



МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-  
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ» (ФГБОУ ВО «ЮУрГГПУ»)

ФАКУЛЬТЕТ ИНОСТРАННЫХ ЯЗЫКОВ  
КАФЕДРА АНГЛИЙСКОЙ ФИЛОЛОГИИ

**Грамматические трансформации в устном и письменном научно-  
техническом переводе с английского языка на русский**

**Выпускная квалификационная работа по направлению**

**45.03.02 Лингвистика**

**Направленность программы бакалавриата**

**«Перевод и переводоведение»**

**Форма обучения очная**

Проверка на объем заимствований:

62,36 % авторского текста

Работа рекомендована к защите  
рекомендована/не рекомендована

«17» июня 2022 г.

зав. кафедрой английской филологии  
Афанасьева Ольга Юрьевна

Выполнила:

студентка группы ОФ-403-074-4-1

Баннова Надежда Андреевна

Научный руководитель:

преподаватель

Солоницына Анастасия Сергеевна

Челябинск

2022 год



МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-  
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ» (ФГБОУ ВО «ЮУрГГПУ»)

ФАКУЛЬТЕТ ИНОСТРАННЫХ ЯЗЫКОВ  
КАФЕДРА АНГЛИЙСКОЙ ФИЛОЛОГИИ

**Грамматические трансформации в устном и письменном научно-  
техническом переводе с английского языка на русский**

**Выпускная квалификационная работа по направлению**

**45.03.02 Лингвистика**

**Направленность программы бакалавриата**

**«Перевод и переводоведение»**

**Форма обучения очная**

Проверка на объем заимствований:

\_\_\_\_\_ % авторского текста

Работа \_\_\_\_\_ к защите

рекомендована/не рекомендована

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

зав. кафедрой английской филологии  
Афанасьева Ольга Юрьевна

Выполнила:

студентка группы ОФ-403-074-4-1

Баннова Надежда Андреевна

Научный руководитель:

преподаватель

Солоницына Анастасия Сергеевна

Челябинск

2022 год

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ИССЛЕДОВАНИЯ .....	7
1.1 Понятие и классификация грамматических трансформаций.....	7
1.2 Особенности устного перевода .....	9
1.3 Особенности письменного перевода.....	12
1.4 Особенности научно-технического перевода .....	16
1.5 Особенности грамматического строя английского и русского языка	18
ВЫВОДЫ ПО ПЕРВОЙ ГЛАВЕ.....	20
ГЛАВА 2. АНАЛИЗ ПРИМЕНЕНИЯ ГРАММАТИЧЕСКИХ ТРАНСФОРМАЦИЙ В УСТНОМ И ПИСЬМЕННОМ НАУЧНО- ТЕХНИЧЕСКОМ ПЕРЕВОДЕ .....	23
2.1 Применение грамматических трансформаций в устном переводе видеофрагментов «Блокчейн и деньги».....	24
2.2 Применение грамматических трансформаций в письменном переводе .....	36
2.3 Сопоставление грамматических трансформаций в устном и письменном переводе научно-технических материалов.....	49
ВЫВОДЫ ПО ВТОРОЙ ГЛАВЕ.....	52
ЗАКЛЮЧЕНИЕ .....	54
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	57
ПРИЛОЖЕНИЯ.....	61

## ВВЕДЕНИЕ

Исследование научно-технического перевода – важная и актуальная задача, направленная на достижение адекватного перевода, способствующая решению многих прикладных задач и ускорению обмена информацией в области новейших достижений науки и техники среди специалистов и ученых разных стран. Данная выпускная квалификационная работа посвящена рассмотрению и анализу грамматических трансформаций в устном и письменном научно-техническом переводе.

Актуальность данной работы заключается в необходимости теоретического осмысления проблем грамматических трансформаций в связи с высокой ролью научно-технического перевода с английского на русский язык в российском производстве. При этом важно рассмотреть как устный, так и письменный научно-технический перевод, так как в связи с ускоряющимся научно-техническим прогрессом оба этих вида переводческой деятельности становятся все более востребованными, поэтому предпринятое нами исследование представляется важной и актуальной задачей.

Объектом исследования служат устный и письменный перевод научно-технических материалов в условиях непрерывного развития технологий.

Предметом исследования являются грамматические трансформации в переводе научно-технических текстов на тему блокчейна.

Цель данной работы заключается в изучении грамматических трансформаций при переводе английских научно-технических материалов на русский язык.

В соответствии с целью были поставлены следующие задачи:

1. Рассмотреть существующие классификации грамматических трансформаций.

2. Выявить специфику устного и письменного научно-технического перевода.

3. Рассмотреть особенности грамматического строя английского и русского языков и их влияние на процесс перевода.

4. Проанализировать и определить наиболее часто встречающиеся грамматические переводческие трансформации в материалах научно-технического характера на тему блокчейна.

5. Сопоставить особенности применения грамматических трансформаций в устном и письменном переводе.

6. Сформулировать на базе полученных данных практические рекомендации по переводу текстов на тему блокчейна с английского на русский язык.

Для решения задач нами применялись следующие методы: контекстуальный анализ сплошной выборки, метод сопоставительного анализа оригинала и перевода и количественный метод. На основе сопоставительного анализа англо-русских научно-технических переводов исследовался характер переводных эквивалентов и проводился анализ основных приемов перевода.

Новизна исследования заключается в выявлении специфики применения грамматических трансформаций в устном и письменном научно-техническом переводе материалов на тему блокчейна.

Теоретическая ценность заключается в возможном расширении данных, используемых в прикладном переводоведении, при помощи анализа переводческих трансформаций на материалах активно развивающихся сфер научно-технического прогресса, таких, как блокчейн.

Практической значимостью выпускной квалификационной работы является то, что результаты данного исследования в дальнейшем могут быть использованы при обучении теории и практике устного и письменного научно-технического перевода на материалах текстов о развивающейся технологии блокчейна.

На защиту выносятся следующие положения:

1. При переводе научно-технических текстов одинаковой тематики используемые грамматические трансформации в устном и письменном переводе практически не разнятся.

2. В устном переводе чаще используются опущения и добавления, в письменном переводе они встречаются реже, но, тем не менее, присутствуют.

3. В устном переводе научно-технических материалов наиболее часто применяются такие грамматические трансформации как замена членов предложения, добавление, замена частей речи, опущение и замена типа синтаксической связи.

4. В письменном переводе научно-технических текстов чаще применяются такие грамматические трансформации как замена членов предложения, замена частей речи и замена типа синтаксической связи.

Теоретическую основу настоящей работы составили исследования Л. С. Бархударова, А. Д. Швейцера, Я. И. Рецкера, В. Н. Комиссарова, В. П. Беркова, И. С. Алексеевой, Е. В. Бреус, Т. В. Поповой, Т. А. Казаковой, В. С. Виноградова, Л. В. Бирюковой и других учёных в области устного и письменного научно-технического перевода.

Практическим материалом исследования послужили научно-технические видео- и письменные материалы, полученные с сайтов [ocw.mit.edu](http://ocw.mit.edu), [bitcoin.org](http://bitcoin.org), [archive.org](http://archive.org) и аутентичных англоязычных материалов с переводами на русский язык.

Структура и объем работы определяются спецификой ее целей, задач и методов. Данная выпускная квалификационная работа состоит из введения, двух глав с выводами по каждой, заключения и списка использованных источников.

Во введении обосновывается актуальность данного исследования, формулируются цели и задачи, определяются объект и предмет работы, указываются методы и материалы исследования.

В первой главе описывается общий и структурированный теоретический и методологический материал по исследуемой теме. В первой части главы представлены понятие грамматических трансформаций и их классификация. Во второй части были выявлены особенности устного, письменного научно-технического перевода и грамматического строя английского и русского языка.

Вторая глава посвящена переводам устных и письменных материалов на тему блокчейна. В этой главе проводится сравнительный анализ использования грамматических трансформаций при переводе текстовых и устных научно-технических материалов с английского на русский язык и анализ причин и целей использования различных типов данных трансформаций.

Каждая глава сопровождается выводами, а общий итог подводится в заключении.

# ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ИССЛЕДОВАНИЯ

## 1.1 Понятие и классификация грамматических трансформаций

Одним из наиболее частых способов перевода является применение грамматических трансформаций. Переводчик очень часто сталкивается с несоответствиями между грамматическими структурами исходного языка и языка перевода. Для того чтобы быстро употребить верный вариант перевода, переводчику необходимо знать все виды грамматических трансформаций. Несомненно, русский язык является синтетическим языком, в то время как английский – аналитическим. Поэтому верный порядок слов и точная передача смысловых логических ударений является значительной отличительной особенностью при переводе с английского языка.

Как отмечает Я. И. Рецкер, грамматические трансформации – это преобразования структуры предложения в процессе перевода в соответствии с нормами ПЯ [15, с. 45].

При этом Рецкер говорил о том, что трансформация может быть полной или частичной. Обычно, когда заменяются главные члены предложения, происходит полная трансформация, если же заменяются лишь второстепенные члены предложения, происходит частичная трансформация [15, с. 47].

На данный момент существует множество подходов к классификации грамматических трансформаций на виды и типы, предложенных различными авторами.

Согласно Л. С. Бархударову, к грамматическим трансформациям относятся следующие типы:

1. Замена форм слова.
2. Замена частей речи, к примеру, наиболее распространенным приемом данной грамматической трансформации является замена



английских существительных русскими глаголами. Это явление связано с тем, что у русского языка более богатая и гибкая глагольная система.

3. Замена членов предложения (перестройка синтаксической структуры предложения).

4. Синтаксические замены в сложном предложении, такие, как замена простого предложения сложным, и наоборот, а также замена придаточного предложения главным и замена главного предложения придаточным [3, с. 183].

А. Д. Швейцер определил следующую классификацию грамматических трансформаций:

1) объединение предложений – это способ перевода, при котором синтаксическая структура предложения в оригинале преобразуется путём соединения двух простых предложений в одно сложное;

2) членение предложения – способ, при котором синтаксическая структура предложения в оригинале преобразуется в две или более предикативные структуры переводящего языка;

3) добавление грамматикализованных единиц: союзов, местоимений и т.п.;

4) опущение грамматикализованных элементов [18, с. 180].

Таким образом, можно сказать, что грамматические трансформации заключаются в преобразовании структуры предложения в процессе перевода в соответствии с нормами переводящего языка. Они могут происходить на синтаксическом уровне:

1) синтаксическое уподобление;

2) перестановки;

3) замена членов предложения;

4) преобразование активных конструкций в пассивные и наоборот;

5) членение предложения;

6) объединение предложений;

- 7) замена типа синтаксической связи;
- 8) опущения;
- 9) дополнения.

Также они могут происходить на морфологическом уровне:

- 1) замены частей речи (преобразование местоимения);
- 2) грамматическая замена числа;
- 3) замена грамматического времени.

Может происходить сочетание одних видов грамматических трансформаций с другими [28].

Принимая во внимание все сказанное выше, можно сделать вывод, что перевод с одного языка на другой невозможен без грамматических трансформаций, так как они помогают адаптировать иноязычный текст, позволяя переводчику составлять предложения по правилам переводящего языка, сохранив смысл текста.

## 1.2 Особенности устного перевода

Устный перевод – один из самых сложных видов перевода, требующий не только наличия отличного знания иностранного и родного языков и тематики, но и умения быстро, полно и доступно передавать смысл сказанного. Задачи устного перевода во многом пересекаются с задачами других видов переводческой деятельности. Однако с точки зрения основных механизмов и условий осуществления устного перевода, он имеет определенные отличия от письменного перевода. Осознание этого факта, связанное с тем, что во второй половине XX века сфера использования устного перевода значительно расширилась, привело к систематическому изучению закономерностей и особенностей устного перевода и появлению специальной теории устного перевода. Более того, в рамках теории устного перевода стали различать теорию последовательного перевода и теорию синхронного перевода [21].

Устный перевод как процесс и результат переводческой деятельности имеет множество определений. Приведем некоторые из них. Берков В. П. определяет устный перевод как вид перевода, при котором оригинал и его перевод предстают в нефиксированном виде в процессе перевода. Этот факт предопределяет однократность восприятия переводчиком отрезков оригинала и невозможность сравнения перевода после его завершения. При устном переводе создание текста перевода может происходить либо параллельно восприятию оригинала, либо после завершения восприятия оригинала [4, с. 91].

Согласно В. Н. Комиссарову устный перевод – такой вид перевода, при котором оригинал и его перевод выступают в процессе перевода в нефиксированной (устной) форме, что предопределяет однократность восприятия переводчиком отрезков оригинала и невозможность последующего сопоставления или исправления перевода после его выполнения [10, с. 111].

В рамках проводимых исследований устного перевода можно выделить три направления:

1. Изучение факторов, влияющих на извлечение переводчиком информации, содержащейся в оригинале. Восприятие устной речи кратковременно, одноразово и дискретно. Полнота понимания зависит от ритма, паузации и темпа речи; извлечение информации происходит в виде отдельных частей по мере развертывания цепочки языковых единиц в речи оратора, восприятие осуществляется на основе «смысловых опорных пунктов» [11, с. 75]. Переводчик прогнозирует последующее содержание текста на основе уже воспринятых «квантов» информации.

2. Рассмотрение устного перевода как особого вида речи на ПЯ. Описание специфики устной речи переводчика, которая отличается от обычной непереводной речи. Отличия заключаются в том, что речь переводчика ориентирована на оригинал и формируется в процессе

перевода. В рамках этого направления изучаются также особенности синхронного перевода.

3. Изучение устного перевода как особого вида перевода, в противопоставлении письменному переводу. Основное внимание при этом уделяется выявлению количественных и качественных особенностей устного перевода, которые отличают его от перевода письменного и составляют его специфику [1, с. 67].

Существует значительное количество видов устного перевода. Основными являются:

Последовательный перевод – это способ устного перевода, при котором переводчик начинает переводить после того, как оратор перестал говорить, закончив свою речь или часть ее. Последовательный перевод следует за текстом подлинника, либо уже отзвучавшим, то есть полностью произнесенным, либо произносимым с перерывами звучания обычно по-абзацно, то есть по группам из нескольких предложений, реже по-фразно – отдельными предложениями с паузой после каждого предложения. Следовательно, последовательный перевод имеет две разновидности: абзацно-фразовый перевод и собственно последовательный перевод.

Синхронный перевод – это способ устного перевода, при котором переводчик, слушая речь оратора, практически одновременно (с небольшим отставанием – 2-3 секунды) проговаривает перевод. Как правило, синхронный перевод осуществляется с использованием технических средств, в специальной кабине, где речь оратора подается переводчику через наушники, а сам переводчик говорит в микрофон, откуда перевод транслируется для рецепторов. Синхронный перевод требует от переводчика умения одновременно выполнять разнородные речевые действия: слушать на одном языке, переводить на другой язык и говорить на этом языке. Если при последовательном переводе переводчику необходима хорошо натренированная быстродействующая память, то при

синхронном переводе на первый план выступает умение одновременного слушания и говорения.

Более подробная классификация подвидов включает следующее:

- 1) устный последовательный односторонний перевод (перевод выступлений);
- 2) устный последовательный абзацно-фразовый односторонний перевод (перевод выступлений, лекций и т.п.);
- 3) устный последовательный абзацно-фразовый двусторонний перевод (перевод переговоров, интервью, бесед и т.п.);
- 4) синхронный перевод [6, с. 22].

### 1.3 Особенности письменного перевода

Деятельность письменного переводчика, продуктом которой является текст на языке перевода, носит смещенный характер и имеет иные координаты, по сравнению с речевой деятельностью, осуществляемой на иностранном языке. Во-первых, письменный перевод выполняется другим субъектом с другими субъективными характеристиками личности. Во-вторых, иными являются внешние условия его деятельности, так как осуществление происходит уже в ином социокультурном контексте. В-третьих, иным является и средство осуществления этой деятельности (перевод осуществляется на ином языке, система, норма и узус которого имеют уже иной характер, чем те же параметры исходного языка). Однако данные смещения (расхождения) могут рассматриваться как вынужденные [13, с. 27].

Особенность письменного перевода заключается в концентрации внимания, как на доминирующих аспектах текста на иностранном языке, так и на фоновых явлениях, поскольку деятельность переводящего ограничена текстом, и он вынужден переводить всю информацию, заложенную в тексте.

Рассматривая письменный перевод как деятельность и как результат деятельности (интерпретация исходного текста), опираясь на исследования Т. А. Казаковой, В. С. Виноградова и И. С. Алексеевой, можно выделить такие его особенности, как:

- 1) ограничение «письменного переводчика» текстом (отсутствие контакта, как с источником информации, так и с ее получателем);
- 2) возможность многократного обращения к тексту с целью точного его понимания благодаря отсутствию дефицита времени (письменный перевод не ставит переводчика в жесткие временные рамки);
- 3) возможность поиска верного эквивалента слова или группы слов, с которыми данное слово может сочетаться (создавать «ситуативный контекст»);
- 4) возможность использования вспомогательных источников информации (словарей, справочников, консультаций со специалистами);
- 5) возможность исправлять и редактировать собственный текст перевода.

Письменный перевод связан с целым рядом процессов, определяющих знаковый характер данного вида речевой деятельности. К ним относятся восприятие графической формы слов, узнавание их, то есть соотнесение с прошлым речевым опытом знакомства с данными или подобными формами; восприятие и распознавание синтаксических структур; восприятие и распознавание структуры образа или понятия; понимание смысловой связи между словами, предложениями и целым текстом. Любое из указанных действий при переводе сопровождается переводческим поиском, подысканием межъязыковых соответствий любому знаку, опознанному в исходном тексте [9, с. 30].

Л. В. Бирюкова предлагает строить процесс письменного перевода в три этапа:

1. Допереводной анализ оригинала в целом.

2. Собственно перевод как процесс установления и нахождения межъязыковых и контекстуальных соответствий и перевыражение содержания оригинала средствами языка перевода.

3. Общее редактирование [5, с. 17].

Суть первого этапа заключается в уточнении коммуникативной ситуации (кем, для кого, когда, в каких условиях, с какой целью создан оригинал), а также в определении тематики текста, его основного содержания, структуры и стилистического своеобразия.

Содержание второго этапа представляет собой последовательный ряд операций по переводу отдельных небольших отрезков оригинала. Обычно таким отрезком оказывается 1 фраза – простое или сложное предложение средней длины.

