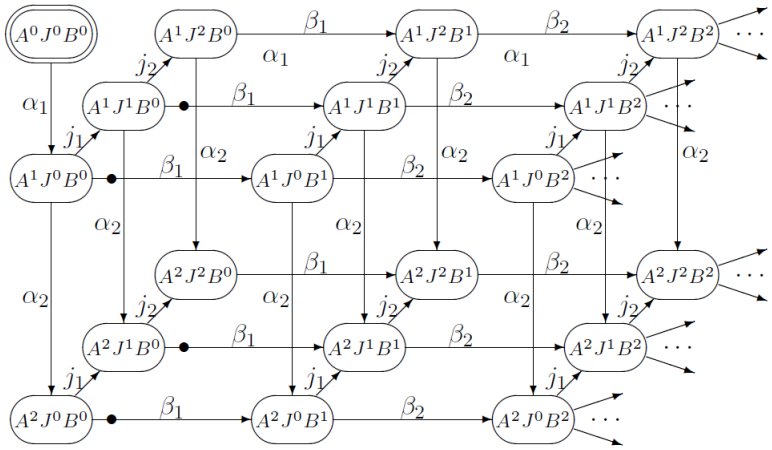
Рассмотрим теперь задачу доказательства свойства (52) для РП $\mathcal{P}_3$, описываемого схемой (44), в которой действия определяются согласно (43). ГП $G_{\mathcal{P}_3}$ имеет вид (49).

Теорема 10 и третье соотношение в (50), которое имеет вид

$$A^0J^0B^0 \models \{\varphi_3, [c_{AJ}] = \emptyset, [c_{BJ}] = \emptyset, [\bar{c}] = \emptyset\} \tag{57}$$

обосновывают нереализуемость отмеченных черными кружочками ребер в ГП (49). По теореме 11, отсюда следует недостижимость всех вершин верхнего яруса в ГП (49) за исключением вершины $A^0J^0B^0$.

После редукции ГП (49) путем удаления недостижимых вершин верхнего яруса и связанных с ними ребер получаем редуцированный ГП



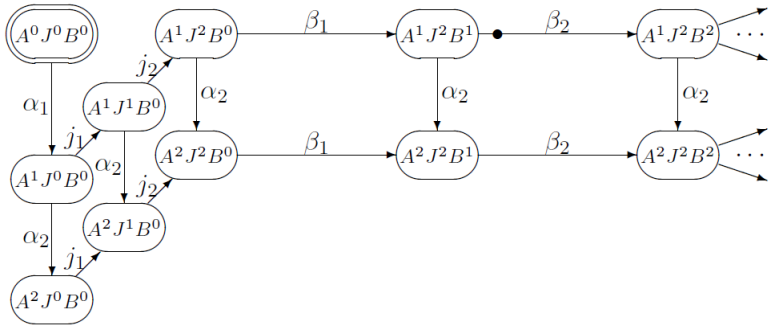
(58)

Далее мы приводим список соотношений, каждое из которых следует из предыдущих (первое следует из (57)) по теоремам 8 и 9:

$$\begin{array}{l} A^1J^0B^0 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \emptyset, [\bar{c}] = \emptyset\} \\ A^2J^0B^0 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \emptyset, [\bar{c}] = \{x\}\} \\ A^1J^1B^0 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \emptyset, [\bar{c}] = \emptyset, u = \bar{c}\} \\ A^2J^1B^0 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \emptyset, [\bar{c}] = \{x\}, u = \bar{c}\} \end{array} \tag{59}$$

По теореме 10, из данных соотношений следует нереализуемость отмеченных черными кружочками ребер в ГП (59). Удаляя эти ребра и соответствующие недостижимые вершины (используя теорему 11), получаем редуцирован-

ный ГП

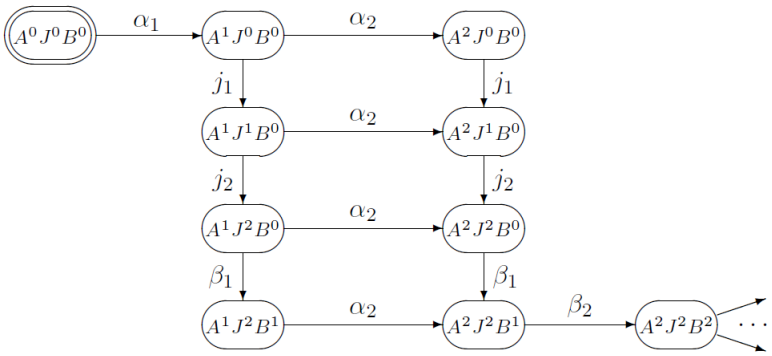


(60)

Из последнего соотношения в (59) при помощи теорем 8 и 9 получаем:

$$\begin{array}{l} A^1J^2B^0 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \{u\}, [\bar{c}] = \emptyset, u = \bar{c}\} \\ A^1J^2B^1 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \{u\}, [\bar{c}] = \emptyset, u = \bar{c}, v = u\} \end{array} \tag{61}$$

По теореме 10, из последнего соотношения в (61) следует нереализуемость отмеченного черным кружочком ребра в ГП (60). Удаляя это ребро и соответствующие недостижимые вершины (для нахождения которых используем теорему 11), получаем редуцированный ГП



(62)

Применяя теоремы 8 и 9, вычисляем формулы, соответствующие оставшимся вершинам:

$$\begin{array}{l} A^2J^2B^0 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \{u\}, [\bar{c}] = \{x\}, u = \bar{c}\} \\ A^2J^2B^1 \models \{\varphi_3, [c_{AJ}] = \{\bar{c}\}, [c_{BJ}] = \{u\}, [\bar{c}] = \{x\}, u = \bar{c}, v = u\} \\ A^2J^2B^2 \models \{x = y\} \end{array} \tag{63}$$

Поскольку

- в ГП (62) вершина $A^2J^2B^2$ является единственной вершиной, удовлетворяющей условию в (52), и

- согласно последнему соотношению в (63), для этой вершины верно утверждение в (52),

то задача доказательства свойства (52) для РП $\mathcal{P}_3$ решена.

3.1.6. Теоремы, используемые для верификации распределенных процессов с защищенными ключами

В этом пункте будем использовать следующее обозначение:

$$\begin{array}{l}\forall\, E \subseteq Tm, \forall\, \alpha \in Act, \forall\, k \in Keys \\ E_{k,\alpha} = E \cup \{e\},\ \text{если}\ \alpha = !k(e),\ \text{и}\ E_{k,\alpha} = E,\ \text{иначе.}\end{array} \tag{64}$$

Теорема 12.

Пусть заданы РП $\mathcal{P}$, где $Var(\mathcal{P})_{\mathbf{C}} = \{\circ\}$, подмножество $E \subseteq \langle\mathcal{P}\rangle_0$, состояния $s, s' \in \Sigma_{\mathcal{P}_\dagger}$, такие, что $s \xrightarrow{\alpha_P} s'$, где $P \in \mathcal{P}$, и если $\alpha = !e$, то

$$\forall\, x \in E_{\mathbf{X}}\ \ x \perp_{\mathbf{K},E} e^s. \tag{65}$$

Тогда $\forall\, E', E'' \subseteq Tm$, $\forall\, k \in E_{\mathbf{K}} : k \neq public\_key(\ldots)$ верна импликация

$$s \models \left\{ \begin{array}{l} E \perp_{\mathbf{K}} P_\dagger, k^{-1}[P_\dagger] \subseteq k^{-1}[\circ] \\ E' \subseteq k^{-1}[\circ] \subseteq E'' \end{array} \right\} \Rightarrow s' \models \left\{ \begin{array}{l} E \perp_{\mathbf{K}} P_\dagger, k^{-1}[P_\dagger] \subseteq k^{-1}[\circ] \\ E'_{k,\alpha} \subseteq k^{-1}[\circ] \subseteq E''_{k,\alpha} \end{array} \right\}$$

Доказательство.

Значения формул $E \perp_{\mathbf{K}} P_\dagger$ и $k^{-1}[P_\dagger] \subseteq k^{-1}[\circ]$ в s зависит только от множеств $[P_\dagger]^s$ и $[\circ]_s$ (для $E \perp_{\mathbf{K}} P_\dagger$ это верно согласно (5)), и

- если α имеет вид $?e$ или $e := e'$, то при переходе от s к s' данные множества не изменяются, и
- если α имеет вид $!e$, то при переходе от s к s' может измениться лишь множество $[\circ]_s$ путем добавления к нему терма e^s,

поэтому из (65) следует импликация

$$s \models \{E \perp_{\mathbf{K}} P_\dagger, k^{-1}[P_\dagger] \subseteq k^{-1}[\circ]\} \Rightarrow s' \models \{E \perp_{\mathbf{K}} P_\dagger, k^{-1}[P_\dagger] \subseteq k^{-1}[\circ]\}.$$

Импликация $s \models E' \subseteq k^{-1}[\circ] \subseteq E'' \Rightarrow s' \models E'_{k,\alpha} \subseteq k^{-1}[\circ] \subseteq E''_{k,\alpha}$ следует из определения (64). ■

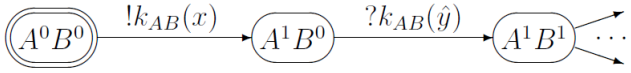
Кроме того, верны аналоги теоремы 8, с заменой

- $E \perp_{\mathbf{C}} P_\dagger$ на $\{E \perp_{\mathbf{K}} P_\dagger, k^{-1}[P_\dagger] \subseteq k^{-1}[\circ]\}$,
- $[c]$ на $k^{-1}[\circ]$, $E_{i,c}$ на $E_{i,k}$, $(E_{i,c})_{c,\alpha_i}$ на $(E_{i,k})_{k,\alpha_i}$,

и теоремы 9, с заменой $c?\hat{x}$ на $?k(\hat{x})$, $[c']$ на $(k')^{-1}[\circ]$, $c = c'$ на $k = k'$.

3.1.7. Верификация распределенного процесса $\mathcal{P}_2$

Доказательство свойства (52) для РП $\mathcal{P}_2$, описываемого схемой (42), проводится аналогично доказательству этого свойства для РП $\mathcal{P}_1$ в пункте 3.1.4. ГП $G_{\mathcal{P}_2}$ имеет вид (48). Так же обосновывается нереализуемость отмеченного черным кружочком ребра в ГП (48). По теореме 11, отсюда следует недостижимость вершины A^0B^1. После редукции ГП (48) получаем граф



(66)

В (66) существует единственная вершина (A^1B^1), удовлетворяющая условию в (52). Таким образом, требуется доказать, что $A^1B^1 \models x = y$. Это свойство следует из соотношения $A^1B^0 \models \{\varphi_2, k_{AB}^{-1}[\circ] = \{x\}\}$.

3.1.8. Верификация распределенного процесса $\mathcal{P}_4$

Доказательство свойства (52) для РП $\mathcal{P}_4$, описываемого схемой (44), где действия α_i, β_i, j_i $(i = 1, 2)$ определяются согласно (45), проводится аналогично доказательству этого свойства для РП $\mathcal{P}_3$ в пункте 3.1.5. ГП $G_{\mathcal{P}_4}$ имеет вид (49). Так же обосновывается нереализуемость отмеченных черными кружочками ребер в ГП (49). После редукции ГП (49) получаем такие же ГП (58), (60), (62), как и в случае верификации РП $\mathcal{P}_3$. Мы не будем излагать детально решение задачи верификации РП $\mathcal{P}_4$, приведем лишь соотношения, связанные с вершинами ГП (62) для данного случая.

$$
\begin{aligned}
&A^1J^0B^0 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \emptyset, \bar{k}^{-1}[\circ] = \emptyset\}\\
&A^2J^0B^0 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \emptyset, \bar{k}^{-1}[\circ] = \{x\}\}\\
&A^1J^1B^0 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \emptyset, \bar{k}^{-1}[\circ] = \emptyset, u = \bar{k}\}\\
&A^2J^1B^0 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \emptyset, \bar{k}^{-1}[\circ] = \{x\}, u = \bar{k}\}\\
&A^1J^2B^0 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \{u\}, \bar{k}^{-1}[\circ] = \emptyset, u = \bar{k}\}\\
&A^1J^2B^1 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \{u\}, \bar{k}^{-1}[\circ] = \emptyset, u = \bar{k}, v = u\}\\
&A^2J^2B^0 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \{u\}, \bar{k}^{-1}[\circ] = \{x\}, u = \bar{k}\}\\
&A^2J^2B^1 \models \{\varphi_4, k_{AJ}^{-1}[\circ] = \{\bar{k}\}, k_{BJ}^{-1}[\circ] = \{u\}, \bar{k}^{-1}[\circ] = \{x\}, u = \bar{k}, v = u\}\\
&A^2J^2B^2 \models \{x = y\}.
\end{aligned}
$$

3.2. Верификация протокола Yahalom

В этом и следующем параграфе рассматривается другой метод верификации КП, основанный на использовании теоремы 6. Данный метод не описывается явно, т.к. его содержание можно понять по приводимым ниже примерам его применения в задаче верификации КП аутентификации Yahalom (в этом параграфе) и КП передачи ШС между несколькими агентами (в параграфе 3.3).

3.2.1. Описание протокола Yahalom

КП Yahalom предназначен для аутентификации (т.е. проверки подлинности) агентов, взаимодействующих по открытому каналу $\circ$, и передачи сеансовых ключей между этими агентами.

Предполагается что

- заданы множество агентов Ag, а также агент J, называемый **доверенным посредником**, данные агенты могут взаимодействовать друг с другом по открытому каналу $\circ$,
- каждый агент $A \in Ag$ имеет общий секретный ключ k_{AJ} с доверенным посредником J, на котором A и J могут шифровать и дешифровать сообщения, используя симметричную систему шифрования, причем только A и J знают ключ k_{AJ}.

В каждом сеансе КП Yahalom принимают участие следующие агенты: **инициатор** $A \in Ag$, доверенный посредник J, и **респондер** $B \in Ag$. Каждый агент из Ag в одних сеансах м.б. инициатором, а в других – респондером. Один и тот же агент м.б. и инициатором и респондером в одном и том же сеансе (т.е. возможно, что $A = B$). Выполнение сеанса КП Yahalom с инициатором A, респондером B и доверенным посредником J представляет собой совокупность четырех пересылок сообщений:

$$\begin{array}{lcl} 1. \quad A \to B & : & A, n_A \\ 2. \quad B \to J & : & B, k_{BJ}(A, n_A, n_B) \\ 3. \quad J \to A & : & k_{AJ}(B, k, n_A, n_B), k_{BJ}(A, k) \\ 4. \quad A \to B & : & k_{BJ}(A, k), k(n_B) \end{array} \tag{67}$$

Пересылки в (67) имеют следующий смысл:

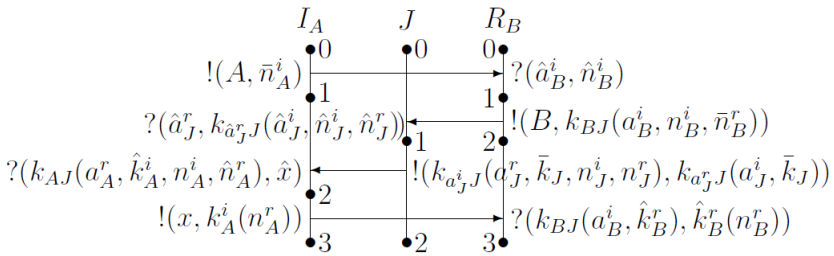
1) A посылает B запрос на аутентификацию и генерацию сеансового ключа k, этот запрос состоит из имени агента A и нонса n_A,

2) B посылает J запрос на генерацию сеансового ключа k, в свой запрос он включает своё имя, имя агента A, для связи с которым нужен этот ключ, полученный нонс n_A, и свой нонс n_B,

3) J генерирует сеансовый ключ k и посылает A пару сообщений,
   - из первого сообщения A может извлечь сеансовый ключ k,
   - а второе предназначено для того, чтобы A переслал его B,

4) A посылает B пару сообщений,
   - первое из которых было получено им от J, агент B может извлечь из этого сообщения сеансовый ключ k, и
   - используя ключ k, агент B дешифрует второе сообщение, если результат дешифрования совпадает с его нонсом n_B, то это является для него доказательством того, что отправителем этого сообщения был именно A.

