О семантическом анализе юридических текстов

Перпер Е.М., Гасанов Э.Э., Кудрявцев В.Б.

Рассматривается задача создания программы, осуществляющей семантический анализ юридических текстов на русском языке. Получив на вход текст, программа должна выдать схему вычисления значений всех сущностей, описанных в тексте. В работе приведены приемы, позволяющие автоматически строить такую схему при наличии морфологической информации о всех словах текста.

Ключевые слова: семантический анализ, синтаксический анализ, юридический текст, логическая формула, прием.

1. Введение

Обычно проверить для текста, правильно ли выделен его смысл, сложно, но в некоторых частных случаях сделать это проще. Одним из таких частных случаев является построение по тексту какого-либо закона, посвященного бухгалтерскому учету, схем вычисления значений упомянутых в законе объектов. Можно считать, что смысл закона выделен правильно, если и только если вычисленные по этим схемам значения объектов совпадают со значениями, вычисленными в соответствии с текстом закона.

Программе, создающей такие схемы, легко найти практическое применение, например, она могла бы использоваться в ERP-системах (Enterprise Resource Planning — Управление ресурсами предприятия), позволяющих бухгалтерам автоматически заполнять формы отчетности. При изменениях в законодательстве, касающихся изменения правил заполнения форм отчетности, производителям ERP-систем ([1, 2] и т.д.) приходится вручную менять программу. Этого не потребуется, если программа будет создаваться автоматически по тексту нормативно-правового акта.

Задачу построения данных схем предлагается решать в несколько этапов.

Первым этапом является создание по каждому предложению рассматриваемого текста его синтаксического дерева (дерева зависимостей) — дерева связей между словами в предложении. Определение дерева зависимостей см., например, на сайте Национального корпуса русского языка [3].

Большая часть парсеров — программ, осуществляющих синтаксический анализ — основывается либо на машинном обучении на синтаксически размеченных корпусах [4], либо на использовании строго определенных правил, позволяющих находить связи между словами [5].

Для программ первого типа необходима большая база синтаксически размеченных текстов (то есть, предложений, для которых синтаксический граф уже построен). Для русского языка можно выделить две базы синтаксически размеченных текстов — это Национальный корпус русского языка (доступ к этому корпусу ограничен) и проект Universal Dependencies [6]. Проект содержит корпуса для более чем 60 языков; цель проекта — разработка единой модели синтаксической разметки для разных языков [7]. Корпус русского языка проекта Universal Dependencies содержит более 66000 предложений, и его можно свободно использовать для обучения парсеров.

Что касается программ, работающих на основе правил, то они не требуют базы данных синтаксически размеченных текстов. Тем не менее, создание набора правил, который позволял бы проводить синтаксический анализ произвольных предложений — сложная задача. Одной из моделей языка, используемых программами, осуществляющими обработку естественного языка на основе правил, является модель «Смысл $\Leftrightarrow$ Текст» (Мельчук [8]). Вводя эту модель, Мельчук постулирует, что естественный язык — это преобразователь из текста в смысл и обратно. Мельчук строит формальную модель языка, состоящую из нескольких уровней (в т.ч., семантического и синтаксического), и вводит наборы правил, с помощью которых должен совершаться переход обрабатываемого представления текста на другой уровень. В идеале эти наборы правил должны обеспечивать переход от текста к его смыслу и обратно. Строго описанные правила оказалось удобно использовать в различных программах, осуществляющих обработку естественного языка (не обязательно русского), например, для автоматической генерации текста по определенным правилам [9].

Вероятно, наиболее развитой системой, осуществляющей обработку русского языка, в настоящее время является Abbyy Compreno. Она проводит синтаксический и семантический анализ текста [10], а также ис-

пользуется в программах, извлекающих информацию из неструктурированного текста, и программах, проводящих классификацию документов. Доступ к этой системе ограничен.

Для автора работы основным недостатком парсеров, имеющихся в свободном доступе, оказалась крайняя трудоемкость исправления совершаемых ими ошибок. По этой причине автор был вынужден отказаться от их использования.

В данной работе парсер строится на основе правил. Тот факт, что в работе рассматриваются не произвольные тексты, а тексты нормативно-правовых актов, касающихся бухгалтерского учета, значительно упрощает создание необходимого набора правил.

После синтаксического анализа текста осуществляется семантический анализ — выделение в тексте семантических (смысловых) связей между его фрагментами. В настоящей работе семантический анализ состоит в том, что по каждому синтаксическому графу строится формула логики предикатов, которая определяет условия, накладываемые на рассматриваемые в тексте объекты.

Наконец, по всем формулам вместе создается модель закона, представляющая собой совокупность схем вычисления значений объектов, упомянутых в законе. Каждая такая схема является своеобразным аналогом алгебраического дерева вычислений [11]. Построение формул логики предикатов и создание модели закона осуществляются, как и синтаксический анализ, на основе строго определенных правил (также называемых в настоящей работе приемами).

Общая идея алгоритма построения модели закона впервые была опубликована в [12], а идея используемого в работе метода синтаксического анализа — в работе [13]. Часть приемов, приведенных в этих работах, используется и в настоящей работе.

2. Основные понятия и формулировка результатов

Введем понятие модели юридического документа в соответствии с тем, как это было сделано в [12].

Рассмотрим ориентированный граф, каждой вершине которого сопоставлена некоторая процедура одного из следующих видов:

1)получение значения из конкретного поля в памяти. Считается, что это значение передается по всем ребрам, выходящим из такой вершины;

2)запись значения в конкретное поле в памяти. В такую вершину должно вести единственное ребро. Считается, что записываемое значение поступает в данную вершину по этому ребру;

3)вычисление некоторой арифметической функции одного или двух аргументов, логической функции одного или двух аргументов либо унарного или бинарного отношения. В такую вершину должно входить столько ребер, сколько аргументов у функции или отношения. По каждому из этих ребер в вершину поступает значение соответствующего аргумента функции (отношения). Значение функции (отношения) передается по всем ребрам, выходящим из этой вершины.

4)выбор одного значения из нескольких. В эту вершину для некоторого натурального числа $m \geq 2$ должно вести $m + 1$ ребро. Эти ребра должны быть пронумерованы числами от 0 до m. По ребру с номером m в вершину поступает число $i \in Z, 0 \leq i \leq m - 1$. По каждому ребру, выходящему из вершины, передается значение, поступившее в вершину по ребру с номером i.

Будем относить вершину к i-му виду вершин, $i \in \{1, 2, 3, 4\}$, если ей сопоставлена процедура i-го вида.

Из вершин графа 2-го вида выделим одну.

Этому графу сопоставляется процедура его вычисления: для выделенной вершины вычисляется значение, поступающее в нее по единственному входящему ребру. Это значение вычисляется с помощью процедуры, сопоставленной вершине, из которой выходит соответствующее ребро.

Для любой вершины 1-го, 3-го, 4-го вида вычисление значения, передаваемого по ребрам, выходящим из этой вершины, происходит следующим образом.

Для вершины 1-го вида это значение берется из соответствующего поля в памяти.

Для вершины 3-го вида сначала вычисляется значение, поступающее в вершину по одному из входящих в нее ребер. Если этого недостаточно для вычисления значения функции (отношения), то вычисляется значение, поступающее в вершину по другому ребру. В любом случае, дальше вычисляется значение функции (отношения) на полученных значениях аргументов, это значение и передается по выходящим из вершины ребрам.

Для вершины 4-го вида сначала вычисляется число i, поступающее в вершину по входящему в него ребру с максимальным номером, затем

вычисляется значение, поступающее в вершину по ребру номер i. Это значение и передается по выходящему из вершины ребру.

Если результат процедуры вычисления графа при любых исходных значениях полей совпадает с результатом заполнения соответствующего поля в форме отчетности в соответствии с текстом закона, назовем этот граф моделью вычисления поля. Граф, содержащий в себе в качестве подграфа модель вычисления любого поля из формы отчетности, назовем моделью юридического документа.

Первым этапом построения модели юридического документа по его тексту является синтаксический анализ.

Синтаксическое отношение — это отношение между парой слов предложения, причем одно из слов является главным в этом отношении, а другое — зависимым. Каждое синтаксическое отношение в зависимости от частей речи участвующих в отношении слов, их морфологических характеристик и т.д., может быть отнесено к определенному классу синтаксических отношений.

На вход программе, осуществляющей синтаксический анализ, поступает последовательность *лексем*. Лексема, в свою очередь, представляет собой последовательность символов. Это может быть слово, число, знак препинания. Текст предложения разбивается на последовательность лексем в процессе *лексического анализа*.

Помимо последовательности лексем, на вход синтаксическому анализу поступает последовательность *токенов*. Токен создается для каждого слова предложения в процессе морфологического анализа и представляет собой тройку, в которую входят: само слово; *лемма* — каноническая форма слова (например, для существительного это будет то же слово, но в именительном падеже и единственном числе); набор *морфологических характеристик* (для существительного это род, падеж, число и т.д., для глагола это вид, время и т.д).

Выходом синтаксического анализа является *синтаксическое дерево*, также называемое *деревом зависимостей*. Это ориентированное дерево. Каждой его вершине сопоставлен токен. Дуга в синтаксическом дереве ведет из вершины A в вершину B тогда и только тогда, когда сопоставленные этим вершинам слова связаны в русском языке синтаксическим отношением, причем главным в этом отношении является слово, соответствующее вершине A. Дуге при этом сопоставлено название отношения.

В данной работе, как и в [13], для синтаксического анализа предлагается использовать (возможно, в несколько упрощенном виде) подход, применяемый А.С. Подколзиным для автоматического решения различ-

ных математических задач [14]. В применении к синтаксическому анализу этот подход состоит в следующем. В программе имеется список *правил*, которые позволяют находить синтаксические связи между словами. Для каждого токена и каждого правила проверяется, применимо ли это правило к данному токену; если да, то создается продиктованное этим правилом синтаксическое отношение.

Каждое правило состоит из трех частей. Первая часть проверяет, подходит ли слово для этого правила: обладает ли оно нужным набором морфологических характеристик. В большинстве случаев значение леммы не проверяется, однако есть и правила, которые работают с конкретными леммами. В тех случаях, когда целью правила является построения синтаксического отношения, в котором рассматриваемое слово было бы зависимым, проверяется также, что слово еще не является зависимым ни в каком построенном синтаксическом отношении. Объясняется эта проверка просто: каждое слово может входить в какое угодно число синтаксических отношений в качестве главного, но лишь в одно — в качестве зависимого.

Вторая часть заключается в поиске слова, которое может входить в синтаксическое отношение с рассматриваемым словом. В некоторых правилах ищется не одно слово, а несколько, притом таких, что каждое из них могло бы образовывать синтаксические отношения либо с другим найденным словом, либо с рассматриваемым словом.

Наконец, третья часть строит синтаксические отношения между рассматриваемым словом и найденными словами. В том случае, если слово прошло проверку в первой части правила, и для него были найдены подходящие слова во второй части правила, будем говорить, что правило *применимо* к слову. Таким образом, если правило применимо к слову, то в результате применения правила к этому слову строится одно или несколько новых синтаксических отношений.

Надо заметить, что описанный в работе алгоритм включает в себя элементы семантического анализа — это видно по названию некоторых отношений, по использованию списков существительных, формируемых исходя из смыслового значения этих существительных, и по нескольким другим признакам. Дело в том, что определенную работу по семантическому анализу удобно выполнять одновременно с синтаксическим анализом.