На третьем этапе перевода устраняются погрешности, выявляющиеся при прочтении всего текста: неэкономность формулировок, громоздкость конструкций, повторы и т. д. [14].

Виноградов В. С. отмечает, что в практике перевода выделяются два этапа работы. Один из них связан с осмыслением текста на иностранном языке, а другой с воспроизведением его на родной язык.

Первый этап, который называется восприятием текста, представляет собой чрезвычайно сложный сенсорно-мыслительный процесс, основанный на различных видах и формах анализа и синтеза работы органов чувств и мозга. На этом этапе переводчик стремится как можно полнее понять исходный текст и осознать его эстетическую ценность. Для переводчика как рецептора очень важно достичь такого уровня знаний и эстетической восприимчивости, который позволял бы объективно воспринимать весь объем смыслового содержания текста [7, с. 32].

У каждого типа и вида перевода этап восприятия имеет свои особенности и характеристики, которые делятся на две фазы:

1. Допереводное восприятие, то есть восприятие произведения в первом чтении, когда переводчик старается глубоко осмыслить,

«прочувствовать» произведение, осознать его информативную ценность и определить его стилистическое своеобразие.

2. Собственно переводное восприятие, то есть непосредственное восприятие конкретных слов, предложений, фраз, абзацев и т. д. в момент перевода.

Во второй фазе восприятия, когда происходит по-фразная рецепция иностранного текста перед его воссозданием на другом языке, переводчик оперирует анализом и синтезом, воспринимая смысл отдельных элементов (слов, словосочетаний, предложений) текста.

Второй этап процесса перевода – этап воссоздания на языке перевода воспринятой информации оригинала, и снова в сознании переводчика осуществляются сложные процессы анализа и синтеза, связанные с передачей из сферы мышления смысла уже в иной материальной словесной форме. Данный этап тоже состоит, по меньшей мере, из двух фаз: перевыражения и идентификации.

Восприняв семантическую и когнитивную информацию, заключенной в подлежащем переводу тексте, переводчик воссоздает эту информацию в материальных единицах языка перевода, стремясь сохранить ее полный объем. Не подыскивает, как иногда принято думать, соответствия каждому слову и словосочетанию исходной фразы, а «перевыражает» ее смысл [7, с. 34].

Таким образом, письменный перевод, являясь особым видом речевой деятельности, имеет свои особенности, но, при многочисленном разнообразии письменных текстов, имеет всегда одну и ту же схему и предполагает обычно следующую последовательность действий: сначала знакомство переводчика с исходным текстом; затем после предварительного предпереводческого анализа, т. е. выявления типа текста, жанровых и стилистических признаков, темы и области знаний, с которыми связан текст, переход к созданию текста перевода [14].



#### 1.4 Особенности научно-технического перевода

Особенности научно-технической литературы распространяются на ее стилистику, лексику и грамматику. Основная цель научно-технического перевода – точность передачи информации. В научной и технической литературе обнаруживают ряд грамматических и лексических особенностей.

К характерным особенностям научно-технической литературы относится высокая информативность, четкая логика, структура, точность и ясность. Предполагается использование большого количества терминов. Терминологическая лексика позволяет четко и коротко изложить суть. Что касается синтаксической структуры научной и технической литературы, то чаще всего они характеризуются сложными предложениями: много причастных, инфинитивных, герундиальных оборотов [6, с. 20].

Термины в научно-технической литературе составляют примерно 25 % текста. Кроме терминов, научно-технический текст содержит общенаучные (общетехнические) и общеупотребительные слова. В отличие от терминов эта лексика характеризуется тем, что она не обладает свойством обозначения понятия и объекта определенной сферы науки и техники, а от общеупотребительной лексики ее отличает то, что она функционирует исключительно в научно-технической сфере и вместе с терминами определяет специфику научно-технического стиля. Кроме того, иногда в английской научно-технической литературе можно столкнуться с жаргоном и сленгом. Арго, жаргон и сленг, как маргинальные проявления языка, отчасти являются следствием попыток найти «свежие», своеобразные выразительные слова для наименования понятий, действий и предметов в данной сфере и придания речи определенного, «нелитературного» колорита [3, с. 45].

Важной особенностью английского научно-технического стиля, проявляющейся в отборе и использовании языковых средств, является

также стремление к краткости и компактности изложения, что выражается особенно в достаточно широком использовании эллиптических конструкций. Неверное понимание этих конструкций нередко приводит к ошибкам в переводе. Эта тенденция находит отражение и в ряде других грамматических особенностей. Для научно-технического стиля характерна, например, замена определительных придаточных предложений прилагательными в постпозиции [9, с. 37].

Следует также отметить множество случаев опущения в научно-технических материалах артикля, особенно определенного, там, где в текстах другого типа его употребление считается абсолютно обязательным.

В научно-техническом стиле широко используется употребление множественного числа вещественных существительных. Наблюдается также частое использование причинно-следственных союзов и логических связок [17].

Для научно-технических материалов английского языка характерно преобладание простых предложений, которые, составляют в среднем свыше 50 % общего числа предложений в тексте. В то же время количество сложных предложений относительно невелико. Это явление несвойственно соответствующему стилю в русском языке, где сложные предложения используются очень широко. В связи с этим в англо-русских технических переводах часто используется объединение предложений, в результате чего двум или более простым предложениям английского оригинала соответствует одно сложное предложение в русском переводе [16, с. 38].

Перевод научно-технического текста должен верно передавать смысл оригинала. Отступления от оригинала должны быть оправданы особенностями русского языка, требованиями стиля. Важно не допускать потери важной информации оригинала.

## 1.5 Особенности грамматического строя английского и русского языка

Становление фонетики, грамматики (морфологии и синтаксиса), лексики русского и английского языков прошли многие этапы развития исторических периодов Англии и России. В ходе политических, экономических, социальных, военных и культурных взаимодействий Англия перенимала некоторые слова из русского диалекта, а Россия дополняла лексику русского языка, опираясь на английский диалект. И русский, и английский языки имеют свои языковые структуры, существенно отличающиеся друг от друга [19].

Сходство русского и английского языков состоит в том, что они принадлежат к индоевропейской языковой семье. Русский язык – к восточнославянской ветви и английский – к германской. Однако сами языковые структуры этих языков строятся на принципах аналитической и синтетической грамматики. Поэтому главная ошибка людей, изучающих иностранный язык, – это попытка дословного перевода с одного языка на другой. Это приводит к неправильному построению предложения и, как следствие, искажает смысл сказанного [27].

Синтетические языки (греч. *synthesis* – составление – «получающийся в результате синтеза», «объединяющий»). К ним относятся древнегреческий, латинский, старославянский, русский, немецкий, литовский и др. Синтетическая грамматика подразумевает под собой свободный порядок слов, имеющих суффиксы, окончания, склонения, спряжения и прочее. Грамматические значения синтетики выражаются в пределах изменений внутри самого слова [12].

Аналитические языки (греч. *analysis* – развязывание – «получающийся в результате анализа», «разъединяющий»). Грамматические отношения выражаются служебными словами, порядком слов, интонацией и т. п., а не словоизменением, т. е. не грамматическим

чередованием морфем в пределах словоформы, как в синтетических языках. К аналитическим языкам относятся английский, французский, итальянский, испанский, болгарский, новоперсидский, датский язык и др [8, с. 13].

Принимая во внимание изложенные выше принципы построения языковых структур английского и русского языков, можно сделать вывод, что английский язык в силу своей строгой структуры является в большей степени аналитическим языком. Русский язык преимущественно синтетический из-за свободного порядка слов в предложении и непосредственных изменений самих слов через их склонение, спряжение и другие грамматические компоненты. Конечно, языки не могут быть полностью аналитическими или синтетическими. В некоторых контекстах английский язык может содержать признаки синтетического языка, а русский язык может содержать признаки аналитического языка [12].

Описанные различия грамматического строя английского и русского языка приводят к необходимости применения грамматических трансформаций для достижения максимально возможной адекватности перевода.

## ВЫВОДЫ ПО ПЕРВОЙ ГЛАВЕ

В теоретической части исследования мы проанализировали понятие и классификации грамматических трансформаций, предложенные различными авторами, а также особенности устного и письменного научно-технического перевода.

В современной теории перевода существует несколько точек зрения, которые имеют некоторое сходство, но в то же время противопоставлены друг другу. Мы поддерживаем точки зрения Л. С. Бархударова и А. Д. Швейцера, поскольку используем метод сравнительного анализа научно-технических переводов с английского на русский язык. Основываясь на теории, представленной в первой главе, можно сделать следующие выводы.

1. Ряд ученых, изучающих теорию перевода, разделяют грамматические трансформации в соответствии с актуальными представлениями. Л. С. Бархударов делит грамматические трансформации на следующие типы: замена форм слова, замена частей речи, замена членов предложения, синтаксические замены в сложном предложении. А. Д. Швейцер же, в свою очередь, выделяет следующую классификацию: объединение предложений, членение предложения, добавление грамматикализованных единиц, опущение грамматикализованных элементов. При этом существуют общепринятые способы классификации на синтаксическом уровне, такие как синтаксическое уподобление, перестановки, замена членов предложения, преобразование активных конструкций в пассивные и наоборот, членение предложения, объединение предложений, замена типа синтаксической связи, опущения, дополнения. Также они могут происходить на морфологическом уровне: замены частей речи (преобразование местоимения), грамматическая замена числа, замена грамматического времени.

2. Устный перевод является одним из самых сложных видов перевода, поскольку требует от переводчика наличия отличного знания разговорного языка и определённой научной тематики, умения быстро, полно и доступно передавать смысл сказанного.

3. Письменный перевод имеет множество особенностей, таких как возможность многократного обращения к тексту с целью точного его понимания благодаря отсутствию дефицита времени, возможность поиска верного эквивалента слова или группы слов, с которыми данное слово может сочетаться, возможность использования вспомогательных источников информации, возможность исправлять и редактировать собственный текст перевода. Но при его выполнении существует определенная последовательность действий: ознакомление с текстом оригинала, произведение предварительного предпереводческого анализ, т. е. выявление типа текста, жанровых и стилистических признаков, тем и областей знаний, с которыми связан текст, и, после указанных этапов, переход к созданию текста перевода.

4. Научно-технический стиль имеет свои уникальные особенности, такие как высокая информативность, четкая логика, структура, точность и ясность, а также использование большого количества терминов. Поэтому появляется необходимость применения грамматических трансформаций, таких как синтаксическое уподобление, перестановки, замена членов предложения, членение предложения, объединение предложений, замена типа синтаксической связи, опущения, добавления, замены частей речи, грамматическая замена числа и времени. Также в научно-техническом переводе не допускается значительное отступление от исходного содержания текста оригинала. Это действия влияют на устный и письменный перевод научно-технических текстов с английского на русский язык и должны быть оправданы грамматическим строем русского языка.

5. Несмотря на то, что английский и русский язык относятся к одной индоевропейской языковой семье, они значительно отличаются по языковым структурам, поскольку английский язык является аналитическим, а русский – синтетическим, это влияет на перевод текстов с английского языка на русский и приводит к необходимости использовать грамматические трансформации.

На основе полученной информации мы провели анализ особенностей использования грамматических трансформаций при переводе научно-технических материалов с английского на русский язык, основываясь на аутентичных текстовых материалах и видеоматериалах и их существующих официальных переводах.

## ГЛАВА 2. АНАЛИЗ ПРИМЕНЕНИЯ ГРАММАТИЧЕСКИХ ТРАНСФОРМАЦИЙ В УСТНОМ И ПИСЬМЕННОМ НАУЧНО- ТЕХНИЧЕСКОМ ПЕРЕВОДЕ

В устном и письменном научно-техническом переводе важную роль играет правильно подобранный вариант использования грамматических трансформаций. Для выполнения качественного перевода переводчик обращается к правилам языка перевода, а затем подбирает наиболее подходящий вид грамматических трансформаций.

Для написания практической части данной выпускной квалификационной работы был проведен сопоставительный анализ оригиналов и переводов научно-технических материалов о технологии блокчейн.

Материалами для анализа устного и письменного перевода научно-технических текстов послужили лекции профессора Гари Генслера «Блокчейн и деньги» (Gary Gensler «Blockchain and money») в переводе Cryptus [22; 23; 24; 25; 26; 29], и для письменного: книга Антонопулоса Андреаса М. «Осваиваем биткойн. Программирование блокчейна» (Andreas M. Antonopoulos «Mastering the Bitcoin. Programming the Open Blockchain») в переводе А. В. Снастина [2; 30] и статья Сатоши Накамото, создателя криптовалюты биткойн, «Биткойн: система цифровой пиринговой наличности» (Satoshi Nakamoto «Bitcoin: A Peer-to-Peer Electronic Cash System») в переводе arvicco и grich [20; 31].

Для этого мы следовали следующей схеме:

- 1) контекстуальный анализ сплошной выборки предложений, которые были переведены с использованием грамматических трансформаций;
- 2) определение видов использованных грамматических трансформаций;



3) подсчет частотности и процентного соотношения видов грамматических трансформаций;

4) анализ особенностей использования грамматических трансформаций при переводе устных и письменных научно-технических материалов;

5) сопоставительный анализ переводческих трансформаций, применяемых в устном и письменном научно-техническом переводе;

6) составление практических рекомендаций по применению переводческих трансформаций при переводе устных и письменных научно-технических текстов.

## 2.1 Применение грамматических трансформаций в устном переводе видеофрагментов «Блокчейн и деньги»

Для начала рассмотрим лекции Гари Генслера «Блокчейн и деньги» (Gary Gensler «Blockchain and money»). Данный материал был переведен командой Cryptus.

Обратимся к таблице 1.

Таблица 1 – Блокчейн и деньги

Оригинал	Перевод	Грамматическая трансформация
That's the protocol that mounts <u>on top (1)</u> of the TCP/IP stack <u>to provide encryption (3) using (1, 2)</u> asymmetric case.	Это протокол, который накладывается <u>поверх (1)</u> уровня TCP/IP и <u>используется (1, 2)</u> для асимметричного шифрования.	1. Замена части речи. 2. Замена членов предложения. 3. Опускание.
But these time-stamped blocks of data creates a database, an auditable database.	Эти отмеченные временем блоки информации создают базу данных, прозрачную базу данных, <u>которую можно просматривать.</u>	Добавление.
The data can be actually	Эти данные могут быть	Замена членов

<p>computer code. And <u>you could</u> have the data being verified computer code and algorithms.</p>	<p>компьютерным кодом. То есть <u>можно</u> переводить как деньги, так и подтвержденный компьютерный код.</p>	<p>предложения.</p>
<p><u>It's hard to update the software of a blockchain (1), because if you create a <b>decentralized (2)</b> where <u>no one's in control (5), no one can collect economic rents (3), you also don't have sort of somebody with the ability (1) to necessarily update the software (5).</u></u></p>	<p><u>Блокчейн на данный момент очень трудно обновлять (1), потому что если мы создаем определенную платформу, определенный блокчейн (3), на котором не будет <b>никакого централизатора (2, 5), то очень сложно дать кому-то необходимые полномочия (1), которые будут обновлять постоянно блокчейн (4) до новейшего программного обеспечения(5).</b></u></p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена части речи.</li> <li>3. Опускание.</li> <li>4. Добавление.</li> <li>5. Замена типа синтаксической связи.</li> </ol>
<p>The Depository Trust Corporation is looking at <u>blockchain-inspired</u> solutions for some of their data warehouses, but they, too, are going to control the nodes.</p>	<p>Депозитарные трастовые корпорации тоже собираются использовать <u>блокчейн технологии</u> в хранении своей базы данных, однако они тоже собираются контролировать все свои ноды.</p>	<p>Замена части речи.</p>
<p><u>So the word (1) «nonce» means a random number that is used once. N for (2) number, and «once».</u></p>	<p>«Nonce» – это случайное число, которое может использоваться всего лишь один раз. «N» <u>означает (2) число, и остальное означает (3) «once», то есть «один» (3).</u></p>	<ol style="list-style-type: none"> <li>1. Опускание.</li> <li>2. Замена части речи.</li> <li>3. Добавление.</li> </ol>
<p>So <u>if you take (1) a certain set of data, it will always give you (1) the same hash. And you can efficiently compute it (1). You've got to do it in short periods of time (1). And in Bitcoin's case, it's done in nanoseconds or less (2), because they're one computer, one CPU (1) can do couple of terahashes a</u></p>	<p>Если <u>взять (1) один и тот же набор данных, то всегда получается (1) один и тот же хэш. Также хэш-функция очень эффективно высчитывается (1). Это должно быть сделано в коротком промежутке времени (1).</u> И в случае с биткоином, хэш-функции высчитываются за</p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Опускание.</li> </ol>

second.	наносекунды, потому что один компьютер, один процессор может сделать несколько терахэшей в секунду.	
<p>A zip code wouldn't make it (1). Well (2), the computer scientists use this term preimage resistant. You can only go one way (3), meaning (2) it's infeasible to determine the input from the output (4). It's infeasible (5) to determine the x from the hash of x.</p>	<p><u>Почтовый индекс не справится с теми задачами, которые нам нужны (1).</u> Информатики используют термин «устойчивость к прообразу». <u>То есть можно двигаться только в одну сторону (3), и невозможно определить, какой был ввод, имея только вывод (4).</u> Это невыполнимая задача (5) определить x от хэша от x.</p>	<ol style="list-style-type: none"> <li>1. Добавление.</li> <li>2. Опушения.</li> <li>3. Замена членов предложения.</li> <li>4. Замена типа синтаксической связи.</li> <li>5. Замена части речи.</li> </ol>
<p>In his paper, he's talking about <u>how hard it is computationally to do (2)</u> what some people call a 51% attack, to basically take over all the nodes.</p>	<p>В документах он пишет о том, что некоторые люди называют «атака 51%». <u>Это значит то, что благодаря высокой компьютерной мощи в теории можно захватить больше половины нод, на которых работает биткойн (1).</u></p>	<ol style="list-style-type: none"> <li>1. Добавление.</li> <li>2. Опушение.</li> </ol>
<p>It's called an avalanche effect, <u>meaning (1, 3) you change one little difference and the whole thing looks different (2).</u> <u>If you changed one thing (1), it (3) all looked so different.</u> And why that's important <u>is it makes it (3) more secure (1).</u></p>	<p>Это называется эффект лавины, <u>то есть (1, 3), если Вы измените хотя бы одну небольшую мелочь, то все значение в общем поменяется (2).</u> <u>Если что-то совсем чуть-чуть менялось (1), то вся хэш-функция (3) выглядела совсем по-другому.</u> И это важно, потому что, таким образом, <u>хэш-функция (3) получается более безопасной (1).</u></p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена типа синтаксической связи.</li> <li>3. Замена части речи.</li> </ol>
<p>There is a bunch of cryptography <u>underneath (2)</u> this. And the key is <u>it (2) is not 100% immutable.</u> <u>It's probably one in, you know</u></p>	<p>В криптовалютах есть много криптографий, как такой <u>невидимый слой (2).</u> И ключ к <u>пониманию (3)</u> – это то, что <u>блокчейн (2), он</u></p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена части речи.</li> <li>3. Добавление.</li> </ol>