Формальное описание сеанса КП Yahalom изображается схемой



(68)

В этой схеме левая и правая нити соответствуют ПП I_A и R_B, описывающим поведение инициатора A и респондера B соответственно, средняя нить соответствует ПП, описывающему поведение посредника J, этот ПП обозначается тем же символом J. Смысл переменных в этих ПП усматривается из сопоставления действий в этих ПП с соответствующими действиями в (67). Верхний индекс i или r при какой-либо переменной означает, что она предположительно содержит информацию об инициаторе (i) или респондере (r) данного сеанса.

Предполагаем, что $Agent(I_A) = A$, $Agent(R_B) = B$, $Agent(J) = J$.

РП $\mathcal{P}$, соответствующий КП Yahalom, имеет вид

$$\mathcal{P} = \{\{I^*_A \mid A \in Ag\}, \{R^*_B \mid B \in Ag\}, J^*\}. \tag{69}$$

Ниже мы будем использовать следующие обозначения:

- если $\mathcal{P}$ – РП, и π – путь в $\Sigma_{\mathcal{P}_\dagger}$, то запись $\pi \ni P^{i,i'} : s \overset{\alpha}{\to} s'$ означает, что π содержит ребро $s \overset{\alpha_P}{\to} s'$, и $at_{s_P} = i$, $at_{s'_P} = i'$,
- запись $s \models E \perp_{\mathbf{K}} e$ обозначает утверждение $\forall\, x \in E_{\mathbf{X}} \;\; x \perp_{\mathbf{K},E} e^s$.

Нетрудно доказать, что

$$s \models E \perp_{\mathbf{K}} (e, e') \Leftrightarrow s \models E \perp_{\mathbf{K}} e \quad \text{и} \quad s \models E \perp_{\mathbf{K}} e'. \tag{70}$$

3.2.2. Свойства протокола Yahalom

Свойства РП (69), которые будут верифицированы:

- **секретность** ключей и нонсов n^r_B:

$$\forall\, s \in \Sigma_{\mathcal{P}_\dagger} \quad s \models E \perp_{\mathbf{K}} P_\dagger, \quad \text{где } E = \{k_{BJ}, k_J, n^r_B \mid B \in Ag\} \tag{71}$$

- **аутентификация инициатора перед респондером**: для любых $R_B \in \mathcal{P}$, $s \in \Sigma_{\mathcal{P}_\dagger}$, если $s \models at_{R_B} = 3$, то $\exists\, I_A \in \mathcal{P}$:

$$s \models \{at_{I_A} = 3, a^r_A = B, a^i_B = A, n^i_A = n^i_B, n^r_A = n^r_B, k^i_A = k^r_B\}, \tag{72}$$

- **аутентификация респондера перед инициатором**: для любых $I_A \in \mathcal{P}$, $s \in \Sigma_{\mathcal{P}_\dagger}$, если $s \models at_{I_A} = 2$, то $\exists\, R_B \in \mathcal{P}$:

$$s \models \{at_{R_B} = 2, a^r_A = B, a^i_B = A, n^i_A = n^i_B, n^r_A = n^r_B\}. \tag{73}$$

3.2.3. Секретность ключей и нонсов n^r_B

Докажем (71) от противного.

Предположим, что $S = \{s \in \Sigma_{\mathcal{P}_\dagger} \mid s \not\models \varphi\} \neq \emptyset$, где φ – формула в (71).

$\forall\, s \in S$ обозначим записью π_s путь минимальной длины из 0 в s. Пусть s – состояние из S с наименьшей длиной π_s. Т.к. $0 \models \varphi$, то $s \neq 0$.

Пусть $s' \overset{\alpha_P}{\to} s$ – ребро из π_s с концом в s.

Из определения s следует, что $s' \models \varphi$, $s \not\models \varphi$. Если бы было верно $P = P_\dagger$, то из теоремы 3 следует, что $s \models \varphi$, т.е. имеем противоречие.

Поэтому $P \in \{I_A, R_B, J \mid A, B \in Ag\}$, и

$$\alpha_P = !e,\ [\circ]_s = [\circ]_{s'} \cup \{e\},\ \exists\, y \in E_{\mathbf{X}} : \neg(y \perp_{\mathbf{K},E} e^s). \tag{74}$$

Перебором всех вариантов обоснования существования ребра $s' \overset{\alpha_P}{\to} s$ со свойствами (74) находим единственное возможное обоснование:

$$\pi_s \ni I^{2,3}_A : s' \overset{!e}{\to} s, \quad \text{где } e = (x, k^i_A(n^r_A)). \tag{75}$$

Т.к. $s' \models at_{I_A} = 2$, то $\exists\, s_1 \leq_{\pi_s} s'$:

$$\pi_s \ni I^{1,2}_A : s'_1 \overset{?e_1}{\to} s_1, \text{ где } e_1 = (k_{AJ}(a^r_A, \hat{k}^i_A, n^i_A, \hat{n}^r_A), \hat{x}). \tag{76}$$

Т.к. $s_1 \leq_{\pi_s} s'$ и $s' \models \varphi$, то $s_1 \models \varphi$. В частности, $s_1 \models E \perp_{\mathbf{K}} e_1$. По (70), отсюда следует, что $s_1 \models E \perp_{\mathbf{K}} x$.

По теореме 6, из $s_1 \models \varphi$, $e^s_1 \in [\circ]_{s_1}$ и $k_{AJ} \in E$ следует, что $\exists\, s_2 \leq_{\pi_s} s'_1$: π_s содержит ребро $s'_2 \overset{(!e_2)_P}{\longrightarrow} s_2$, где $P \in \mathcal{P}$ и первая компонента $k_{AJ}(\ldots)$ терма e^s_1 входит в e^s_2. Перебором всех вариантов обоснования существования ребра с такими свойствами находим единственное обоснование:

$$\begin{cases} \pi_s \ni J^{1,2} : s'_2 \overset{!e_2}{\to} s_2, \text{ где } e_2 = (k_{a^i_J J}(a^r_J, \bar{k}_J, n^i_J, n^r_J), \ldots) \\ k_{(a^i_J)^s J}((a^r_J)^s, \bar{k}_J, \ldots) = k_{AJ}(a^r_A, (k^i_A)^s, \ldots) \end{cases} \tag{77}$$

(многоточие в (77) и ниже обозначает компоненту пары, не представляющую интерес для рассмотрения). Из (77) следует, что $(k^i_A)^s = \bar{k}_J$, поэтому $s \models E \perp_{\mathbf{K}} k^i_A(n^r_A)$. Учитывая установленное выше свойство $s_1 \models E \perp_{\mathbf{K}} x$, на основании (70) получаем: $s \models E \perp_{\mathbf{K}} (x, k^i_A(n^r_A))$, т.е. $s \models E \perp_{\mathbf{K}} e$, что противоречит предположению $s \not\models E \perp_{\mathbf{K}} e$. ■

Доказанное свойство $\forall\, s \in \Sigma_{\mathcal{P}_\dagger}\ \ s \models \varphi$ будет использоваться ниже.

В излагаемых ниже доказательствах при каждом применении теоремы 6 имеется единственный вариант обоснования существования ребра (35) в графе $\Sigma_{\mathcal{P}_\dagger}$, и мы будем сразу будем излагать это обоснование, без упоминания о единственности варианта такого обоснования. Эта единственность обеспечивается подходящим определением действий вида $!e$ в ПП, входящих в рассматриваемый РП.

3.2.4. Аутентификация инициатора перед респондером

Пусть ПП $R_B \in \mathcal{P}$ и состояние $s \in \Sigma_{\mathcal{P}_\dagger}$ таковы, что $s \models at_{R_B} = 3$.

Докажем, что $\exists\, I_A \in \mathcal{P}$: выполнено (72).

Пусть π – путь из 0 в s. Из $s \models at_{R_B} = 3$ следует, что $\exists\, s_1 \leq_\pi s$:

$$\pi \ni R_B^{2,3} : s_1' \overset{?e_1}{\to} s_1, \text{ где } e_1 = (k_{BJ}(a_B^i, \hat{k}_B^r), \hat{k}_B^r(n_B^r)).$$

По теореме 6, из $s_1 \models \varphi$, $e_1^s \in [\circ]_{s_1}$ и $k_{BJ} \in E$ следует, что $\exists\, s_2 \leq_\pi s_1'$:

$$\begin{cases} \pi \ni J^{1,2} : s_2' \overset{!e_2}{\to} s_2, \text{ где } e_2 = (\ldots, k_{a_J^r J}(a_J^i, \bar{k}_J)) \\ k_{(a_J^r)^s J}((a_J^i)^s, \bar{k}_J) = k_{BJ}((a_B^i)^s, (k_B^r)^s) \end{cases} \tag{78}$$

Из второго равенства в (78) следует, что

$$(a_J^r)^s = B, (a_J^i)^s = (a_B^i)^s, \bar{k}_J = (k_B^r)^s. \tag{79}$$

Из $s_2' \models at_J = 1$ следует, что $\exists\, s_3 \leq_\pi s_2'$:

$$\pi \ni J^{0,1} : s_3' \overset{?e_3}{\to} s_3, \text{ где } e_3 = (\ldots, k_{\hat{a}_J^r J}(\hat{a}_J^i, \hat{n}_J^i, \hat{n}_J^r)). \tag{80}$$

Из (79) и (80) следует, что $k_{BJ}(3 \text{ терма}) \subseteq e_3^s \in [\circ]_{s_3}$, откуда по теореме 6, с учетом $s_3 \models \varphi$ и $k_{BJ} \in E$ получаем: $\exists\, s_4 \leq_\pi s_3'$:

$$\begin{cases} \pi \ni R_{\grave{B}}^{1,2} : s_4' \overset{!e_4}{\to} s_4, \text{ где } e_4 = (\ldots, k_{\grave{B}J}(a_{\grave{B}}^i, n_{\grave{B}}^i, \bar{n}_{\grave{B}}^r)) \\ k_{\grave{B}J}((a_{\grave{B}}^i)^s, (n_{\grave{B}}^i)^s, \bar{n}_{\grave{B}}^r) = k_{BJ}((a_B^i)^s, (n_J^i)^s, (n_J^r)^s) \end{cases} \tag{81}$$

Из второго равенства в (81) следует, что

$$\grave{B} = B, (n_B^i)^s = (n_J^i)^s, \bar{n}_B^r = (n_J^r)^s. \tag{82}$$

По теореме 6, из $s_1 \models \varphi$, $(k_B^r(n_B^r))^s \subseteq e_1^s \in [\circ]_{s_1}$, и $(k_B^r)^s = \bar{k}_J \in E$ следует, что $\exists\, s_5 \leq_\pi s_1'$:

$$\begin{cases} \pi \ni I_A^{2,3} : s_5' \overset{!e_5}{\to} s_5, \text{ где } e_5 = (\ldots, k_A^i(n_A^r)) \\ (k_A^i(n_A^r))^s = \bar{k}_J(n_B^r) \end{cases} \tag{83}$$

Из второго равенства в (83) следует, что

$$(k_A^i)^s = \bar{k}_J, (n_A^r)^s = n_B^r. \tag{84}$$

Из $s_5 \models at_{I_A} = 2$ следует, что $\exists\, s_6 \leq_\pi s_5'$:

$$\pi \ni I_A^{1,2} : s_6' \overset{?e_6}{\to} s_6, \text{ где } e_6 = (k_{AJ}(a_A^r, \hat{k}_A^i, n_A^i, \hat{n}_A^r), \ldots). \tag{85}$$

Из (84) и (85) следует, что

$$k_{AJ}(a_A^r,(k_A^i)^s,n_A^i,(n_A^r)^s)=k_{AJ}(a_A^r,\bar{k}_J,n_A^i,n_B^r)\subseteq e_6^s\in[\circ]_{s_6}. \tag{86}$$

По теореме 6, из $s_6\models\varphi$, $k_{AJ}\in E$, и (86) следует, что $\exists\, s_7\leq_\pi s_6'$:

$$\left\{\begin{array}{l}\pi\ni \grave{J}^{1,2}:s_7'\overset{!e_7}{\to}s_7,\ \text{где } e_7=(k_{a_{\grave{J}}^i\grave{J}}(a_{\grave{J}}^r,\bar{k}_{\grave{J}},n_{\grave{J}}^i,n_{\grave{J}}^r),\ldots)\\ k_{(a_{\grave{J}}^i)^sJ}((a_{\grave{J}}^r)^s,\bar{k}_{\grave{J}},(n_{\grave{J}}^i)^s,(n_{\grave{J}}^r)^s)=k_{AJ}(a_A^r,\bar{k}_J,n_A^i,n_B^r)\end{array}\right. \tag{87}$$

Из второго равенства в (87) следует, что

$$(a_{\grave{J}}^i)^s=A,(a_{\grave{J}}^r)^s=a_A^r,\grave{J}=J,(n_J^i)^s=n_A^i,(n_J^r)^s=\bar{n}_B^r. \tag{88}$$

Свойство (72) следует из (79), (82), (84), (88). ■

3.2.5. Аутентификация респондера перед инициатором

Пусть ПП $I_A\in\mathcal{P}$ и состояние $s\in\Sigma_{\mathcal{P}_\dagger}$ таковы, что $s\models at_{I_A}=2$.

Докажем, что $\exists\, R_B\in\mathcal{P}$: выполнено (73).

Пусть π – путь из 0 в s. Из $s\models at_{I_A}=2$ следует, что $\exists\, s_1\leq_\pi s$:

$$\pi\ni I_A^{1,2}:s_1'\overset{?e_1}{\to}s_1,\ \text{где } e_1=(k_{AJ}(a_A^r,\hat{k}_A^i,n_A^i,\hat{n}_A^r),\ldots). \tag{89}$$

По теореме 6, из $s_1\models\varphi$, $k_{AJ}(4\text{ терма})\subseteq e_1^s\in[\circ]_{s_1}$ и $k_{AJ}\in E$, следует, что $\exists\, s_2\leq_\pi s_1'$:

$$\left\{\begin{array}{l}\pi\ni J^{1,2}:s_2'\overset{!e_2}{\to}s_2,\ \text{где } e_2=(k_{a_J^iJ}(a_J^r,\bar{k}_J,n_J^i,n_J^r),\ldots)\\ k_{(a_J^i)^sJ}((a_J^r)^s,\bar{k}_J,(n_J^i)^s,(n_J^r)^s)=k_{AJ}(a_A^r,(k_A^i)^s,n_A^i,(n_A^r)^s)\end{array}\right. \tag{90}$$

Из второго равенства в (90) следует, что

$$(a_J^i)^s=A,(a_J^r)^s=a_A^r,\bar{k}_J=(k_A^i)^s,(n_J^i)^s=n_A^i,(n_J^r)^s=(n_A^r)^s. \tag{91}$$

Из $s_2\models at_J=1$ следует, что $\exists\, s_3\leq_\pi s_2'$:

$$\pi\ni J^{0,1}:s_3'\overset{?e_3}{\to}s_3,\ \text{где } e_3=(\ldots,k_{\hat{a}_J^rJ}(\hat{a}_J^i,\hat{n}_J^i,\hat{n}_J^r)). \tag{92}$$

Из (91) и (92) следует, что $k_{a_A^rJ}(A,n_A^i,(n_A^r)^s)\subseteq e_3^s\in[\circ]_{s_3}$, откуда по теореме 6, учитывая $s_3\models\varphi$, и $k_{a_A^rJ}\in E$, получаем: $\exists\, s_4\leq_\pi s_3'$:

$$\left\{\begin{array}{l}\pi\ni R_B^{1,2}:s_4'\overset{!e_4}{\to}s_4,\ \text{где } e_4=(\ldots,k_{BJ}(a_B^i,n_B^i,\bar{n}_B^r))\\ k_{BJ}((a_B^i)^s,(n_B^i)^s,\bar{n}_B^r)=k_{a_A^rJ}(A,n_A^i,(n_A^r)^s).\end{array}\right. \tag{93}$$

Второе равенство в (93) влечёт равенства, из которых следует (73):

$$B=a_A^r,(a_B^i)^s=A,(n_B^i)^s=n_A^i,\bar{n}_B^r=(n_A^r)^s.\ \blacksquare$$

3.3. Верификация протокола передачи шифрованных сообщений между несколькими агентами

В этом пункте рассматривается пример верификации КП, предназначенного для передачи ШС по открытому каналу между несколькими агентами. Данный КП является обобщением рассмотренного в пункте 2.6.3 КП Wide-Mouth Prog, представляемого РП $\mathcal{P}_4$.

