О поиске натурального классического логического вывода с использованием частичной скулемизации

Охотников О.А.

Рассматривается метод поиска вывода в односукцедентном секвенциальном варианте классического исчисления предикатов. В этом алгоритме используются метапеременные и частичная скулемизация. Для рассматриваемого алгоритма доказываются теоремы о корректности и полноте.

Ключевые слова: автоматическое доказательство теорем, поиск вывода, язык первого порядка, исчисление предикатов, исчисление секвенций, продукционная система, искусственный интеллект.

Введение

В работе [1] сформулирован алгоритм автоматического доказательства теорем, в котором впервые была использована частичная скулемизация для решения задачи поиска натурального вывода. Раньше скулемизация использовалась только в алгоритмах установления выводимости [2], но не поиска вывода. Практика использования этого алгоритма показала, что его эффективность не уступает существующим аналогам [3], [4]. После статьи [1] до сих пор не было публикаций, посвященных этому алгоритму. В данной работе решается задача теоретического обоснования этого алгоритма.

Для достижения этой цели мы формулируем указанный алгоритм в виде продукционной системы дедуктивных задач с частичной скулемизацией. Это означает, что указываются примитивные задачи и правила сведения задач к подзадачам. При этом процесс поиска решения дедуктивной задачи сопряжен с построением дерева поиска типа И/ИЛИ. Такая формулировка алгоритма нам представляется наиболее удобной для построения этой теории.

В данной работе доказываются теоремы о корректности и полноте формулируемой продукционной системы со скулемизацией. Эти две теоремы обеспечивают для всякой выводимой в исчислении секвенции возможность извлечения ее логического вывода из деривационного поддерева поискового дерева. Тем самым устанавливается взаимосвязь между поиском решения задач в продукционной системе со скулемизацией и поиском вывода в классическом односукцедентном исчислении секвенций [5]. Доказательство корректности содержит в себе эффективный алгоритм получения вывода в односукцедентном исчислении секвенций из деривационного поддерева поискового дерева. В то же время существует эффективный алгоритм получения натурального вывода из односукцедентного секвенциального вывода. Все это вместе дает точную формулировку и обоснование новой процедуры поиска натурального вывода [1].

Статья состоит из введения, пяти разделов и списка использованной литературы. В первом разделе определяются используемые в этой статье основные понятия и вводятся обозначения.

Во втором разделе описывается односукцедентное секвенциальное классическое исчисление предикатов, для которого в статье решается задача поиска вывода. Для этого формулируется сопряженная исчислению продукционная система с метапеременными. Здесь же формулируются теоремы о корректности и полноте этой продукционной системы по отношению к исчислению. Указанная продукционная система служит эталоном для установления корректности и полноты продукционной системы со скулемизацией.

В третьем разделе формулируется продукционная система со скулемизацией. Для этого для псевдоформул определяется скулемовская нормальная форма, которая используется в формулировке дедуктивных задач. Такая скулемовская нормальная форма позволяет эффективно решать задачу поиска логического вывода в классическом односукцедентном исчислении секвенций.

В четвертом разделе доказывается теорема о корректности продукционной системы со скулемизацией. Полнота этого метода поиска вывода устанавливается в теореме о полноте, которая доказывается в пятом разделе.

1. Основные понятия и обозначения

В краткой форме введем используемую в статье терминологию. Мы следуем определениям и обозначениям [5]. Кроме того, нам необходимо понятие продукционной системы. Продукционная система представляет собой способ решения задач в отдельной области путем указания примитивных задач и правил сведения задач к подзадачам [6]. Для этого фиксируется конечный алфавит A и разрешимое подмножество $B \subseteq A^*$ слов в этом алфавите, элементы которого кодируют задачи в рассматриваемой области. Среди задач выделяется разрешимое подмножество $C \subseteq B$ примитивных задач, решение которых известно. Кроме того, задается конечное количество правил сведения задач к подзадачам, которые представляют собой разрешимые отношения на B: $R_1 \subseteq B^{n_1}, \ldots, R_k \subseteq B^{n_k}$. В рамках продукционной системы процесс поиска решения задачи сопряжен с формированием дерева поиска типа И/ИЛИ. Ниже в этом разделе мы в краткой форме уточняем, каким образом формируется такое дерево поиска.