<p>(1), I don't know (4), a quadrillion immutable (1).</p>	<p>(3) не на 100% неизменный. У нас все равно есть шанс (1), может быть (1), он очень маленький, может быть, один к нескольким квадриллионам (1), что что-то пойдет не так (3).</p>	<p>4. Опускание.</p>
<p>The actual probability should be actually <u>1 over 2</u> (2) to the power of 128. So much more than one quadrillion.</p>	<p>Вероятность взлома (1), она должна быть половина (2) в 128 степени. Так что это намного больше, чем 1 к нескольким (1) квадриллионам.</p>	<p>1. Добавление. 2. Замена части речи.</p>
<p>He said if one of <u>them</u> (1) is broken at least the other one is <u>less likely</u> (1) to be broken. You have to hash <u>something</u> (1) twice. <u>And he was just making it that much more secure</u> (3), even knowing <u>it</u> (1) was one out of 10 to the 40th chance.</p>	<p>Он лично (2) написал, что, если одну из <u>этих функций</u> (1) взломают, то <u>хотя бы есть вероятность</u> (1), что не взломают вторую из них. Он хотел зашифровать <u>некие вещи</u> (1) дважды. Таким образом, это было <u>бы более безопасно</u> (3), <u>хотя вероятность взлома</u> (1) <u>и так очень мала</u> (2) и составляет всего лишь 10 в 40 степени.</p>	<p>1. Замена части речи. 2. Добавление. 3. Замена членов предложения.</p>
<p>A light node or a wallet <u>that</u> anyone here could download on your cell phone probably does not download the millions of transactions that have happened in the <u>history</u> (3) of Bitcoin. You are <u>unlikely</u> (2) to download <u>what's called</u> (4) a full node. But you might download all the headers, <u>this bit of information that's all of the headers</u> (2).</p>	<p>Облегченная нода, или же <u>так называемый</u> (1) кошелек, который каждый из здесь присутствующих может скачать на свой телефон, он, скорее всего, не скачивает <u>в сеть</u> (1) миллионы транзакций из прошлого, которые <u>ранее</u> (3) произошли на блокчейне. <u>Это очень маловероятно</u> (2), что Вы скачаете целую ноду. Однако Вы можете скачать все заголовки <u>предыдущих нодов</u> (1), <u>то есть получить всю информацию, которая в этих заголовках содержится</u> (2).</p>	<p>1. Добавление. 2. Замена членов предложения. 3. Замена части речи. 4. Опускание.</p>
<p>All of the <u>information in</u></p>	<p>Вся <u>информация, которая</u></p>	<p>Замена типа</p>

<p><u>Bitcoin</u> is still not that large. It's less than 200 gigs.</p>	<p><u>находится в биткойне</u>, она не столь тяжелая. Она весит не менее 200 Гб.</p>	<p>синтаксических связей.</p>
<p>And this is really important when you get to like Ethereum where there's a lot of data, <u>a lot of computation</u> (2) down in each of these blocks.</p>	<p>И это очень важно <u>знать</u> (1), особенно, когда мы будем <u>говорить в будущем</u> о <u>таких тяжелых блокчейнах</u> (1) как Эфириум, где содержится очень много информации в каждом блоке.</p>	<ol style="list-style-type: none"> <li>1. Добавление.</li> <li>2. Опущение.</li> </ol>
<p>The papers mentioned that <u>it's possible</u> (1, 2) that two the hash of x <u>equal</u> (2) to hash of y.</p>	<p>В документах упоминалось то, что <u>может быть такая ситуация</u> (1, 2), что два хэша-х <u>будут равны</u> (2) хэшу-у.</p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена части речи.</li> </ol>
<p>But if the miners are working at the same time, if the same information are not treated at the same exact time, it won't be a problem <u>because then they just continue just like two different</u> (2).</p>	<p><u>Но также там говорится</u> (1), что не все майнеры работают с четкостью одновременно, если вся информация не будет обрабатываться в одно и то же время, то это не будет проблемой. <u>Если у нас одна и та же информация не обрабатывается в одно и то же время, то никаких проблем у нас не возникнет</u> (1).</p>	<ol style="list-style-type: none"> <li>1. Добавление.</li> <li>2. Опущение.</li> </ol>
<p><u>Having said that, the real way that timestamping happens</u> (2, 3) is if a <u>block is mined</u> (2) and it's the 540,000th block and <u>it's sort of accepted in all the nodes</u> (2), these 10,000 nodes start mining the 540,000 and 1st block, <u>in essence, it's just think of it as almost like a stack</u> (2, 4).</p>	<p><u>Поняв это, мы можем усвоить следующую вещь</u> (1, 2): а именно, что <u>когда мы майним</u> (2), если, <u>допустим, мы сейчас майним</u> (1) 540-тысячный блок, и, <u>если его подтверждают все остальные ноды</u> (2), то есть все остальные 10 тысяч нод начинают майнить 540-тысячный и первый блок, <u>то у нас получается такой процесс, в котором блоки накладываются друг за другом, поверх друг друга</u></p>	<ol style="list-style-type: none"> <li>1. Добавление.</li> <li>2. Замена членов предложения.</li> <li>3. Опущение.</li> <li>4. Замена части речи.</li> </ol>

	(1, 2, 4).	
<p>If there were two that hashed (4) to the same kind of has digest that it would be <b>unlikely</b> (1, 2, 4) that they'd both be valid in the context. So given <u>what's a valid blockchain transaction that that could even further</u> (3) reduce the likelihood of any problems.</p>	<p>Если мы получим два <u>одинаковых хэша (4), то они могут отличаться (4) по своему контексту (1, 2),</u> то есть иметь разное значение в программном коде. То есть, соответственно, благодаря этому у нас еще меньше вероятность какой-либо ошибки.</p>	<ol style="list-style-type: none"> <li>1. Замена типа синтаксической связи.</li> <li>2. Замена членов предложения.</li> <li>3. Опускание.</li> <li>4. Замена части речи.</li> </ol>
<p>The blockchain context would even <u>make two hashes of the same value</u> (2) even more unlikely because of the context.</p>	<p>Контекст хэширования <u>имеет значение (1), так что, если различные хэши будут в разном контексте друг к другу (2), то вероятность столкновения еще меньше (1).</u></p>	<ol style="list-style-type: none"> <li>1. Добавление.</li> <li>2. Замена членов предложения.</li> </ol>
<p>And that doesn't mean (1, 2) that <u>they (1) got identical hashes (2) because the puzzle is not geared to getting an exact hash (2).</u> The <u>Bitcoin puzzle is having a certain number of leading zeros (2).</u> So <u>it's (1) literally started, I think (4), it was nine or 10 leading zeros. I'm talking about 10 years ago (4). And now, you have to hash to something with, I think, it's about 20 (4) or 26 leading zeros (2).</u></p>	<p>И даже в этом случае (1, 2) <u>майнерам (1) не обязательно было находить одинаковый хэш (2), потому что задача в майнинге биткойна (3) не заключается в том, чтобы найти какой-то конкретный хэш (2), а немного в другом (3). Задача майнеров заключается в том, чтобы упростить этот хэш до определенного количества нулей (2).</u> Когда <u>майнинг (1) только начался, там было примерно 9 или 10 нулей. А сейчас их порядка 26 нулей (2), вот столько нужно сейчас в хэш-функции упростить (3).</u></p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена членов предложения.</li> <li>3. Добавление.</li> <li>4. Опускание.</li> </ol>
<p>Hash functions <u>are (1, 2) basically a way to compress (1) a lot of (1) data, have a fingerprint (2), make sure that it's basically commitment (1).</u></p>	<p>Хэш-функции – <u>это (1, 2), в общем, способ сжатия (1) большого количества (1) информации, поставить туда электронный отпечаток пальца (2) и убедиться в том, что все</u></p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена членов предложения.</li> </ol>

	<u>правильно работает (1).</u>	
<u>You have to generate a key pair (2). And when a key pair is generated, a public key and a private key are generated at the same time (2). And they (3) need a random number to go into it (4).</u>	<u>Во-первых (1), необходимо сгенерировать пару ключей (2). И когда создается пара ключей, у нас также создается приватный и публичный ключ в одно и то же время (2). И эти ключи (3) должны создаваться на основе рандомного числа.</u>	1. Добавление. 2. Замена членов предложения. 3. Замена части речи. 4. Опускание.
So you pick a random number actually between 0 and 256, that's <u>your (1) private key. To pick a public key, you derive it directly from the private key (2).</u>	На самом деле Вы выбираете рандомное число от 0 до 256, и это будет приватный ключ. <u>А публичный ключ создается напрямую из приватного ключа (2).</u>	1. Опускание. 2. Замена членов предложения.
Also, a public key is very long. <u>It's about (1) 512 bits. And so you can shrink the data and make the data more compressed by hashing it , which took it to 256 bits. He hashes it twice (3), and then he does this base 58 and makes it even a little tighter (4).</u>	Также публичный ключ, он довольно длинный. <u>Он занимает (1) где-то 512 бит. Поэтому мы можем уменьшить данные и сделать эти данные более сжатыми путем того, что все это хэшируется и весит всего лишь 256 бит. После этого (2) он даже 2 раза (3) все это хэширует, и потом он использует еще «base 58», чтобы все это укоротить (4).</u>	1. Замена членов предложения. 2. Добавление. 3. Замена части речи. 4. Замена синтаксических связей.
So cryptographic hash function <u>is a way to take not only a lot of information and put it into a fixed form, but the key thing here is the hash functions are what tie the blocks together because hash functions can point to previous information.</u>	Итак, криптографическая хэш-функция – <u>это не просто способ для того, чтобы взять большой объем данных и скомпоновать его в фиксированную форму, но также ключевой момент в хэш-функциях: они являются тем, что объединяет блоки друг с другом, потому что хэш-функции могут указывать на предыдущую</u>	Замена синтаксических связей.

	информацию.	
There is also a problem with performance and scalability.	<u>В блокчейне</u> есть проблема с производительностью и с масштабируемостью.	Добавление.
They're so small, <u>they were just _____ essentially inconsistently.</u>	Они очень маленькие, и, вроде, <u>они бессмысленно нагружают систему.</u>	Замена членов предложения.
It's been attacked by so many <u>viruses and bugs</u> , you have some reason to trust it.	На них нападало столько хакерских атак, <u>и их пытались взломать столько раз</u> , что у нас есть очень много причин им доверять.	1. Опускание. 2. Добавление.
<u>And the transactions code and ledgers shifts</u> (3) largely dependent upon whether <u>it's (2)</u> a transaction ledger or an account ledger.	<u>И также у нас сюда входит тема транзакций кода и реестров (1).</u> <u>Здесь технология может (3)</u> значительно варьироваться от того, является ли <u>реестр (2)</u> реестром транзакций или реестром балансов.	1. Добавление. 2. Замена части речи. 3. Замена членов предложения.
But either you have to record a transaction, or you have to record <u>a change in the state</u> or a change in the account.	Но в любом случае, нам нужно будет либо записывать транзакцию, либо нам нужно будет записывать изменения в балансе аккаунта.	Опускание.
But it's embedded in this, and <u>they</u> could have different scripting.	Ну и, конечно же, <u>различные блокчейны</u> могут иметь различные скрипты.	Замена части речи.
So Bitcoin has a very robust and well-established code base.	У биткойна, на самом деле, очень эффективная база для кода, <u>на которой можно настраивать свои вещи.</u>	Добавление.
Bitcoin has a lot of high-quality code so <u>you can build up on it.</u>	У биткойна есть очень много качественного кода, <u>на котором можно настраивать свой код.</u>	Замена членов предложения.



<p>One estimate is that <u>it's (1) 200 million (2) kilowatt hours per day.</u></p>	<p>По некоторым подсчётам <u>майнинг (1) расходует где-то 200 миллионов (2) киловатт часов за сутки.</u></p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена членов предложения.</li> </ol>
<p>Bitcoin Cash has a bigger block size and a shorter 2.5 minute processing time.</p>	<p>У Bitcoin Cash <u>сейчас более короткий интервал между блоками, а именно 2,5 минуты,</u> и более высокий размер блока.</p>	<p>Замена типа синтаксической связи.</p>
<p>Another problem is <u>that if you have bigger blocks, they take longer to propagate to the network.</u></p>	<p>Другая проблема в том, <u>что чем больше у нас блоки, тем сложнее их распространить по сети.</u></p>	<p>Замена членов предложения.</p>
<p>And thus, you might end up inadvertently having more chains that are discredited, in a sense, because there was <u>work being done until the first one gets propagated.</u></p>	<p>Таким образом, у нас могут просто возникнуть несколько ответвлений или цепей, которые потом просто дискредитируют, несмотря на то, что работа <u>была сделана для того, чтобы их намайнить, эти блоки потом пропадут, потому что они окажутся невалированными.</u></p>	<ol style="list-style-type: none"> <li>1. Добавление.</li> <li>2. Замена членов предложения.</li> </ol>
<p>And <u>it's (2) written almost like a smart contract but it's not called smart contracts to sort of partition, or you might loosely think of it as escrowing (1), even though technically it might be different (3).</u></p>	<p>Этот <u>скрипт (2) очень похож на смарт контракт, он контролирует движение активов между сетями, но он не называется смарт контрактом, то есть технически у вас получается такое условное депонирование (1) и вы не сможете потратить деньги на основной сети и одновременно на сайте сети (4).</u></p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена части речи.</li> <li>3. Опускание.</li> <li>4. Добавление.</li> </ol>
<p><u>But generally (1), all the alternatives have some way to randomize (3) or delegate the node (2) that will do the next block.</u></p>	<p><u>Мы говорим о механизме, который позволяет в случайном порядке (3) выбрать ноду (2), которая выберет следующий блок.</u></p>	<ol style="list-style-type: none"> <li>1. Опускание.</li> <li>2. Замена типа синтаксической связи.</li> <li>3. Замена части речи.</li> </ol>

Open source software updates, <u>which are not backward compatible</u> .	Обновления в открытом доступе, <u>которые не будут работать с предыдущей версией обновления</u> .	Замена членов предложения.
Turing complete allows you to do loops <u>inside</u> of computer programs.	Полнота по Тьюрингу позволяет вам создавать повторы <u>при программировании</u> программного кода.	Замена части речи.
So you can find any transaction on an entire blockchain <u>by knowing the block</u> , and then within the block, which transaction.	То есть вы можете найти любую транзакцию внутри блокчейна, <u>если вы знаете номер блока</u> , а также, какая эта транзакция, номер транзакции в этом блоке.	Замена членов предложения.
The actual inputs (1) disappear <u>once they go through this</u> (2).	Они (2) стираются после того, <u>как у нас совершается процесс транзакции</u> (2).	1. Замена части речи. 2. Замена членов предложения.
<u>The digital signatures have to be validated</u> (1).	<u>Также нам необходимо валидировать</u> _____ и <u>подтвердить</u> (2) цифровую подпись (1).	1. Замена членов предложения. 2. Добавление.
<u>Instead of going back and looking for all these things</u> (2), <u>there's actually a database that has all the unspent transactions</u> (1).	Нам не нужно отматывать весь блокчейн назад и смотреть, <u>где какие балансы</u> (2), <u>мы просто видим в отдельной базе данных, кто сколько денег может потратить</u> (1).	1. Замена членов предложения. 2. Замена типа синтаксической связи.
<u>81% are transactions that send to</u> (2) (1) a hash of a Bitcoin address.	<u>81% всех транзакций отправляется</u> (2) (1) на хэш какого-то биткойн-адреса.	1. Замена членов предложения. 2. Замена типа синтаксической связи.
In computer technology, all data can be broken down <u>to a series of registries of 0's or 1's</u> (2, 3).	В компьютерной технологии любые данные могут быть разбиты <u>на более мелкие значения</u> (1), <u>и они состоят из серий из единиц и нулей</u> (2, 3).	1. Добавление. 2. Замена типа синтаксической связи. 3. Замена членов предложения.

<p>You could hash all 100,000 books, <u>and then store the hashes in the blockchain and not the books themselves.</u></p>	<p>Вы можете захешировать все 100 000 книг и сохранить лишь хэш на блокчейне, а книги будут храниться где-то в другом месте.</p>	<p>Замена членов предложения.</p>
<p><b><u>The storing of the hash allows (1) you to prove (2) that you have the data (2).</u></b></p>	<p><b><u>Владение хэшем означает лишь то, что (1) вы можете просто перепроверить (2), правильные ли у вас вводные данные (2).</u></b></p>	<p>1. Замена типа синтаксической связи. 2. Замена членов предложения.</p>
<p>Maybe it's immutable except for one out of 10 to the 40th times it could be broken <u>or something.</u></p>	<p>То есть блокчейн неизменный, но есть вероятность 1/10 в 40 степени, что блокчейн может сломаться.</p>	<p>Опущение.</p>
<p><u>Part of the reason it was hashed (1), and part of the reason it goes through that extra code, is to make it (2) even more secure.</u></p>	<p><u>Причина, по которой он хэшируется (1), и причина, по которой накладывается дополнительная форма кода – это для того, чтобы сделать адрес (2) более безопасным.</u></p>	<p>1. Замена членов предложения. 2. Замена части речи.</p>
<p>If you take a 24-digit hash, and you contract it to a 4-digit hash, <u>there's fewer options.</u></p>	<p>Если вы, к примеру, возьмёте 24-значный хэш и вы сожмёте его до 4-значного хэша, <u>то вариантов для подбора станет меньше.</u></p>	<p>Замена членов предложения.</p>

Всего в таблице 1 было рассмотрено 100 примеров грамматических трансформаций, использованных в устном научно-техническом переводе с английского языка на русский. Статистический анализ показал, что большинство предложений были переведены с использованием нескольких грамматических трансформаций, общее число которых составило 221, чаще всего использовалась замена членов предложения (50 раз, 23% от общего количества примененных трансформаций), а именно преобразование активных конструкций пассивными, замена личных

предложений безличными, замена подлежащего обстоятельством или дополнением и наоборот; добавление (47 раз, 21%); замена части речи (43 раза, 19%): замена существительного глаголом, местоимения и причастия существительными; опущение (35 раз, 16%) и замена типа синтаксической связи (27 раз, 12%): замена простых предложений сложными и замена сочинительных связей подчинительными и наоборот. Наименее частотные грамматические трансформации, такие как объединение и членение предложений, перестановка и замена грамматического числа представлены в приложении 1.