3.3.1. Описание протокола

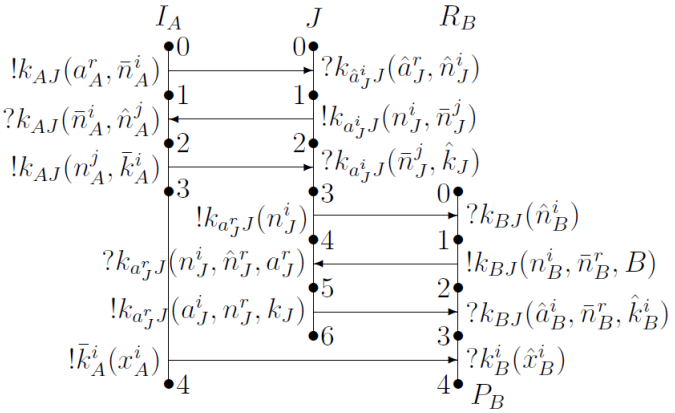
Участники этого протокола – агенты из множества $Ag \subseteq Agents$ и доверенный посредник J. Каждый агент $A \in Ag$ использует для связи с J ключ k_{AJ}, доступный только A и J. Сеанс передачи сообщения x в зашифрованном виде от агента $A \in Ag$ агенту $B \in Ag$ включает в себя следующие действия:

- обмен сообщениями между A и J, в результате чего J узнает имя A отправителя, имя B получателя, и ключ k, на котором будет зашифровано сообщение x от A для получателя B,
- обмен сообщениями между J и B, в результате чего B узнает имя A отправителя сообщения, которое B получит от A, и ключ k, на котором будет зашифровано это сообщение,
- пересылка ШС $k(x)$ от A к B.

Выполнение сеанса данного КП с инициатором A, респондером B и доверенным посредником J представляет собой следующую совокупность пересылок сообщений:

$$\begin{array}{lllll} 1. & A \to J & : & k_{AJ}(A, n_A) \\ 2. & J \to A & : & k_{AJ}(n_A, n_J) \\ 3. & A \to J & : & k_{AJ}(n_J, k) \\ 4. & J \to B & : & k_{BJ}(n_A) \\ 5. & B \to J & : & k_{BJ}(n_A, n_B, B) \\ 6. & J \to B & : & k_{BJ}(A, n_B, k) \\ 7. & A \to B & : & k(x) \end{array} \tag{94}$$

Данный сеанс представляется следующей схемой:



(95)

РП $\mathcal{P}$, соответствующий этому КП, имеет вид (69).
Свойства этого КП, которые должны быть верифицированы:

- **секретность** ключей, передаваемых сообщений и нонсов:

$$\forall\, s \in \Sigma_{\mathcal{P}_\dagger}\ \ s \models E \perp_{\mathbf{K}} P_\dagger,\ \ \text{где } E = \{k_{AJ}, k_A^i, x_A^i, n_A^i \mid A \in Ag\} \tag{96}$$

- **целостность** передаваемых сообщений:

$$\begin{array}{l}\forall\, R_B \in \mathcal{P},\, \forall\, s \in \Sigma_{\mathcal{P}_\dagger}, \text{ если } s \models at_{R_B} = 4, \text{ то } \exists\ I_A \in \mathcal{P}: \\ s \models \{at_{I_A} = 4, a_A^r = B, a_B^i = A, n_A^i = n_B^i, k_A^i = k_B^i, x_A^i = x_B^i\}\end{array} \tag{97}$$

3.3.2. Верификация протокола

Доказательство свойства секретности (96) дословно повторяет начало рассуждений в пункте 3.2.3 по доказательству аналогичного свойства протокола Yahalom, с тем лишь отличием что не существует ни одного варианта обоснования существования ребра $s' \overset{\alpha_P}{\to} s$ со свойствами (74).

Докажем свойство целостности (97). Будем использовать в этом доказательстве доказанное выше свойство (96) (не упоминая об этом).

Пусть ПП $R_B \in \mathcal{P}$ и состояние $s \in \Sigma_{\mathcal{P}_\dagger}$ таковы, что $s \models at_{R_B} = 4$. Докажем, что $\exists\, I_A \in \mathcal{P}$: выполнено утверждение во второй строчке (97).

Пусть π – путь из 0 в s. Из $s \models at_{R_B} = 4$ следует, что

$$\begin{array}{l}\exists\, s_1 \le_\pi s:\ \pi \ni R_B^{3,4} : s_1' \overset{?e_1}{\to} s_1,\ \text{где } e_1 = k_B^i(\hat{x}_B^i) \\ \exists\, s_2 \le_\pi s_1':\ \pi \ni R_B^{2,3} : s_2' \overset{?e_2}{\to} s_2,\ \text{где } e_2 = k_{BJ}(\hat{a}_B^i, \bar{n}_B^r, \hat{k}_B^i)\end{array} \tag{98}$$

По теореме 6, из второй строки в (98), $e_2^s \in [\circ]_{s_2}$, $k_{BJ} \in E$, следует:

$$\left\{\begin{array}{l}\exists\, s_3 \le_\pi s_2' : \pi \ni J^{5,6} : s_3' \overset{!e_3}{\to} s_3,\ \text{где } e_3 = k_{a_J^r J}(a_J^i, n_J^r, k_J) \\ k_{(a_J^r)^s J}((a_J^i)^s, (n_J^r)^s, (k_J)^s) = k_{BJ}(\hat{a}_B^i, \bar{n}_B^r, \hat{k}_B^i)\end{array}\right. \tag{99}$$

Из второй строки в (99) следует, что

$$(a_J^r)^s = B,\ (a_J^i)^s = (a_B^i)^s,\ (n_J^r)^s = \bar{n}_B^r,\ (k_J)^s = (k_B^i)^s. \tag{100}$$

Из первой строки в (99), с учетом (100), получаем:

$$\exists\, s_4 \le_\pi s_3' : \pi \ni J^{4,5} : s_4' \overset{?e_4}{\to} s_4,\ \text{где } e_4 = k_{BJ}(n_J^i, n_B^r, B). \tag{101}$$

По теореме 6, из (101), и того, что $e_4^s \in [\circ]_{s_4}$, $k_{BJ} \in E$, следует:

$$\left\{\begin{array}{l}\exists\, s_5 \le_\pi s_4' : \pi \ni \grave{B}^{1,2} : s_5' \overset{!e_5}{\to} s_5,\ \text{где } e_5 = k_{\grave{B}J}(n_{\grave{B}}^i, \bar{n}_{\grave{B}}^r, \grave{B}) \\ k_{\grave{B}J}((n_{\grave{B}}^i)^s, \bar{n}_{\grave{B}}^r, \grave{B}) = k_{BJ}((n_J^i)^s, \bar{n}_B^r, B)\end{array}\right. \tag{102}$$

Из второй строки в (102) следует, что

$$\bar{n}^r_{\grave{B}} = \bar{n}^r_B,\ \grave{B} = B,\ (n^i_B)^s = (n^i_J)^s. \tag{103}$$

Из (101) следует, что

$$\exists\, s_6 \leq_\pi s'_4 : \pi \ni J^{2,3} : s'^{\,?e_6}_6 \to\ s_6,\ \text{где}\ e_6 = k_{a^i_J J}(n^j_J, k_J). \tag{104}$$

По теореме 6, из (104), и того, что $e^s_6 \in [\circ]_{s_6}$, $k_{(a^i_J)^s J} \in E$, следует:

$$\begin{cases} \exists\, s_7 \leq_\pi s'_6 : \pi \ni A^{2,3} : s'^{\,!e_7}_7 \to\ s_7,\ \text{где}\ e_7 = k_{AJ}(n^j_A, \bar{k}^i_A) \\ k_{AJ}((n^j_A)^s, \bar{k}^i_A) = k_{(a^i_J)^s J}(\bar{n}^j_J, (k_J)^s) \end{cases} \tag{105}$$

Из второй строки в (105) получаем:

$$A = (a^i_J)^s,\ (n^j_A)^s = \bar{n}^j_J,\ \bar{k}^i_A = (k_J)^s \tag{106}$$

Из первой строки в (105) получаем:

$$\exists\, s_8 \leq_\pi s'_7 : \pi \ni I^{1,2}_A : s'^{\,?e_8}_8 \to\ s_8,\ \text{где}\ e_8 = k_{AJ}(\bar{n}^i_A, \hat{n}^j_A). \tag{107}$$

По теореме 6, из (107), и того, что $e^s_8 \in [\circ]_{s_8}$, $k_{AJ} \in E$, следует:

$$\begin{cases} \exists\, s_9 \leq_\pi s'_8 : \pi \ni \grave{J}^{1,2} : s'^{\,!e_9}_9 \to\ s_9,\ \text{где}\ e_9 = k_{a^i_{\grave{J}} J}(n^i_{\grave{J}}, \bar{n}^j_{\grave{J}}) \\ k_{(a^i_{\grave{J}})^s J}((n^i_{\grave{J}})^s, \bar{n}^j_{\grave{J}}) = k_{AJ}(\bar{n}^i_A, (n^j_A)^s) \end{cases} \tag{108}$$

Из второй строки в (108) получаем:

$$(a^i_{\grave{J}})^s = A,\ (n^i_{\grave{J}})^s = \bar{n}^i_A,\ \bar{n}^j_{\grave{J}} = (n^j_A)^s \tag{109}$$

Из (106) и (109) получаем:

$$\bar{n}^j_{\grave{J}} = \bar{n}^j_J = (n^j_A)^s,\ \grave{J} = J,\ (a^i_J)^s = A,\ (n^i_J)^s = \bar{n}^i_A. \tag{110}$$

Из (100) и (106) получаем:

$$(k^i_B)^s = (k_J)^s = \bar{k}^i_A \in E, \tag{111}$$

поэтому по теореме 6, из первой строки в (98) и $e^s_1 \in [\circ]_{s_1}$ следует:

$$\begin{cases} \exists\, s_{11} \leq_\pi s'_1 : \pi \ni \grave{A}^{3,4} : s'^{\,!e_{11}}_{11} \to\ s_{11},\ \text{где}\ e_{11} = \bar{k}^i_{\grave{A}}(x^i_{\grave{A}}) \\ \bar{k}^i_{\grave{A}}(x^i_{\grave{A}}) = (k^i_B)^s((x^i_B)^s) \end{cases} \tag{112}$$

Из второй строки в (112) и (111) получаем:

$$\bar{k}^i_{\grave{A}} = (k^i_B)^s = \bar{k}^i_A,\ \grave{A} = A,\ x^i_A = (x^i_B)^s. \tag{113}$$

Из первой строки в (108) и (110) получаем:

$$\exists\, s_{10} \leq_\pi s'_9 : \pi \ni J^{0,1} : s'_{10} \overset{?e_{10}}{\rightarrow} s_{10},\ \text{где}\ e_{10} = k_{\hat{a}^i_J J}(\hat{a}^r_J, \hat{n}^i_J). \tag{114}$$

Из (100) и (110) следует, что $e^s_{10} = k_{AJ}(B, \bar{n}^i_A)$.
По теореме 6, из (114), $e^s_{10} \in [\circ]_{s_{10}}$, $k_{AJ} \in E$, следует:

$$\begin{cases} \exists\, s_{12} \leq_\pi s'_{10} : \pi \ni \grave{A}^{0,1} : s'_{12} \overset{!e_{12}}{\rightarrow} s_{12},\ \text{где}\ e_{12} = k_{\grave{A}J}(a^r_{\grave{A}}, \bar{n}^i_{\grave{A}}) \\ k_{\grave{A}J}(a^r_{\grave{A}}, \bar{n}^i_{\grave{A}}) = k_{AJ}(B, \bar{n}^i_A) \end{cases} \tag{115}$$

Из второй строки в (115) получаем:

$$\bar{n}^i_{\grave{A}} = \bar{n}^i_A,\ \grave{A} = A,\ a^r_A = B. \tag{116}$$

Утверждение (97) обосновывается следующим образом:

- $s \models at_{I_A} = 4$ следует из (112), (113): $s_{11} \models at_{\grave{A}} = 4$, $\grave{A} = A$, $s_{11} \leq_\pi s$,
- $s \models a^r_A = B$ следует из (116),
- $s \models a^i_B = A$ следует из (100) и (106),
- $s \models n^i_A = n^i_B$ следует из (103) и (110),
- $s \models k^i_A = k^i_B$ следует из (111),
- $s \models x^i_A = x^i_B$ следует из (113). ■

4. Заключение

В настоящей работе была построена новая модель КП, и показаны примеры ее использования для решения задач верификации свойств целостности, секретности и соответствия.

Для дальнейшей деятельности по развитию данной модели и основанных на ней методов верификации можно назвать следующие задачи:

- развитие языков спецификаций свойств КП, позволяющих выражать например свойства нулевого разглашения в КП аутентификации, свойства неотслеживаемости в КП электронных платежей, свойства анонимности и правильности подсчета голосов в КП электронного голосования, и разработка методов верификации свойств, выражаемых на этих языках,
- построение методов автоматизированного синтеза КП по описанию свойств, которым они должны удовлетворять.

Список литературы

[21CDS] Veronique Cortier, Stephanie Delaune, and Vaishnavi Sundararajan. A Decidable Class of Security Protocols for Both Reachability and Equivalence Properties. Journal of Automated Reasoning, 65:479–520, April 2021.

[21RCSSS] Roggenbach, M., Cerone, A., Schlingloff, H., Schneider, G., Shaikh, S.A., Formal verification of security protocols, in: Formal Methods for Software Engineering: Languages, Methods, Application Domains (Texts in Theoretical Computer Science. An EATCS Series) 1st ed., Springer International Publishing, 2021.

[17CW] Veronique Cortier and Cyrille Wiedling. A formal analysis of the Norwegian E-voting protocol. Journal of Computer Security, 25(15777):21–57, 2017.

[16ABF] M. Abadi, B. Blanchet, C. Fournet. The Applied Pi Calculus: Mobile Values, New Names, and Secure Communication. [Research Report] ArXiv. 2016, pp.110. `hal-01423924`, `https://arxiv.org/abs/1609.03003`

[16B] Bruno Blanchet, Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif, 2016.

[16YEMM] Fan Yang, Santiago Escobar, Catherine A Meadows, Jose Meseguer. Strand Spaces with Choice via a Process Algebra Semantics. PPDP '16: Proceedings of the 18th International Symposium on Principles and Practice of Declarative Programming, September 2016, pages 76–89.

[14CK] Veronique Cortier, Steve Kremer. Formal Models and Techniques for Analyzing Security Protocols: A Tutorial. Foundations and Trends in Programming Languages, 1(3):151–267, (2014)

[13LP] Yongjian Li, Jun Pang. An inductive approach to strand spaces. Formal Aspects of Computing, Vol. 25, No. 4, 2013.

[12CM] Cas Cremers, Sjouke Mauw. Operational Semantics and Verification of Security Protocols, Springer-Verlag Berlin Heidelberg, 2012.