Для большей части создаваемых алгоритмом синтаксических отношений обозначения классов, к которым относятся эти отношения, взяты из [5], но для некоторых отношений используются специальные названия:

это позволяет облегчить работу с синтаксически разобранным предложением на следующих этапах.

Второй этап построения модели закона состоит в упрощении синтаксического графа каждого предложения. Цель этого упрощения — получить граф, в котором каждой вершине соответствует одна сущность (либо одно или несколько служебных слов).

На третьем этапе построения модели юридического документа происходит отождествление сущностей из различных предложений. В результате одну и ту же сущность во всех построенных на следующем этапе формулах будет выражать одна и та же переменная.

На четвертом этапе по каждому предложению исходного текста и соответствующему синтаксическому графу строится логическая формула. В результате связи между сущностями, о которых идет речь в предложении, оказываются выраженными с помощью математических операций.

На пятом этапе по совокупности всех формул для каждого поля из формы отчетности строится модель вычисления этого поля. Каждая такая модель фактически является программой, вычисляющих значение поля в соответствии с необходимыми и достаточными для этого данными. Совокупность всех таких моделей — модель закона — и будет той программой, которую нужно создать.

Для каждого этапа решения задачи приведены приемы, с помощью которых этот этап осуществляется. Приведенных приемов достаточно, чтобы построить фрагмент модели положения ПБУ 6/01 [15] по пунктам этого закона, касающимся годовой суммы амортизационных отчислений по объектам основных средств. Дальнейшие исследования будут касаться накопления большого числа приемов, что позволит строить модели различных нормативно-правовых актов.

3. Построение связного синтаксического графа предложения

Прежде чем перейти к описанию правил, заметим, что на слова, которые могли бы образовывать синтаксические отношения друг с другом, накладываются определенные ограничения. Если некоторая часть предложения заключена в круглые скобки, то никакое слово из этой части не может быть главным в синтаксическом отношении, в котором зависимое слово находится вне этих скобок. Применению основного алгоритма построения синтаксического дерева предшествует определение глубины

вложенности каждого слова предложения в круглые скобки (в дальнейшем будем коротко именовать ее глубиной). Эта глубина вычисляется как разность числа открывающих и числа закрывающих круглых скобок перед словом.

Подробнее остановимся на том, как осуществляется вторая часть правила. Происходит последовательный перебор всех слов предложения, осуществляемый либо в сторону начала предложения, либо в сторону его конца. Перебор начинается со слова, следующего после рассматриваемого слова в том направлении, в котором осуществляется перебор. Если рассматриваемое слово должно быть главным в синтаксическом отношении и имеет большую глубину, чем слово, до которого дошел перебор, то перебор завершается неудачей. Если рассматриваемое слово должно быть зависимым в синтаксическом отношении и имеет меньшую глубину, чем слово, до которого дошел перебор, то перебор также завершается неудачей. Если перебор дошел до начала либо конца предложения, а в некоторых случаях — до начала либо конца *клаузы* (простого предложения в составе сложного), в которой содержалось рассматриваемое слово, и нужное слово не найдено, перебор снова заканчивается неудачей. Если же нужное слово найдено, перебор успешно завершается, и найденное слово используется в следующей части правила для построения синтаксического отношения. В тех случаях, когда ищется несколько слов, может быть осуществлено несколько переборов.

Будем называть слово *допустимым*, если его глубина не является препятствием для включения этого слова в синтаксическое отношение.

Перед основным алгоритмом построения синтаксических отношений, совершается несколько подготовительных действий.

1) Для каждого тире в предложении создается токен, который в качестве леммы содержит само тире, а его набор морфологических характеристик состоит из одного элемента — указания на то, что тире следует рассматривать как глагол. Действительно, в большинстве случаев тире заменяет глагол «есть» в какой-либо форме.

2) Предложение разбивается на сегменты по некоторым знакам препинания — запятым, точкам с запятой, двоеточиям, притом рассматриваются только те из них, что имеют нулевую глубину. Во многих случаях сегмент совпадает с клаузой, но не всегда, так как запятые не только разделяют простые предложения в составе сложного, но и выполняют другие роли (например, отделяют друг от друга однородные члены предложения).

3) В предложении выделяются группы идущих подряд токенов, которые с точки зрения синтаксического анализа рассматриваются как единое целое. Делается это по той причине, что применение общих правил построения синтаксических отношений к словам из этой группы может привести к ошибкам, поэтому необходимо выделить эти группы до основного цикла построения синтаксических отношений.

   Группа токенов, о которой идет речь, может представлять собой

   а) предлог: «исходя из», «в течение», «в отношении», «в соответствии с», «в связи с» и др.;

   б) союз: «а также», «прежде чем» и др.;

   в) устойчивое словосочетание: «иметь место» и др.;

   г) графическое сокращение: «и т.д.», «и т.п.» и др.

   Алгоритм использует отдельный словарь, содержащий рассматриваемые группы токенов. Каждая группа токенов из рассматриваемого предложения ищется в словаре, и в случае успеха поиска каждая пара идущих подряд слов из этой группы соединяется отношением «НЕДЕЛИМ», где главным объявляется то из двух слов, которое находится ближе к началу предложения. Кроме того, если группа представляет собой предлог или союз, то считается, что каждое слово из группы является соответственно предлогом или союзом.

4) При синтаксическом анализе юридических текстов удобно также выделить группы токенов, обозначающие статьи, пункты, подпункты нормативных актов. Пусть в предложении непосредственно после слова «статья», «пункт», «подпункт» находятся один или несколько токенов, каждый из которых соответствует некоторой букве или числительному. Тогда для каждого такого токена создается отношение «ПУНКТ», в котором этот токен является зависимым, а токен, соответствующий слову «статья», «пункт» или «подпункт» — главным.

В основном алгоритме синтаксического анализа, помимо попыток применения к слову правил построения синтаксических отношений, для этого слова производятся некоторые вычисления, которые могут быть в дальнейшем использованы при рассмотрении не только этого слова, но и

последующих слов. Например, таким образом находится и запоминается последний встретившийся в рассмотренной части предложения глагол, а также последний встретившийся глагол перед последним встретившимся двоеточием.

Приступим, наконец, к описанию правил, из которых состоит основной алгоритм синтаксического анализа. Заметим, что некоторые части речи в процессе применения правил приравниваются к другим частям речи. В большинстве случаев это явно оговаривается. Исключение — местоимения: местоимения-существительные во всех правилах приравниваются к существительным, а местоимения-прилагательные — к прилагательным, если находятся непосредственно перед прилагательным или существительным, и к существительным в противном случае. В правилах это явно не оговаривается, но, например, когда речь в правиле идет о существительном, предполагается, что это может быть также и местоимение-существительное, и местоимение-прилагательное, если оно приравнено к существительному.

Правила 1 и 2 применяются к тире. С их помощью осуществляется попытка соединить тире с глаголом или причастием, которые были заменены этим тире. Эти правила (как и некоторые другие) относятся к числу сложных, так как с их помощью создаются связи между различными сегментами предложения.

1) Правило, создающее для тире синтаксическое отношение «ВАРИАНТ» между этим тире и последним глаголом, находящимся перед последним двоеточием перед тире. Если такой глагол существует, то он был найден ранее. В создаваемом отношении этот глагол является главным элементом, а тире — зависимым.

2) Правило, которое для тире проверяет, является ли союз первым элементом сегмента, в котором это тире находится. Если является, то происходит перебор сегментов от предыдущего к первому в предложении, пока первым элементом сегмента является тот же союз и не достигнуто начало предложения. Если у сегмента, на котором перебор прекратился, вторым словом является тот же союз, а первым — глагол или причастие, то между этим глаголом (причастием) и тире создается синтаксическое отношение «ВАРИАНТ», в котором глагол (причастие) — главный элемент, а тире — зависимый.

3) Правило, которое для слова «не» ищет ближайшее находящееся после него допустимое слово и создает синтаксическое отношение «ОТР» («отрицание»), в котором найденное слово является главным элементом, а «не» — зависимым.

4) Правило, которое для существительного ищет ближайшее находящееся после него допустимое слово, не являющееся прилагательным в родительном падеже. Если такое слово найдено, и это — существительное в родительном падеже, то создается синтаксическое отношение «ГЕНИТ-ИГ» («генитивная именная группа») с рассматриваемым словом в качестве главного элемента и найденным существительным в родительном падеже в качестве зависимого элемента.

5) Правило, ищущее для предлога «на» ближайшее к нему среди находящихся перед ним слов сегмента допустимое слово из списка существительных, обозначающих величину или значение («стоимость», «цена», «размер» и т.д.). Если между этим существительным и рассматриваемым предлогом «на» не расположены глагол или тире, создается синтаксическое отношение «ОПР» («определительное») с найденным существительным в качестве главного слова и предлогом «на» в качестве зависимого слова. Это правило, как и следующие два, касаются случаев присоединения предлога к существительному. Случай присоединения предлога к глаголу рассматривается после этих правил и действует «по умолчанию».

6) Правило, ищущее для предлога «от» ближайшее к нему среди находящихся перед ним слов сегмента допустимое слово из списка существительных, обозначающих доход («доход», «выручка» и т.д.). Если между этим существительным и рассматриваемым предлогом не расположены глагол или тире, создается синтаксическое отношение «ОПР» с найденным существительным в качестве главного слова и предлогом «от» в качестве зависимого слова.

7) Правило, ищущее для предлога «по» ближайшее к нему среди находящихся перед ним слов сегмента допустимое слово из списка существительных («положение», «списание» и т.д.). Если между этим существительным и рассматриваемым предлогом «по» не расположены глагол или тире, создается синтаксическое отношение «ОПР» с найденным существительным в качестве главного слова и предлогом «по» в качестве зависимого слова.

8) Правило, ищущее для предлога «о» («об») ближайшее к нему среди находящихся перед ним слов сегмента допустимое слово из списка существительных («законодательство», «информация», «решение» и т.д.). В случае успеха поиска создается синтаксическое отношение «ОПР» с найденным существительным в качестве главного слова и предлогом «о» («об») в качестве зависимого слова.

9) Правило, проверяющее для существительного в именительном падеже, есть ли глагол в сегменте, где это существительное находится. Пусть глагола там нет, и непосредственно перед последним двоеточием, встретившимся до этого сегмента, находится то же самое существительное (возможно, в другом падеже). Тогда создается отношение «А_ИМЕННО», где главным словом является найденное перед двоеточием существительное, а зависимым — рассматриваемое слово.

10) Правило, рассматривающее союз «то есть» или его графическое сокращение «т.е.». Если непосредственно перед этим союзом находится существительное, то осуществляется поиск допустимого существительного после этого союза, и в случае удачи создаются два отношения: «А_ИМЕННО», где главным словом является существительное, находящееся перед союзом, а зависимым — сам союз, и «СОЮЗ», где главным словом является этот союз, а зависимым — существительное, находящееся после него.

11) Правило, ищущее для количественного числительного ближайшее к нему среди находящихся после него допустимых существительных в родительном падеже и подходящем числе. Если такое существительное нашлось, и между ним и рассматриваемым количественным числительным нет ничего, кроме союзов и других количественных числительных, то строится синтаксическое отношение «КОЛИЧ» («количественное»), где найденное слово является главным, а числительное — зависимым словом. Если последнее допустимое слово предложения, находящееся перед числительным, обозначает количественное отношение («больше», «ниже», «равное» и т.д.), то создается также синтаксическое отношение «ДОП» («дополнение»). В этом отношении главным объявляется слово, обозначающее количественное отношение, а зависимым — ранее найденное существительное.