Обратимся к диаграмме 1.

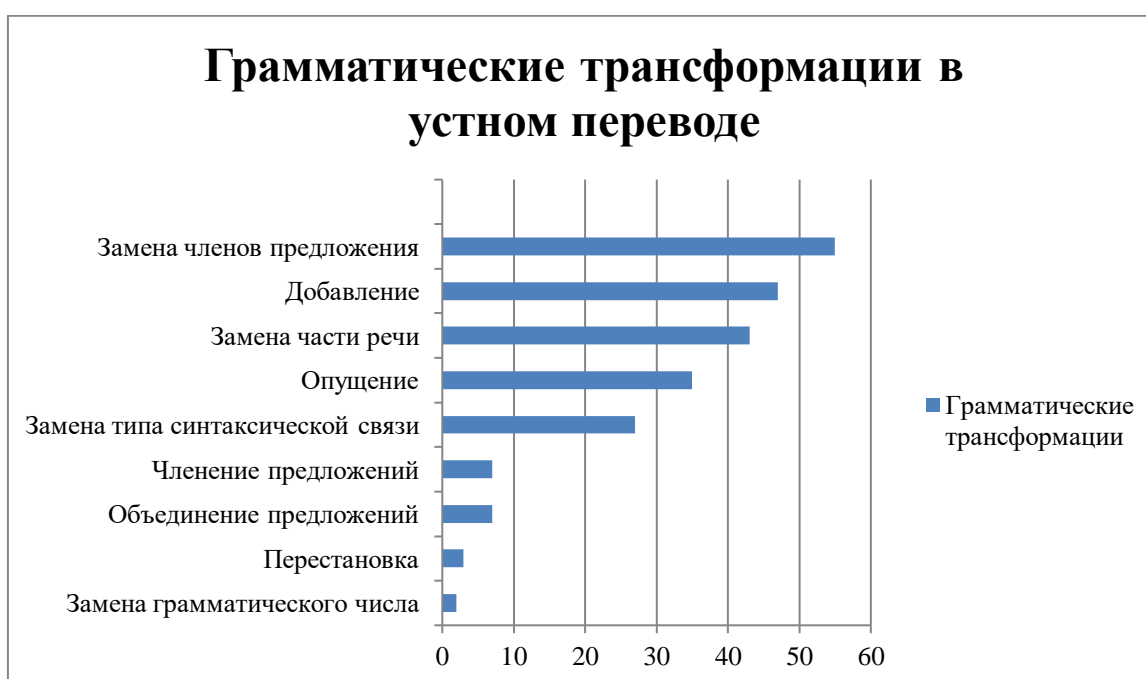


Диаграмма 1 – Грамматические трансформации в устном переводе

Согласно приведенным данным на диаграмме 1, можно сделать вывод, что замена членов предложения и добавление являлись наиболее употребляемыми грамматическими трансформациями в устном научно-техническом переводе. Поскольку устный перевод считается одним из самых сложных, он требует частого применения трансформаций, вследствие чего переводчику необходимо менять определенные члены предложения, перестраивая структуру предложения, и добавлять комментарии, чтобы информация была понятна реципиенту. Также можно

отметить, что частыми грамматическими трансформациями является замена части речи и опущение. Это обуславливается тем, что необходимо опускать определенные детали, не влияющие на общий смысл высказывания, чтобы успевать за говорящим, и изменять части речи для конкретизации мысли. Использование указанных выше грамматических трансформаций позволяет достигнуть эквивалентности при устном научно-техническом переводе. Кроме того, для достижения адекватности перевода, переводчику необходимо сохранять информативность и скорость перевода, что приводит к применению таких трансформаций, как добавление и опущение.

## 2.2 Применение грамматических трансформаций в письменном переводе

В качестве письменных материалов для исследования были подобраны: книга Антонопулоса Андреаса М. «Осваиваем биткойн. Программирование блокчейна» (Andreas M. Antonopoulos «Mastering the Bitcoin. Programming the Open Blockchain») и статья Сатоши Накамото, создателя криптовалюты биткойн, «Биткойн: система цифровой пиринговой наличности» (Satoshi Nakamoto «Bitcoin: A Peer-to-Peer Electronic Cash System»).

Для начала проанализируем статью Сатоши Накамото «Биткойн: система цифровой пиринговой наличности» (Satoshi Nakamoto «Bitcoin: A Peer-to-Peer Electronic Cash System»). Авторами любительского перевода являются arvicco и grich, официальный перевод отсутствует.

Обратимся к таблице 2.

Таблица 2 – Биткойн: система цифровой пиринговой наличности

Оригинал	Перевод	Грамматическая трансформация
The longest chain not only	Самая длинная версия	Замена членов

<p>serves as proof of the sequence of events witnessed, but proof that <u>it came from the largest pool of CPU power.</u></p>	<p>цепочки служит не только подтверждением очередности событий, но и доказывает, что <u>над ней произвел работу самый большой вычислительный сегмент сети.</u></p>	<p>предложения.</p>
<p>The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash.</p>	<p>Метка времени показывает, что в данный момент конкретные данные существовали и потому попали в хэш блока.</p>	<p>Замена типа синтаксической связи.</p>
<p>The proof-of-work involves scanning for a value that when <u>hashed (1, 2)</u>, such as with SHA-256, the hash begins with a number of zero bits.</p>	<p>Суть заключается в поиске такого значения, чей <u>хэш (1, 2)</u> (например, SHA-256) начинался бы с некоторого числа нулевых битов.</p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена членов предложения.</li> </ol>
<p><u>The average work required is exponential in the number of zero bits required (1)</u> and can be verified by <u>executing (2)</u> a single hash (3).</p>	<p><u>Требуется выполнить объем работы, экспоненциально зависящий от числа нулей (1), но для проверки найденного значения (4) достаточно вычислить (2) лишь один хэш (3).</u></p>	<ol style="list-style-type: none"> <li>1. Замена типа синтаксической связи.</li> <li>2. Замена членов предложения.</li> <li>3. Замена части речи.</li> <li>4. Добавление.</li> </ol>
<p>Once the CPU effort has been expended to make it satisfy (1) the proof-of-work (2), the block cannot be changed <u>without redoing the work (3).</u></p>	<p>Как только блок, <u>удовлетворяющий (1) условию, найден (2), его содержимое нельзя изменить, не выполнив заново всей работы (3).</u></p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена членов предложения.</li> <li>3. Замена типа синтаксической связи</li> </ol>
<p><u>To compensate (1) for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour.</u></p>	<p><u>Для компенсации (1) возрастающей вычислительной мощности процессоров и колебания числа работающих узлов в сети, сложность хэширования должна изменяться, чтобы обеспечивать равномерную скорость генерации блоков.</u></p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена членов предложения.</li> </ol>

<u>To facilitate this without breaking the block's hash,</u> transactions are hashed in a Merkle Tree, <u>with only the root included in the block's hash (2).</u>	<u>Чтобы хэш блока остался неизменным,</u> все транзакции в блоке хранятся в виде хэш-дерева Меркла <u>и лишь его корень</u> включается в хэш блока (2).	1. Замена членов предложения. 2. Замена типа синтаксической связи.
A block header <u>with no transactions</u> would be about 80 bytes.	Заголовок <u>пустого</u> блока будет составлять около 80 байт.	Замена части речи.
<u>If we suppose blocks are generated every 10 minutes,</u> 80 bytes * 6 * 24 * 365 = 4.2MB per year.	<u>Из расчета скорости генерации блока раз в десять минут</u> получаем $80*6*24*365=4.2$ Мб в год.	Замена членов предложения.
<u>They vote with their CPU power, expressing their acceptance of valid blocks (1) by working on extending them (3) (2) and rejecting invalid blocks by refusing to work on them (2).</u>	<u>Они выражают свое согласие</u> <u>принять</u> <u>корректный блок в цепочку,</u> <u>используя свои</u> <u>вычислительные мощности</u> (1) <u>для удлинения этой цепи</u> (3) (2), или <u>несогласие</u> (если блок содержит <u>неверные данные</u> ) (4), <u>не продолжая эту цепочку</u> (2).	1. Замена типа синтаксической связи. 2. Замена членов предложения. 3. Замена части речи. 4. Добавление.

Обратимся к таблице 3, перевод книги Антонопулоса Андреаса М. «Осваиваем биткойн. Программирование блокчейна» (Andreas M. Antonopoulos «Mastering the Bitcoin. Programming the Open Blockchain»). Перевод книги осуществил Снастин А. В.

Таблица 3 – Осваиваем биткойн. Программирование блокчейна

Оригинал	Перевод	Грамматические трансформации
Bitcoin are created through a process called «mining,» which involves competing <u>to find</u> solutions to a mathematical problem while processing bitcoin transactions.	Биткойны создаются с помощью процесса, называемого майнингом (mining), который подразумевает конкуренцию <u>в поиске</u> решений для математической задачи при обработке транзакций	Замена части речи.

	биткойнов.	
Any participant in the bitcoin network (i.e., <u>anyone using a device running the full bitcoin protocol stack</u> ) may operate as a miner, using their computer's processing power to verify and record transactions.	Любой член биткойн-сети (то есть <u>любой, использующий устройство, на котором работает полный стек протоколов биткойна</u> ) может выступить в роли майнера (miner), используя вычислительные мощности своего компьютера для проверки и фиксации транзакций.	Замена типа синтаксической связи.
A full client, or «full node,» is a client that stores the entire history of bitcoin transactions ( <u>every transaction by every user, ever</u> ), manages users' wallets, and can initiate transactions directly on the bitcoin network.	Полноценный клиент (full client), или «полноценный узел» (full node), – это клиент, который хранит полную хронологию транзакций биткойнов ( <u>каждую транзакцию, когда-либо выполненную любым пользователем</u> ), управляет кошельками пользователей и может непосредственно начать выполнение транзакций в биткойн-сети.	Замена членов предложения.
Her bitcoin address is simply a number <u>that corresponds (1, 2) to a key that she can use to control (1) access to the funds.</u>	Этот биткойн-адрес – пока просто число, <u>соответствующее (1, 2) ключу, который Алиса может использовать для управления (1) доступом к своим денежным средствам.</u>	1. Замена части речи. 2. Замена членов предложения.
This allows a bitcoin wallet application to prefill the information used <u>to send (1) the payment while showing a human-readable description to the user (2).</u>	Это позволяет приложению биткойн-кошелек заполнить заранее информационные поля, используемые <u>при отправке (1) платежа, с передачей пользователю описания, понятного человеку (2).</u>	1. Замена части речи. 2. Замена членов предложения.
The bitcoin network is a peer- to-peer network, <u>with</u>	Биткойн-сеть – это пиринговая сеть, <u>в которой</u>	Замена членов предложения.



<u>each bitcoin client participating by connecting to several other bitcoin clients.</u>	<u>каждый биткойн-клиент соединен с несколькими другими биткойн-клиентами.</u>	
The purpose of the bitcoin network is <u>to propagate</u> transactions and blocks to all participants.	Задача биткойн-сети – <u>распространения</u> транзакций и блоков между всеми членами сети.	Замена части речи.
<u>Make sure (1) you have enough disk space, bandwidth, and time to complete (2) the initial synchronization.</u>	<u>Вы должны быть абсолютно уверены (1) в том, что действительно имеете достаточное дисковое пространство, соединение с высокой пропускной способностью и время для завершения (2) начальной синхронизации.</u>	1. Замена членов предложения. 2. Замена части речи.
<b><u>When the git cloning operation has completed (1), you will have a complete local copy (2) of the source code repository in the directory bitcoin.</u></b>	<b><u>После завершения операции клонирования командой git (1) вы получаете полную локальную копию (2) репозитория исходного кода в каталоге bitcoin.</u></b>	1. Замена членов предложения. 2. Замена типа синтаксической связи.
By default, the local copy will be synchronized with the most recent code, which might be an unstable or beta version of bitcoin.	По умолчанию локальная копия синхронизируется с самым свежим исходным кодом, который может быть нестабильным или <u>представлять</u> бета-версию биткойна.	Добавление.
<u>Before compiling the code, select a specific version by checking out a release tag.</u>	<u>Перед компиляцией кода выберите конкретную версию, явно указав номер релиза tag.</u>	Замена типа синтаксических связей.
The autogen.sh script creates a set of automatic configuration scripts that will interrogate your system to discover the correct settings and <b><u>ensure you have all (1) the necessary libraries to compile (2) the</u></b>	Скрипт autogen.sh создает набор скриптов автоматического конфигурирования, которые опрашивают целевую систему, чтобы получить правильные параметры конфигурации и	1. Замена типа синтаксической связи. 2. Замена части речи. 3. Замена членов предложения.

code (3).	<b><u>убедиться в том, что все (1) необходимые для компиляции (2) кода библиотеки установлены (3).</u></b>	
Next, <u>run the configure script (1) to automatically discover (2) all the necessary libraries and create (2) a customized build script for your system (3).</u>	После этого <u>запускается скрипт configure (1) для автоматического определения (2) наличия всех требуемых библиотек и создание (2) скрипта сборки, специализированного для конкретной системы (3).</u>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена части речи.</li> <li>3. Замена типа синтаксической связи.</li> </ol>
When you first run bitcoind (1, 2), it will remind you (1) to create (3) a configuration file with a strong password for the JSON-RPC interface.	При первом запуске bitcoind (1, 2) <u>будет выведено напоминание (1) о необходимости создания (3) файла конфигурации с надежным паролем для интерфейса JSON-RPC.</u>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена типа синтаксической связи.</li> <li>3. Замена части речи.</li> </ol>
<b>Once</b> you are happy with the configuration <u>options you have selected (1), (2) you should add (1) bitcoin to the startup scripts in your operating system, so that it (3) runs continuously and restarts when (2) the operating system restarts (1).</u>	<b>Если</b> вы вполне удовлетворены <u>выбранными значениями (1) параметров конфигурации, то (2) следует добавить (1) команду запуска биткойн-системы в скрипты автозапуска вашей операционной системы, чтобы программа Bitcoin Core (3) работала постоянно и автоматически перезапускалась при (2) каждой перезагрузке операционной системы (1).</u>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена типа синтаксической связи.</li> <li>3. Замена части речи.</li> </ol>
This is known as «transaction malleability,» because transaction hashes can be modified prior to confirmation in a block.	Это <u>состояние</u> известно под названием «деформативность транзакции» (transaction malleability), потому что транзакция может быть изменена до момента ее подтверждения в блоке.	Добавление.

<p><u>To decode (1) that, (3) we use the decoderawtransaction command (2), passing the hex data as a parameter (2).</u></p>	<p>Для ее <u>расшифровки (1) (3) используется команда decoderawtransaction (2)</u>, в которую как параметр <u>передаются данные в шестнадцатеричном формате (2).</u></p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена членов предложения.</li> <li>3. Замена типа синтаксических связей.</li> </ol>
<p>These are implemented in a variety of programming languages, <u>offering programmers (1, 2) native interfaces in their (2) preferred (1) language.</u></p>	<p>Они написаны на различных языках программирования и <u>предлагают (1, 2) разработчикам интерфейсы, специализированные для конкретных языков программирования (3), предпочитаемых (1) тем или иным программистом (2).</u></p>	<ol style="list-style-type: none"> <li>1. Замена типа синтаксической связи.</li> <li>2. Замена части речи.</li> <li>3. Добавление.</li> </ol>
<p>The digital signature used to <u>spend (1) funds is (2) also referred to as a witness, a term used in cryptography.</u></p>	<p>Цифровая подпись применяется <u>при расходовании (1) денежных средств и (2) также обозначается термином «свидетельство» (witness), используемым в криптографии.</u></p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена типа синтаксической связи.</li> </ol>
<p>In bitcoin, <u>we use public key cryptography (1) to create (2) a key pair that controls access to bitcoin (1, 3).</u></p>	<p>В биткойн-системе <u>криптография с открытым ключом применяется (1) для создания (2) пары ключей, управляющих доступом (1, 3).</u></p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена части речи.</li> <li>3. Замена типа синтаксической связи.</li> </ol>
<p><u>When spending bitcoin (1), the current bitcoin owner presents her public key and a signature (different each time, but created from the same private key) in a transaction to spend those bitcoin (1, 3).</u></p>	<p>При <u>расходовании биткойнов (1) текущий владелец предоставляет свой открытый ключ и цифровую подпись (подписи (2) различные для каждого случая, но создаются с помощью одного и того же секретного ключа) в транзакции, определяющей расход биткойнов (1, 3).</u></p>	<ol style="list-style-type: none"> <li>1. Замена типа синтаксической связи.</li> <li>2. Добавление.</li> <li>3. Замена членов предложения.</li> </ol>