[12G] Joshua D. Guttman. State and Progress in Strand Spaces: Proving Fair Exchange. Journal of Automated Reasoning, 48(2): 159-195, 2012.

[11CK] V. Cortier and S. Kremer, editors. Formal Models and Techniques for Analyzing Security Protocols, volume 5 of Cryptology and Information Security Series. IOS Press, 2011.

[11DMRS] A. Datta, J.C. Mitchell, A. Roy, S. Stiller, Protocol composition logic, in Formal Models and Techniques for Analyzing Security Protocols, ed. by V. Cortier, S. Kremer (IOS Press, Lansdale, 2011)

[11RS] Mark D. Ryan and Ben Smyth, Applied pi calculus, in: Formal Models and Techniques for Analyzing Security Protocols, Edited by Veronique Cortier, 2011 IOS Press, p. 112-142.

[08C] C.J.F. Cremers, On the protocol composition logic PCL, in ACM Symposium on Information, Computer & Communication Security (ASIACCS'08), ed. by M. Abe, V. Gligor, Tokyo, Japan (ACM, New York, 2008), pp. 66–76

[08CJSTW] Cervesato I., Jaggard A.D., Scedrov A., Tsay J.-K., Walstad C., Breaking and fixing public-key Kerberos, Information and Computation Volume 206, Issues 2-4, (2008), Pages 402-424.

[07ABF] M. Abadi, B. Blanchet, C. Fournet, Just Fast Keying in the Pi Calculus. In ACM Transactions on Information and System Security, 10(3), 2007.

[07DDMR] A. Datta, A. Derek, J.C. Mitchell, A. Roy, Protocol Composition Logic (PCL), in Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin, ed. by L. Cardelli, M. Fiore, G. Winskel. Electronic Notes in Theoretical Computer Science, vol. 172, (2007), pp. 311– 358

[07DGT1] S. Doghmi, J.D. Guttman, F.J. Thayer, Skeletons and the shapes of bundles, in 7th International Workshop on Issues in the Theory of Security (WITS'07), Braga, Portugal (2007)

[07DGT2] S.F. Doghmi, J.D. Guttman, F.J. Thayer, Skeletons, homomorphisms, and shapes: characterizing protocol executions, in 23rd Conference on the Mathematical Foundations of Programming Semantics (MFPS XXIII), New Orleans, USA. Electronic Notes in Theoretical Computer Science, vol. 173 (Elsevier, Amsterdam, 2007), pp. 85–102

[07DGT3] S.F. Doghmi, J.D. Guttman, F.J. Thayer, Searching for shapes in cryptographic protocols, in 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'07), ed. by O. Grumberg, M. Huth, Braga, Portugal. Lecture Notes in Computer Science, vol. 4424 (Springer, Berlin, 2007), pp. 523–537

[05AB] M. Abadi and B. Blanchet. Analyzing Security Protocols with Secrecy Types and Logic Programs. In Journal of the ACM, 52(1), pp. 102-146, 2005.

[05CDLMS] I. Cervesato, N. Durgin, P. Lincoln, J. Mitchell, A. Scedrov. A Comparison between Strand Spaces and Multiset Rewriting for Security Protocol Analysis. Journal of Computer Security, vol. 13, no. 2, pp. 265-316, 2005

[05KR] S. Kremer, M. Ryan. Analysis of an Electronic Voting Protocol in the Applied Pi Calculus. In 14th European Symposium on Programming (ESOP), pp. 186-200, 2005.

[02GT] J.D. Guttman, F.J. Thayer, Authentication tests and the structure of bundles. Theor. Comput. Sci. 283(2), 333–380 (2002)

[02SW] S.G. Stubblebine, R.N. Wright, An authentication logic with formal semantics supporting synchronization, revocation, and recency. IEEE Trans. Softw. Eng. 28(3), 256–285 (2002)

[01AF] M. Abadi, C. Fournet, Mobile values, new names, and secure communication, in 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'01), ed. by C. Hankin, D. Schmidt, London, UK (ACM, New York, 2001), pp. 104–115

[01B] B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In 14th IEEE Computer Security Foundations Workshop (CSFW), pp. 82-96, 2001.

[01DMP] N.A. Durgin, J.C. Mitchell, D. Pavlovic, A compositional logic for protocol correctness, in 14th IEEE Computer Security Foundations Workshop (CSFW'01), Cape Breton, Canada (IEEE Computer Society, Los Alamitos, 2001), pp. 241–272

[00AR] M. Abadi, P. Rogaway, Reconciling two views of cryptography (the computational soundness of formal encryption), in IFIP International Conference on Theoretical Computer Science (IFIP TCS'00), ed. by J. van Leeuwen, O. Watanabe, M. Hagiya, P.D. Mosses, T. Ito, Sendai, Japan (2000), pp. 3–22

[00B] G. Bella. Inductive Verification of Cryptographic Protocols. PhD thesis, Cambridge University, 2000.

[00GT2] J. D. Guttman and F. J. Thayer. Authentication tests and the normal, efficient penetrator. IEEE Computer Society Symposium on Research in Security and Privacy, 2000.

[00RSGLR] P.Y.A. Ryan, S.A. Schneider, M.H. Goldsmith, G. Lowe and A.W. Roscoe. The Modelling and Analysis of Security Protocols: the CSP Approach, Addison-Wesley, 2000.

[00RS] P. Y. A. Ryan and S. A. Schneider. Process algebra and non-interference. Journal of Computer Security, 2000.

[99AG] M. Abadi, A.D. Gordon, A calculus for cryptographic protocols: the Spi calculus. Inf. Comput. 148, 1–70 (1999)

[99P] L. C. Paulson. Inductive Analysis of the Internet Protocol TLS. In ACM Trans. on Information and System Security, 2(3), pp. 332-351, 1999.

[99THG1] F. J. Thayer, J. C. Herzog, and J. D. Guttman. Strand spaces: Proving security protocols correct. Journal of Computer Security, 7(2/3):191-230, 1999.

[99THG2] F.J. Thayer, J.C. Herzog, J.D. Guttman, Mixed Strand Spaces, in 12th IEEE Computer Security Foundations Workshop (CSFW'99), IEEE Computer Society, Los Alamitos, 1999, pp. 72–82

[98P] L.C. Paulson, The inductive approach to verifying cryptographic protocols. J. Comput. Secur. 6(1–2), 85–128 (1998)

[98THG1] F.J. Thayer, J.C. Herzog, J.D. Guttman, Honest ideals on Strand Spaces, in 11th IEEE Computer Security Foundations Workshop (CSFW'98), Rockport, USA (IEEE Computer Society, Los Alamitos, 1998), pp. 66–77

[98THG2] F. J. Thayer, J. C. Herzog, and J. D. Guttman. Strand spaces: why is a security protocol correct? IEEE Computer Society Symposium on Security and Privacy, 1998.

[98S] S. A. Schneider. Verifying authentication protocols in CSP. IEEE Transactions on Software Engineering, 1998.

[97DS] B. Dutertre and S. A. Schneider. Embedding CSP in PVS. An application to authentication protocols. Theorem proving in Higher Order Logics, number 1275 in LNCS. Springer, 1997.

[97LR] G. Lowe and A. W. Roscoe. Using CSP to detect errors in the TMN protocol. IEEE Transactions in Software Engineering, 23(10), 1997.

[97P] L.C. Paulson, Proving properties of security protocols by induction, in 10th IEEE Computer Security Foundations Workshop (CSFW'97), Rockport, Massachusetts (IEEE Computer Society, Los Alamitos, 1997), pp. 70–83

[96S] S. Schneider, Security properties and CSP, in 17th IEEE Symposium on Security & Privacy (S&P'96), Oakland, USA (IEEE Computer Society, Los Alamitos, 1996), pp. 174–187.

[96SvO] P.F. Syverson, P.C. van Oorschot, A unified cryptographic protocol logic. CHACS Report 5540-227 NRL (1996)

[96SS] S. A. Schneider and A. Sidiropoulos. CSP and anonymity. European Symposium on Research in Computer Security, 1996.

[95AN] R. Anderson and R. Needham. Programming Satan's computer. In J. van Leeuwen (ed.) Computer Science Today, volume 1000 of LNCS. Springer, 1995.

[95L] Gavin Lowe. An attack on the Needham-Schroeder public key authentication protocol. Information Processing Letters, 56(3):131–136, November 1995.

[94KMM] R.A. Kemmerer, C. Meadows, J.K. Millen, Three systems for cryptographic protocol analysis. J. Cryptol. 7, 79–130 (1994)

[93vO] P.C. van Oorschot, Extending cryptographic logics of belief to key agreement protocols, in 1st ACM Conference on Computer and Communications Security (ACM CCS'93), ed. by D.E. Denning, R. Pyle, R. Ganesan, R.S. Sandhu, V. Ashby, Fairfax, USA (ACM, New York, 1993), pp. 232–243

[93SM] Syverson P., Meadows C., A Logical Language for Specifying Cryptographic Protocol Requirements, Proceedings of the 1993 IEEE Computer Security Symposium on Security and Privacy, (1993) 165-177, IEEE Computer Society Press.

[91AT] M. Abadi, M. Tuttle, A semantics for a logic of authentication, in 10th ACM Symposium on Principles of Distributed Computing (PODC'91), Montreal, Canada (ACM, New York, 1991), pp. 201–216

[90BAN] Burrows M., Abadi M., Needham R., A Logic of Authentication. In ACM Transactions on Computer Systems, 8(1), (1990) 18-36.

[90GNY] L. Gong, R.M. Needham, R. Yahalom, Reasoning about belief in cryptographic protocol analysis, in 11th IEEE Symposium on Security & Privacy (S&P'90), Oakland, USA (IEEE Computer Society, Los Alamitos, 1990), pp. 234–248

[87NS] Needham R., Schroeder M., Authentication revisited, Operating Systems Review, Vol. 21, No. 1, (1987).

[85H] C. A. R. Hoare. Communicating Sequential Processes. Prentice-Hall, 1985.

[81DS] Denning D., Sacco G., Timestamps in Key Distribution Protocols, Communications of the ACM, Vol. 24, No. 8, (1981) 533-536.

[80M] R. Milner, A Calculus of Communicating Systems, Springer Verlag, 1980.

[78NS] Roger Needham and Michael Schroeder. Using encryption for authentification in large networks of computers. Communications of the ACM, 21(12), December 1978.

Mathematical model and methods of verification of cryptographic protocols

Mironov A.M.

In this paper, a new mathematical model of cryptographic protocols is presented, and examples of the application of this model for solving problems of verification of cryptographic protocols are given. Cryptographic protocols are distributed algorithms designed to enable the transmission of confidential information in an insecure environment. They are used, for example, in electronic payments, electronic voting procedures, systems for accessing confidential data, etc. Errors in cryptographic protocols can lead to great damage, therefore it is necessary to use mathematical methods to substantiate the various properties of correctness and security of cryptographic protocols. The paper outlines new methods for formal verification of cryptographic protocols. *Keywords:* cryptographic protocols, sequential processes, distributed processes, verification.

О выразимости кусочно-постоянных функций в пространстве кусочно-параллельных

А. Д. Отрощенко[1]

Для конечной системы кусочно-параллельных функций, реализуемых схемами из линейных элементов и функций Хэвисайда, дополненной всеми одноместными линейными функциями получен критерий выразимости кусочно-постоянных функций. Таким образом получен критерий выразимости бинарного классификатора, реализованного нейронной схемой МакКаллока-Питтса.

Ключевые слова: Кусочно-постоянная функция, кусочно-параллельная функция, проблема полноты, проблема выразимости, нейронные-схемы МакКаллока-Питтса.

1. Определение кусочно-параллельной функции

В соответствии с [2], мы рассматриваем класс PP кусочно-параллельных функций, которые строятся из линейных функций
$a_1x_1 + ... + a_nx_n + a_0 : R^n \to R, a_i \in R, i = 0, 1, ..., n, n \in \mathbb{N}$ и функции Хэвисайда
$\theta(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$ с использованием операций суперпозиции. Как показано в [2], функция f из PP может быть представлена в следующем виде: $f = f_L + f_{PC}$, где f_L -линейная функция, а f_{PC} - кусочно-постоянная функция. Будем обозначать $\vec{a} = (a_1, a_2, \ldots, a_n)$,
$(a, \vec{b}) = (a, b_1, b_2, \ldots, b_n)$,
$\langle \vec{a}, \vec{b} \rangle = \sum_{i=1}^{n} a_i b_i$,
$\vec{e}_i = (0, \ldots, 0, 1, \ldots, 0)$, $\vec{e}_0 = (1, 1, 1, \ldots, 1, 1)$
∂A граница множества A
В соответствии с [1], кусочно-параллельная функция имеет вид

$$f(\vec{x}) = \langle \vec{a}_0, \vec{x} \rangle + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(\langle \vec{a}_j, \vec{x} \rangle + c_j) = \sigma_{ij}) - k). \qquad (1)$$

[1] *Отрощенко Александр Дмитриевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: iskander.aka@mail.ru.

Otroschenko Alexander Dmitrievich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

где $\sigma_{ij} \in \{-1, 0, 1\}$, $\chi(A) = \begin{cases} 1, \text{условие } A \text{ выполнено} \\ 0, \text{условие } A \text{ не выполнено} \end{cases}$.
В дальнейшем, вместо $\sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(\langle \vec{a}_j, \vec{x} \rangle + c_j) = \sigma_{ij}) - k)$ мы будем писать $NL_f(\vec{x})$. Пусть $l = max|d_i|$. Иногда, мы будем писать $NL_f^l(\vec{x})$. Нижний индекс мы также иногда будем опускать, обозначая таким образом произвольную кусочно-постоянную функцию, максимум модуля которой не больше l. Также будем теперь обозначать $f_\epsilon(\vec{x}) = \epsilon f(\frac{\vec{x}}{\epsilon})$.
Будем называть множество значений аргумента соответствующую определенному d_i в дальнейшем носителем сигнатуры i, а сам d_i - сдвигом. Носитель сигнатуры, неограниченный хотя бы с одной стороны по каждой из координат, будем называть нограниченным. Плоскости разделяющие носители сигнатуры будем называть разрезами. Мы будем рассматривать дальше кусочно-параллельные функции с конечным числом сдвигов.
Обозначим множество функций с линейной частью, зависящей от не более чем одной переменной NLL_1. В соответствии с [3], это замкнутый предполный класс функций. С-финитно-линейные функции в соответствии с В. С. Половниковым [1], будем обозначать как FL. Напомним: Кусочно-параллельнаая функция $f(x_1, x_2, \ldots, x_n)$, такая, что $\forall a_i, b_i \exists C, A, B$, такие, что при $|t| > C : f(a_1 t + b_1, a_2 t + b_2, \ldots, a_n t + b_n) = At + B$ называется С-финитно-линейной. Класс С-финитно-линейных функций замкнут и предполон. Само утверждение, которое будет доказано, звучит следующим образом:

Теорема 1. *Пусть $U = L_1 \cup U_{add}$, где L_1 - все линейные одноместные функции, а U_{add} - конечное множество кусочно-параллельных. Замыкание U содержит все кусочно-постоянные функции тогда, и только тогда, когда $U \not\subseteq NLL_1, FL$*

2. Доказательство достаточности

2.1. Принадлежность функции Хэвисайда и суммы Хэвисайдов замыканию системы

В следующем доказательстве, мы фактически повторим часть доказательства полноты из [2], правда, для этого нам нужно будет его немного подправить.