12) Правило, ищущее для порядкового числительного ближайшее к нему среди находящихся после него и согласованных с ним допустимых существительных. Если такое существительное нашлось, и между ним и рассматриваемым количественным числительным нет ничего, кроме союзов и других количественных числительных, то строится синтаксическое отношение «НОМЕР», где найденное слово является главным, а числительное — зависимым словом.

13) Правило, проверяющее, является ли последнее допустимое слово предложения, идущее в предложении до числительного, словом, которое может обозначать количественное отношение («больше», «ниже», «равное» и т.д.). Если это так, то строится синтаксическое отношение «КОЛИЧ», где найденное слово является главным, а числительное — зависимым словом.

14) Правило, которое для предлога, а также для слов «исходя» (если следующее за ним слово – «из») и «как», ищет допустимый глагол, чтобы связать их синтаксическим отношением «ГЛ_ДОП» («глагол – дополнение»), в котором глагол будет главным словом, а рассматриваемое правилом слово — зависимым. Глагол ищется в том же сегменте, где находится рассматриваемое слово, сначала в направлении от этого слова к концу сегмента, а в случае неудачи — в направлении от этого слова к началу сегмента. В данном правиле к глаголу приравниваются отглагольные существительные и причастия. Если нужного слова найти не удалось, а рассматриваемое правилом слово — «по» или «при», ищется в направлении от него к концу предложения ближайший допустимый глагол, не находящийся в клаузе, начинающейся со слова «который». В случае, когда нужное слово все еще не найдено, в качестве него берется последний глагол, находящийся перед последним (до рассматриваемого правилом слова) двоеточием, если такой существует.

15) Правило, ищущее для существительного, не находящегося в именительном падеже, ближайшее к нему среди находящихся перед ним слов сегмента допустимое слово, являющееся глаголом, существительным, страдательным причастием (не согласованным с существительным), кратким причастием, предлогом либо союзом (но не союзом «или», «и», «а»). Если такого слова нет и сегмент отделен от предыдущего точкой с запятой, то в качестве искомого слова рассматривается последний глагол в первом сегменте пред-

ложения, а при наличии предлога непосредственно после этого глагола — этот предлог. Создается синтаксическое отношение «ДОП» («дополнение»), в котором главным является найденное слово, а зависимым — рассматриваемое существительное.

16) Правило, ищущее для не находящегося в именительном падеже существительного, чей сегмент отделен от предыдущего запятой, первое в предыдущем сегменте (считая от его начала) существительное в том же падеже. Если такое существительное найдено, и оно является зависимым в каком-либо синтаксическом отношении, то добавляется такое же отношение, в котором зависимым словом является рассматриваемое правилом существительное.

17) Правило, ищущее для существительного в именительном падеже допустимый глагол, чтобы связать их синтаксическим отношением «ПОДЛ» («подлежащее»), в котором глагол будет главным словом, а существительное — зависимым. Глагол ищется так же, как и в предыдущем правиле, за тем исключением, что отглагольное существительное в данном случае к глаголу не приравнивается.

18) Правило, которое ищет для наречия допустимый глагол, чтобы связать их синтаксическим отношением «ГЛ_ОБСТ» («глагол – обстоятельство»), где глагол будет главным словом, а наречие — зависимым. Глагол ищется так же, как и в правиле 14.

19) Правило, ищущее для слова «если» допустимый глагол, чтобы связать их синтаксическим отношением «ЕСЛИ», в котором глагол будет главным словом, а существительное — зависимым. Глагол ищется так же, как и в правиле 14.

20) Правило, которое для слова «если», являющегося зависимым в каком-либо синтаксическом отношении, ищет ближайший к нему допустимый глагол, находящийся в предложении до сегмента, в котором находится рассматриваемое слово. Если такое слово найдено, то создается отношение «УСЛ» («условие»), где найденный глагол будет главным словом, а зависимым будет слово, являющееся главным в отношении, где «если» является зависимым.

21) Правило, которое для полного причастия, находящегося непосредственно после запятой, ищет ближайшее к нему в сторону начала предложения существительное, имеющее тот же падеж и число,

причем если это число — единственное, то и тот же род. Если такое существительное нашлось, создается синтаксическое отношение «ПРИЧ_СУЩ» («причастие – существительное»), в котором главным словом является существительное, а зависимым — причастие.

22) Правило, которое для прилагательного или полного причастия ищет ближайшее к нему в сторону конца предложения существительное, имеющее тот же падеж и число, причем если это число — единственное, то и тот же род. Если такое существительное нашлось, создается синтаксическое отношение «ПРИЛ_СУЩ» («прилагательное – существительное») (если рассматриваемое слово — прилагательное) или «ПРИЧ_СУЩ» (если рассматриваемое слово — причастие), где главным словом является существительное, а зависимым — рассматриваемое слово.

23) Правило, обрабатывающее союзы «и», «или», «а», не находящиеся в начале сегмента. Сначала ищется слово w_1 — допустимое слово, ближайшее к рассматриваемому союзу по направлению к началу предложения. Кроме того, ищется слово w_2 — допустимое слово, ближайшее к рассматриваемому союзу по направлению к концу предложения. Если w_2 — полное причастие, краткое причастие, глагол либо отглагольное существительное, то от рассматриваемого союза в сторону начала предложения ищется слово w_3 — ближайшее допустимое слово, удовлетворяющее следующим условиям: если w_2 — полное причастие, то w_3 должно быть полным причастием либо прилагательным, имеющим тот же падеж, что w_2; если w_2 — глагол либо краткое причастие, то найденное слово также должно быть глаголом либо кратким причастием и иметь то же число, что и w_2; если w_2 — отглагольное существительное, то w_3 должно быть существительным в том же падеже.

    Если w_2 не является полным причастием, кратким причастием, глаголом либо отглагольным существительным, ищется ближайшее к союзу по направлению к концу предложения допустимое слово w_4, удовлетворяющее следующим условиям: если w_1 — прилагательное либо полное причастие, то найденное слово также должно быть прилагательным либо полным причастием и иметь тот же падеж, что и w_1; если w_1 — существительное, то найденное слово должно быть существительным и иметь тот же падеж, что и w_1; если w_1 — глагол либо краткое причастие, то найденное слово также должно быть глаголом либо кратким причастием и иметь то же

число, что и w_1. Заметим, что w_1 может совпадать с w_3, а w_2 — с w_4.

Если подходящее слово найдено, то создается два синтаксических отношения «ОДНОР_ИГ» («однородные именные группы»), которые содержит рассматриваемый союз в качестве главного слова и найденные слова w_1 и w_4 (либо w_2 и w_3, если w_2 — полное причастие) в качестве зависимых. Если слово w_1 (w_3) до применения данного правила участвовало в каком-либо синтаксическом отношении в качестве зависимого, то оно заменяется в этом отношении на рассматриваемый союз.

24) Правило, обрабатывающее союзы «и», «или», «а», находящиеся в начале сегмента. Правило в основном аналогично предыдущему, отличие заключается в том, что в качестве w_2 выбирается не любое допустимое слово, ближайшее к рассматриваемому союзу по направлению к концу предложения, а только глагол или краткое причастие. Затем так же, как это делается в предыдущем правиле, для w_2 ищется слово w_3, и с участием этих слов и рассматриваемого союза создаются синтаксические отношения «ОДНОР_ИГ».

25) Правило, проверяющее, является ли последнее допустимое слово предложения, идущее в предложении до количественного числительного в именительном падеже, существительным. Если да, то создается отношение «НОМЕР», в котором найденное существительное — главное слово, а числительное — зависимое.

26) Правило, проверяющее, является ли существительным последнее отличное от частицы «не» допустимое слово предложения, идущее в предложении до слова, обозначающего количественное отношение. Если это так, то строится синтаксическое отношение «ОПР», где найденное существительное — главный элемент, а слово, обозначающее количественное отношение — зависимый.

27) Правило, обрабатывающее слово, чья лемма — «который». Сначала в предыдущем сегменте ищется ближайшее к рассматриваемому слову допустимое существительное, имеющее то же число, что и рассматриваемое слово. Затем ищется ближайшее к рассматриваемому слову по направлению к концу предложения допустимое слово, являющееся глаголом, кратким причастием или тире. Если нужные слова найдены, создается синтаксическое отноше-

ние «ПРИДАТ_ОПР» («придаточное определительное»), в котором главным словом является найденное существительное, а зависимым — найденный глагол, краткое причастие или тире.

Описанные выше правила расположены в том же порядке, в каком их рассматривает алгоритм. Если несколько правил независимо друг от друга пытаются создать разные отношения, в которых одно и то же слово было бы зависимым, сработает только то из правил, которое было рассмотрено раньше. Таким образом, при изменении порядка применения правил результат может измениться.

Помимо правил, касающихся построения синтаксических отношений, имеются три правила, изменяющих уже созданные синтаксические отношения. Все эти правила так или иначе касаются построения множественного актанта — отношения между однородными членами предложения. Перечислим их.

28) Правило, которое для каждого союза «и», «или», «а» просматривает все отношения «ОДНОР_ИГ», где этот союз является главным словом. Если среди зависимых слов есть слова $w_1, w_2, ..., w_k, k \in N$ из списка существительных, обозначающих величину или значение, и ровно одно слово не из этого списка, то отношения, в которых рассматриваемый союз — главный, разрываются, а в том отношении, где этот союз является зависимым, он заменяется на единственное найденное слово не из списка существительных, обозначающих величину или значение. Далее происходит перебор вершин ориентированного дерева от вершины, которой соответствует рассматриваемый союз, по направлению к корню дерева. Пусть вершине, получаемых при этом переборе, соответствует слово w из списка существительных, обозначающих величину или значение. Тогда перебор прекращается; найденное слово в том отношении, где оно является зависимым, заменяется на рассматриваемый союз; создаются отношения «ОДНОР_ИГ», в которых главным словом является рассматриваемый союз, а зависимыми — слова $w, w_1, w_2, \ldots, w_k$.

Данное правило введено по следующей причине: оказывается полезным считать, что существительные, обозначающие значение, могут быть однородными только с существительными, обозначающими значение. Это правило применяется сразу после применения правила 23 или 24.

Следующие три правила применяются тогда, когда для предложения уже построено дерево зависимостей. Эти правила применяются к каждой подходящей вершине дерева.

29) Правило, создающее множественный актант из слов, являющихся зависимыми от одного и того же слова в отношении «ОДНОР_ИГ». Множественный актант удобно представлять как вершину, из которой исходит ребро с меткой «МНА» к каждой вершине, соответствующей слову из множественного актанта. Основная работа по созданию такого множественного актанта уже проделана при применении правил 23 и 24. Теперь же достаточно указать, что каждая вершина, из которой исходит хотя бы одно ребро с меткой «ОДНОР_ИГ» (в действительности, как видно из построения отношений «ОДНОР_ИГ», не бывает вершин, из которых выходило бы ровно одно такое ребро), соответствует множественному актанту, а метку ребра поменять на « МНА ».