<p>From the public key (K), <u>we use a one-way cryptographic hash function (1) to generate (2) a bitcoin address (A).</u></p>	<p>К полученному открытому ключу (K) <u>применяется односторонняя криптографическая хэш-функция (1) для генерации (2) биткойн-адреса (A).</u></p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена части речи.</li> </ol>
<p>However, <u>anyone with access (1) to the public key and the transaction fingerprint can use them to verify the signature (1).</u></p>	<p>Но <u>каждый может получить доступ к открытому ключу (1), а (2) цифровой отпечаток транзакции в сочетании с открытым ключом (3) позволяет проверить достоверность подписи (1).</u></p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена типа синтаксической связи.</li> <li>3. Добавление.</li> </ol>
<p>Usually, the OS random number generator is initialized by a human source of randomness, which is why <u>you may be asked to wiggle your mouse around for a few seconds (2).</u></p>	<p>Обычно генератор случайных чисел ОС инициализируется некоторым произвольным значением, заданным человеком, вот почему, например, <u>ОС (1) может попросить вас подвигать мышь в течение нескольких секунд (2).</u></p>	<ol style="list-style-type: none"> <li>1. Добавление.</li> <li>2. Замена членов предложения.</li> </ol>
<p><u>The reverse operation, known (1) as “finding the discrete logarithm” (2) – calculating k if (1) you know K (2) – is as difficult as trying all possible values of k (2), i.e., a brute-force search.</u></p>	<p><u>Обратная операция под названием (1) «поиск дискретного логарифма» (2) – вычисление k по известному значению K (1, 2) – является вычислением (3), трудоемкость которого сравнима с перебором всех известных значений k (2), то есть с методикой полного перебора (3) (или «грубой силы»).</u></p>	<ol style="list-style-type: none"> <li>1. Замена типа синтаксической связи.</li> <li>2. Замена членов предложения.</li> <li>3. Добавление.</li> </ol>
<p>This mathematical trick becomes the basis for <u>unforgeable</u> and secure digital signatures that prove ownership of bitcoin funds.</p>	<p>Этот хитрый математический прием стал основой для надежно защищенных цифровых подписей, <u>которые невозможно подделать</u> и которые служат доказательством права</p>	<p>Замена типа синтаксических связей.</p>

	владения биткойнами.	
Because this curve is defined over a finite field of prime order instead of over the real numbers, it looks like a pattern of dots scattered in <u>two dimensions</u> (2), which <u>makes it difficult to visualize</u> (3).	Поскольку эта кривая определена над конечным полем простого порядка, а не над полем действительных <u>(или вещественных)</u> (1) чисел, она выглядит как набор точек, распределенных по <u>двумерному пространству</u> (2), что <u>затрудняет ее визуальное представление</u> (3).	1. Добавление. 2. Замена части речи. 3. Замена членов предложения.
In order to <u>represent</u> (1) long numbers in a compact way, <u>using</u> (1) <u>fewer symbols</u> , many (2) computer systems use mixed-alphanumeric representations with a base (or radix) higher than 10.	Для более компактного <u>представления</u> (1) длинных чисел <u>с использованием</u> (1) <u>меньшего количества символов</u> многие (2) компьютерные системы используют смешанные алфавитно-цифровые представления с основанием (base или radix), большим 10.	1. Замена части речи. 2. Замена типа синтаксической связи.
<u>When presented with Base58Check code</u> , the decoding software will calculate the checksum of the data and compare it to the checksum included in the code.	<u>При получении кода в формате Base58Check</u> декодирующее программное обеспечение вычисляет контрольную сумму данных и сравнивает ее с контрольной суммой, включенной в код.	1. Замена типа синтаксической связи. 2. Замена членов предложения.
<u>To convert</u> (1) data (a number) into a Base58Check format, <u>we first add a prefix to the data</u> (2), called the «version byte,» which serves to <u>easily identify</u> (1) the type of <u>data that is encoded</u> (3).	<u>Для преобразования</u> (1) данных (числа) в формат Base58Check <u>сначала к данным добавляется префикс</u> (2), называемый байтом версии (version byte), который служит для <u>простой идентификации</u> (1) <u>типа кодируемых данных</u> (3).	1. Замена части речи. 2. Замена членов предложения. 3. Замена типа синтаксической связи.
Next, <u>we compute the «double-SHA» checksum</u> ,	Далее <u>вычисляется</u> <u>двукратная</u> <u>SHA</u>	1. Замена членов предложения.

<p><u>meaning (2) we apply the SHA256 hashalgorithm (2) twice (3) on the previous result (prefix and data).</u></p>	<p>контрольная сумма, то есть (2) к предыдущему результату (префикс и данные) <u>два раза (3) применяется хэш-алгоритм SHA256 (2).</u></p>	<p>2. Замена типа синтаксической связи. 3. Замена части речи.</p>
<p>In bitcoin, most of the data presented to the user is <u>Base58Check-encoded to make it compact, easy to read (2), and easy to detect (2) errors (1).</u></p>	<p>В биткойн-системе большинство данных предъявляется пользователю <u>в кодировке Base58Check для компактности представления, удобства чтения (2) и быстрого обнаружения (2) ошибок (1).</u></p>	<p>1. Замена членов предложения. 2. Замена части речи.</p>
<p>The version prefix in Base58Check encoding is used <u>to create (1) easily distinguishable formats, which when encoded (2) in Base58 contain specific characters at the beginning of the Base58Check-encoded payload.</u></p>	<p>Префикс версии в кодировке Base58Check используется <u>для создания (1) легко различаемых форматов, которые при кодировании (2) в Base58 содержат специальные символы в самом начале блока основных закодированных данных.</u></p>	<p>1. Замена части речи. 2. Замена членов предложения.</p>
<p>The Bitcoin Explorer commands make it easy <u>to write shell scripts and command-line «pipes» that manipulate (2, 3) bitcoin keys, addresses, and transactions.</u></p>	<p>Команды проводника Bitcoin Explorer упрощают <u>написание скриптов командной оболочки и конвейеров в командной строке для обработки (2, 3) ключей, адресов и транзакций биткойна.</u></p>	<p>1. Замена части речи. 2. Замена типа синтаксической связи. 3. Замена членов предложения.</p>
<p><u>You can use (1) Bitcoin Explorer to decode (2) the Base58Check format on the command line.</u></p>	<p><u>Можно воспользоваться (1) проводником Bitcoin Explorer для декодирования (2) из формата Base58Check в командной строке.</u></p>	<p>1. Замена членов предложения. 2. Замена части речи.</p>
<p>Here's the public key generated by the private key <u>we created earlier</u>, shown as the coordinates x and y.</p>	<p>Открытый ключ, сгенерированный по секретному ключу, <u>созданному ранее</u>, показан</p>	<p>Замена членов предложения.</p>

	в виде пары координат.	
So, while <u>we can omit</u> (1) the y coordinate <u>we have to store</u> (1) the sign of y (positive or negative); or in other words, <u>we have to remember</u> (1) if <u>it</u> (2) was above or below the x-axis because each of those options represents a different point and a different public key.	Таким образом, несмотря на то что <u>координата у пропускается</u> (1), <u>необходимо хранить</u> (1) ее знак (плюс или минус), другими словами, <u>необходимо запомнить</u> (1), располагается ли <u>точка</u> (2) выше или ниже оси X, потому что каждый вариант представляет совершенно различные точки, <u>соответственно</u> (3), и различные открытые ключи.	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена части речи.</li> <li>3. Добавление.</li> </ol>
Compressed public keys are gradually becoming the default across bitcoin clients, <u>which</u> (1) is having a significant impact on reducing the size of transactions and therefore the blockchain.	Сжатые открытые ключи постепенно становятся форматом по умолчанию для всех пользователей биткойна, <u>и это</u> (1) оказывает значительное воздействие на сокращение размеров транзакций, следовательно, <u>и на уменьшение размера структуры данных</u> (2) блокчейна.	<ol style="list-style-type: none"> <li>1. Замена типа синтаксической связи.</li> <li>2. Добавление.</li> </ol>
This is especially important when a wallet application is importing private keys from another bitcoin wallet application, because the new wallet needs to scan the blockchain <u>to find</u> (2) transactions corresponding to these imported keys.	Это особенно важно, если приложение кошелек импортирует секретные ключи из другого приложения биткойн-кошелек, потому что новый кошелек должен просканировать <u>структуру данных</u> (1) блокчейна <u>в поисках</u> (2) транзакций, соответствующих этим импортируемым ключам.	<ol style="list-style-type: none"> <li>1. Добавление.</li> <li>2. Замена части речи.</li> </ol>

В письменном научно-техническом переводе с английского на русский язык всего было рассмотрено 100 примеров в таблицах 2 и 3 с использованием 206 грамматических трансформаций. Согласно

статистике, наиболее часто использовались такие грамматические трансформации как замена членов предложения (66 раз, 32% от общего количества использованных трансформаций), а именно: замена личных предложений безличными, замена пассивных конструкций активными, и наоборот, и замена второстепенных членов предложения главными; замена части речи (61 раз, 30%): замена местоимений, глаголов и наречий существительными; и замена типа синтаксических связей (45 раз, 22%): замена сложных предложений простыми предложениями и наоборот, а также замена сочинительных связей подчинительными и наоборот. Наименее частотные грамматические трансформации, такие как добавление и опущение, объединение и членение предложений, перестановка и замена грамматического числа и времени представлены в приложении 2 и 3.

Обратимся к диаграмме 2.

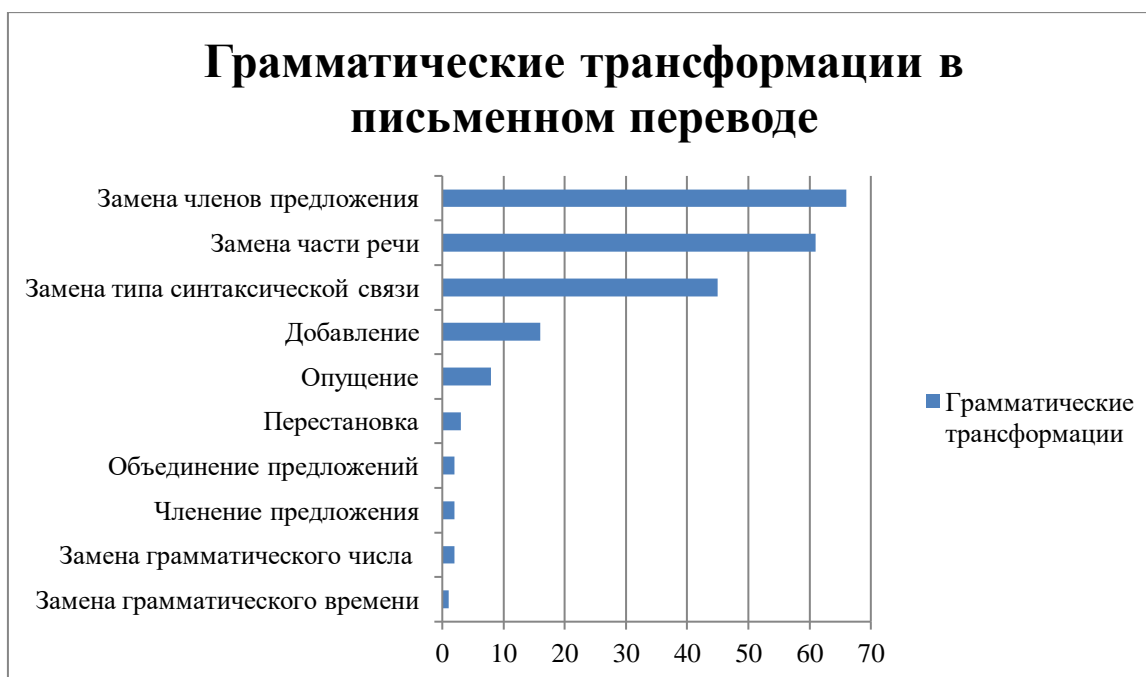


Диаграмма 2 – Грамматические трансформации в письменном переводе

По приведенным данным на диаграмме 2 можно отметить, что применение такой грамматической трансформации как замена членов предложения является наиболее часто употребляемым. Это обуславливается тем, что построение предложений на английском языке



может значительно отличаться от русского языка, поэтому члены предложения могут меняться достаточно часто при переводе. Замена части речи также является грамматической трансформацией, к которой переводчик нередко вынужден прибегать в процессе перевода, поскольку, как показало исследование, во многих случаях местоимения, глаголы и наречия в английском языке заменяются существительным при переводе на русский язык. На третьем месте стоит замена типа синтаксической связи. Переводчику необходимо использовать эту грамматическую трансформацию, поскольку синтаксические связи английского и русского языка могут часто различаться. Поэтому для соблюдения грамматических правил русского языка при переводе следует обратиться к данной грамматической трансформации. Менее употребляемыми видами в письменном научно-техническом переводе оказались добавления и опущения. Причиной могут быть особенности научно-технических текстов, где нежелательно прибегать к вольностям и необходимо передавать все содержимое текста оригинала.

По результатам выборок мы выяснили, что и в устном переводе и в письменном переводе на тему блокчейна преимущественно использовались такие грамматические трансформации как замена членов предложения, замена части речи, добавление, опущение и замена типа синтаксической связи. В основе таких переводческих решений лежат различия в грамматическом строе английского, как аналитического, и русского, как синтетического, языков. При этом в устном переводе, в отличие от письменного, достаточно часто использовались также добавление и опущение. Это объясняется тем, что в устном переводе переводчику необходимо добавлять или опускать детали, не влияющие на передачу главной мысли, так как приоритетом устного переводчика является не только перевод, как таковой, но и соблюдение строгих временных рамок. Наименее востребованными трансформациями в устном переводе явились объединение и членение предложений, перестановка и

замена грамматического числа. В письменном переводе реже использовались такие грамматические трансформации как добавление, опущение, членение и объединение предложений, перестановка и замена грамматического числа и времени.

### 2.3 Сопоставление грамматических трансформаций в устном и письменном переводе научно-технических материалов

Процентное соотношение анализа грамматических трансформаций устного и письменного научно-технического перевода представлено в наглядном виде в таблице 4.

Обратимся к таблице 4.

Таблица 4 – Грамматические трансформации в устном и письменном научно-техническом переводе

Грамматическая трансформация	Устный перевод	Письменный перевод
Замена членов предложения.	50 раз, 23% от общего количества примененных трансформаций, а именно преобразование активных конструкций пассивными, замена личных предложений безличными, замена подлежащего обстоятельством или дополнением и наоборот.	66 раз, 32% от общего количества использованных трансформаций, а именно: замена личных предложений безличными, замена пассивных конструкций активными, и наоборот, и замена второстепенных членов предложения главными.
Замена части речи.	43 раза, 19%: замена существительного глаголом, местоимения и причастия существительными.	61 раз, 30%: замена местоимений, глаголов и наречий существительными.
Замена типа синтаксической связи.	27 раз, 12%: замена простых предложений сложными и замена сочинительных связей	45 раз, 22%: замена сложных предложений простыми предложениями и

	подчинительными и наоборот.	наоборот, а также замена сочинительных связей подчинительными и наоборот.
Добавление.	47 раз, 21%.	—
Опущение.	35 раз, 16%.	—

Для более грамотного употребления грамматических трансформаций была составлена схема, которой специалист может придерживаться при анализе и переводе устных и письменных научно-технических материалов:

1) в устном переводе учитывать избыточность речи говорящего и применять опущения для таких грамматических элементов как местоимения, вводные конструкции, слова-паразиты, избыточная информация;

2) при появлении в устном высказывании терминов и реалий, незнакомых реципиенту переводного текста, применять грамматическую трансформацию добавления, например, формулировать придаточные предложения;

3) при наличии местоимений, сложных прилагательных и глаголов в инфинитивной форме в устном и письменном научно-техническом переводе с английского языка на русский, применять замену части речи, например, производить замену на существительное для уточнения информации и адаптации структуры фраз к строю русского языка;

4) учитывать различия в грамматическом строе английского, как аналитического, и русского, как синтетического, языка и применять такую грамматическую трансформацию как замена типа синтаксической связи, например, заменять придаточное предложение на главное, сочинительную связь на подчинительную и преобразовывать простые предложения в сложные и наоборот для создания понятного реципиенту и логически связного сообщения;

5) при наличии пассивных конструкций, избыточных применений местоимений и т.д., необходимо применять замену членов предложения, например, преобразовать пассивную конструкцию в активную, личное предложение в безличное, а также при необходимости производить замену второстепенных членов предложения главными. Данные трансформации обеспечивают достижение высокой информативности, четкой логики, структуры, точности и ясности переводного текста.

Данная схема позволяет сделать правильный анализ научно-технического текста и успешно применять разнообразные грамматические трансформации в устном и письменном переводе.

## ВЫВОДЫ ПО ВТОРОЙ ГЛАВЕ

Проанализировав грамматические трансформации, использовавшиеся в устном и письменном переводе научно-технических материалов на тему блокчейна, мы пришли к следующим выводам:

1. В устном научно-техническом переводе использовалось сразу несколько грамматических трансформаций, а именно: замена членов предложения, замена части речи, замена типа синтаксической связи, добавления, опущение и т.д., в одной фразе, ввиду того что данный вид перевода является одним из самых трудных, поскольку требует не только наличия отличного знания иностранного и родного языков и тематики, но и умения быстро, полно и доступно передавать смысл сказанного, адаптируя новые термины и неизвестные реалии для правильного их восприятия реципиентом на языке перевода.

2. В письменном переводе, в отличие от устного перевода, применялось меньшее количество грамматических трансформаций, поскольку этот вид перевода предоставляет переводчику большее количество времени на изучение текста и сам перевод в целом, позволяя ему более полно воспроизвести исходный текст на языке перевода. Но они также необходимы в процессе перевода, в частности такие трансформации как замена членов предложения, замена части речи и замена типа синтаксической связи.

3. Частота использования таких видов грамматических трансформаций как замена членов предложения (в частности, преобразование активных конструкций пассивными, замена личных предложений безличными, замена подлежащего обстоятельством или дополнением и наоборот) и замена части речи (в частности, замена существительного глаголом, местоимения, глагола и причастия существительными) в научно-техническом переводе на тему блокчейна применялось чаще всех остальных трансформаций.

4. Устный и письменный перевод научно-технических материалов не может быть осуществлен без какого-либо использования грамматических трансформаций, так как это важная составляющая в переводе данного стиля текстов. Данная необходимость вызвана такими особенностями научного стиля в английском языке как наличие менее формальных лексических и грамматических единиц и стремление к краткости и компактности изложения, что выражается особенно в достаточно широком использовании эллиптических конструкций. Эти явления несвойственны соответствующему стилю в русском языке, где требуется строгая формальность и широко используются развернутые сложносочиненные и сложноподчиненные предложения. Следовательно, переводчику было нужно адаптировать англоязычные научные тексты к нормам русского научно-технического функционального стиля.