Теорема 2. *Пусть $U \not\subseteq NLL_1, FL$. Тогда $\theta(x) \in [U]$*

Доказательство. Пусть $f \in U/NLL_1$.
Значит, в $f(\vec{x}) = \langle \vec{a}_0, \vec{x} \rangle + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(\langle \vec{a}_j, \vec{x} \rangle + c_j) = \sigma_{ij}) - k)$ линейная часть зависит от не менее чем двух переменных, поэтому у $\vec{a}_0$

есть две ненулевые компоненты. Пусть эти компоненты по первой и второй переменной. Для простоты, подставим во все остальные переменные ноль, в $x_1 = x/a_{01}$, $x_2 = y/a_{02}$ и получим $f_{01}(x, y)$. Итак

$$f_{01}(x,y) = x + y + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(\frac{a_{j1}}{a_{01}}x + \frac{a_{j2}}{a_{02}}y + c_j) = \sigma_{ij}) - k) =$$

$$= x + y + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(A_j x + B_j y + c_j) = \sigma_{ij}) - k) = x + y + NL_f^F.$$

Пусть $g \in U/FL$. Пусть $g = \langle \vec{g_l}, \vec{y} \rangle + NL_g^G(\vec{y})$

Рассмотрим

$$h(\vec{y}) = f_{01\epsilon}(g(\vec{y}), g_\delta(-\vec{y})) =$$

$$= \langle \vec{g_l}, \vec{y} \rangle + NL_g^G(\vec{y}) + \langle \vec{g_l}, -\vec{y} \rangle + \delta NL_g^G(-\frac{\vec{y}}{\delta}) + \epsilon NL_f^F(\frac{g(\vec{y})}{\epsilon}, \frac{g_\delta(-\vec{y})}{\epsilon}) =$$

$$= NL_g^G(\vec{y}) + NL^{\delta G + \epsilon F}(\vec{y})$$

Так как $g \notin FL$, то $\exists \vec{a}, \vec{c}, R, C_+ \neq C_-$, что при $t > R, g(\vec{a}t + \vec{c}) = \langle \vec{g_l}, \vec{a} \rangle t + C_+$, а при $t < -R, g(\vec{a}t + \vec{c}) = \langle \vec{g_l}, \vec{a} \rangle t + C_-$, т.е. $NL_g^G(\vec{a}t) = C_+$ при $t > R$, и $NL_g^G(\vec{a}t) = C_-$ при $t < -R$. Тогда, подберем $\delta > 0, \epsilon > 0$ так, чтобы

$$|C_+ - C_-| > 2(\delta G + \epsilon F),$$

подойдут $\delta_0 = \frac{|C_+ - C_-|}{5G}, \epsilon_0 = \frac{|C_+ - C_-|}{5F}$, таким образом,

$$2(\delta_0 G + \epsilon_0 F) = 2(\frac{|C_+ - C_-|}{5G}G + \frac{|C_+ - C_-|}{5F}F) = \frac{4}{5}|C_+ - C_-| < |C_+ - C_-|.$$

Теперь заметим, что $h(\vec{a}t + \vec{c}) = NL_g^G(\vec{a}t + \vec{c}) + NL^{\delta_0 G + \epsilon_0 F}(\vec{a}t + \vec{c})$. Пусть при $t > R_2, h(\vec{a}t + \vec{c}) = C_+^{'}$, а при $t < -R_2, h(\vec{a}t + \vec{c}) = C_-^{'}$, Тогда считая $t_1 > max(R, R_2)$, а $t_2 < -max(R, R_2)$ распишем

$$|C_+^{'} - C_-^{'}| = |h(\vec{a}t_1 + \vec{c}) - h(\vec{a}t_2 + \vec{c})| =$$

$$= |NL_g^G(\vec{a}t_1 + \vec{c}) + NL^{\delta_0 G + \epsilon_0 F}(\vec{a}t_1 + \vec{c}) - NL_g^G(\vec{a}t_2 + \vec{c}) - NL^{\delta_0 G + \epsilon_0 F}(\vec{a}t_2 + \vec{c})| =$$

$$= |C_+ - C_- + NL^{\delta_0 G + \epsilon_0 F}(\vec{a}t_1 + \vec{c}) - NL^{\delta_0 G + \epsilon_0 F}(\vec{a}t_2 + \vec{c})| \geq$$

$$\geq |C_+ - C_-| - |NL^{\delta_0 G + \epsilon_0 F}(\vec{a}t_1 + \vec{c}) - NL^{\delta_0 G + \epsilon_0 F}(\vec{a}t_2 + \vec{c})| \geq$$
$$\geq |C_+ - C_-| - |NL^{\delta_0 G + \epsilon_0 F}(\vec{a}t_1 + \vec{c})| - |NL^{\delta_0 G + \epsilon_0 F}(\vec{a}t_2 + \vec{c})| \geq$$
$$\geq |C_+ - C_-| - 2(\delta_0 G + \epsilon_0 F) = \frac{1}{5}|C_+ - C_-| > 0$$

Обозначим $p(t) = h(\vec{a}t + \vec{c})$. Таким образом, получили, что $p \in [U], p \in PC, p \notin FL$.
Пусть $p(t) = p_-$, при $t < -R$, $p(t) = p_+$, при $t > R$. $p_+ \neq p_-$ по доказанному выше. t_{min} - какая-либо точка в которой функция принимает минимальное значение p_{min}. t_{max} - какая-либо точка в которой функция принимает максимальное значение p_{max}. $p_{max} \neq p_{min}$ так, как

$$p_{max} \geq max(p_+, p_-) > min(p_+, p_-) \geq p_{min}.$$

Отсюда же следует, что $t_{max} \neq t_{min}$. Теперь рассмотрим следующую функцию.

$$p^{'}(t) = p(\frac{p(t) - p_-}{p_+ - p_-}(t_{max} - t_{min}) + t_{min})$$

Заметим, что при $t < -R$, то

$$p^{'}(t) = p(\frac{p(t) - p_-}{p_+ - p_-}(t_{max} - t_{min}) + t_{min}) =$$

$$= p(\frac{p_- - p_-}{p_+ - p_-}(t_{max} - t_{min}) + t_{min}) = p(t_{min}) = p_{min},$$

а при $t > R$,

$$p^{'}(t) = p(\frac{p(t) - p_-}{p_+ - p_-}(t_{max} - t_{min}) + t_{min}) =$$

$$= p(\frac{p_+ - p_-}{p_+ - p_-}(t_{max} - t_{min}) + t_{min}) = p(t_{max} - t_{min} + t_{min}) = p_{max}.$$

Теперь рассмотрим $p''(t) = \frac{p'(t) - p_{min}}{p_{max} - p_{min}}$. Ясно, что

$$p''(t) = \begin{cases} 1, t > t_+ \\ 0, t < t_- \\ [0,1], t_+ \geq t \geq t_- \end{cases},$$

также заметим, что у $p''(t)$ есть конечное число неустранимых точек разрыва. Рассмотрим самую правую(максимальную по x) из них - t_0. Заметим, что $p''(t_0 + \epsilon') = 1$, где $\epsilon' > 0$ выбрана достаточно малой, чтобы p'' была постоянной в интервалах $(t_0 - \epsilon', t_0)$ и $(t_0, t_0 + \epsilon')$. Это верно, т.к. мы рассматриваем самую правую неустранимую точку разрыва. Пусть

$p''(t_0) = p''_0$, а $p''(t_0 - \epsilon') = p''_-$. $p''_- < 1$, т.к. рассматриваемая точка разрыва - неустранима, а 1 - максимальное значение p''.
Рассмотрим $v(x,y) = f_{01\tau}(x,y) = x + y + NL^{\tau F}(x,y)$, пока не опредееляя τ. Теперь определим $a > 0, b, G > 0$ так, чтобы

$$q(t) = v(a(p''(t+t_0)+b), \frac{Gt}{\epsilon'}) = a(p''(t+t_0)+b) + \frac{Gt}{\epsilon'} + NL^{\tau F}(t)$$

при $t > 0$ было положительно, а при $t < 0$ отрицательно.
Положим пока $b > -1$. При $\epsilon > t > 0$

$$q(t) = a(p''(t+t_0)+b) + \frac{Gt}{\epsilon'} + NL^{\tau F}(t) = a(1+b) + \frac{Gt}{\epsilon'} + NL^{\tau F}(t) > a(1+b) - \tau F$$

Теперь наложим ограничение, что $a > \frac{\tau F}{1+b}$. Тогда при $\epsilon > t > 0$ $q(t) > a(1+b) - \tau F = q_+ > 0$.

При $t > \epsilon$

$$q(t) = a(p''(t+t_0)+b) + \frac{Gt}{\epsilon'} + NL^{\tau F}(t) > a(p''(t+t_0)+b) + G - \tau F$$

Теперь наложим ограничение, что $G > \tau F - a(\min_{t>0} p''(t+t_0)+b)$. Тогда при $t > \epsilon$ $q(t) > a(p''(t+t_0)+b) + G - \tau F = Q_+ > 0$.

Далее $p''_- < 1$, а значит интервал $(-1, -p''_-)$ непуст. Пусть $b \in (-1, -p''_-)$. При $-\epsilon' < t < 0$

$$q(t) = a(p''(t+t_0)+b) + \frac{Gt}{\epsilon'} + NL^{\tau F}(t) =$$

$$= a(p''_- + b) + \frac{Gt}{\epsilon'} + NL^{\tau F}(t) < a(p''_- + b) + \tau F$$

Теперь учтя, что $p''_- + b < 0$ наложим ограничение, что $a > \frac{\tau F}{p''_- + b}$.
Тогда при $\epsilon > t > 0$ имеем, что $q(t) < a(p''_- + b) - \tau F = q_- < 0$.

При $t \le -\epsilon'$

$$q(t) = a(p''(t+t_0)+b) + \frac{Gt}{\epsilon'} + NL^{\tau F}(t) =$$

$$= a(p''(t+t_0)+b) + \frac{Gt}{\epsilon'} + NL^{\tau F}(t) \le a(p''(t+t_0)+b) - G + \tau F$$

Теперь наложим ограничение, что $G > \tau F + \max_{t<0} a(p''(t+t_0)+b)$.
Тогда при $t \le -\epsilon$ имеем, что $q(t) < \max_{t<0} a(p''(t+t_0)+b) - G + \tau F = Q_- < 0$.
Пусть $q(0) = q_0$.

Выберем, $b = \frac{-1-p''_-}{2}$, $\tau = \frac{\epsilon'}{2F}$,
затем $a > max(\frac{\tau F}{1+b}, \frac{\tau F}{(p''_-+b)}) + 1 = \|\frac{\epsilon'}{1-2p''_-}\| + 1$, затем
$G > max(\tau F - a(\min_{t>0} p''(t+t_0) + b), \tau F + \max_{t<0} a(p''(t+t_0)+b))$.
Пусть

$$M = max(|\frac{t_+ - t_-}{q_-}|, |\frac{t_+ - t_-}{q_+}|, |\frac{t_+ - t_-}{Q_-}|, |\frac{t_+ - t_-}{Q_+}|, |\frac{t_+ - t_-}{q_0}|).$$

Тогда

$$\theta(x) = \begin{cases} p''(t_- - 1/2 + M(q(t) + 1/M)), q_0 > 0 \\ 1 - p''(t_- - 1/2 + M(q(t) + 1/M)), q_0 \leq 0 \end{cases}.$$

□

Теорема 3. *Пусть $U \not\subseteq NLL_1, FL$. Тогда $\forall n, d_i : 1 \leq i \leq n$ функция $\Theta_{d_1,d_2,\ldots,d_n}(\vec{x}) = \sum_{i=1}^{n} d_i\theta(x_i) \in [U]$*

Доказательство. По теореме 2 мы имеем, что $\theta(x) \in [U]$. Зафиксируем n и d_i. Пусть $f \in U/NLL_1$, то есть имеет линейную часть, зависящую хотя бы от двух входов. Подставим константы во все входы, так чтобы функция зависела только от двух переменных от которых бы и зависела линейная часть, т.е. $f'(x,y) = x + y + NL(x,y)$. Теперь

$$f(\vec{x}) = f'(x_1, f'(x_2, f(x_3, \ldots, f'(x_{n-2}, f'(x_{n-1}, x_n))\ldots))) = \sum_{i=1}^{n} x_i + NL(\vec{x}).$$

Подставим $x_i = \epsilon d_i\theta(y_i) + C_i$ и рассмотрим $h(\vec{y}) = f((\epsilon d_i\theta(y_i) + C_i))$, пока не определяя ϵ и C_i. Тогда

$$h(\vec{y}) = \sum_{i=0}^{n}(\epsilon d_i\theta(y_i) + C_i) +$$

$$+ \sum_{i=1}^{s} d_i\theta(\sum_{j=1}^{k} \chi(sgn(\langle \vec{a}_j, (\epsilon d_i\theta(y_i) + C_i)_{i=1}^{n} \rangle + c_j) = \sigma_{ij}) - k) =$$

$$= \epsilon \sum_{i=0}^{n} d_i\theta(y_i) + \sum_{i=0}^{n} C_i +$$

$$+ \sum_{i=1}^{s} d_i\theta(\sum_{j=1}^{k} \chi(sgn(\sum_{i=0}^{n} a_{ji}(\epsilon d_i\theta(y_i) + C_i) + c_j) = \sigma_{ij}) - k) =$$

$$= \epsilon \sum_{i=0}^{n} d_i\theta(y_i) + \sum_{i=0}^{n} C_i +$$

$$+\sum_{i=1}^{s} d_i\theta(\sum_{j=1}^{k}\chi(sgn(\epsilon\sum_{i=0}^{n} a_{ji}d_i\theta(y_i)+\sum_{i=0}^{n} a_{ji}C_i+c_j)=\sigma_{ij})-k)$$

Заметим, что мы можем выбрать C'_i так, чтобы $\forall j : \sum_{i=0}^{n} a_{ji}C'_i \neq 0$. Пусть тогда $M = 2\frac{\max\|c_j\|+1}{\min_j\|\sum_{i=0}^{n} a_{ji}C'_i\|}$. и выберем $_i = MC'_i$, а $\epsilon = \frac{1}{2\max_j\sum_{i=0}^{n}\|a_{ji}d_i\|}$. Тогда $\forall j$ имеем:

$$\|\sum_{i=0}^{n} a_{ji}C_i+c_j\| = \|2\frac{\max\|c_j\|+1}{\min_j\|\sum_{i=0}^{n} a_{ji}C'_i\|}\sum_{i=0}^{n} a_{ji}C'_i+c_j\| >$$

$$> \|2\frac{\max\|c_j\|+1}{\min_j\|\sum_{i=0}^{n} a_{ji}C'_i\|}\sum_{i=0}^{n} a_{ji}C'_i\| - \|c_j\| > \|2\max_j\|c_j\|+1\| - \|c_j\| \geq$$

$$\geq 2\max_j\|c_j\| - \|c_j\| + 2 \geq 2$$

с одной стороны, и

$$\|\epsilon\sum_{i=0}^{n} a_{ji}d_i\theta(y_i)\| = \|\frac{1}{2\max_j\sum_{i=0}^{n}\|a_{ji}d_i\|}\sum_{i=0}^{n} a_{ji}d_i\theta(y_i)\| \geq 1/2.$$

Отсюда следует, что $\forall j : sgn(\epsilon\sum_{i=0}^{n} a_{ji}d_i\theta(y_i)+\sum_{i=0}^{n} a_{ji}C_i+c_j) =$ $= sgn(\sum_{i=0}^{n} a_{ji}C_i+c_j)$ и не зависит от $\vec{y}$, а значит

$$\sum_{i=1}^{s} d_i\theta(\sum_{j=1}^{k}\chi(sgn(\epsilon\sum_{i=0}^{n} a_{ji}d_i\theta(y_i)+\sum_{i=0}^{n} a_{ji}C_i+c_j)=\sigma_{ij})-k) = S$$

и не зависит от $\vec{y}$. Тогда

$$\Theta_{d_1,d_2,\ldots,d_n}(\vec{y}) = \frac{1}{\epsilon}(h(\vec{y}) - \sum_{i=0}^{n} C_i - S) = \sum_{i=0}^{n} d_i\theta(y_i).$$