30) Правило, создающее множественный актант из слов, являющихся зависимыми от одного и того же слова в отношении «ВАРИАНТ» (т.е. из нескольких тире, заменяющих одно и то же слово) и из самого этого слова. Достаточно указать, что каждая вершина, из которой исходит хотя бы одно ребро с меткой «ВАРИАНТ», соответствует множественному актанту, а метку ребра поменять на « МНА ».

31) Правило, создающее множественный актант из всех слов $w_1, w_2, \ldots, w_k$, являющихся зависимыми от одного и того же слова w в отношении с одним и тем же названием v (но не «ОДНОР_ИГ», так как этот случай рассматривается отдельно, и не «ГЛ_ДОП», так как такие слова не являются однородными). В дерево зависимостей добавляется новая вершина w_0, куда из вершины, соответствующей слову w, опускается ребро с меткой v. Все ребра, ведущие в вершины, соответствующие словам $w_1, w_2, \ldots, w_k$, удаляются, и в каждую из этих вершин из w_0 проводится ребро с меткой « МНА ». Наконец, вершина w_0 помечается как соответствующая множественному актанту.

Приведем пример предложения, для которого описанный алгоритм построил дерево зависимостей. Рассмотрим следующее предложение: «При способе уменьшаемого остатка годовая сумма амортизационных

отчислений определяется исходя из остаточной стоимости объекта основных средств на начало отчетного года и нормы амортизации, исчисленной исходя из срока полезного использования этого объекта и коэффициента не выше 3, установленного организацией». Это предложение — несколько измененная часть пункта 19 ПБУ 6/01 [15].

Будем считать, что на вход программы, осуществляющей синтаксический анализ, поступила последовательность лексем, на которые было разбито предложение, а также следующие токены (они получены благодаря морфологическому разбору, произведенному программой, созданной на проекте АОТ [5]):

1) При ПРИ ПРЕДЛ,
2) способе СПОСОБ С,но,мр,пр,ед,
3) уменьшаемого УМЕНЬШАТЬ ПРИЧАСТИЕ,стр,пе,нс,но,од,нст,мр,рд,ед,
4) остатка ОСТАТОК С,но,мр,рд,ед,
5) годовая ГОДОВОЙ П,но,од,жр,им,ед,
6) сумма СУММА С,но,жр,им,ед,
7) амортизационных АМОРТИЗАЦИОННЫЙ П,но,од,рд,мн,
8) отчислений ОТЧИСЛЕНИЕ С,но,ср,рд,мн,
9) определяется ОПРЕДЕЛЯТЬСЯ Г,дст,нп,нс,3л,нст,ед,
10) исходя ИСХОДИТЬ нп,нс,
11) из ИЗ
12) остаточной ОСТАТОЧНЫЙ П,кач,но,од,жр,рд,ед,
13) стоимости СТОИМОСТЬ С,но,жр,рд,ед,
14) объекта ОБЪЕКТ С,но,мр,рд,ед,
15) основных ОСНОВНОЙ П,но,од,рд,мн,
16) средств СРЕДСТВО С,но,ср,рд,мн,
17) на НА ПРЕДЛ,

18) начало НАЧАЛО С,но,ср,вн,ед,

19) отчетного ОТЧЕТНЫЙ П,кач,но,од,мр,рд,ед,

20) года ГОД С,но,мр,рд,ед,

21) и И СОЮЗ,

22) нормы НОРМА С,но,жр,вн,рд,им,ед,мн,

23) амортизации АМОРТИЗАЦИЯ С,но,жр,рд,ед,

24) , ,

25) исчисленной ИСЧИСЛИТЬ ПРИЧАСТИЕ,стр,пе,св,но,од,прш,жр,пр,тв, дт,рд,ед,

26) исходя ИСХОДИТЬ нп,нс,

27) из ИЗ

28) срока СРОК С,но,мр,рд,ед,

29) полезного ПОЛЕЗНЫЙ П,кач,но,од,ср,рд,ед,

30) использования ИСПОЛЬЗОВАНИЕ С,но,ср,рд,ед,

31) этого ЭТОТ МС-П,но,од,мр,рд,ед,

32) объекта ОБЪЕКТ С,но,мр,рд,ед,

33) и И СОЮЗ,

34) коэффициента КОЭФФИЦИЕНТ С,но,мр,рд,ед,

35) не НЕ ЧАСТ,

36) выше ВЫШЕ Н,

37) 3 3 ЧИСЛ-П,но,од,жр,тв,ед,

38) , ,

39) установленного УСТАНОВИТЬ ПРИЧАСТИЕ,стр,пе,св,но,од,прш,ср,мр, вн,рд,ед,

40) организацией ОРГАНИЗАЦИЯ С,но,жр,тв,ед,

41) . .

В результате применения описанного алгоритма без использования правил 29 и 31 создаются следующие синтаксические отношения (у отношения сначала указывается главное слово, а потом зависимое):

1) НЕДЕЛИМ(исходя, из) по правилу 3а предварительной обработки (таких отношений создается два, так как «исходя из» встречается в предложении дважды);
2) ГЛ_ДОП(определяется, при) по правилу 14;
3) ГЕНИТ_ИГ(способе, остатка) по правилу 4;
4) ДОП(при, способе) по правилу 15;
5) ПРИЧ_СУЩ(остатка, уменьшаемого) по правилу 22;
6) ПРИЛ_СУЩ(сумма, годовая) по правилу 22;
7) ГЕНИТ_ИГ(сумма, отчислений) по правилу 4;
8) ПОДЛ(определяется, сумма) по правилу 17;
9) ПРИЛ_СУЩ(отчислений, амортизационных) по правилу 22;
10) ГЛ_ДОП(определяется, исходя) по правилу 14;
11) ПРИЛ_СУЩ(стоимости, остаточной) по правилу 22;
12) ГЕНИТ_ИГ(стоимости, объекта) по правилу 4;
13) ДОП(из, стоимости) по правилу 15 (однако затем по правилу 23 это отношение заменяется на отношение
14) ДОП(из, и));
15) ГЕНИТ_ИГ(объекта, средств) по правилу 4;
16) ПРИЛ_СУЩ(средств, основных) по правилу 22;
17) ОПР(стоимости, на) по правилу 5;
18) ГЕНИТ_ИГ(начало, года) по правилу 4;
19) ДОП(на, начало) по правилу 15;

20) ПРИЛ_СУЩ(года, отчетного) по правилу 22;

21) ОДНОР_ИГ(и, стоимости) по правилу 23;

22) ОДНОР_ИГ(и, нормы) по правилу 23;

23) ГЕНИТ_ИГ(нормы, амортизации) по правилу 4;

24) ПРИЧ_СУЩ(амортизации, исчисленной) по правилу 21;

25) ГЛ_ДОП(исчисленной, исходя) по правилу 14;

26) ГЕНИТ_ИГ(срока, использования) по правилу 4;

27) ДОП(из, срока) по правилу 15 (однако затем по правилу 23 это отношение заменяется на отношение

28) ДОП(из, и));

29) ПРИЛ_СУЩ(использования, полезного) по правилу 22;

30) ГЕНИТ_ИГ(использования, объекта) по правилу 4;

31) ПРИЛ_СУЩ(объекта, этого) по правилу 22;

32) ОДНОР_ИГ(и, срока) по правилу 23;

33) ОДНОР_ИГ(и, коэффициента) по правилу 23;

34) ОТР(выше, не) по правилу 3;

35) ОПР(коэффициента, выше) по правилу 26;

36) КОЛИЧ(выше, 3) по правилу 13;

37) ПРИЧ_СУЩ(коэффициента, установленного) по правилу 21;

38) ДОП(установленного, организацией) по правилу 15.

В случае применения правил 29 и 31 к вершинам построенного дерева зависимостей, метки ребер, соответствующих отношениям 21, 22, 32 и 33, поменяются с «ОДНОР_ИГ» на «МНА». Кроме того, вершины, из которых эти ребра выходят, будут помечены как соответствующие множественному актанту. Как можно заметить, правило 29 будет применено дважды, а правило 31 — ни разу.

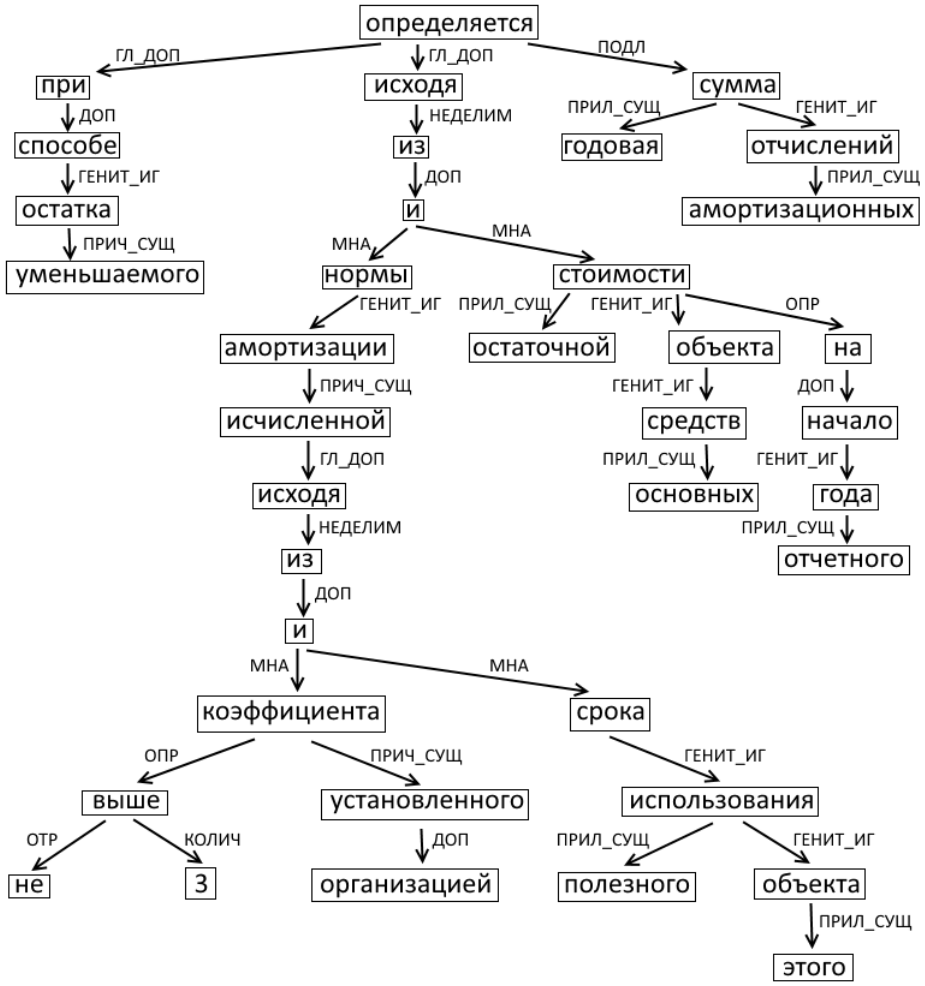


Рис. 1. Синтаксический граф

4. Упрощение синтаксического графа предложения

После синтаксического разбора каждая вершина в графе, за исключением вершин, добавленных в граф по правилу 31, будет соответствовать

некоторому токену. Однако некоторой сущности в предложении может соответствовать не одно слово, а сочетание слов.