5. На основании проведенного сопоставительного анализа нами были сформулированы практические рекомендации, которых специалист может придерживаться при переводе устных и письменных научно-технических материалов: в устном переводе учитывать избыточность речи говорящего и применять опущения для грамматических элементов; при появлении в устном высказывании терминов и реалий, незнакомых реципиенту переводного текста, применять грамматическую трансформацию добавления; при наличии местоимений, сложных прилагательных и глаголов в инфинитивной форме в устном и письменном научно-техническом переводе, применять замены частей речи; учитывать различия в грамматическом строе английского, как аналитического, и русского, как синтетического, языков и применять такую грамматическую трансформацию как замена типа синтаксической связи; при наличии пассивных конструкций, избыточных применений местоимений и т.д., необходимо применять замену членов предложения.

## ЗАКЛЮЧЕНИЕ

Проведенное исследование было нацелено на выявление наиболее частотных грамматических трансформаций в устном и письменном научно-техническом переводе. Для достижения этой цели требовалось решение ряда задач.

В рамках работы над теоретической главой исследования нами был осуществлен сбор и анализ теоретической информации на тему рассмотрения существующих классификаций грамматических трансформаций, выявления специфики устного и письменного научно-технического перевода и рассмотрения особенностей грамматического строя английского и русского языков и их влияния на процесс перевода.

На основе изученной теоретической информации мы пришли к промежуточным выводам и на их основе приступили к решению практических задач, поставленных нами в начале исследования, а именно: анализ и определение наиболее часто встречающихся грамматических переводческих трансформаций в материалах научно-технического характера на тему блокчейна, сопоставление особенностей применения грамматических трансформаций в устном и письменном переводе, формулирование на базе полученных данных практических рекомендаций по переводу текстов на тему блокчейна с английского на русский язык.

Мы проанализировали 200 примеров в устном и письменном научно-техническом переводе из лекций профессора Гари Генслера «Блокчейн и деньги» (Gary Gensler «Blockchain and money») в переводе Cryptus, книги Антонопулоса Андреаса М. «Осваиваем биткойн. Программирование блокчейна» (Andreas M. Antonopoulos «Mastering the Bitcoin. Programming the Open Blockchain») в переводе А. В. Снастина и статьи Сатоши Накамото, создателя криптовалюты биткойн, «Биткойн: система цифровой пиринговой наличности» (Satoshi Nakamoto «Bitcoin: A Peer-to-Peer Electronic Cash System») в переводе arvicco и grich.

На основе проведенного анализа мы сформулировали практические рекомендации для студентов и профессиональных переводчиков, которые в дальнейшем могут облегчить процесс обучения и профессиональной переводческой деятельности. Таким образом, все поставленные задачи были успешно выполнены и достигнута цель исследования

Научно-технические тексты характеризуются логической последовательностью изложения, упорядоченной системой связей между частями высказывания. В данных типах текста автор стремится добиться сжатости, точности, однозначности путем сохранения насыщенности содержания в таких текстах, что является также необходимостью для переводчика. При переводе научно-технических текстов одинаковой тематики используемые грамматические трансформации в устном и письменном переводе практически не различаются.

Поскольку устный перевод, в отличие от письменного, ограничивал переводчика во времени и скорости, в нем чаще использовались опущения и добавления, а в письменном переводе они встречались реже, но, тем не менее, присутствовали.

Существуют различные мнения авторов о разграничении видов грамматических трансформаций, но при этом существуют общепринятые способы классификации, на которые мы опирались, на синтаксическом уровне, такие как синтаксическое уподобление, перестановки, замена членов предложения, преобразование активных конструкций в пассивные и наоборот, членение предложения, объединение предложений, замена типа синтаксической связи, опущения, дополнения. Также они могут происходить на морфологическом уровне: замены частей речи, грамматическая замена числа, замена грамматического времени.

Значительные особенности устного и письменного научно-технического перевода – это краткость, лаконичность, логичность, но в то же время и полное изложение материала, в котором не должна употребляться эмоционально насыщенная лексика, присущая другим



стилям текстов. При этом в английских и русских научно-технических функциональных стилях есть свои особенности: наличие менее формальных лексических и грамматических единиц и стремление к краткости и компактности изложения, что выражается особенно в достаточно широком использовании эллиптических конструкций в английском. Это явление несвойственно соответствующему стилю в русском языке, где присутствует строгая формальность и широко используются развернутые сложносочиненные и сложноподчиненные предложения. На основе этого в устном переводе научно-технических материалов наиболее часто применялись замена членов предложения, добавления, замена частей речи, опущения и замена типа синтаксической связи. В письменном же переводе научно-технических текстов чаще применялись такие грамматические трансформации как замена членов предложения, замена частей речи и замена типа синтаксической связи.

Это позволило нам сделать выводы, что наиболее часто употребляемыми видами грамматических трансформаций являются замена членов предложения, замена части речи, замена типа синтаксической связи, опущение и добавление.

Исходя из результатов проделанной нами работы, можно сделать вывод, что использование грамматических трансформаций в устном и письменном научно-техническом переводе представляется многогранной теоретической и практической проблемой. Данный аспект научно-технического перевода требует подробного рассмотрения в рамках прикладного переводоведения и ознакомления студентов переводческих специальностей и профессиональных переводчиков с особенностями применения различных грамматических трансформаций в профессиональной деятельности.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Алексеева И. С. Введение в переводоведение : учебное пособие / И. С. Алексеева. – Москва : Изд. центр Академия, 2004. – 352 с. – ISBN 978-5-7695-4392-0.
2. Антонопулос А. М. Осваиваем биткойн. Программирование блокчейна / А. М. Антонопулос. – пер. с англ. А. В. Снастина. – Москва : ДМК Пресс, 2018. – 428 с. – ISBN 978-5-94074-965-3.
3. Бархударов Л. С. Язык и перевод: вопросы общей и частной теории перевода / Л. С. Бархударов. – Изд. 2-е. – Москва : Изд-во ЛКИ, 2008. – 235 с. – ISBN 978-5-382-00577-5.
4. Берков В. П. Мастерство перевода : Сб. 8 / В. П. Берков. – Москва : Ред. коллегия: А. Гатов [и др.], 1971. – 488 с. – ISBN 978-5-458-62273-8.
5. Бирюкова Л. В. Учись переводить с немецкого языка на русский : учеб. пособие / Л. В. Бирюкова. – Москва : Междунар. назависимый экол.-политол. ун-т. – Изд-во МНЭПУ, 1999. – 41 с.
6. Бреус Е. В. Основы теории и практики перевода с русского языка на английский. / Е. В. Бреус. – Москва : Изд-во УРАО, 2001. – 120 с.
7. Виноградов В. С. Перевод: общие и лексические вопросы : учеб. пособие / В. С. Виноградов. – 2-е изд., перераб. – Москва : КДУ, 2004. – 240 с. – ISBN 5-98227-018-0.
8. Воронцова Г. Н. Очерки по грамматике английского языка / Г. Н. Воронцова. – Москва : Изд-во литературы на иностранных языках, 1960. – 400 с.
9. Казакова Т. А. Аспекты теории письменного перевода: учеб. пособие к спецкурсу / Т. А. Казакова. – Москва : Свердл. гос. пед. ин-т, 1988. – 49 с.
10. Комиссаров В. Н. Современное переводоведение / В. Н. Комиссаров. – Москва : ЭТС, 1999. – 288с. – ISBN5-93386-015-8.

11. Мильчин А. Э. Методика редактирования теста / А. Э. Мильчин. – Изд. 3-е, перераб. и доп. – Москва : Логос, 2005. – 524 с. – ISBN 5-98704-033-7.
12. Мартынюк К. В. Сравнительные особенности грамматического строя в русском и английском языках на примере временных глаголов и артиклей / К. В. Мартынюк, А. Д. Туманова. – Текст : непосредственный // Молодой ученый, 2021. – № 29 (371 с.). – URL: <https://moluch.ru/archive/371/83275/> (дата обращения: 06.06.2022).
13. Попова Т. В. Методика обучения студентов неязыковых вузов письменному переводу научных текстов (на материале немецкого языка) / Т. В. Попова. – Санкт-Петербург : 2008. – 253 с. – URL: <https://new-disser.ru/avtoreferats/01004293382.pdf> (дата обращения: 17.04.2022).
14. Попова Т. В. Об особенностях письменного перевода / Т. В. Попова. – Санкт-Петербург : Международный научно-исследовательский журнал. – 2014. – URL: <https://cyberleninka.ru/article/n/ob-osobennostyah-pismennogo-perevoda> (дата обращения 15.04.2022).
15. Рецкер Я. И. Теория перевода и переводческая практика / Я. И. Рецкер. – Москва : Международные отношения, 1974. – 216 с.
16. Толстой С. С. Основы перевода с английского языка на русский / С. С. Толстой. – Москва : Ин-т междунар. отношений, 1957. – 57 с.
17. Херина А. А. Особенности научно-технического перевода / А. А. Херина. – Москва : Academy, 2015. – URL: <https://cyberleninka.ru/article/n/osobennosti-nauchno-tehnicheskogo-perevoda> (дата обращения 17.04.2022).
18. Швейцер А. Д. Теория перевода: Статус, проблемы, аспекты / А. Д. Швейцер. – Москва : Наука, 1988. – 212 с.
19. Аналитические и синтетические языки : [сайт]. – 2014. – URL: <https://sites.google.com/site/stgeneralinguist/analiticeskie-i-sinteticeskie-azyki> (дата обращения 15.05.2022).

20. Биткойн: система цифровой пиринговой наличности : [сайт]. – 2018. – URL: [https://bitcoin.org/files/bitcoin-paper/bitcoin\\_ru.pdf](https://bitcoin.org/files/bitcoin-paper/bitcoin_ru.pdf) (дата обращения: 07.05.2022).
21. Виды и особенности устного перевода : [сайт]. – 2017. – URL: [https://studwood.net/1430833/literatura/vidy\\_osobennosti\\_ustnogo\\_perevoda](https://studwood.net/1430833/literatura/vidy_osobennosti_ustnogo_perevoda) (дата обращения 15.04.2022).
22. Генслер Г. 1 лекция MIT – блокчейн и деньги / Г. Генслер [русская озвучка Cryptus]. – 2021. – URL: <https://www.youtube.com/watch?v=pZxp--SpWF8> (дата обращения: 06.05.2022).
23. Генслер Г. 3 лекция MIT – криптография и основы блокчейна / Г. Генслер [русская озвучка Cryptus]. – 2021. – URL: [https://www.youtube.com/watch?v=WEmdRTF5\\_IQ&t=3047s](https://www.youtube.com/watch?v=WEmdRTF5_IQ&t=3047s) (дата обращения: 15.05.2022).
24. Генслер Г. 4 лекция MIT – консенсус и основы блокчейна / Г. Генслер [русская озвучка Cryptus]. – 2021. – URL: <https://www.youtube.com/watch?v=j1XdgNbXSY0&t=1539s> (дата обращения: 16.05.2022).
25. Генслер Г. 5 лекция MIT – транзакции, UTXO, скрипты и основы блокчейна / Г. Генслер [русская озвучка Cryptus]. – 2021. – URL: [https://www.youtube.com/watch?v=Lu\\_LstTaohU](https://www.youtube.com/watch?v=Lu_LstTaohU) (дата обращения: 20.05.2022).
26. Генслер Г. 7 лекция MIT –технические испытания, блокчейн и деньги / Г. Генслер [русская озвучка Cryptus]. – 2021. – URL: <https://www.youtube.com/watch?v=ogCj24ip6uM> (дата обращения: 21.05.2022).
27. Грамматические особенности английского и русского языка : [сайт]. – 2018. – URL: <https://multiurok.ru/files/grammaticheskie-osobennosti-angliiskogo-i-russkogo.html> (дата обращения 16.05.2022).

28. Классификация грамматических трансформаций : [сайт]. – 2013. – URL: <https://study-english.info/article059.php> (дата обращения 30.04.2022).

29. Лекция: «Блокчейн и деньги» : [сайт]. – 2018. – URL: <https://ocw.mit.edu/courses/15-s12-blockchain-and-money-fall-2018/> (дата обращения: 05.05.2022).

30. Antonopoulos A. M. Mastering Bitcoin. Programming the Open Blockchain : [сайт]. – 2017. – URL: <https://archive.org/details/MasteringBitcoin2nd/page/n1/mode/2up> (дата обращения: 22.05.2022).

31. Satoshi Nakamoto Bitcoin: A Peer-to-Peer Electronic Cash System : [сайт]. – 2018. – URL: <https://bitcoin.org/bitcoin.pdf> (дата обращения: 07.05.2022).

## ПРИЛОЖЕНИЯ

### ПРИЛОЖЕНИЕ 1

Анализ перевода лекций Гари Генслера «Блокчейн и деньги» («Blockchain and money»)

Таблица 1 – Блокчейн и деньги

Оригинал	Перевод	Грамматическая трансформация
<p>It's the technology <u>or the science behind</u> (1) encryption and decryption <u>for fortification</u>, so how do you <u>encrypt a particular text</u> (1) <u>such that it's not readable</u> (2) by someone else without having the decryption code.</p>	<p>Это технология шифрования и расшифровки <u>той или иной информации</u>. То есть это используется для фортификации, <u>чтобы люди не могли прочесть</u> (2) ту или иную информацию без наличия определенного ключа или кода.</p>	<ol style="list-style-type: none"> <li>1. Опускание.</li> <li>2. Замена типа синтаксической связи.</li> </ol>
<p>So it's <u>time-stamped</u> (4) append logs, <u>meaning</u> (1, 3, 5) you can add a little bit of information to this. <u>And it's time-stamped</u> (4). <u>So these are these blocks being added</u> (2).</p>	<p>Это база данных <u>с отметками во времени</u> (4). (1) Это означает то (3), что (5) Вы можете добавлять туда небольшие куски информации. <u>И они будут содержать время</u> (4), <u>когда они были добавлены</u> (2).</p>	<ol style="list-style-type: none"> <li>1. Членение предложений.</li> <li>2. Объединение предложений.</li> <li>3. Замена членов предложения.</li> <li>4. Замена части речи.</li> <li>5. Замена типа синтаксической связи.</li> </ol>
<p>A modern payment system, you need to be <u>able</u> (1) to move about 100,000 payments a second. <u>You do</u> (1) about 30,000 transactions a second. And your computers have to be <u>resilient</u> (2) to 100,000</p>	<p>В современной платежной системе <u>нужно</u> (1) совершать где-то 100 тысяч транзакций в секунду. <u>Сейчас</u> (3) <u>проходит</u> (1) где-то 30 тысяч транзакций в секунду, однако компьютеры должны иметь</p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена части речи.</li> <li>3. Добавление.</li> <li>4. Объединение предложений.</li> </ol>

<p>transactions a second (4).</p>	<p><u>пропускную способность (2) в 100 тысяч (4).</u></p>	
<p>The Australian Stock Exchange is updating their clearing and settling. They announced <u>they're doing a blockchain project (1).</u> They're doing it with digital assets. <u>(2)</u></p>	<p>Австралийские фондовые биржи анонсировали, что они будут обновлять свою систему и переходить на более новые технические решения. (2)</p>	<ol style="list-style-type: none"> <li>1. Опускание.</li> <li>2. Объединение предложений.</li> </ol>
<p><u>And they're using the Hyperledger blockchain, which is an IBM software (1), open-source (2) software.</u> But the Australian Stock Exchange is going to put it on three computers, <b>which is called three nodes (4), that they control all three of them (3).</b></p>	<p><u>Тем самым они будут использовать блокчейн «Hyperledger». Это разработка компании IBM (1), разработка с открытым исходным кодом (2).</u> Они поставят ее на три компьютера, <u>но при этом все эти три компьютера они будут сами контролировать(3).</u></p>	<ol style="list-style-type: none"> <li>1. Членение предложений.</li> <li>2. Замена части речи.</li> <li>3. Замена типа синтаксической связи.</li> <li>4. Опускание.</li> </ol>
<p>So <u>information (1), blocks going on. And that leads to basically a database. But it's the blocks of data. Bitcoin right now is about 550,000 blocks and the blocks are added on average every 10 minutes (3).</u></p>	<p>То есть постоянно накапливаются блоки. <u>Их становится больше (2).</u> И это приводит к базе данных. Эта база данных состоит из блоков. И биткойн сейчас содержит в себе где-то 550 тысяч блоков, <u>при этом новый блок создается примерно раз в 10 минут (3).</u></p>	<ol style="list-style-type: none"> <li>1. Опускание.</li> <li>2. Добавление.</li> <li>3. Замена типа синтаксической связи.</li> </ol>
<p><u>And similarly (1), the key thing is that (1) two sets of data are — it's infeasible that x and y would hash to the same thing. It's not impossible. It's infeasible (4). And if you look at the history of hash functions, this is usually the thing (1), that at some point in time these hash functions will not be collision resistant. Some quantum computing will come along, or something (2) will come</u></p>	<p>Невыполнимо то, что два набора данных <u>дадут одинаковый результат (4),</u> то есть это невыполнимо, что x и y захэшируются в одно и то же значение. <u>И я, опять же, сказал(3), что это невыполнимо, а не невозможно (4), потому что на текущий момент это сделать нельзя (3).</u> Однако, возможно, в будущем что-то изменится, появятся либо квантовые компьютеры, либо <u>какие-то</u></p>	<ol style="list-style-type: none"> <li>1. Опускание.</li> <li>2. Замена части речи</li> <li>3. Добавление.</li> <li>4. Объединение предложений.</li> </ol>

<u>along. (4)</u>	<u>более продвинутые технологии (2, 3), которые смогут разорвать концепцию (3) устойчивости к столкновениям.</u>	
And then there's something called puzzle friendliness. Even if you know a little bit of the input, it doesn't mean that you're going to get the output.	И также присутствует термин <u>как</u> дружелюбная головоломка. Это значит, что даже если Вы знаете <u>часть</u> вводных данных, Вы не получите выходные данные.	Замена части речи.
But the headers and the Merkle trees use something called SHA 256, which is a standard which is literally 256 bits long. <u>That's like (2) zeros and ones for (2) 256 registries.</u>	<u>Но биткойн и хэш-функции (1) используют заголовки и Дерево Меркла, которые используют стандарт SHA 256, который имеет буквально длину в 256 битов. Это значит (2): там чередуются (1) единички и нули длиной (2) в 256 символов.</u>	1. Добавление. 2. Замена части речи.
<u>This concept of it's really keeping the security (1) by a little bit of information in something called a header (2, 3) and then pushing all the meat (1) of the transaction and data down.</u>	<u>Они придерживаются концепции того, что сохраняют всю главную информацию в заголовке блокчейна, который называется «шапкой» (2, 3), а все транзакции, всю тяжелую информацию они сталкивают в глубину блокчейна.</u>	1. Опускание. 2. Замена членов предложения. 3. Замена типа синтаксической связи.
<u>So it's harder to find a block, the more miners there are. So every block header needs to have some what's called a difficulty target.</u>	<u>Чем больше у нас майнеров, тем труднее найти блок. Каждая шапка блока, она обязана иметь что-то, что называется «цель сложности».</u>	Перестановка.
But there is another piece of it as well is that the hash function, <u>if it's a good cryptographic secure hash function (1), is what's called</u>	<u>Однако у нас есть еще и другая сторона, и в ней говорится то, что хорошая хэш-функция, правильно сделанная хэш-функция (1),</u>	1. Замена членов предложения. 2. Замена части речи. 3. Добавление.