□

3. Различные вспомогательные утверждения

Лемма 1. *Пусть* $L_1 \cup \{\Theta_{d_1,d_2,\ldots,d_k}(\vec{y})|\forall k \in N, \vec{d} \in R^k\} \subseteq U$. *Тогда* $[U]$ *содержит все одноместные кусочно-постоянные функции.*

Доказательство. Так как $\chi(sgn(ax+c)=1) = 1-\theta(-ax-c)$, $\chi(sgn(ax+c)=-1) = 1-\theta(ax+c)$, $\chi(sgn(ax+c)=0) = 1-\theta(ax+c)-\theta(-ax-c)$, то очевидно, что $\sum_{j=1}^{k}\chi(sgn(a_jx+c_j)=\sigma_{ij}) = \Theta'_i(x)$ выражается через

Θ и одноместные линейные. Заметим теперь, что

$$\sum_{i=1}^{s} d_i\theta(\sum_{j=1}^{k}\chi(sgn(a_jx+c_j)=\sigma_{ij})-k) = \Theta_{d_1,d_2,\ldots,d_s}(\Theta'_1(x),\Theta'_2(x),\ldots,\Theta'_s(x)).$$

□

Теорема 4. *Пусть $U \not\subseteq NLL_1, FL$. Тогда $x+\theta(x) \in [U]$*

Доказательство. Как было показано при доказательстве прошлых теорем, $f_\tau(x,y) = x+y+NL_\tau(x,y) \in [U]$.
Пусть $g_\epsilon^C = f_\epsilon(x, C\theta(x)) - C/2$, при этом ϵ, τ, C определим позднее. Рассмотрим теперь $h(x) = f_\tau(g_\epsilon^C(x), NL_\mu(x) - M)$, одноместную кусочно-постоянную функцию $NL_\mu^F(x)$ мы определим также позднее, так как нам это будет удобно. Распишем подробнее:

$$h(x) = x + C\theta(x) + NL_\epsilon(x) - C/2 + NL_\mu^F(x) - M + NL^h(x),$$

где

$$NL^h(x) = \sum_{i=1}^{s} d_i\theta(\sum_{j=1}^{k}\chi(sgn(a_j(x+C\theta(x)-C/2+NL_\epsilon(x))+$$

$$+b_j(NL_\mu^F(x)-M)+c_j)=\sigma_{ij})-k) =$$

$$= \sum_{i=1}^{s} d_i\theta(\sum_{j=1}^{k}\chi(sgn(a_j(x+C\theta(x)-C/2)+NL_{\epsilon a_j+\mu b_j}(x)-b_jM+c_j)=\sigma_{ij})-k)$$

Так как, $NL^h(x) = NL_\tau(g_\epsilon^C(x), NL_\mu(x)-M)$, то ясно, что $\|NL^h(x)\| \le \tau$.
Положим $M = \max_{j:a_j=0}\|\frac{c_j}{b_j}\|+1$, $\epsilon < min(\min_{a_j\neq 0}\|\frac{1}{4a_j}\|, 1/4)$,
$\mu < \min(\min_{b_j\neq 0}\|\frac{1}{2b_j}\|, 1/2)$, $C = 2\max_{a_j\neq 0}\frac{\|\epsilon a_j\|+\|\mu b_j\|+\|b_jM\|+\|c_j\|}{\|a_j\|}$. Тогда получим, что при $a_j = 0$,

$$sgn(a_j(x+C\theta(x)-C/2)+NL_{\epsilon a_j+\mu b_j}(x)-b_jM+c_j) =$$

$$= sgn(NL_{\epsilon a_j+\mu b_j}(x)-b_jM+c_j) =$$

$$= sgn(NL_{\epsilon a_j+\mu b_j}(x) - b_j\max_{j:a_j=0}\|\frac{c_j}{b_j}\|+1+c_j) = 1$$

и не зависит от NL_μ^F, а при $a_j \neq 0$, по выбору констант
$\|NL_{\epsilon a_j+\mu b_j}(x) - b_jM + c_j\| < \|a_jC/2\|$. Тогда

$$sgn(a_j(x+C\theta(x)-C/2)+NL_{\epsilon a_j+\mu b_j}(x)-b_jM+c_j) = sgn(a_j(x+C\theta(x)-C/2))$$

и снова не зависит от выбора $NL^F_\mu(x)$. Теперь

$$h(x) = x + C\theta(x) + NL_{\epsilon+\tau}(x) - C/2 + NL^F_\mu(x) - M$$

и положив $\mu = min(\min_{a_j \neq 0} \| \frac{1}{4a_j} \|, 1/4, \min_{b_j \neq 0} \| \frac{1}{2b_j} \|)$, $\epsilon = \tau = \mu/3$, получим, что можно взять $NL^F_\mu(x) = -NL_{\epsilon+\tau}(x)$, и тогда т.к. $C > 0$ $H(x) = h(Cx)/C + M = x + \theta(x)$. □

Пусть $U \not\subseteq NLL_1, FL$. Тогда $[U]$ содержит все одноместные кусочно-постоянные функции.

Доказательство. Легко следует из 3 и 1. □

Лемма 2. *Пусть* $L_1 \cup \{\Theta_{d_1,d_2,\ldots,d_k}(\vec{y}) | \forall k \in N, \vec{d} \in R^k\} \subseteq U$. *Тогда если* $\theta(\sum_{i=1}^n x_i) \in [U]$, *то и все n-местные кусочно-постоянные функции принадлежат* $[U]$.

Доказательство. Заметим, что раз $\theta(\sum_{i=1}^n x_i) \in [U]$, то и $\theta(\sum_{i=1}^n a_i x_i + c_i) \in [U], \forall a_i, c$. Также по доказанному выше $\Theta_{\vec{d}}(\vec{y}) = \sum_{i=1}^n d_i \theta(y_i) \in [U]$ Теперь остается заметить, что произвольная кусочно-постоянная функция имеет вид $\sum_{i=1}^s d_i \theta(\sum_{j=1}^k \chi(sgn(\langle \vec{a}_j, \vec{x} \rangle + c_j) = \sigma_{ij}) - k)$. □

Пусть $U \not\subset NLL_1, FL$. Тогда если $\theta(\sum_{i=1}^n x_i) \in [U]$, то и все n-местные кусочно-постоянные функции принадлежат $[U]$.

Доказательство. Легко следует из 3 и 1. □

Лемма 3. *Пусть* $L_1 \cup \{\theta(x) + \theta(y)\} \subseteq U$, $G_i \subset R^p, 1 \leq i \leq k$ - *множества значений n-мерного параметра, такие, что* $\chi(\vec{x} \in G_i) \in [U]$. *Тогда* $\chi(\vec{x} \in \cup_{i=1}^k G_i) \in [U]$, $\chi(\vec{x} \in \cap_{i=1}^k G_i) \in [U]$, $\chi(\vec{x} \in R^n / G_i) \in [U]$.

Доказательство. Очевидно:
1) $\chi(\vec{x} \in R^n / G_i) = 1 - \chi(\vec{x} \in G_i)$
3)$\chi(\vec{x} \in G_1 \cap G_2) = \theta(\theta(\chi(\vec{x} \in G_1) - 0.5) + \theta(\chi(\vec{x} \in G_2) - 0.5) - 2)$.
3)$\chi(\vec{x} \in G_1 \cup G_2) = \theta(\theta(\chi(\vec{x} \in G_1) - 0.5) + \theta(\chi(\vec{x} \in G_2) - 0.5) - 0.5)$. □

Более того, мы можем заметить, что $\theta(f(\vec{x})) = \chi(\vec{x} : f(\vec{x}) \geq 0)$,
$1 - \theta(-f(\vec{x})) = \chi(\vec{x} : f(x) > 0)$, а $\chi(\vec{x} \in \cap_{i=1}^k G_i) = \prod_{i=1}^k \chi(\vec{x} \in G_i)$. Таким образом, если $\theta(f(x)) \in [U], \theta(g(x)) \in [U]$, то и $\theta(f(x))\theta(g(y)) \in [U]$. Отсюда легко следует, что если $f(x) \in [U] \cap PC, g(y) \in [U] \cap PC$, то и $f(x)g(y) \in [U] \cap PC$.
Также заметим, что:
$\theta(\theta(a - 0.5) + \theta(b - 0.5) - 2) = a \wedge b$,
$\theta(\theta(a - 0.5) + \theta(b - 0.5) - 0.5) = a \vee b$,
$1 - \theta(a - 0.5) = \neg a$ для $a, b \in \{0, 1\}$. Назовем множество $G \subset R^k$ выразимым в T, если
$\exists f : f \in [T], f(\vec{x}) = \chi(\vec{x} \in G)$.

Лемма 4. *Пусть* $L_1 \cup \{\theta(x) + \theta(y)\} \subseteq U$,*а* $1 \leq i \leq k, G_i$ *- выразимые множества значений n-мерного параметра. Тогда если* $F \in [A \setminus B, A \cup B, A \cap B]$, *то* $\chi(\vec{x} \in F(G_1, G_2, \ldots, G_k)) \in [U]$

Доказательство. Очевидно следует из леммы 3. □

Пусть $U \not\subseteq NLL_1, FL$, $G_i, 1 \leq i \leq k$ - выразимые множества значений n-мерного параметра. Тогда если $F \in [A \setminus B,$
$A \cup B, A \cap B]$, то $\chi(\vec{x} \in F(G_1, G_2, \ldots, G_k)) \in [U]$

Лемма 5. *Пусть функция* f *принимает бесконечное количество значений, а функция* g *- конечное* $C_1, C_2, C_3, \ldots, C_p$. *Тогда*

$$\theta(f+g) = \sum_{i=1}^{p} \theta(f + C_i)\theta(g - C_i)\theta(C_i - g)$$

Доказательство. Очевидно, что

$$\sum_{i=1}^{p} \theta(f + C_i)\theta(g - C_i)\theta(C_i - g) = \sum_{i=1}^{p} \theta(f + C_i)\chi(g = C_i) = \theta(f + g)$$

□

Лемма 6. *Функция* $\langle \vec{a}, \vec{x} \rangle + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(\langle \vec{a}_j, \vec{x} \rangle + c_j) = \sigma_{ij}) - k)$ *может быть записана в такой форме, что* $a_{1j} \in \{0, 1\}$.

Доказательство. Рассмотрим следующее преобразование: $\forall j : B_j > 0$ положим $\forall i, t : A'_{tj} = A_{tj}/B_j, c'_j = c_j/B_j B'_j = 1$, и $\forall j : B_j < 0 : \sigma'_{ij} = -\sigma_{ij}$, $\forall B_j > 0 : \sigma'_{ij} = \sigma_{ij}$, в случае же $B_j = 0$ положим $\forall i, t : A'_{tj} = A_{tj}, B'_j = 0, c'_j = c_j, \sigma'_{ij} = \sigma_{ij}$. Заметим, что после такого преобразования, $h(\vec{x}, y)$ не поменяется. Итак, будем считать, что все $B_j \geq 0$.

□

Лемма 7. *Пусть* n *и* k *местные* $f(\vec{x}), g(\vec{y})$ *имеют векторы нормалей к разделяющим плоскостям сигнатур* $\vec{a}_j, \vec{b}_j$ *соответственно, и*

$$f(\vec{x}) = \langle \vec{a}_0, \vec{x} \rangle + \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(\langle \vec{a}_j, \vec{x} \rangle + c_j) = \sigma_{ij}) - k),$$

$$g(\vec{y}) = \langle \vec{b}_0, \vec{x} \rangle + \sum_{i=1}^{s'} p_i \theta(\sum_{j=1}^{k'} \chi(sgn(\langle \vec{b}_j, \vec{y} \rangle + c'_j) = \sigma'_{ij}) - k'),$$

Тогда у $h(x_1, x_2, \ldots = f(x_1, x_2, \ldots, x_{n-1}, g(x_n, x_{n+1}, \ldots, x_{n+k-1}))$ *множество векторов нормалей к разделяющим плоскостям сигнатур -* $\{(a_{1j}, a_{2j}, \ldots, a_{(n-1)j}, a_{nj}\vec{b}_0) \forall j \neq 0\} \cup \{(0, 0, \ldots, 0, \vec{b}_i), \forall i \neq 0\}$ *и линейной частью* $(a_{10}, a_{20}, \ldots, a_{(n-1)0}, a_{n0}\vec{b}_0)$

Доказательство. Распишем h:

$$h(x_1, x_2, \ldots = f(x_1, x_2, \ldots, x_{n-1}, g(x_n, x_{n+1}, \ldots, x_{n+k-1})) =$$

$$= \sum_{i=1}^{n-1} a_{i0}x_i + a_{n0} \sum_{i=1}^{k} b_{i0}x_{i+n-1} +$$

$$+ \sum_{i=1}^{s'} p_i \theta(\sum_{j=1}^{k'} \chi(sgn(\sum_{l=1}^{k} b_{lj}x_{l+n-1} + c'_j) = \sigma'_{ij}) - k') +$$

$$+ \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(\sum_{l=1}^{n-1} a_{lj}x_l + a_{nj} \sum_{l=1}^{k} b_{l0}x_{l+n-1} +$$

$$+ c_j + \sum_{i'=1}^{s'} p_{i'} \theta(\sum_{j=1}^{k'} \chi(sgn(\sum_{l=1}^{k} b_{lj}x_{l+n-1} + c'_j) = \sigma'_{ij}) - k')) = \sigma_{ij}) - k)$$

С учетом 5 имеем:

$$NL_2(x_1, x_2, \ldots, x_{n+k-1}) = \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \chi(sgn(\sum_{l=1}^{n-1} a_{lj}x_l + a_{nj} \sum_{l=1}^{k} b_{l0}x_{l+n-1} +$$

$$+ c_j + \sum_{i'=1}^{s'} p_{i'} \theta(\sum_{j=1}^{k'} \chi(sgn(\sum_{l=1}^{k} b_{lj}x_{l+n-1} + c'_j) = \sigma'_{ij}) - k')) = \sigma_{ij}) - k) =$$

$$= \sum_{i=1}^{s} d_i \theta(\sum_{j=1}^{k} \sum_{i'=1}^{s'} \chi(sgn(\sum_{l=1}^{n-1} a_{lj}x_l + a_{nj} \sum_{l=1}^{k} b_{l0}x_{l+n-1} + c_j +$$

$$+ p_{i'}) = \sigma_{ij}) \theta(\sum_{j=1}^{k'} \chi(sgn(\sum_{l=1}^{k} b_{lj}x_{l+n-1} + c'_j) = \sigma'_{ij}) - k') - k),$$

Отсюда следует, что нелинейная часть g в нелинейной части f не создает плоскостей сигнатур с нормалями какими-либо еще нормалями к ним. □

Лемма 8. *Пусть $U \not\subseteq NLL_1$. Тогда $\forall n \geq 2$*
$\exists f_1(x_1, x_2, \ldots, x_n), f_2(x_1, x_2, \ldots, x_n), \ldots f_{n-1}(x_1, x_2, \ldots, x_n)$, такие, что их линейные части $L(\vec{x}) = \langle \vec{e}_0, \vec{x} \rangle$, а векторы нормалей не содержащие нулевых компонент к разделяющим плоскостям сигнатур f_i имеют вид

$$(\overbrace{1, 1, 1, \ldots, 1}^{i}, \overbrace{b_k, b_k, b_k, \ldots, b_k}^{n-i})$$

для некоторого набора $b_k \neq 0$(возможно пустого).