Рассмотрим произвольное поддерево синтаксического графа, состоящее из двух или более вершин. Пусть его вершины — $\alpha_1, \alpha_2, \ldots, \alpha_n$, причем α_1 — его корень. Тогда для вершин $\alpha_1, \alpha_2, \ldots, \alpha_n$ определена операция *объединения вершин*, состоящая из следующих шагов:

1) в граф добавляется вершина α;

2) ребро, входящее в вершину α_1, перенаправляется в вершину α;

3) для каждого ребра e из каждой вершины $\alpha_i, i \leq n$, кроме ребер, ведущих в какую-либо вершину $\alpha_j, j \leq n$, проводится ребро из вершины α, ведущее в ту же вершину, что и e, и обладающее той же меткой;

4) вершине α сопоставляется набор всех токенов, сопоставленных вершинам $\alpha_1, \alpha_2, \ldots, \alpha_n$.

5) вершины $\alpha_1, \alpha_2, \ldots, \alpha_n$ и все входящие в них и выходящие из них ребра удаляются.

Граф будем называть *упрощенным синтаксическим графом*, если он получен из синтаксического графа предложения путем нескольких (в т.ч., 0) применений операции объединения вершин. Мы будем строить упрощенный граф таким образом, чтобы в нем каждой вершине соответствовала либо одна сущность, либо одно или несколько служебных слов (предлогов, союзов и т.д).

На объединяемые вершины могут быть наложены определенные ограничения:

а) вершину α может быть запрещено объединять с любой вершиной, в которую из α ведет ребро, если вершине α сопоставлены слова (сочетания слов) «за исключением», «кроме», «числитель», «знаменатель», «соотношение», «а также», «исходя», «один из», «в течение», «по», «при»; если вершине α сопоставлено количественное отношение («больше», «свыше» и т.д.);

б) вершину α может быть запрещено объединять с вершиной, из которой в α ведет ребро, если вершина α соответствует множественному актанту; если ребру, ведущему в вершину α, сопоставлено одно из следующих синтаксических отношений: «ПРИДАТ_ОПР», «В_СЛУЧАЕ», «В_ОТНОШЕНИИ», «ОТР», «УСЛ», «ЕСЛИ», «КОЛИЧ»; если хотя бы одному ребру, выходящему из вершины α, сопоставлено синтаксическое отношение «ПРИДАТ_ОПР».

Цель данных ограничений - избежать объединения вершин, которым приписаны слова, обозначающие логические функции и отношения. В некоторых случаях эти ограничения могут игнорироваться.

Упрощение графа производится с помощью перечисленных ниже приемов. Во всех случаях указано, игнорирует ли прием ограничения а и б.

1) Пусть «способ» — одно из слов, соответствующих некоторой вершине α, и пусть из α в некоторую вершину β ведет ребро, соответствующее синтаксическому отношению «ГЕНИТ_ИГ». Тогда объединяются вершины α, β и все вершины, в которые можно перейти из β. Ограничения а и б игнорируются.

2)Пусть из некоторой вершины α в некоторую вершину β ведет ребро, соответствующее синтаксическому отношению «НЕДЕЛИМ». Тогда объединяются вершины α и β. Ограничения а и б игнорируются.

3) Пусть некоторой вершине α сопоставлен один токен, являющийся предлогом (но не «по», «при», «кроме»), и из α выходит лишь одно ребро. Тогда α объединяется с вершиной, в которую из α ведет ребро. Ограничение а учитывается, ограничение б - игнорируется.

4) Путь несколько идущих подряд слов предложения составляют некоторую сущность из хранящегося в отдельном файле списка сущностей (например, «объект основных средств», «физическое лицо» и т.д). Тогда объединяются все вершины, соответствующие этим словам. Ограничения а и б игнорируются.

5) Пусть из вершины α в вершину β ведет ребро. Тогда α и β объединяются. Ограничения а и б учитываются.

Приемы используются в следующем порядке: сначала происходит попытка применить прием 1 там, где это возможно; затем то же делается с приемом 2. Далее в цикле по всем вершинам происходит попытка применения приема 3, после чего в цикле по всем словам предложения происходит попытка использовать прием 4. Наконец, везде, где возможно, используется прием 5.

Нетрудно заметить, что прием 5 фактически является основным, тогда как почти все остальные приемы нужны лишь для случаев, когда необходимо проигнорировать хотя бы одно из ограничений а и б.

В результате применения указанных приемов к синтаксическому графу, изображенному на рисунке 1, мы получим упрощенный синтаксический граф, изображенный на рисунке 2.

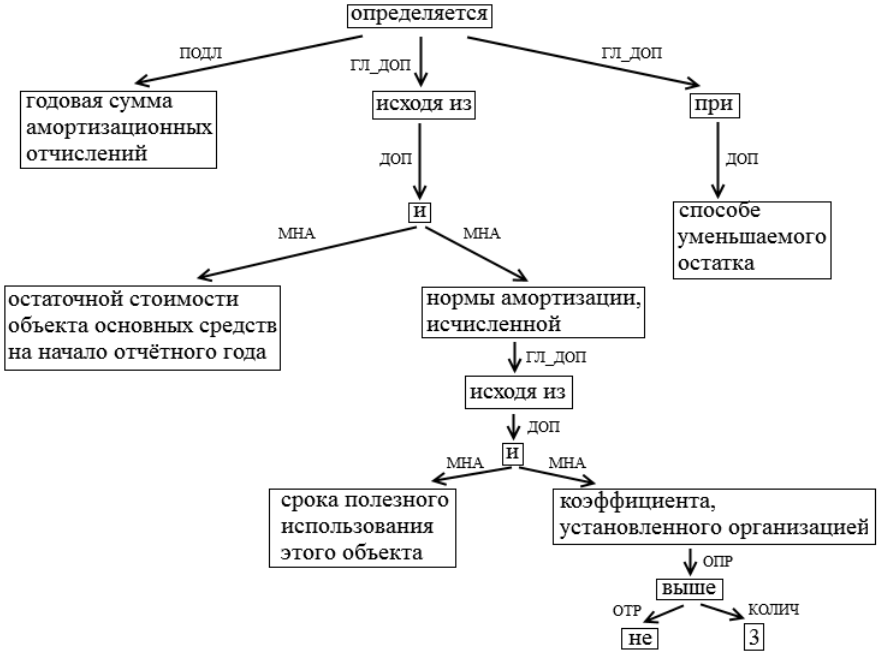


Рис. 2. Упрощенный синтаксический граф

5. Отождествление сущностей

Пусть по каждому предложению текста закона уже построена формула. Переменные этих формул соответствуют определенным текстовым фрагментам.

Перед построением формулы каждой сущности необходимо сопоставить переменную. Для того, чтобы одной и той же сущности в разных предложениях была сопоставлена одна и та же переменная, проводится процедура отождествления сущностей. Часто сущность полностью определяется текстовым фрагментом (предполагается, что если слова в двух фрагментах текста отличаются только своей формой, например, находятся в разных падежах, то эти два фрагмента определяют одну и ту же сущность). В некоторых случаях этого, однако, не происходит. Для представления множества всех сущностей удобно использовать ориентированное дерево, каждой вершине которого (кроме корня) сопоставлена некоторая сущность и некоторый текст. Если в этом дереве из вершины,

которой сопоставлена сущность A, ведет ребро в вершину, которой сопоставлена сущность B, то это означает, что B — атрибут (т.е. свойство или действие) A.

Перед построением дерева сущностей каждое личное местоимение («он», «она», «его» и др.) заменяется на слово, на которое это местоимение указывает. Для этого ищется ближайшее к личному местоимению в сторону начала предложения слово, согласованное с этим местоимением в роде и числе.

Дерево сущностей строится по упрощенному синтаксическому графу предложения. Этот граф представляет собой ориентированное дерево. Алгоритм вначале рассматривает его корень, затем — все вершины, которые из корня ведет ребро, и т.д. Вначале дерево сущностей состоит из двух вершин: корня α_0, которому не сопоставлена никакая сущность, и вершины α_1, в которую из корня ведет ребро и которой сопоставлена сущность «объект основных средств».

Обозначим рассматриваемую вершину упрощенного синтаксического графа через α. Если вершине α соответствует только одно или несколько служебных слов, то этой вершине не соответствует никакая сущность, и рассмотрение этой вершины сразу же прекращается. Каждой из остальных вершин упрощенного синтаксического графа будет соответствовать одна вершина в дереве сущностей.

Рассмотрим в упрощенном синтаксическом графе цепь, ведущую из корня в вершину α. Обозначим ближайшую к α (но отличную от нее) вершину этой цепи, которой соответствует некоторая вершина в строящемся дереве сущностей, через β (если таковая вершина найдется). Вершину цепи, в которую из β ведет ребро, обозначим через δ (эта вершина может совпадать с α). Пусть вершине β в строящемся дереве сущностей соответствует вершина β'.

Рассмотрение вершины α заключается в попытке использования следующих приемов.

1) Если из α в некоторую вершину γ ведет ребро, соответствующее синтаксическому отношению «ПОДЛ», то для целей построения дерева сущностей считается, что в упрощенном синтаксическом графе есть ребро из γ в α, но нет ребра из α в γ, и что ребро, ведущее (в действительности) в α (если оно есть), ведет в γ. Таким образом, мы считаем, что сказуемое является атрибутом подлежащего, а не наоборот.

2) Пусть в тексте, соответствующем вершине α, есть слово «объект», причем из соответствующей этому слову вершины первоначального (т.е. неупрощенного) синтаксического графа не выходит ребро, которому сопоставлено синтаксическое отношение «ГЕНИТ_ИГ»; либо пусть в тексте, соответствующем α, есть слова «объект основных средств». Тогда вершине α соответствует вершина дерева сущностей, в которую ведет ребро из α_1, и которой сопоставлена сущность, соответствующая α, и сопоставленный α текст без слова «объект» (во втором случае — без слов «объект основных средств»).

3) Если вершина β отсутствует, то вершине α соответствует вершина дерева сущностей, в которую ведет ребро из корня, и которой сопоставлены текст и сущность, соответствующие вершине α.

4) Пусть ребру из β в δ сопоставлено синтаксическое отношение «ДОП», «ГЛ_ДОП» либо «КОЛИЧ», а вершине β сопоставлены слова «в случае», «кроме», «за исключением», «в отношении», «исходя из», «при», «по», «для», «пониматься», «являться», либо слова, обозначающие количественное отношение («больше», «равный» и т.д). Тогда вершине α соответствует вершина дерева сущностей, в которую ведет ребро из корня, и которой сопоставляются текст и сущность, соответствующие вершине α.

5) Пусть ребру из β в δ сопоставлено синтаксическое отношение «УСЛ», а из α не выходит ребро, соответствующее синтаксическому отношению «ПОДЛ». Тогда вершине α соответствует вершина дерева сущностей, в которую ведет ребро из корня, и которой сопоставляются текст и сущность, соответствующие вершине α.

6) Вершине α в строящемся дереве сущностей будет соответствовать вершина, в которую ведет ребро из β', и которой сопоставлены те же текст и сущность, что и вершине α.

Порядок применения приемов следующий: сначала используется прием 1; далее происходит попытка применения приемов 2–6 (в порядке возрастания номера приема). Если попытка использования одного из приемов 2–6 успешна, то рассмотрение вершины α сразу же прекращается. Нетрудно заметить, что ситуация, когда все приемы 2–6 неприменимы, невозможна, так как ровно один из приемов 3 и 6 всегда применим.