<p><u>collision resistant (1, 2), so infeasible, in fact, 1 divided by 10 to the 40th, that's a 1 with 40 zeroes after it (4). It's so infeasible (2) to happen, it's possible but infeasible to happen (4).</u></p>	<p>она <u>устойчива</u> к столкновениям (1, 2), поэтому это настолько маловероятно (3). На самом деле, это единица, деленная на 10 в 40 степени, то есть, по сути, это единица с 40 нулями после нее (4). Такая маленькая вероятность (2), что это произойдет. Это возможно в теории, но очень маловероятно (4).</p>	<p>4. Членение предложений.</p>
<p><u>So time stamps are not a particularly important part of Bitcoin. They (4) are timestamped (2, 3).</u></p>	<p><u>Временные отметки, они не особо важны для функционала (1) биткойна, хотя, конечно же (1), биткойн (4) использует временные отметки (2, 3).</u></p>	<p>1. Добавление. 2. Объединение предложений. 3. Замена членов предложения. 4. Замена части речи.</p>
<p><u>But sometimes if somebody puts something off and it's off by a few minutes or even up to two hours (1, 2), there's a check in the technology (3) in the scripting function if the timestamp's off more than a couple hours (1). So literally, it's not that precise.</u></p>	<p><u>Но иногда бывает такое, что какие-то действия, они показывают неправильное время в несколько минут, а иногда и в несколько часов (1, 2). В биткойне (4) существует скрипт, который проверяет расхождения отметок, если они отличаются больше, чем на несколько часов (1). То есть, в буквальном смысле, эти временные отметки, они не очень точные.</u></p>	<p>1. Замена типа синтаксической связи. 2. Членение предложений. 3. Опущение. 4. Добавление.</p>
<p><u>Meaning (1, 2) it's gotten more and more difficult, and the result of the hash has to have a bunch of leading zeros (1).</u></p>	<p><u>Это означает (1, 2), что майнинг сейчас стал намного-намного труднее, и решение хэш-функции должно включать в себя еще большее количество нулей (1).</u></p>	<p>1. Замена части речи. 2. Замена членов предложения.</p>
<p><u>There is (1) software that helps (2), takes transactions, puts them basically (2) into this binary tree called a Merkle tree, uses (1) hash</u></p>	<p><u>Это (1) программное обеспечение будет брать эти транзакции, вымещать их в бинарное дерево, называемое Деревом</u></p>	<p>1. Замена части речи. 2. Опущение. 3. Добавление.</p>

<p>functions, and basically skinnies <u>it</u> (1) all the way up to the top.</p>	<p>Меркла, и <u>используя</u> (1) хэш-функции, оно будет сжимать всю <u>эту информацию</u> (1) к верхушке, <u>где она будет хэширована</u> (3).</p>	
<p>She wants to encrypt <u>it</u>. She encrypts it with Bob's public key, sends it to him. <u>He</u> decrypts it with his private key.</p>	<p>Она хочет зашифровать <u>сообщение</u>. Она его шифрует с помощью публичного ключа Боба, отправляет его ему. И <u>Боб</u> расшифровывает его с помощью <u>приватного ключа</u>.</p>	<p>Замена части речи.</p>
<p>To do a digital signature, you need a private key and a message. And it's a function of the <u>message and the private key</u> (2).</p>	<p>Для того чтобы поставить цифровую подпись, нам нужен приватный ключ и сообщение. И <u>мы используем</u> (1) <u>функцию приватного ключа и самого сообщения</u> (2).</p>	<ol style="list-style-type: none"> <li>1. Добавление.</li> <li>2. Перестановка.</li> </ol>
<p>Elliptic curve digital signature algorithm, that's the actual algorithm that Bitcoin uses <u>to take</u> (1) <u>the private key</u> (2).</p>	<p>Алгоритм цифровой подписи на эллиптической кривой – это тот алгоритм, который использует биткойн для <u>создания</u> (1) <u>приватных ключей</u> (2).</p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена грамматического числа.</li> </ol>
<p>The key to random number generation <u>is</u> (2) if you're generating any length that it truly <u>is not clumpier</u> (1), <u>that there's let's say</u> (3, 5) <u>it's what maximum entropy</u> (4, 5), and that you really don't have any <u>clumps</u> (1). If <u>it</u> (1) all clumps in one area, then that's not great randomness.</p>	<p>Ключ создания случайных чисел <u>заключается в том</u> (2), что если у Вас есть любое количество чисел, чтобы они <u>минимальное количество раз повторялись</u> (1), <u>и чтобы одинаковых чисел было меньше</u> (2). <u>Это называют «максимальная энтропия</u> (4, 5), <u>и важно</u> (2), чтобы одинаковых чисел было как можно меньше (1). Если в <u>какой-то линейке цифр</u> (1) очень много одинаковых цифр, то это не очень хорошая случайность.</p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Добавление.</li> <li>3. Опущение.</li> <li>4. Замена членов предложения.</li> <li>5. Членение предложений.</li> </ol>

<p>He literally hashed it <u>twice (2)</u>, <u>once (2) with this hash function called SHA256 (3)</u>, <u>another hash function (3, 4, 5)</u>, <u>then concatenates, and puts a little check sum at the end</u>, <u>and then uses something called a base 58 to make it (2) even shorter (4)</u>.</p>	<p>Он использует публичный ключ (1) и, буквально, хэширует его 2 раза (2). Первый раз (2) он хэшируется (3), используя функцию SHA256, и потом используется другая хэш-функция (3, 4, 5). Эти функции (1) объединяются, и далее на конце добавляются еще дополнительные символы (4), чтобы все усложнить (1), и далее используется «base 58», чтобы результат (2) сократить (4).</p>	<ol style="list-style-type: none"> <li>1. Добавление.</li> <li>2. Замена части речи.</li> <li>3. Членение предложений.</li> <li>4. Замена членов предложения.</li> <li>5. Замена типа синтаксической связи.</li> </ol>
<p><u>Time-stamped (1) append-only logs are essentially a record of transactions or a block as blockchain uses it with a time (2, 3)</u>. <u>And that can't be changed in the future (4)</u>.</p>	<p>Неудаляемые записи с временными отметками (1) – это записи, которые блокчейн на себя записывает (2). Они содержат в себе информацию о транзакциях, а также, во сколько они произошли, в какое время (3), и их нельзя удалить и изменить в будущем (4).</p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена членов предложения.</li> <li>3. Членение предложений.</li> <li>4. Объединение предложений.</li> </ol>
<p>There are <u>issues with micro payments and how they're inconsistently confirmed</u>.</p>	<p>Есть проблема с микро платежами и то, как они плохо влияют на систему.</p>	<ol style="list-style-type: none"> <li>1. Грамматическая замена числа.</li> <li>2. Замена членов предложения.</li> </ol>
<p>So the question <u>is about possible solutions</u> to address performance and scalability.</p>	<p>То есть вопрос сводится к тому, как мы можем (1)(2) улучшить производительность и масштабируемость блокчейна.</p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена типа синтаксической связи.</li> </ol>
<p>But very few people are actually going <u>to investigate it enough to be assured there's not a bug or an error (2)</u>.</p>	<p>Но я сомневаюсь (1) то, что очень много людей туда зашли (1) и начали проверять его на какие-нибудь ошибки или багги (2), на то, что их там нет</p>	<ol style="list-style-type: none"> <li>1. Добавление.</li> <li>2. Замена членов предложения.</li> </ol>

	(1).	
<u>Account ledgers have to have some way (1) to record the change in accounts (2).</u>	<u>В реестре аккаунтов нужна технология (1), которая будет записывать все изменения в балансе аккаунтов (2).</u>	1. Замена членов предложения. 2. Замена типа синтаксической связи.
So it's natural to add privacy (3) on top of Bitcoin (2) in your fork rather than write it from scratch.	Соответственно, это естественно, что разработчики используют уже готовый код (1) и просто меняют какие-то детали (3) в процессе хард-форка, вместо того, чтобы с нуля писать свой собственный код.	1. Добавление. 2. Опускание. 3. Замена членов предложения.
Proof-of-work is also has a bunch of energy consumption.	Консенсус proof-of-work потребляет очень много энергии.	Замена части речи.
In Bitcoin, it's every 10 minutes, and you're not really sure until 3, 4, 5, some would say 6 blocks to an hour (2) go by.	В биткойне это, например, время блока 10 минут, и вы никогда не можете быть уверены, что транзакция действительно прошла (1), пока не пройдет хотя бы 3,4, может быть, 5 блоков.	1. Добавление. 2. Опускание.
The bigger blocks you have, only the miners that can handle the enormous blocks will be able to stay in business.	Чем больше у нас размер блоков, тем меньше у нас майнеров, которые обладают оборудованием, чтобы эти блоки обрабатывать.	Замена типа синтаксической связи.
<u>And in sort of unintuitive ways, if that happens (1), you get more accidental forks to the blockchain.</u>	<u>И чем дольше блоки доходят до майнеров по сети (2), тем больше у нас вероятности создать случайный и незапланированный форк.</u>	1. Опускание. 2. Добавление.
So the side chains are not recorded gross on the main chain.	Итак, сайдчейны, они по большей мере не записываются, их транзакции не	Добавление.

	<u>записываются</u> на основной цепи.	
Proof-of-work <u>is one of the issues</u> (2) about scalability.	<u>На самом деле</u> (1), proof-of-work <u>сильно тормозит</u> (2) масштабируемость сети.	1. Добавление. 2. Замена части речи.
<u>Blockchain applications derive part of their value from participation of multiple parties on the network as well</u> , that multiple people are involved.	<u>Блокчейны получают свою основную ценность как раз из-за того, что у них есть множество участников сети</u> , которые работают над поддержкой их программного кода.	Замена типа синтаксической связи.
And the input <u>inside of Bitcoin</u> (1) is an ID of a previous output.	Эти вводные данные являются <u>не чем иным, как хешированным</u> (2) ID <u>предыдущей транзакции, то есть</u> (2) предыдущих выходных данных.	1. Опускание. 2. Добавление.
So not only is Bitcoin a series of blocks of information – each block that has 1,000 to 2,000 individual transactions – but in a sense, there's a separate chain that's going on.	Таким образом, <u>у нас получается, что</u> (1) биткойн не просто серия из информации из различных блоков, в каждом из которых есть примерно 1000-2000 транзакций, но также <u>у нас там есть ещё и отдельная цепь</u> , которая находится в блокчейне.	Добавление.
And so inputs always have to equal outputs. When a transaction is validated.	Поэтому входные данные всегда должны быть равны выходным данным, когда транзакция валидируется.	Объединение предложений.
<u>You can add</u> 100 bytes of arbitrary data in a Coinbase.	<u>Можно добавить</u> 100 байтов производных данных в Coinbase транзакцию.	Замена членов предложения.
So it all rolls into a database <u>called</u> the unspent transaction	Итак, это все попадает в такую базу данных, <u>которая называется</u>	Замена типа синтаксической связи.

output.	неизрасходованные выходные данные транзакции.	
The 54 million all reside in a database within the Bitcoin software called <u>(2) the unspent transaction output, UTXO (3).</u>	Эти 54 миллиона транзакций, они все хранятся в <u>специальной таблице (1)</u> , в специальной базе данных, <u>которая называется (2) UTXO или выходные данные неизрасходованных транзакций (3).</u>	1. Добавление. 2. Замена типа синтаксической связи. 3. Перестановка.
There's several hundred, but not several thousand, little operations and codes <u>that you can use in the Bitcoin script.</u>	В блокчейне есть более ста, но менее тысячи различных операций, которые вы можете использовать <u>в коде блокчейна, для того чтобы запустить какой-то скрипт.</u>	Замена типа синтаксической связи.
So it's either to a hash of a Bitcoin address, a hash of a conditional script, and a little less than 1% now go to multiple signatures.	То есть <u>большинство скриптов</u> , они идут на хэш биткойн-адреса или на хэш кондиционального скрипта, и чуть менее 1% идут сразу на несколько подписей.	Добавление.
It's taking a lot of data and shrinking it down, compressing it, shrinking it, to maybe a series of numbers.	<u>В этом процессе</u> мы берём много данных и уменьшаем их, сжимаем, уменьшаем, иногда даже в цепочку цифр.	Добавление.
They use the hash function to <u>compress (1)</u> a bunch of data, what I call the Merkle Tree, but it's taking 1,500 or 2,000 transactions and squeezing it into one hash.	Хэш-функции используются <u>для сжатия (1)</u> кучи данных, я называю это древо Меркла, но <u>то, что она делает – (2)</u> она берет 1500 или 2000 транзакций и сжимает их в один единый хэш.	1. Замена части речи. 2. Добавление.
<u>And it's just</u> , you put any text in, <u>you</u> put a password in, and it turns it into a blockchain address.	Вы просто можете вставить какой-то текст, вставить пароль, и он сразу превратит его в биткойн-	Опущение.

	адрес.	
So if you're an internet service provider, or if you're China, you can actually fork Bitcoin with zero mining power <u>by just controlling the network.</u>	То есть, если вы интернет-провайдер, либо, <u>допустим</u> , Китай, вы можете форкнуть биткойн с нулевой майнинговой мощностью, просто владея самой сетью.	Опущение.

## ПРИЛОЖЕНИЕ 2

Анализ перевода статьи Сатоши Накамото «Биткойн: система цифровой пиринговой наличности» («Bitcoin: A Peer-to-Peer Electronic Cash System»)

Таблица 2 – Биткойн: система цифровой пиринговой наличности

Оригинал	Перевод	Грамматическая трансформация
<p>The network <u>timestamps (2) transactions by hashing them into an ongoing chain (1) of hash-based (2) proof-of-work (3), forming a record that cannot be changed (1) without redoing the proof-of-work (1).</u></p>	<p>Сеть <u>ставит метки (2) времени на транзакции, соединяя их в цепочку (1) доказательств проделанной работы на основе хэширования (2, 3). Сформированные таким образом записи невозможно изменить (1), не выполнив заново всего объема вычислений (1).</u></p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена части речи.</li> <li>3. Членение предложения.</li> </ol>
<p>As long as a majority of CPU power <u>is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers.</u></p>	<p>До тех пор пока большая часть <u>вычислительных мощностей контролируется узлами, не объединенными с целью атаковать сеть, они будут генерировать самую длинную цепочку, опережая любых злоумышленников.</u></p>	<p>Замена членов предложения.</p>
<p>The <u>network itself (1) requires minimal structure (2, 3). Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network (1) at will, accepting the longest proof-of-work chain as proof of what happened while they were gone (2).</u></p>	<p><u>Устройство самой сети (1) очень простое (2, 3): сообщения рассылаются на основе принципа «наименьших затрат», а узлы могут покидать сеть и снова подключаться к ней (1) в любой момент, принимая самую длинную версию цепочки для восстановления пропущенной истории</u></p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена членов предложения.</li> <li>3. Объединение предложений.</li> </ol>



	<u>транзакций (2).</u>	
Each timestamp includes the previous timestamp in <u>its hash (1), forming a chain, with each additional timestamp reinforcing the ones before it (2, 3).</u>	Каждый хэш включает в себя предыдущую метку: <u>так выстраивается цепь, где очередное звено укрепляет все предыдущие (2, 3).</u>	1. Опускание. 2. Замена членов предложения. 3. Замена типа синтаксической связи.
To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof- of-work system similar to Adam Back's Hashcash, <u>rather than newspaper or Usenet posts.</u>	Чтобы реализовать распределенный одноранговый сервер меток времени, мы используем схему «доказательства работы», подобную системе Hashcash Адама Бека.	Опускание.
<u>For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block (2) until a value is found that gives the block's hash the required zero bits (1).</u>	<u>В нашем сервере меток времени поиск значения с нужным хэшем происходит путем перебора значения итерируемого поля-добавки (nonce) в блоке данных (2).</u>	1. Опускание. 2. Замена членов предложения.
As later <u>blocks (1, 3) are chained after it (2), the work to change the block (4) would include (5) redoing all the blocks after it (2).</u>	И если он (1, 3) не является последним в цепочке (2), эта работа <u>включает в себя (5) и перевычисление всех блоков, следующих за ним (2).</u>	1. Замена части речи. 2. Замена членов предложения. 3. Замена грамматического числа. 4. Опускание. 5. Замена грамматического времени.
If <u>the majority (1) were based on one-IP-address-one-vote, it (3) could be subverted by anyone able to allocate many IPs (2).</u>	Если голосом считается один IP-адрес, то такую <u>схему (3) можно скомпроментировать, если контролировать большой диапазон адресов (2).</u>	1. Опускание. 2. Замена членов предложения. 3. Замена части речи.
Once the latest transaction in a coin is buried under enough <u>blocks (1), the spent transactions before it can be discarded to save (2) disk</u>	Как только последняя транзакция в монете-цепочке окажется внутри достаточно старого <u>блока (1), все предшествующие ей</u>	1. Замена грамматического числа. 2. Замена части

space.	транзакции в цепочке могут быть удалены <u>в целях очистки (2)</u> дискового пространства.	речи.
A user only <u>needs (1, 2)</u> to keep a copy of the block headers of the longest proof-of-work chain, <u>which he can get by querying network nodes (1) until he's convinced he has the longest chain (3)</u> , and obtain the Merkle branch linking the transaction <u>to the block it's timestamped in (3)</u> .	<b>Пользователю необходимо (1, 2)</b> лишь хранить заголовки блоков самой длинной цепочки, <u>которую он получил от других узлов (1)</u> , и запрашивать хэш-поддереву для необходимой транзакции.	1. Замена членов предложения. 2. Замена части речи. 3. Опускание.
He can't check the transaction for himself, but <u>by linking it to a place in the chain (2)</u> , he can see that a <u>network node has accepted it, and blocks added after it further confirm the network has accepted it (2, 3)</u> .	Он не может проверить <u>корректность (1)</u> транзакции самостоятельно, <u>но получив ссылку на блок, в котором она находится (2)</u> , он может убедиться в том, <u>что этот блок и все последующие приняты и подтверждены сетью (2)</u> .	1. Добавление. 2. Замена членов предложения. 3. Опускание.
One strategy <u>to protect against (1)</u> this would be <u>to accept (1)</u> alerts from network nodes when they detect an invalid block, <u>prompting the user's software to download the full block (4)</u> and <u>alerted transactions to confirm the inconsistency (4, 5)</u> .	Одной из стратегий <u>противодействия (1)</u> этому может быть <u>рассылка (1)</u> сигналов тревоги от обычных пиров, которые получают «ложный» блок (2). <u>Такой сигнал (3) будет заставлять программу-клиент загружать блок полностью (4)</u> , <u>чтобы самостоятельно подтвердить некорректность данных (4, 5)</u> .	1. Замена части речи. 2. Членение предложения. 3. Добавление. 4. Замена членов предложения. 5. Замена типа синтаксической связи.