Доказательство. Итак, так как $U \not\subseteq NLL_1$, то по 6 $\exists h \in [U]$:
$h(x,y) = x+y+\sum_{i=1}^{s_1} d_i\theta(\sum_{j=1}^{k_1} \chi(sgn(a_jx+B_jy+c_j) = \sigma_{ij})-k), a_j \in \{0,1\}$.
Докажем индукцией по количеству входов n. Возьмем набором $b_k = B_j$:
$a_j \neq 0, b_j \neq 0$
При $n = 2$ утверждение доказано ($f_1(x_1,x_2) = h(x_1,x_2)$).
Пусть утверждение доказано при $n = k$. Тогда для $n = k+1$, рассмотрим

$$g_i(\vec{x}, x_{k+1}) = f_i(x_1, x_2, \ldots, x_{k-1}, h(x_k, x_{k+1})), i = 1 \ldots n-1$$

$$g_k(\vec{x}, x_{k+1}) = f_{k-1}(h(x_1,x_2), x_3, \ldots, x_{k-1}, x_k, x_{k+1})$$

Линейные части этих функций по 7 будут иметь вид $L(\vec{x}) = \langle \vec{e}_0, \vec{x} \rangle$, а векторы нормалей не содержащие нулевых компонент к разделяющим плоскостям сигнатур по 7 будут иметь вид для $g_i, i = 1 \ldots n-1$:
$(\overbrace{1,1,1,\ldots,1}^{i}, \overbrace{b_k, b_k, b_k, \ldots, b_k}^{n-i}, b_k)$,
а для g_k:
$(\overbrace{1,1,1,\ldots,1}^{k}, b_k)$. □

4. Завершение доказательства достаточности

Теорема 5. *Пусть $U \not\subseteq NLL_1, FL$. Тогда $PC \subset [U]$*

Доказательство. Будем вести доказательство теоремы по индукции по количеству входов. Пусть $U \not\subseteq NLL_1, FL$.
Для n=1 доказано в 1.
Для n=2 утверждение следует из теорем 4 и 5 публикации [3] (ее результат очевидным образом можно повторить пользуясь функциями $\sum_{i=1}^{p} d_i\theta(x_i)$ и $x + \theta(x)$) и 2.

Пусть построено для всех $k < n$. Ясно, что $H(y,\vec{x}) = y - \sum_{i=1}^{n-1} x_i +$
$+ \sum_{i=1}^{s} d_i\theta(\sum_{j=1}^{k} \chi(sgn(-\sum_{l=1}^{n-1} A_{lj}x_i + B_jy + c_j) = \sigma_{ij}) - k) \in [U]$.
В соответствии с 6 считаем, что $B_j \in \{0,1\}$. Зафиксируем $\epsilon_0 > 0$.
Зафиксируем $\vec{x}_0$ единичной длины, такой что $\forall j : B_j \neq 0$ выполнено $|\langle \vec{x}_0, \vec{e}_0 \rangle - \langle \vec{x}_0, \vec{A}_j \rangle| \geq \epsilon_0$, если $\vec{A}_j \neq \vec{e}_0$. Так мы зафиксировали компакт K_{ϵ_0}. Этот компакт - поверхность сферы, с вырезанными полосами вдоль конечного набора линий на сфере. Будем рассматривать $\vec{x} = t(\vec{x}_0 + \vec{\delta})$, $\|\vec{x}_0 + \vec{\delta}\|_{l_2} = 1$, $\|\vec{\delta}\|_{l_1} < \tau$, $\|\vec{x}\|_{l_1} > X_0$.
Запишем усиление условия $\vec{x} = t(\vec{x}_0 + \vec{\delta})$, $\|\vec{x}_0 + \vec{\delta}\|_{l_2} = 1$, $\|\vec{\delta}\|_{l_1} < \tau$ так, чтобы можно было выбрать новую τ усиливающую наше усиление. Рассмотрим набор из $2n-2$ векторов $V_{con} = (\vec{x}_0 + \gamma\vec{e}_1, \vec{x}_0 + \gamma\vec{e}_2, \ldots,$

$\vec{x}_0 + \gamma\vec{e}_{n-1}, \vec{x}_0 - \gamma\vec{e}_1, \vec{x}_0 - \gamma\vec{e}_2, \ldots, \vec{x}_0 - \gamma\vec{e}_{n-1})$ и "натянем"на них пирамиду без основания. Заметим, что условие принадлежности этой пирамиде через хар-функции может быть записано с помощью определителей, как

$$\chi^{\gamma}_{x_0}(\vec{x}) = \prod_{u=1}^{n-1} \chi(sgn(\begin{vmatrix} \vec{x} \\ V_{con,u} \\ V_{con,u+1} \\ \ldots \\ V_{con,u+n-3} \end{vmatrix}) = sgn(\begin{vmatrix} \vec{x}_0 \\ V_{con,u} \\ V_{con,u+1} \\ \ldots \\ V_{con,u+n-3} \end{vmatrix})) \in [U],$$

считая, что если $g > 2n-2$, то $V_{con,g} = V_{con,g \mod (2n-1)+1}$.
Для условия $\|\vec{x}\|_{l_1} > X_0$ запись его усиления в виде характеристической функции примет вид

$$\chi_{|X_0|}(\vec{x}) = \prod_{i=1}^{n-1} \chi(x_i : x_i < -X_0 \vee x_i > X_0) \in [U].$$

Запишем $H(y, \vec{x})$ в следующей форме:

$$H(y, \vec{x}) = y - t(R_0 + \langle\vec{\delta}, \vec{e}_0\rangle) + \sum_{i=1}^{s} d_i \prod_{j=1}^{k} \chi_{ij}(B_j y - t(R_j + \langle\vec{A}_j, \vec{\delta}\rangle) + c_j),$$

где $R_0 = \langle x_0, e_0\rangle$, $R_j = \langle x_0, \vec{A}_j\rangle$, а $\chi_{ij}(f) = \chi(sgn(f) = \sigma_{ij})$.
Для $j : B_j = 0$ положим

$$\chi^{-1}_j(\vec{x}) = \chi(-\sum_{t=1}^{n-1} A_{tj}x_i + c_j < 0) \in [U],$$

$$\chi^{0}_j(\vec{x}) = \chi(-\sum_{t=1}^{n-1} A_{tj}x_i + c_j = 0) \in [U],$$

$$\chi^{1}_j(\vec{x}) = \chi(-\sum_{t=1}^{n-1} A_{tj}x_i + c_j > 0) \in [U].$$

Заметим, что если для множества $\vec{x}$ выполнено, что $\chi^s_j(\vec{x}) = 1$, то на нем $\chi_{ij}(B_j y - t(R_j + \langle\vec{A}_j, \vec{\delta}\rangle) + c_j) = V_{ij}$, т.е. константе.

Пусть $B_j \neq 0$.
Для $j : \exists i : A_{ij} = 0$ положим

$$\chi^{-1}_j(\vec{x}) = \chi(-\sum_{t=1, t\neq i}^{n-1} A_{tj}x_i + c_j < 0) \in [U],$$

$$\chi_j^0(\vec{x}) = \chi(-\sum_{t=1,t\neq i}^{n-1} A_{tj}x_i + c_j = 0) \in [U],$$

$$\chi_j^1(\vec{x}) = \chi(-\sum_{t=1,t\neq i}^{n-1} A_{tj}x_i + c_j > 0) \in [U].$$

Заметим, что если для множества $\vec{x}$ выполнено, что $\chi_j^s(\vec{x}) = 1$, то на нем
$\chi_{ij}(B_j y - t(R_j + \langle \vec{A}_j, \vec{\delta}\rangle) + c_j) = V_{ij}$, т.е. константе.
Пусть $B_j \neq 0, A_{ij} \neq 0$.
По ограничению на $\vec{x}_0$ $R_j \neq R_0$. Тогда:
1)Если $R_j > R_0$, то $\exists \tau > 0 : R_j - \langle \vec{A}_j, \vec{\delta}\rangle > R_0 + \langle \vec{\delta}, \vec{e}_0\rangle$, а $\exists t_j$, что при $t > t_j$, из того, что $y - t(R_j + \langle \vec{\delta}, \vec{e}_0\rangle) < D$, будет следовать,
$\forall i : y - t(R_0 + \langle \vec{\delta}, \vec{e}_0\rangle) + d_i < D$. Значит, при $t > t_j$, когда $\chi_{ij}(y - t(R_j + \langle \vec{A}_j, \vec{\delta}\rangle) + c_j)$ меняет своё значение, $\chi(H(y, t(\vec{x}_0 + \delta)) > D)$ своего значения не меняет. Значит $\exists V_{ij}$, что при $t > t_j$

$$\chi(H(y, t(\vec{x}_0 + \delta)) > D) = \chi(y - t(R_0 + \langle \vec{\delta}, \vec{e}_0\rangle) +$$

$$+ \sum_{i=1}^{s} d_i V_{ij} \prod_{l=1,l\neq j}^{k} \chi_{il}(B_l y - t(R_l + \langle \vec{A}_l, \vec{\delta}\rangle) + c_l)).$$

2)Если $R_j < R_0$, то $\exists \tau > 0 : R_j - \langle \vec{A}_j, \vec{\delta}\rangle < \langle \vec{x}_0 + \vec{\delta}, \vec{e}_0\rangle$, а значит при $t > t_j$. из того, что $y - t\langle \vec{x}_0 + \delta, \vec{A}_j\rangle > D$, будет следовать, $\forall i : y - t\langle \vec{x}_0 + \delta, \vec{e}_0\rangle + d_i > D$, , а значит $\chi(H(y, t(\vec{x}_0 + \delta)) > D)$ при $t > t_j, \forall i$ не зависит от χ_{ij}. Значит, при $t > t_j$, когда $\chi_{ij}(y - t(R_j + \langle \vec{A}_j, \vec{\delta}\rangle) + c_j)$ меняет своё значение, $\chi(H(y, t(\vec{x}_0 + \delta)) > D)$ своего значения не меняет. Значит $\exists V_{ij}$, что при $t > t_j$

$$\chi(H(y, t(\vec{x}_0 + \delta)) > D) = \chi(y - t(R_0 + \langle \vec{\delta}, \vec{e}_0\rangle) +$$

$$+ \sum_{i=1}^{s} d_i V_{ij} \prod_{l=1,l\neq j}^{k} \chi_{il}(B_l y - t(R_l + \langle \vec{A}_l, \vec{\delta}\rangle) + c_l)).$$

3) Отдельно рассмотрим случай, когда $\vec{A}_j = \vec{e}_0$. Положим в этом случае $D > -\min_j c_j + \max_i d_i + 1$. Тогда из $H(y, \vec{x}) > D$, будет следовать, что $y - \sum_{t=1}^{n-1} x_i + NL_{d_i}(y, \vec{x}) > -\min_j c_j + \max_i d_i + 1$, а значит
$y - t(R_j + \langle \vec{A}_j, \delta\rangle) + c_j > (c_j - \min_j c_j) + (\max_i d_i - NL_{d_i}(y, \vec{x})) + 1$ т.е.
$y - t(R_j + \langle \vec{A}_j, \delta\rangle) + c_j > 0$.
Тогда $\chi_{ij}(B_j y - t(R_j + \langle \vec{A}_j, \vec{\delta}\rangle) + c_j) = V_{ij}$, т.е. константе.

Тогда по вышесказанному, при наложенных нами ограничениях

$$\sum_{i=1}^{s} d_i \prod_{j=1}^{k} \chi_{ij}(B_j y - t(R_j + \langle \vec{A}_j, \vec{\delta} \rangle) + c_j) =$$

$$= \sum_{i=1}^{s} d_i \prod_{(B_j=0 \vee \exists t: A_{tj}=0) j=1}^{k} \chi_{ij}(B_j y - t(R_j + \langle \vec{A}_j, \vec{\delta} \rangle) + c_j) *$$

$$* \prod_{(B_j \neq 0 \wedge \forall t: A_{tj} \neq 0), j=1}^{k} \chi_{ij}(B_j y - t(R_j + \langle \vec{A}_j, \vec{\delta} \rangle) + c_j) =$$

$$= \sum_{i=1}^{s} d_i \prod_{(B_j=0 \vee \exists t: A_{tj}=0), j=1}^{k} V_{ij} *$$

$$* \prod_{(B_j \neq 0 \wedge \forall t: A_{tj} \neq 0), j=1}^{k} \chi_{ij}(B_j y - t(R_j + \langle \vec{A}_j, \vec{\delta} \rangle) + c_j) =$$

$$= \sum_{i=1}^{s} d_i^1 \prod_{(B_j \neq 0 \wedge \forall t: A_{tj} \neq 0), j=1}^{k} \chi_{ij}(B_j y - t(R_j + \langle \vec{A}_j, \vec{\delta} \rangle) + c_j) =$$

$$= \sum_{i=1}^{s} d_i^1 \prod_{(B_j \neq 0 \wedge \forall t: A_{tj} \neq 0), R_j < R_0, j=1}^{k} \chi_{ij}(B_j y - t(R_j + \langle \vec{A}_j, \vec{\delta} \rangle) + c_j) *$$

$$* \prod_{(B_j \neq 0 \wedge \forall t: A_{tj} \neq 0), R_j > R_0, j=1}^{k} \chi_{ij}(B_j y - t(R_j + \langle \vec{A}_j, \vec{\delta} \rangle) + c_j) =$$

$$* \prod_{(B_j \neq 0, \vec{A}_j = \vec{e}_0), j=1}^{k} \chi_{ij}(B_j y - t(R_j + \langle \vec{A}_j, \vec{\delta} \rangle) + c_j) =$$

$$= \sum_{i=1}^{s} d_i^1 \prod_{(B_j \neq 0 \wedge \forall t: A_{tj} \neq 0), R_j < R_0, j=1}^{k} V_{ij} \prod_{(B_j \neq 0 \wedge \forall t: A_{tj} \neq 0), R_j > R_0, j=1}^{k} V_{ij} *$$

$$* \prod_{(B_j \neq 0, \vec{A}_j = \vec{e}_0), j=1}^{k} V_{ij} = Const$$

Добавим начальные ограничения на малость, и обозначим $G_{x_0}^{s_1,s_2,\ldots,s_k} = \{\vec{x} | \chi_{sect}(\vec{x}) \chi_{|X_0|}(\vec{x}) \prod_{u=1}^{k} \chi_q^{s_q}(\vec{x}) = 1\}$.

$$f_{x_0}^{s_1,s_2,\ldots,s_k}(y, \vec{x}) = \chi(y - \langle \vec{x}_0, \vec{e}_0 \rangle + d_{x_0}^{s_1,s_2,\ldots,s_k} > D) *$$

$$*\chi_{x_0}(\vec{x})\chi_{|X_0|}(\vec{x})\prod_{u=1}^{k}\chi_q^{s_q}(\vec{x})\in[U],$$

а значит

$$g'_{x_0}(y,\vec{x})=\chi(\vec{x}\in\{\vec{x}|\exists(s_1,s_2,\ldots,s_k):f_{x_0}^{s_1,s_2,\ldots,s_k}(y+d_{x_0}^{s_1,s_2,\ldots,s_k},\vec{x})=1\})=$$

$$=\chi(y-\langle\vec{x}_0,\vec{e}_0\rangle>D)\chi(\vec{x}\in\cup_{s_1,s_2,\ldots,s_k}G_{x_0}^{s_1,s_2,\ldots,s_k})=$$

$$=\chi(y-\langle\vec{x}_0,\vec{e}_0\rangle>D)\chi_{x_0}(\vec{x})\chi_{|X_0|}(\vec{x})\in[U]$$

Тогда пусть $g_{x_0}(y-D,\vec{x})=g'_{x_0}(y+D,\vec{x})=$
$=\chi(y-\langle\vec{x}_0,\vec{e}_0\rangle>0)\chi_{x_0}(\vec{x})\chi_{|X_0|}(\vec{x})$.