В процессе построения дерева сущностей каждой сущности сопоставляется переменная.

После того, как дерево сущностей построено, происходит небольшое изменение текста некоторых вершин этого дерева: из текста удаляется слово «исчисленный», если в упрощенном синтаксическом графе из соответствующей этому тексту вершины ведет ребро в вершину со словами «исходя из». Кроме того, из текста удаляются указательные местоимения («этот», «такой» и т.д.).

На рисунке 3 изображено дерево сущностей, построенное по упрощенному синтаксическому графу, изображенному на рисунке 2.

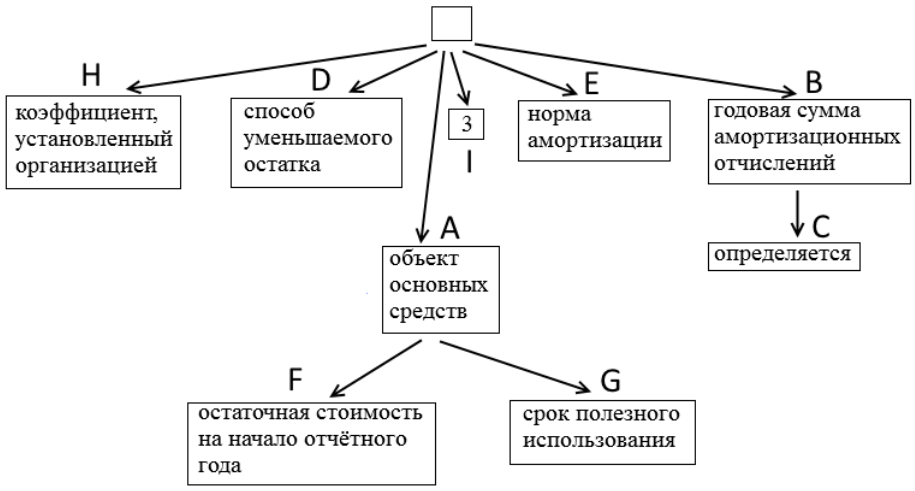


Рис. 3. Дерево сущностей

6. Построение логической формулы

Для каждого предложения по его упрощенному синтаксическому графу (с использованием дерева сущностей) строится логическая формула. При этом используются следующие приемы.

1) Если в предложении присутствует одно из следующих слов или сочетаний слов: «по», «в течение», «в случаях», «при», «в отношении», «если», то управляемая этим словом или сочетанием слов

часть предложения определяет область применения статьи закона либо некоторое условие. Пусть в соответствующую этому слову вершину γ ведет ребро из некоторой вершины α. Вершину, из которой в α ведет ребро (если таковая найдется) обозначим как β. Вершину, в которую из γ ведет ребро (если таковая найдется; таких вершин не может быть более одной) обозначим как δ. Хотя бы одна из вершин β и δ будет существовать. Возможны четыре случая:

а) вершине β сопоставлено тире. Это означает, что в предложении указано сразу несколько различных условий применения статьи закона, причем при каждом условии применяется только соответствующая ему часть статьи. Фактически в этом случае статью можно разделить на несколько независимых частей, что и осуществляется при построении формулы.

В рассматриваемом случае вершина β существует и соответствует множественному актанту. Пусть $\alpha_1, \ldots, \alpha_r$ — вершины, в которые из β ведет ребро с меткой $<<\ >>$ (α — одна из этих вершин). Тогда итоговая формула будет иметь вид $A_1 \& \ldots A_r$, где A_i — формула, построенная по исходному графу, у которого все вершины $\alpha_j, j \neq i$ удалены, а вершины β и α_i объединены.

Все последующие случаи рассматриваются в предположении, что случай а не имеет места.

б) вершина δ существует. Пусть A — формула, соответствующая графу, состоящему из всех вершин, достижимых из δ (будем считать, что каждая вершина достижима из себя), и соединяющих их ребер, B — формула, соответствующая графу, состоящему из всех остальных вершин (кроме γ) и соединяющих их ребер. Тогда итоговая формула будет иметь вид $(A \to B)$;

в) вершина δ не существует, а β не соответствует множественному актанту. Пусть A — формула, соответствующая графу, состоящему из всех вершин, достижимых из α, и соединяющих их ребер, B — формула, соответствующая графу, состоящему из всех остальных вершин (кроме γ) и соединяющих их ребер. Тогда итоговая формула будет иметь вид $(A \to B)$;

г) вершина δ не существует, а β соответствует множественному актанту. Это означает, что слова «по», «в течение», «в случа-

ях», «при», «в отношении», «если» относятся ко всему множественному актанту. Пусть B — формула, соответствующая графу, состоящему из всех вершин, достижимых из β, и соединяющих их ребер, C — формула, соответствующая графу, состоящему из всех остальных вершин и соединяющих их ребер. Тогда итоговая формула будет иметь вид $(B \to C)$.

2) Если в предложении некоторая его часть подчинена действию с помощью слова «кроме», то это означает, что действие выполняется тогда и только тогда, когда не выполняется условие, выраженное частью предложения, управляемой словом «кроме». Пусть это слово сопоставлено вершине α. Вершину, из которой в α ведет ребро, обозначим β. Пусть A — формула, соответствующая подграфу, состоящему из всех вершин, достижимых из β, B — формула, соответствующая графу, состоящему из всех остальных вершин (кроме α) и соединяющих их ребер. Итоговая формула будет иметь вид $(\neg A \sim B)$.

3) Слово «не» означает логическое отрицание. Пусть это слово сопоставлено вершине α, в которую входит ребро, выходящее из вершины β. Если по вершине β построено логическое выражение A, то после применения приема оно превращается в $(\neg A)$.

4) Союз «либо», «или» при перечислении однородных членов предложения,если речь не идет о перечислении аргументов некоторой функции, означает логическое «или» либо функцию, истинную, когда истинно значение ровно одного ее аргумента. Пусть вершина α — это множественный актант, которому соответствует несколько однородных членов предложения, а сами эти однородные члены предложения сопоставлены вершинам $\alpha_1, \ldots, \alpha_r$, в каждую из которых ведет ребро из α. Пусть также в вершину α не ведет ребро из вершины, которой сопоставлены слова «исходя из», и хотя бы одной из вершин $\alpha, \alpha_1, \ldots, \alpha_r$, либо вершине, в которую из $\alpha, \alpha_1, \ldots, \alpha_r$ ведет ребро, сопоставлено слово «или» («либо»). Возможны 2 случая:

   а) среди рассматриваемых вершин $\alpha_j, j \in \{1, \ldots, r\}$, найдется вершина α_i такая, что из нее выходит ребро, соответствующее синтаксическому отношению «УСЛ». Пусть это ребро ведет в вершину γ. Обозначим через A формулу, построенную по графу, состоящему из всех вершин, достижимых из γ, и

соединяющих их ребер; через B — формулу, построенную из всех остальных вершин, достижимых из α_i, и соединяющих их ребер. Если осталась лишь одна вершина $\alpha_j, i \neq j$, то обозначим через формулу, построенную по графу, состоящему из всех вершин, достижимых из α_j, и соединяющих их ребер. Если же таких вершин α_j, что $j \neq i$, больше одной, то обозначим через формулу, полученную применением 4.1 или 4.2 ко всем рассматриваемым вершинам $\alpha_s, s \in \{1, \ldots, r\}$, кроме α_i, которую мы исключим из рассмотрения. Тогда результатом применения приема будет формула $(A\&B \vee \neg A\&C)$;

б) среди рассматриваемых вершин $\alpha_j, j \in \{1, \ldots, r\}$, нет ни одной вершины α_i такой, что из нее выходит ребро, соответствующее синтаксическому отношению «УСЛ». Тогда, если обозначить через $A_j, j \in \{1, \ldots, r\}$, формулу, построенную по графу, состоящему из всех вершин, достижимых из α_j, то в результате применения приема будет создана формула, являющаяся дизъюнкцией всех формул A_j, соответствующих рассматриваемым вершинам $\alpha_j, j \in \{1, \ldots, r\}$;

5) Союз «и» или отсутствие союзов при перечислении однородных членов предложения, если речь не идет о перечислении аргументов некоторой функции, означает логическое «и» либо «или». Пусть вершина α — это множественный актант, которому соответствует несколько однородных членов предложения, а сами эти однородные члены предложения сопоставлены вершинам $\alpha_1, \ldots, \alpha_r$, в каждое из которых ведет ребро из α. Пусть также в вершину α не ведет ребро из вершины, которой сопоставлены слова «исходя из», и никакой из вершин $\alpha, \alpha_1, \ldots, \alpha_r$ и вершин, в которую из $\alpha, \alpha_1, \ldots, \alpha_r$ ведет ребро, не сопоставлены слова «или», «либо», «а также». Если однородные члены предложения — глаголы, в результате применения приема будет создана формула $(A_1\&A_2\& \ldots \&A_r)$. В противном случае результатом применения приема будет формула $(A_1 \vee A_2 \vee \ldots \vee A_r)$.

6) Слова «исходя из» означают некоторую функцию. Пусть эти слова сопоставлены вершине α, а β' — вершина, из которой в α идет ребро. Если вершине β' соответствует глагол, обозначим через β вершину, в которую из β' ведет ребро с меткой «ПОДЛ»; иначе обозначим через β саму вершину β'. Обозначим через γ вершину, в

которую ведет ребро из α (найдется лишь одна такая вершина γ). Пусть B — переменная, соответствующая вершине β. Возможно несколько случаев.

а) Вершине γ не сопоставлен множественный актант. Пусть C — переменная, обозначающая объект, определенный текстовым фрагментом, соответствующим вершине γ. Тогда в результате применения приема графу, состоящему из всех вершин, достижимых из β', и соединяющих их ребер будет поставлена в соответствие формула $((B = f_\beta(C))\&G)$, где f_β — некоторая функция, в тексте закона не определенная явно, G — формула, полученная путем применения приема 8 к вершине γ, либо приемов 6 или 9 к вершине, в которую из γ ведет ребро; если указанные приемы неприменимы, то G — тождественная истина.

$G_i, i \in 1, 2$ — формула, полученная путем применения приема 8 к вершине α'_i, либо приемов 6 или 9 к вершине, в которую из α'_i ведет ребро; если условия применения ни одного из этих приемов не соблюдены, то G_i — тождественная истина.

б) Вершине γ сопоставлен множественный актант. Пусть соответствующие ему однородные члены предложения сопоставлены вершинам $\gamma_1, \ldots, \gamma_r$, в каждое из которых ведет ребро из γ. Пусть никакой из вершин $\gamma, \gamma_1, \ldots, \gamma_r$ и вершин, в которую из $\gamma, \gamma_1, \ldots, \gamma_r$ ведет ребро, не сопоставлены слова «или», «либо», «а также». Пусть $A_j, j \in \{1, \ldots, r\}$ — переменная, соответствующая вершине γ_j, либо поставленная в соответствие вершине γ_j вспомогательная переменная, если γ_j соответствует множественный актант. Тогда в результате применения приема графу, состоящему из всех вершин, достижимых из β', и соединяющих их ребер будет поставлена в соответствие формула $((B = f_\beta(A_1, A_2, \ldots, A_r))\&G_1\& \ldots \&G_r)$, где f_β - некоторая функция, в тексте закона не определенная явно, а G_j — формула, полученная путем применения приема 8 к вершине γ_j, либо приемов 6 или 9 к вершине, в которую из γ_j ведет ребро (если найдется такая вершина, удовлетворяющая условиям применения хотя бы одного из этих приемов), и тождественная истина иначе.