### ПРИЛОЖЕНИЕ 3

Анализ перевода книги Антонопулоса Андреаса М. «Осваиваем биткойн. Программирование блокчейна» («Mastering the Bitcoin. Programming the Open Blockchain»)

Таблица 3 – Осваиваем биткойн. Программирование блокчейна

Оригинал	Перевод	Грамматические трансформации
<p><u>Until the moment this address is referenced (1) as the recipient of value in a transaction posted on the bitcoin ledger, the bitcoin address (2) is simply part of the vast number of possible addresses that are valid in bitcoin (1).</u></p>	<p><u>До того момента, когда этот адрес определяется (1) как получатель денежных единиц в результате транзакций, зафиксированных в реестре биткойна, он (2) является просто частью огромного набора возможных адресов, допустимых в биткойн-системе (1).</u></p>	<ol style="list-style-type: none"> <li>1. Замена типа синтаксической связи.</li> <li>2. Замена части речи.</li> </ol>
<p>Unlike a QR code that simply contains (1, 2) a destination bitcoin address, a payment request is a QR-encoded URL that contains a destination address, a payment amount, and a generic description.</p>	<p>В отличие от QR-кода, содержащего только (1, 2) биткойн-адрес получателя, платежное требование представляет собой QR-закодированный URL, который содержит адрес получателя, сумму платежа и общее описание.</p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена части речи.</li> </ol>
<p>If the wallet application does not maintain a copy of unspent transaction outputs, it can query the bitcoin network to retrieve this information using (1) a variety of APIs available by different providers (2) or by asking a full-node using an</p>	<p>Если приложение кошелька не обеспечивает хранения копий неизрасходованных выходных данных транзакций, то оно может обратиться с запросом к биткойн-сети, чтобы получить эту информацию с помощью (1) множества</p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена членов предложения.</li> </ol>

application programming interface (API) call.	<u>разнообразных прикладных программ интерфейсов (API), предоставляемых различными провайдерами (2), или обратившись к полноценному узлу сети с помощью вызова API-интерфейса.</u>	
This URL will return all the unspent transaction outputs for an address, giving any application the information it needs <u>to construct</u> transaction inputs for spending.	Указанный URL возвращает все неизрасходованные выходные данные транзакции для некоторого адреса, предоставляя любому приложению информацию, <u>необходимую для формирования</u> входных данных транзакции, расходующей денежные средства.	Замена части речи.
This dataset is dozens of gigabytes in size and is downloaded incrementally over several days or weeks, <u>depending</u> on the speed of your CPU and internet connection.	Этот набор данных имеет размер в десятки гигабайтов, поэтому загружается постепенно в течение нескольких дней или даже недель, <u>в зависимости</u> от ресурсов вашего компьютера и пропускной способности интернет - соединения.	Замена части речи.
Tags are used by the developers <u>to mark</u> specific releases of the code by version number.	Тэги используются разработчиками <u>для обозначения</u> каждого релиза кода номером версии.	Замена части речи.
The most important of these is the configure script <u>that offers (1, 2)</u> a number of different options <u>to customize (2)</u> the build process.	Самым важным в этом наборе является скрипт configure, <u>предлагающий</u> разнообразные варианты <u>для настройки (2)</u> процесса компиляции и сборки.	1. Замена типа синтаксической связи. 2. Замена части речи.
<u>If there are any missing libraries or errors (1),</u> the configure command will	<u>Если какие-либо библиотеки не найдены или обнаружены какие-либо</u>	1. Замена членов предложения. 2. Замена типа

<p>terminate with an error <u>instead of (2) creating the build scripts (1).</u></p>	<p><u>проблемы (1), то команда configure завершится с ошибкой и (2) не создаст скриптов сборки (1).</u></p>	<p>синтаксической связи.</p>
<p><u>Once you've edited the configuration file and set the options (1) that (2) best represent (1) your needs, you can test (1) bitcoind with this configuration.</u></p>	<p><u>После редактирования файла конфигурации и установки значений параметров (1), (2) наилучшим образом соответствующих (1) вашим требованиям, можно проверить (1) функционирование bitcoind с заданной конфигурацией.</u></p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена типа синтаксических связей.</li> </ol>
<p><u>You can hit Ctrl-C to interrupt the process once you are satisfied (1) that it is loading the correct settings and running as you expect (2).</u></p>	<p><u>Чтобы прервать выполнение активного процесса, можно нажать комбинацию клавиш CTRL+C (1), если вы убедились в правильности значений всех параметров и программа работает так, как предполагалось (2).</u></p>	<ol style="list-style-type: none"> <li>1. Перестановка.</li> <li>2. Замена членов предложения.</li> </ol>
<p>The command line allows us (1) to experiment <u>interactively (2) with the capabilities that are also available (3) programmatically via the API.</u></p>	<p>Командная строка позволяет экспериментировать в <u>интерактивном режиме (2) с функциональными возможностями, доступными (3) в том числе и через прикладной программный интерфейс (API).</u></p>	<ol style="list-style-type: none"> <li>1. Опускание.</li> <li>2. Замена части речи.</li> <li>3. Замена типа синтаксической связи.</li> </ol>
<p>The data is returned in JavaScript Object Notation (JSON), a format that can easily be «consumed» by all programming languages <u>but is also quite human-readable.</u></p>	<p>Данные возвращаются в формате JavaScript Object Notation (JSON), который без затруднений обрабатывается практически всеми языками программирования, <u>к тому же вполне удобен для чтения человеком.</u></p>	<p>Замена типа синтаксической связи.</p>
<p>When we used the bitcoind cli command <u>to get (1) help</u></p>	<p>Когда мы используем команду <u>bitcoin-cli для</u></p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена членов</li> </ol>

<p>on a command, <u>it showed us an example of using curl (2)</u>, the versatile command-line HTTP client <u>to construct (1)</u> one of these JSON-RPC calls.</p>	<p><u>вывода (1)</u> справочной информации о какой-либо команде, <u>то выводится и пример применения curl (2)</u>, универсального HTTP-клиента для командной строки, <u>для формирования (1)</u> одного из вариантов вызова JSON-RPC.</p>	<p>предложения.</p>
<p>The digital keys in a user's wallet are completely independent of the bitcoin protocol <u>and</u> can be generated and managed by the user's wallet software without reference to the blockchain or access to the internet.</p>	<p>Цифровые ключи в кошельке пользователя абсолютно независимы от протокола биткойна, <u>поэтому</u> могут генерироваться и управляться программным обеспечением пользовательского кошелька без обращений к структуре данных блокчейна и без доступа к Интернету.</p>	<p>Замена типа синтаксической связи.</p>
<p>These digital keys are very rarely seen by the users of bitcoin.</p>	<p>Пользователи биткойн-системы крайне редко видят эти цифровые ключи.</p>	<p>Замена членов предложения.</p>
<p>For the most part, <u>they</u> are stored inside the wallet file and managed by the bitcoin wallet software.</p>	<p>В основном <u>ключи</u> хранятся в файле кошелька и управляются программным обеспечением биткойн-кошелька.</p>	<p>Замена части речи.</p>
<p>There is a mathematical relationship between the public and the private key that allows the private key to be used <u>to generate</u> signatures on messages.</p>	<p>Между открытым и секретным ключами существует математическая зависимость, которая позволяет применять секретный ключ <u>для генерации</u> подписей к сообщениям.</p>	<p>Замена части речи.</p>
<p><u>From the private key (4)</u>, we use <u>elliptic curve multiplication (1)</u>, a one-way cryptographic function, <u>to generate (3)</u> a public key (K).</p>	<p><u>Методика умножения на эллиптических кривых (или умножения точек на эллиптических кривых) (2)</u>, <u>односторонняя криптографическая</u></p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Добавление.</li> <li>3. Замена части речи.</li> <li>4. Перестановка.</li> </ol>

	<p><u>функция, используется</u> (1)  <u>для генерации</u> (3)  открытого ключа (К) <u>по</u>  <u>секретному ключу</u> (4).</p>	
<p>A private key <u>can be applied</u> (1) to the digital fingerprint of a transaction <u>to produce</u> (2) a numerical signature.</p>	<p>Секретный ключ <u>можно применить</u> (1) к цифровым отпечаткам транзакции <u>для получения</u> (2) подписи в числовой форме.</p>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Замена части речи.</li> </ol>
<p>Elliptic curve cryptography is a type of asymmetric or public key cryptography based on the discrete logarithm problem <u>as expressed</u> by addition and multiplication on the points of an elliptic curve.</p>	<p>Криптография с использованием эллиптических кривых – особый тип асимметричной криптографии или криптографии с открытым ключом, основанный на задаче дискретного логарифмирования, <u>которую можно описать</u> как сложение и умножение точек на эллиптической кривой.</p>	<p>Замена типа синтаксической связи.</p>
<p><u>To visualize</u> multiplication of a point with an integer, <u>we will use the simpler elliptic curve</u> (3) over real numbers – <u>remember, the math is the same</u> (3).</p>	<p><u>Для наглядного геометрического представления</u> операции умножения точки на целое число <u>воспользуемся упрощенной эллиптической кривой</u> (3) над полем действительных чисел, <u>поскольку ранее уже отмечалось, что</u> (2) <u>для таких кривых математические методы остаются неизменными</u> (3).</p>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Замена типа синтаксической связи.</li> <li>3. Замена членов предложения.</li> </ol>
<p>Bitcoin addresses are almost always encoded as «Base58Check», which uses 58 characters (a Base58 number system) and a checksum <u>to help</u> human readability, <u>avoid</u> ambiguity, and <u>protect</u> against errors in address transcription and entry.</p>	<p>Биткойн-адреса почти всегда кодируются в формате Base58Check, который использует 58 символов (система счисления Base59) и контрольную сумму <u>для повышения</u> удобства чтения человеком, <u>для устранения</u> неоднозначности и <u>для</u></p>	<p>Замена части речи.</p>

	<u>защиты</u> от ошибок в записи адреса и его ввода.	
<u>To add extra security (1) against typos or transcription errors, Base58Check is a Base58 encoding format, frequently used in bitcoin (2), which (4) has a built-in error-checking code (3).</u>	<u>Для дополнительной защиты (1) от опечаток и ошибок при записи в биткойн-системах часто используется (2) Base58Check – формат кодирования Base58 (4) с добавлением встроенной проверки на ошибки (3).</u>	<ol style="list-style-type: none"> <li>1. Замена части речи.</li> <li>2. Перестановка.</li> <li>3. Замена членов предложения.</li> <li>4. Замена типа синтаксической связи.</li> </ol>
The checksum is derived from the hash of the encoded data and can therefore be used to detect and prevent transcription and typing errors.	Контрольная сумма вычисляется по хэш-значению закодированных данных, следовательно, может использоваться <u>для обнаружения</u> и <u>предотвращения</u> ошибок и опечаток.	Замена части речи.
<u>If the two do not match, an error has been introduced and the Base58Check data is invalid.</u>	<u>Несовпадение контрольных сумм свидетельствует об ошибке,</u> и данные в формате Base58Check считаются некорректными.	<ol style="list-style-type: none"> <li>1. Замена типа синтаксической связи.</li> <li>2. Замена членов предложения.</li> </ol>
For example, in the case of a bitcoin address the prefix is zero (0x00 in hex), whereas the prefix <u>used when encoding a private key (2)</u> is 128 (0x80 in hex) (1).	Например, для биткойн-адреса префикс равен нулю (0x00 в шестнадцатеричном формате), <u>а префикс при кодировании секретного ключа (2) равен 128 (0x80 в шестнадцатеричном формате) (1).</u>	<ol style="list-style-type: none"> <li>1. Замена типа синтаксической связи.</li> <li>2. Замена членов предложения.</li> </ol>
From the resulting 32-byte hash (hash-of-a-hash), <u>we take only the first four bytes (1).</u> (2) <u>These four bytes (3)</u> serve as the error-checking code, or checksum.	Из полученного 32-байтового хэш-значения (хэша от хэша) <u>берутся только первые четыре байта (1), (2) которые (3) служат</u> в качестве кода для проверки ошибок или контрольной суммы (checksum).	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> <li>2. Объединение предложений.</li> <li>3. Замена части речи.</li> </ol>
<u>These formats are primarily used to make it easy for people to read (2) and</u>	Эти форматы в основном используются <u>для повышения</u>	<ol style="list-style-type: none"> <li>1. Замена членов предложения.</li> </ol>



<u>transcribe</u> (2) (1) keys without introducing errors.	<u>удобочитаемости</u> (2) <u>людьми и записи</u> (2) (1) ключей без ошибок.	2. Замена части речи.
Compressed public keys were introduced to bitcoin to <u>reduce</u> the size of transactions and <u>conserve</u> disk space on nodes that store the bitcoin blockchain database.	Сжатые открытые ключи появились в биткойне <u>для уменьшения</u> размеров транзакций и <u>экономии</u> дискового пространства на узлах, которые хранят базу данных структуры блокчейна биткойн-системы.	Замена части речи.
An almost 50% reduction in size in every transaction <u>adds up to a lot of data saved over time.</u>	Почти 50-процентное сокращение размера в каждой транзакции <u>позволяет хранить гораздо больший объем данных.</u>	Замена членов предложения.
<u>Visually</u> , this means that the resulting y coordinate can be above or below the x-axis.	<u>В визуальном представлении</u> это означает, что вычисляемая координата y может располагаться выше или ниже оси X.	Замена части речи.
More importantly, if <u>we convert this compressed public key</u> to a bitcoin address <u>using</u> (2) the double-hash function (RIPEMD160(SHA256(K))) <u>it will produce a different bitcoin address</u> (1).	Но более важно то, что <u>если выполнить преобразование этого сжатого открытого ключа</u> в биткойн-адрес <u>с помощью</u> (2) функции двойного хэширования (RIPEMD160(SHA256(K))), то в результате <u>получится другой биткойн-адрес</u> (1).	1. Замена членов предложения. 2. Замена типа синтаксической связи.
To resolve this issue, <u>when</u> (3) private keys are exported from a wallet (2), the WIF that is used <u>to represent</u> (1) them is implemented differently in newer bitcoin wallets, to indicate that these private keys have been used <u>to produce</u> (1) compressed public keys and therefore compressed bitcoin addresses.	<u>Для решения</u> описанной проблемы (3) <u>при экспортировании секретных ключей</u> из какого-либо кошелька (2) формат WIF (предназначенный <u>для представления</u> (1) секретных ключей) в новых биткойн-кошельках реализован <u>по-другому</u> , указывая, что эти	1. Замена части речи. 2. Замена членов предложения. 3. Замена типа синтаксической связи.

	секретные ключи использовались для генерации (1) сжатых открытых ключей, следовательно, и для сжатых биткойн-адресов.	
This allows <u>the importing wallet</u> to distinguish between private keys originating from older or newer wallets and search the blockchain for transactions with bitcoin addresses corresponding to the uncompressed, or the compressed, public keys, respectively.	Это позволяет <u>кошельку, выполняющему импортирование</u> , различать секретные ключи старых и новых кошельков и проводить поиск в структуре данных блокчейна транзакций с биткойн-адресами, соответствующими несжатым или сжатым открытым ключам соответственно.	Замена типа синтаксической связи.
If a <u>bitcoin wallet is able to implement compressed public keys</u> , it will use those in all transactions.	Если в <u>биткойн-кошельке</u> есть <u>возможность применения сжатых открытых ключей</u> , он будет использовать их во всех транзакциях.	Замена членов предложения.
The private keys in the wallet will be used <u>to derive</u> the public key points on the curve, which will be compressed.	Секретные ключи в таком кошельке будут применяться <u>для генерации</u> открытых ключей как точек на эллиптической кривой, координаты которых будут сжаты.	Замена части речи.
<u>The compressed public keys will be used to produce bitcoin addresses (1) and those (2) will be used in transactions.</u>	С <u>помощью сжатых открытых ключей будут формироваться биткойн-адреса (1), включаемые (2) в транзакции.</u>	1. Замена членов предложения. 2. Замена типа синтаксической связи.
BIP-38 proposes a common standard for encrypting private keys with a passphrase and encoding them with Base58Check <u>so that (1) they can be stored securely on backup media.</u>	Документ BIP-38 предлагает общий стандарт шифрования секретных ключей с помощью парольной фразы (passphrase) и кодирования с помощью Base58Check	1. Замена типа синтаксической связи. 2. Замена членов предложения.

<p><u>transported securely between wallets, or kept in any other conditions (2) where the key might be exposed (2).</u></p>	<p>(1) <u>для обеспечения безопасного их хранения на устройствах резервного копирования, для защищенной передачи между кошельками, а также для хранения в любых других условиях (2), при которых возможно раскрытие ключа (2).</u></p>	
<p>Another way <u>to find</u> a vanity address is <u>to outsource</u> the work to a pool of vanity miners, such as the pool at Vanity Pool.</p>	<p>Другой способ <u>получения</u> «престижного» адреса – <u>передача</u> вычислительной работы в пул майнеров «престижа», например в Vanity Pool (<a href="http://vanitypool.appspot.com">http://vanitypool.appspot.com</a>).</p>	<p>Замена части речи.</p>