Теперь заметим, что при проекции на единичную сферу с центром в нуле множеств $\{\vec{x}:\chi_{x_0}(\vec{x})\chi_{|X_0|}(\vec{x})=1\}$ мы можем выбрать внутри этих проекций на сфере малые окрестности, $O_\phi(\vec{x_0})$, и эти окрестности образуют открытое покрытие K_{ϵ_0}. Выберем конечное подпокрытие этого компакта $(O_{\phi_1}(\vec{x_1}),O_{\phi_2}(\vec{x_2}),\ldots,O_{\phi_a}(\vec{x_a}))$. Обозначим $X_0(g_{x_a})$ константу, которой ограничен модуль аргумента функции g_{x_a}. Тогда получим, что

$$W(y,\vec{x})=\chi(\vec{x}\in\{\vec{x}|\cup_{u=1}^{a}\chi(g_{x_u}(\vec{x})=1)\})\chi(\vec{x}>\max_a X_0(g_{x_a}))=$$

$$=\theta(\sum_{u=1}^{a}\chi(g_{x_u}(\vec{x})=1)-0.5)\chi(\vec{x}>\max_a X_0(g_{x_a}))=$$

$$=\theta(\sum_{u=1}^{a}\chi(y-\langle\vec{x},\vec{e}_0\rangle>0)\chi_{x_u}(\vec{x})\chi_{|X_a|}(\vec{x})-0.5)\chi(\|\vec{x}\|_{l_1}>\max_a X_0(g_{x_a}))=$$

$$=\chi(y-\langle\vec{x},\vec{e}_0\rangle>0)\theta(\sum_{u=1}^{a}\chi_{x_u}(\vec{x})\chi_{|X_a|}(\vec{x})-0.5)\chi(\|\vec{x}\|_{l_1}>\max_a X_0(g_{x_a}))\in[U],$$

но так как проекции бесконечных конусов покрыли K_{ϵ_0}, то

$$W(y,\vec{x})=\chi(y-\langle\vec{x},\vec{e}_0\rangle>0)\theta(\sum_{u=1}^{a}\chi_{|X_u|}(\vec{x})-0.5)\chi(\|\vec{x}\|_{l_1}>\max_a X_0(g_{x_a}))=$$

$$=\chi(y-\langle\vec{x},\vec{e}_0\rangle>0)\chi_{\max_a X_0(g_{x_a})}(\vec{x})*$$

$$*\chi(\vec{x}:|\langle\vec{x}_0,\vec{e}_0\rangle-\langle\vec{x}_0,\vec{A}_j\rangle|\geq\epsilon_0,A_{ij}\neq0,\vec{A}_j\neq\vec{e}_0).$$

Итак, мы научились строить для произвольной $H(y,\vec{x})$ функцию $W(y,\vec{x})$. Обозначим $X_W=\max_a X_0(g_{x_a})$. Построим таким образом W_i, для каждой f_i из 8.

Если при этом набор соответствующих b_i пуст или $\{b_i\} = \{1\}$, то $\forall i$ множитель
$\chi(\vec{x} : |\langle \vec{x}_0, \vec{e}_0\rangle - \langle \vec{x}_0, \vec{A}_j\rangle| \geq \epsilon_0, A_{ij} \neq 0, \vec{A}_j \neq \vec{e}_0)$ у соответствующей W_i равен единице (так как функция из которых она строилась не имеет нормалей к сигнатурам с ненулевыми компонентами и при этом таких, что $\vec{A}_j \neq \vec{e}_0$.

Иначе объединим множества, на которых $W_i = 1$, и домножим на $\chi_{\max_{W_i} X_{W_i}}(\vec{x})$) то есть рассмотрим

$$W_r(y, \vec{x}) = (\vee_{i=1}^{n-1} W_i(y, \vec{x})) \wedge \chi_{\max_{W_i} X_{W_i}}(\vec{x})) =$$

$$= \chi(y - \langle \vec{x}, \vec{e}_0\rangle > D)\chi_{\max_{W_i} X_{W_i}}(\vec{x}))*$$

$$* \vee_{i=1}^{n-1} \chi(\vec{x} : |\langle \vec{x}_0, \vec{e}_0\rangle - \langle \vec{x}_0, \vec{A}_j^{W_i}\rangle| \geq \epsilon_0, A_{lj}^{W_i} \neq 0, \vec{A}_j^{W_i} \neq \vec{e}_0) =$$

$$= \chi(y - \langle \vec{x}, \vec{e}_0\rangle > D)\chi_{\max_{W_i} X_{W_i}}(\vec{x}))*$$

$$*(1 - \wedge_{i=1}^{n-1} \chi(\vec{x} : |\langle \vec{x}_0, \vec{e}_0 - \vec{A}_j^{W_i}\rangle| < \epsilon_0, A_{lj}^{W_i} \neq 0, \vec{A}_j^{W_i} \neq \vec{e}_0))$$

По 8 и так как $\vec{A}_j^{W_i} \neq \vec{e}_0$, то $\vec{A}_j^{W_i}$ у имеют вид
$(\overbrace{1, 1, 1, \ldots, 1}^{i-1}, \underbrace{b_k, b_k, b_k, \ldots, b_k}^{n-i}), b_k \notin \{0, 1\}$.
Найдем $\vec{x}$ для которых выполнено
$\wedge_{i=1}^{n-1} \chi(\vec{x} : |\langle \vec{x}_0, \vec{e}_0 - \vec{A}_j^{W_i}\rangle| < \epsilon_0, A_{lj}^{W_i} \neq 0, \vec{A}_j^{W_i} \neq \vec{e}_0) = 1$. Для этого перепишем в виде набора неравенств вида:

$$\left\| \begin{pmatrix} b_{i_1} - 1, b_{i_1} - 1, b_{i_1} - 1, \ldots, b_{i_1} - 1 \\ 0, b_{i_2} - 1, b_{i_2} - 1, b_{i_2} - 1, \ldots, b_{i_2} - 1 \\ 0, 0, b_{i_3} - 1, b_{i_3} - 1, \ldots, b_{i_3} - 1 \\ \ldots \\ \overbrace{0, 0, 0, \ldots, 0}^{k-1}, \overbrace{b_{i_k} - 1, b_{i_k} - 1, b_{i_k} - 1, \ldots, b_{i_k} - 1}^{n-i} \\ \ldots \\ 0, 0, 0, \ldots, 0, b_{i_{n-1}} - 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \ldots \\ x_k \\ \ldots \\ x_{n-1} \end{pmatrix} \right\| < \epsilon_0 \quad (2)$$

Заметим, что матрица квадратная, верхнетреугольная и невырожденная ($b_i \neq 1$). Значит $\|\vec{x}_0\| < J\epsilon_0$, для некоторого $J > 0$. Возьмем $\epsilon_0 = \frac{1}{2J}$. Тогда $\|\vec{x}_0\| < 1/2$. Но $\|\vec{x}_0\| = 1$. Значит нет $\vec{x}_0$, удовлетворяющих хоть какой-либо из этих систем. Отсюда следует, что

$$\wedge_{i=1}^{n-1} \chi(\vec{x} : |\langle \vec{x}_0, \vec{e}_0 - \vec{A}_j^{W_i}\rangle| < \epsilon_0, A_{lj}^{W_i} \neq 0, \vec{A}_j^{W_i} \neq \vec{e}_0) = 0.$$

Теперь рассмотрим характеристическую функцию от объедининения множества для которого W_r - характеристическая функция, и этого же

множества полученного сдвигом вдоль какого-нибудь вектора, лежащего в $y-\sum_{i=1}^{n-1} x_i = 0$, например вдоль $(n, \vec{e_0})$. Заметим тогда, что для любого $\vec{x}$ или $\chi(\|\vec{x}\|_{l_1} > \max_{W_i} X_{W_i}) = 1$ или $\chi(\|\vec{x} - A\vec{e_0}\|_{l_1} > \max_{W_i} X_{W_i}) = 1$, где $A = \max_{W_i} X_{W_i}$. Тогда

$$1 - \theta(y - \sum_{i=1}^{n-1} x_i) = \theta(W_r(y, \vec{x}) + W_r(y + An, \vec{x} - A\vec{e_0}) - 0.5)$$

Далее из 2 следует, что и все n-местные кусочно-постоянные принадлежат $[U]$.

□

5. Доказательство необходимости

Обозначим PC_n - все кусочно-постоянные функции зависящие от n входов.

Теорема 6. *Пусть* $U = L_1 \cup PC_N$. *Тогда* $\theta(\sum_{i=1}^{N+1} x_i) \notin [U]$

Доказательство. Заметим, что $PC_N \subset [L_1 \cup \cup_n \{\sum_i^n d_i\theta(x_i)\} \cup \{\theta(\sum_{i=1}^N x_i)\}]$. Предположим, что $\theta(\sum_{i=1}^{N+1} x_i)$ принадлежит замыканию. Тогда возьмем формулу над $L_1 \cup \{\cup_n \sum_i^n \theta(x_i), \theta(\sum_{i=1}^N x_i)\}$, и проводя в ней преобразования 5 с каждым выражением вида $\sum a_i x_i + \sum d_i\theta(f_i(\vec{x})$ получим, что

$$\theta(\sum_{i=1}^{N+1} x_i) = F(\theta(\sum_{i=1}^{N} \lambda_{1i_j} x_{i_{j_1}} + c_1), \theta(\sum_{i=1}^{N} \lambda_{2i_j} x_{i_{j_2}} + c_2), \ldots, \theta(\sum_{i=1}^{N} \lambda_{pi_j} x_{i_{j_p}} + c_p)),$$

т.к. количество непрерывных слагаемых в сумматорах при применении этого преобразования не вырастает, а количество сумм вида $\sum a_i x_i + \sum d_i\theta(f_i(\vec{x})$ после каждого такого преобразования внутри формулы уменьшается, а всего их конечно.
Пусть $I_k(\vec{x}) = \theta(\sum_{i=1}^N \lambda_{ki_{j_k}} x_{i_{j_k}} + c_k)$. Так как длина каждой последовательности i_{j_k} равна N, то $\forall k \exists l \in \{1, 2, \ldots, N, N+1\} : i_{j_k} \neq l$. Заметим, тогда что тогда $I_k(\vec{x}) = I_k(\vec{x} + R\vec{e}_l)$. Обозначим $G_k = \{\vec{x} | G_k(\vec{x}) = 1\}$. Заметим, что для ∂G_k верно, что если $\vec{x} \in \partial G_k$, то и $\vec{x} + R\vec{e}_{l(k)} \in \partial G_k$.
Заметим, что F принимает на вход единицы и нули, и ее значение единица или ноль, и мы ей можем сопоставить логическую функцию, то есть

$$\theta(\sum_{i=1}^{N+1} x_i) = F(\chi(\vec{x} \in G_1), \chi(\vec{x} \in G_2), \ldots, \chi(\vec{x} \in G_p))$$

Теперь преобразуем логическое выражение в выражение "пересечений и объединений"в соответствии с 4.

$$\theta(\sum_{i=1}^{N+1} x_i) = \chi(\vec{x} \in F_{/,\cup,\cap}(G_1, G_2, \ldots, G_p))$$

Заметим, что для почти всех точек $\partial F_{/,\cup,\cap}(G_1, G_2, \ldots, G_p)$, кроме множества точек нулевой меры(относительно меры μ на N-мерных объемах, т.е. меры на границе), выполнено, что $\exists \vec{x} \in \partial F_{/,\cup,\cap}(G_1, G_2, \ldots, G_p)$, такой, что $\exists \epsilon > 0, l : \vec{x} + R\vec{e}_l \in \partial F_{/,\cup,\cap}(G_1, G_2, \ldots, G_p), \forall |R| < \epsilon$. Это свойство выполнено, для начальных G_i, которые являются бесконечными многогранниками и продолжает выполняться, при конечном объединении, пересечении и взятии дополнения этих многогранников, так как грани получающихся при таком процессе многогранников - конечные объединения и пересечения подмножеств ∂G_i с границами в виде ломаных, и каждая грань получающегося многогранника - какое-то подмножество одной из граней G_i, при этом, если мера $\mu(\partial G_k \cap \partial F_{/,\cup,\cap}(G_1, G_2, \ldots, G_p)) > 0$, то свойство выше выполнено для всех $\vec{x} \in \partial G_k \cap \partial F_{/,\cup,\cap}(G_1, G_2, \ldots, G_p)$ кроме точек границы этого пересечения (в смысле границы на $\partial F_{/,\cup,\cap}(G_1, G_2, \ldots, G_p)$). Мера таких точек относительно μ - нулевая. Но свойство выше, очевидно, не выполнено для $\partial\{\vec{x} | \theta(\sum_{i=1}^{N+1} x_i) = 1\}$, так как если для $\vec{x} : \sum_{i=1}^{N+1} x_i = \langle \vec{e}_0, \vec{x} \rangle = 0$, то $\forall R \neq 0, l : \langle \vec{e}_0, \vec{x} + R\vec{e}_l \rangle = \langle \vec{e}_0, \vec{x} \rangle + \langle \vec{e}_0, R\vec{e}_l \rangle = R\langle \vec{e}_0, \vec{e}_l \rangle = R \neq 0$, а значит $\forall R \neq 0, l : \vec{x} + R\vec{e}_l \notin \partial\{\vec{x} | \theta(\sum_{i=1}^{N+1} x_i) = 1\}$. Противоречие. □

Обозначим $NLL_{1,n} = [L_1 \cup \{x + NL(x, \vec{y}), NL(x, \vec{y}) \in PC_n\}]$.

Теорема 7. *Пусть $U = L_1 \cup NLL_{1,n}$. Тогда $\theta(\sum_{i=1}^{N+1} x_i) \notin [U]$*

Доказательство. Заметим, что функция с линеным входом в формуле, должна лежать внутри θ после чего работает предыдущее доказательство. □

5.1. Основной результат

Теорема 8. *Пусть U содержит все одноместные линейные и ещё конечное число кусочно-параллельных . Замыкание U содержит все кусочно-постоянные тогда, и только тогда, когда $U \not\subseteq NLL_1, FL$*

Доказательство. Достаточность показана в 5. С другой стороны, очевидно, что есть кусочно-постоянные функции не лежащие в FL, например $\theta(x)$. Кроме того, если $U \subset NLL_1$, то $\exists k : U \subset NLL_{1,k}$, т.к. количество функций которые не одноместные-линейные в U конечно. Но тогда по 7 не все кусочно-постоянные выражаются. □

5.2. Заключение

Таким образом, в настоящей работе найден критерий выразимости кусочно-постоянных функций через одноместные линейные с конечной системой кусочно-параллельных в виде добавки. Автор выражает благодарность своему научному руководителю А.А. Часовских.

Список литературы

[1] В. С. Половников, “О задаче проверки функциональной полноты в классе кусочно-параллельных функций”, *Вестн. Моск. ун-та. Сер. 1. Матем., мех.*, 2008, №6, 31–35.

[2] Половников В.С., “О нелинейной сложности нейронных схем Мак-Каллока-Питтса.”, *М., Интеллектуальные системы.*, 2007, №11, 261-275.

[3] Отрощенко А.Д., “Классы кусочно-параллельных функций, содержащие все одноместные.”, *М., Интеллектуальные системы.*, 2020, №4, 57-74.

On the expressibility of piecewise constant functions in the space of piecewise parallel

Otroschenko A.D.

For a finite system of piecewise parallel functions implemented by schemes of linear elements and Heaviside functions, the criterion for the expressiveness of piecewise constant functions is obtained, supplemented by all single linear functions. Thus, the criterion of expressiveness of the binary classifier implemented by the McCulloch-Pitts neural scheme is obtained.

Keyword: Piecewise constant function, piecewise parallel function, completeness problem, expressibility problem, McCulloch-Pitts neural circuits.

References

[1] V. S. Polovnikov, “On the problem of checking functional completeness in the class of piecewise parallel functions”, *Vestn. Mosk. un-ta. Ser. 1. Mat., fur.*, 2008, №6, 31–35 (In Russian).

[2] Polovnikov V. S., “On the nonlinear complexity of McCulloch-Pitts neural circuits.”, *M., Intelligent systems.*, 2007, №11, 261-275 (In Russian).

[3] Otroschenko A.D., “Classes of piecewise parallel functions containing all single ones.”, *M., Intelligent systems.*, 2020, №4, 57-74 (In Russian).

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: 15.06.2022
Дата выхода: 28.06.2022
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).