в) Вершине γ сопоставлен множественный актант. Пусть соответствующие ему однородные члены предложения сопоставлены

вершинам $\gamma_1, \ldots, \gamma_r$, в каждое из которых ведет ребро из γ, и хотя бы одной из вершин $\gamma, \gamma_1, \ldots, \gamma_r$, либо вершине, в которую из $\gamma, \gamma_1, \ldots, \gamma_r$ ведет ребро, сопоставлено слово «или» («либо»). Тогда в результате применения приема графу, состоящему из всех вершин, достижимых из β', и соединяющих их ребер будет поставлена в соответствие формула $D\&G$, где D — формула, строящаяся с помощью приема из пункта 7, примененного к вершине γ. G — формула, полученная путем применения приема 6 или приема 9 к вершине γ, если она удовлетворяет условиям применения хотя бы одного из этих приемов, и тождественная истина иначе.

7) Пусть вершине γ сопоставлен множественный актант, и из вершины α, которой сопоставлены слова «исходя из», ведет ребро либо в γ, либо в вершину, из которой в γ можно попасть, переходя по ребрам с меткой «МНА» (в соответствии с их направлением). Пусть вершины β' и β ищутся для вершины α так же, как в пункте 6. Пусть также соответствующие γ однородные члены предложения сопоставлены вершинам $\gamma_1, \ldots, \gamma_r$, в каждое из которых ведет ребро из γ, и хотя бы одной из вершин $\gamma, \gamma_1, \ldots, \gamma_r$, либо вершине, в которую из $\gamma, \gamma_1, \ldots, \gamma_r$ ведет ребро, сопоставлено слово «или» («либо»). Пусть $D_j, j \in \{1, \ldots, r\}$ — это $f_\beta(A_j)$, если вершине γ_j не сопоставлен множественный актант; в противном случае это соответствующая множественному актанту формула, строящаяся с помощью приема из пункта 7, примененного к вершине γ_j. Пусть также G_j — формула, полученная путем применения приема 8 к вершине γ_j, либо приемов 6 или 9 к вершине, в которую из γ_j ведет ребро (если найдется такая вершина, удовлетворяющая условиям применения хотя бы одного из этих приемов), и тождественная истина иначе. Тогда будет создана формула $D_1\&G_1 \vee \ldots \vee D_r\&G_r$.

8) Пусть среди слов, соответствующих вершине α, есть слово «соотношение». Это означает, что данной вершине соответствует функция, представляющая собой отношение двух величин. Из вершины α в этом случае ведет ребро в некоторую вершину α', соответствующую множественному актанту, а из вершины α' выходит два ребра с меткой «МНА». Пусть α_1 и α_2 — вершины, в которые входят эти ребра; эти вершины соответствуют числителю и знаменателю отношения. Объект, находящийся в числителе или знаменателе отношения, соответствует либо самой вершине α_i, либо вершине, в

которую из α_i ведет ребро; обозначим вершину, которой соответствует этот объект, через α'_i. В случае, когда α_i и α'_i не совпадают, из вершины α_i должно вести ребро в вершину, которой соответствует слово «числитель» либо слово «знаменатель», что позволяет определить, в числителе или знаменателе отношения находится объект. В том случае, когда α_i и α'_i совпадают, считается, что в числителе оказывается объект, чей текстовый фрагмент находится в предложении раньше.

В результате применения приема графу, состоящему из всех вершин, достижимых из α, и соединяющих их ребер будет поставлена в соответствие формула $((A = C_1/C_2)\&G_1\&G_2)$, где A — переменная, сопоставленная вершине α; C_1 и C_2 — переменные, соответствующие объектам, находящимся в числителе и знаменателе отношения; $G_i, i \in 1, 2$ — формула, полученная путем применения приема 8 к вершине α'_i, либо приемов 6 или 9 к вершине, в которую из α'_i ведет ребро; если условия применения ни одного из этих приемов не соблюдены, то G_i — тождественная истина.

9) Для слов, обозначающих количественные отношения («больше», «меньше», «равный» и т.д.), в формуле будет присутствовать соответствующий предикат. Пусть α — вершина, которой сопоставлено одно из этих слов, γ — вершина, из которой в α ведет ребро. Пусть выходящее из α ребро, которому соответствует синтаксическое отношение «КОЛИЧ», ведет в вершину β. Пусть A – переменная, соответствующая вершине β, C — переменная, соответствующая вершине γ. Тогда графу, состоящему из вершин α, β, γ и соединяющих их ребер, будет соответствовать формула $pr_\alpha(C, A)$, где $pr_\alpha(C, A)$ — соответствующий вершине α предикат.

Применение приема означает проверку для вершины всех условий, указанных в тексте приема, и построение некоторой формулы, если эти условия выполнены. Предполагается, что один и тот же прием не может быть применен к одной и той же вершине (обозначенной во всех приемах как α) более одного раза, за исключением случаев, когда прием обращается рекурсивно к себе.

Алгоритм построения формулы по графу, состоящему из всех вершин, достижимых из данной, и соединяющих их ребер, состоит в следующем. Проверяется, является ли данная вершина вершиной α из приемов 2–6, 8, 9. Если да, то применяется все приемы, которые можно применить

(в порядке возрастания номеров), притом используемые приемы могут вызвать этот же алгоритм для некоторых вершин, к которым из данной идет ребро. Если ни один из приемов 2–6, 8, 9 нельзя применить, то данной вершине сопоставляется переменная, обозначающая объект, определенный текстовым фрагментом, соответствующим этой вершине. Производится конъюнкция этой переменной и всех формул, построенных с помощью этого же алгоритма по вершинам, в которые из данной вершины идет ребро.

Итоговый алгоритм построения формулы в соответствии с описанными приемами следующий.

Если условия приема 1 выполнены для какой-либо вершины (не должно быть более одной такой вершины), то в результате применения этого приема исходный граф разбивается на 2 подграфа (кроме случая, когда выполнены условия приема 1а: в этом случае рассматриваются несколько графов, полученных из исходного, и каждый такой граф разбивается на 2 подграфа). В каждом из этих подграфов выбирается корень, и к ним поочередно применяется описанный выше алгоритм построения формулы по графу, состоящему из всех вершин, достижимых из данной.

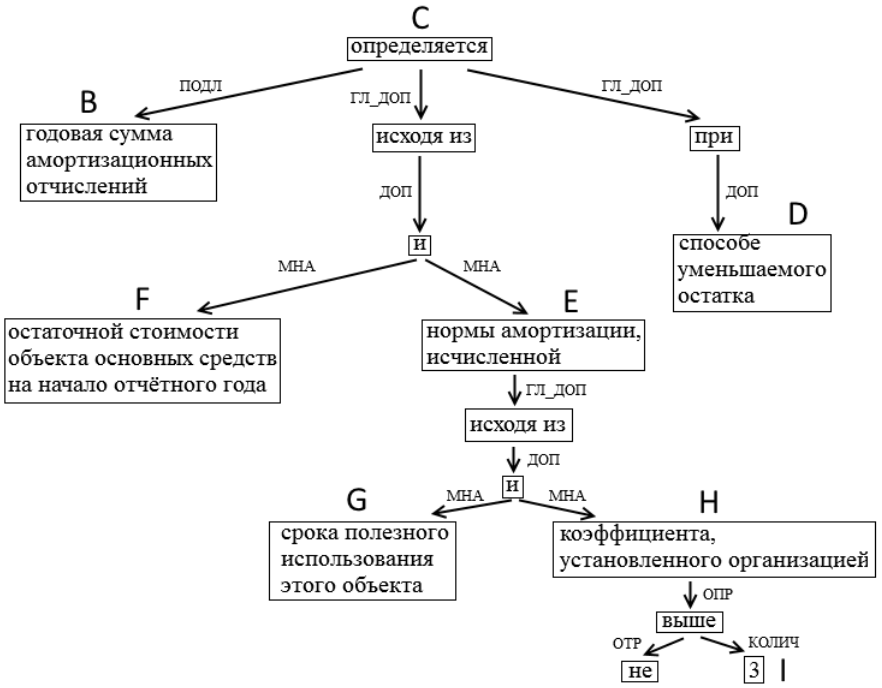
Если условия приема 1 не выполнены ни для какой вершины, то в графе выбирается корень, и к нему применяется описанный выше алгоритм построения формулы по графу, состоящему из всех вершин, достижимых из данной.

На рисунке 4 изображены упрощенный синтаксический граф вместе с сопоставленными его вершинам переменными, а также формула, построенная по упрощенному синтаксическому графу.

7. Построение модели закона по логическим формулам

Когда по каждому предложению закона построена логическая формула, можно перейти непосредственно к построению модели. Как и на всех предыдущих этапах, это можно сделать с помощью применения определенных приемов к имеющимся формулам.

1) Пусть формула имеет вид $A \to B$, где в B имеется выражение вида $C = D$, но нет выражения вида $E = C$. Найдем в уже построенном фрагменте графа модели закона вершину второго вида, в которой



$$D \to ((B = f(F,E)) \,\&\, (E = g(G,H)) \,\&\, !(H > I))$$

Рис. 4. Формула, построенная по упрощенному синтаксическому графу (f, g — еще не определенные функции)

происходит запись значения в поле памяти, соответствующее C из рассматриваемой формулы. Возможны 2 варианта.

а) Такая вершина есть. Обозначим ее α. Рассмотрим вершину, из которой в α ведет ребро. Обозначим ее β. Далее пройдем из β по цепочке ребер с номером 0 против их направления, пока эта цепочка не закончится. Обозначим вершину, в которую мы попадем, как γ. Далее выполним действия, указанные в пункте 1в.

б) Такой вершины нет. Тогда выберем какую-либо не рассмотренную ранее вершину и будем считать ее вершиной второго

вида, соответствующей C из рассматриваемой формулы. Обозначим эту вершину α. Выберем еще одну не рассмотренную ранее вершину, обозначим ее γ. Выпустим из нее ребро в вершину α. Далее выполним действия, указанные в пункте 1в.

в) Будем считать γ вершиной четвертого вида. Рассмотрим какие-либо три новые вершины, выпустим из них в γ по одному ребру, пронумеруем эти ребра числами 0, 1 и 2. Тогда вершине, из которой в γ ведет ребро номер 2, будет соответствовать вычисление логического выражения A. К вершине, из которой в γ ведет ребро номер 1, следует применить прием 3 относительно переменной C. Вершина, из которой в γ ведет ребро номер 0, будет, возможно, рассмотрена в результате применения 1а к одной из оставшихся формул. Если этого не произойдет, значит, процедура вычисления модели никогда не затронет эту вершину.

2) Пусть в формуле имеется выражение вида $A = B$, нет выражения вида $C = A$, и прием 1 к формуле неприменим. Тогда выберем какую-либо не рассмотренную ранее вершину и будем считать ее вершиной второго вида, соответствующей A из рассматриваемой формулы. Обозначим эту вершину α. Выберем еще одну не рассмотренную ранее вершину, обозначим ее γ. Выпустим из нее ребро в вершину α. Вершине γ будет соответствовать вычисление функции B (при помощи приема 3).

3) По фрагменту формулы, соответствующему вычислению значения переменной B, строится один из следующих фрагментов модели (прием 3б используется, если 3а неприменим):

а) Пусть в формуле есть подформула вида $A\&(B = C) \vee \&D$, причем в D входит выражение вида $B = E$. Выберем из всех таких подформул данной формулы (для фиксированного B) ту, что не содержится в другой такой подформуле. Пусть α — вершина, соответствующая вычислению значения B. Будем считать, что это — вершина четвертого вида. Выберем какие-либо три новые вершины, выпустим из них в α по одному ребру, пронумеруем эти ребра числами 0, 1 и 2. Тогда вершине, из которой в α ведет ребро номер 2, будет соответствовать вычисление логического выражения A. Вершине, из которой

в α ведет ребро номер 1, будет соответствовать вычисление функции C (смотрите прием 3б). Если D имеет вид $(B = E)$, вершине, из которой в α ведет ребро номер 0, будет соответствовать вычисление функции E (также с помощью приема 3б); в противном случае следует использовать прием 3а, рассматривая эту вершину в качестве α, а D в качестве выбранной подформулы.

б) Если в формуле есть подформула $B = A$, где A — некоторая переменная, и нет выражений вида $A = C$, где C — некоторая переменная или функция, то будем считать, что рассматриваемая вершина — первого вида, и в ней происходит получение значения A из соответствующего поля памяти. В противном случае речь идет о вычислении некоторой арифметической функции от нескольких аргументов $A_1, \ldots, A_k, k \in N$. Если эта функция в тексте закона не задана явно, то можно, например, потребовать у пользователя ее задания. Если функция уже задана, то ей можно сопоставить несколько вершин третьего типа, моделирующих вычисление этой функции (так, чтобы последний этап вычисления функции проходил в вершине α). Ребра, передающие значения аргументов $A_1, \ldots, A_k$, будут входить в некоторые из этих вершин. К каждой вершине, из которой эти ребра выходят, применяется прием 3 относительно соответствующей переменной.

4) Пусть переменная A — аргумент некоторой функции, и на эту переменную наложено некоторое условие (в виде бинарного или унарного отношения, которому она должна удовлетворять). Тогда выполнение этого условия проверяется в той вершине, в которой вычисляется значение A. В случае невыполнения условия программа требует у пользователя изменить значение переменных, с помощью которых вычисляется A (как правило, это собственно переменная A).

5) Пусть формула имеет вид $A \to B_1 \& \ldots \& B_k, \in N$, где $B_1, \ldots, B_k$ — переменные. Тогда такая формула преобразуется в конъюнкцию формул $A \to B_1, \ldots, A \to B_k$, к которым можно применять прием 6.

6) Пусть формула имеет вид $A \to B$, где B — переменная, A — подформула, не содержащая никаких переменных, кроме булевых.

Пусть B' — переменная B (точнее, соответствующая ей сущность), определенная с помощью всех остальных формул ($B' = 0$, если B не определяется из оставшихся формул). Тогда B определяется как $A \vee B'$.

7) Пусть формула имеет вид $A \to (B \sim C)$ либо $A \to (C \sim B)$, где B — переменная, A, C — подформулы, не содержащие никаких переменных, кроме булевых. Если C — тоже переменная, то для того, чтобы отличить B от C, можно использовать синтаксический граф: в нем соответствующие этим переменным текстовые фрагменты должны быть связаны каким-либо синтаксическим отношением. За B принимается переменная, чей текстовый фрагмент является главным в этом синтаксическом отношении. Пусть B' — переменная B (точнее, соответствующая ей сущность), определенная с помощью всех остальных формул ($B' = 0$, если B не определяется из оставшихся формул). Тогда B определяется как $A\&C \vee B'$.

8) Если функция, определяющая некоторую булеву переменную, не содержит никаких переменных, кроме булевых, то она моделируется с помощью несколько вершин третьего типа. В некоторые из этих вершин будут входить ребра, соответствующие переменным.

Алгоритм построения модели закона с помощью этих приемов следующий. Сначала для каждой формулы, не содержащей никаких переменных, кроме булевых, применяется прием 5; затем ко всем таким формулам применяются приемы 6 и 7; наконец, к каждой такой формуле применяется прием 8. Ко всем остальным формулам применяется прием 1 или 2, а затем — прием 4.

На рисунке 5 изображен фрагмент модели закона, соответствующий предложению «При способе уменьшаемого остатка годовая сумма амортизационных отчислений определяется исходя из остаточной стоимости объекта основных средств на начало отчетного года и нормы амортизации, исчисленной исходя из срока полезного использования этого объекта и коэффициента не выше 3, установленного организацией». Этот фрагмент построен по формуле, приведенной на рисунке 4, с помощью дерева сущностей, откуда были получены соответствующие переменным формулы сущности.

Предполагается, что выяснить, какая именно функция имеется в виду в законе, если эта функция в тексте закона не задана явно, можно, предоставив этот выбор пользователю программы, либо получив эти данные

из каких-либо других нормативно-правовых актов. Последнее решение сложнее реализуется и не всегда может дать результат; в программе в настоящее время реализовано первое решение.

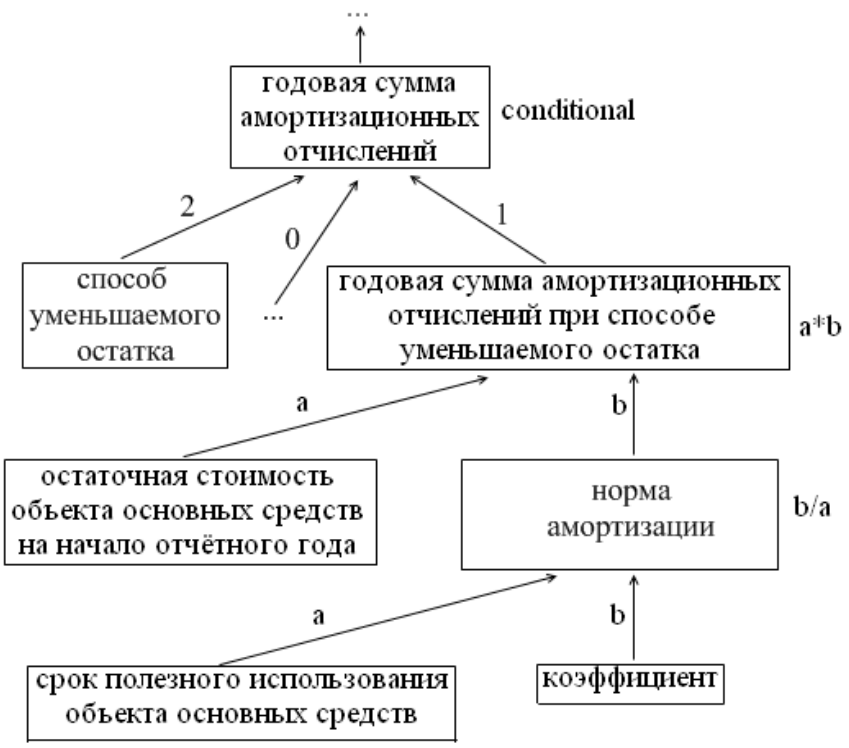


Рис. 5. Фрагмент модели закона.

8. Заключение

В работе описан метод, позволяющий по тексту нормативно-правового акта, касающегося бухгалтерского учета, автоматически строить схемы вычисления значений объектов, упомянутых в данном нормативно-правовом акте. В частности, описаны приемы синтаксического анализа; упрощения синтаксического графа; отождествления сущностей; представления упрощенного синтаксического графа в виде формулы; построения по всем формулам схем вычисления значений объектов. С помощью

данного метода созданы схемы вычисления значений объектов, описанных в положении по бухгалтерскому учету ПБУ 6/01.

Описанные в работе методы обработки текста, написанного на естественном языке, можно использовать при решении различных задач математической лингвистики.

Список литературы

[1] ERP и цифровое ядро. [https://www.sap.com/cis/products/erp.html]

[2] Microsoft Dynamics 365. [https://dynamics.microsoft.com]

[3] Национальный корпус русского языка. Синтаксически размеченный корпус русского языка: информация для пользователей. [http://www.ruscorpora.ru/instruction-syntax.html]

[4] Chen D., Manning C.D. A Fast and Accurate Dependency Parser using Neural Networks // Proceedings of EMNLP 2014. — 2014. — P. 740–750

[5] Сокирко А. Семантические словари в автоматической обработке текста (по материалам системы ДИАЛИНГ). [http://www.aot.ru/docs/sokirko]

[6] Проект Universal Dependencies. [http://universaldependencies.org]

[7] de Marneffe M., Dozat T., Silveira N., Haverinen K., Ginter F., Nivre J., Manning C.D. Universal Stanford Dependencies: A cross-linguistic typology // In Proc. of the 9th Conference on Language Resources and Evaluation (LREC). — 2014. — P. 4585–4592.

[8] Мельчук И. А. Опыт теории лингвистических моделей «Смысл ⇔ Текст». — М.: Школа «Языки русской культуры», 1999. — 368 с.

[9] Iordanskaja L., Kittredge R., Polguere A. Lexical Selection and Paraphrase in a Meaning-Text Generation Model // Natural Language Generation in Artificial Intelligence and Computational Linguistics. SECS. Boston, Springer, 1991. — **119**. — P. 293–312.

[10] Anisimovich K.V., Druzhkin K.Ju., Minlos F.R., Petrova M.A., Selegey V.P., Zuev K.A. Syntactic and semantic parser based on ABBYY Compreno linguistic technologies // Международная конференция по компьютерной лингвистике «Диалог-2012». — 2012. — **2**. — P. 91–103.

[11] Ben-Or M. Lower Bounds For Algebraic Computation Trees // Proc. 15th ACM Annu. Symp. Theory Comput. — 1983. — P. 80—86.

[12] Кудрявцев В.Б., Гасанов Э.Э., Перпер Е.М. Автоматическая генерация компьютерной программы, моделирующей нормативно-правовой акт // Интеллектуальные системы. Теория и приложения (ранее: Интеллектуальные системы по 2014, вып. 2, ISSN 2075-9460). — 2014. — **18**, №2. — С. 133–156.

[13] Перпер Е.М. О синтаксическом анализе юридических текстов // Интеллектуальные системы. Теория и приложения (ранее: Интеллектуальные системы по 2014, вып. 2, ISSN 2075-9460). — 2016. — **20**, №2. — С. 31–49.

[14] Подколзин А. С. Система автоматического решения задач по элементарной алгебре // Дискретная математика. — 1994. — **6**, №4. — С. 35-–57

[15] Приказ Министерства финансов Российской Федерации от 30.03.2001 N 26н «Об утверждении Положения по бухгалтерскому учету «Учет основных средств» ПБУ 6/01» [http://base.consultant.ru/cons/cgi/online.cgi?req=doc;base=LAW;n=111056]

On the semantic analysis of juridical documents

Perper E.M., Gasanov E.E., Kudryavtsev V.B.

The paper considers the task of designing a program which would perform semantic analysis of a juridical document in Russian language. For each entity described in a text, the program must construct a scheme which computes the value of this entity. The paper contains rules which allow to construct such schemes if morphological information about all words of the text is provided.

Keywords: semantic analysis, syntactic analysis, juridical document, logical formula, rule.