Поисковое дерево типа И/ИЛИ представляет собой дерево редукций исходной задачи, которая размещается в корне [6]. Между ребрами дерева существуют "И" и "ИЛИ" отношения. В общем случае из вершины могут выходить несколько ИЛИ-альтернативных И-связок ребер. Ребра из одной связки связываются И-отношением, а ребра из разных связок — ИЛИ-отношением. Вершина, из которой выходят только ребра, связанные И-отношением, называется И-вершиной. Вершина, из которой выходят только ребра, связанные ИЛИ-отношением, называется ИЛИ-вершиной. Всякое И/ИЛИ-дерево можно привести к виду, когда каждая вершина имеет дочерние вершины либо только типа "И", либо только типа "ИЛИ" [6]. Поддерево Θ поискового И/ИЛИ-дерева T называется деривационным, если Θ обладает следующими свойствами:

- исходная задача — это корень дерева Θ;
- если задача S является ИЛИ-вершиной, то в Θ содержится только одна из ее дочерних вершин из T вместе со своим собственным деривационным деревом;
- если задача S является И-вершиной, то все ее дочерние вершины из T вместе со своими деривационными деревьями содержатся в Θ.

Решением исходной задачи считается деривационное поддерево Θ поискового И/ИЛИ-дерева T. Дерево Θ выступает И-деревом, т. к. все его вершины являются вершинами типа И.

По сути продукционная система представляет собой формулировку некоторого алгоритма путем указания примитивных задач и правил декомпозиции задач с точностью до стратегии построения дерева поиска. Зафиксировав конкретную стратегию построения поискового дерева мы получим некоторую реализацию так сформулированного алгоритма. Стратегия построения дерева поиска заключается в методике выбора листа для формирования у него всех дочерних связок вершин. При этом родительская вершина и дочерние вершины из одной связки связаны одним из отношений декомпозиции R_i. Если у листа нельзя породить дочерние вершины, то ему приписывается оценка Yes или No: Yes, если лист представляет собой примитивную задачу из C; оценка No, если ни одно из правил декомпозиции к нему не применимо. Формирование оценки у листа приводит к пересчету всех оценок вершин на ветви, ведущей от листа к корню. При этом И-вершина получает оценку Yes, если все ее дочерние задачи уже имеют оценку Yes; оценка No формируется, если хотя бы одна дочерняя уже имеет оценку No. Если же задача является ИЛИ-вершиной, то она получает оценку Yes, когда хотя бы одна дочерняя уже имеет оценку Yes; оценка же No формируется, когда каждой дочерней приписана оценка No. В остальных случаях у родительской вершины не формируется оценка. Отсутствие оценки у задачи говорит о том, что возможность ее решения еще пока сохраняется. Процесс построения дерева поиска завершается как только корень получает оценку Yes или No. В первом случае для исходной задачи в дереве поиска существует конечное деривационное поддерево. Такая задача называется *разрешимой* в рамках продукционной системы.

В классе всех продукционных систем естественным образом выделяется собственный подкласс тех продукционных систем, которые сопряжены обычным исчислениям, предназначенным для вывода цепочек символов. В такой сопряженной исчислению $\mathcal{C}$ продукционной системе $\mathcal{P}_\mathcal{C}$ примитивными задачами считаются аксиомы исчисления $\mathcal{C}$, а способы сведения задач к подзадачам получаются в результате применения правил вывода исчисления $\mathcal{C}$ снизу вверх, т. е. от заключения к посылкам [7]. При этом далеко не всегда продукционная система сопряжена некоторому исчислению. Дело в том, что порядок формируемых подзадач в отдельной связке может играть принципиальную роль, в то время как порядок посылок в правиле вывода роли не играет.

Имея язык заданной сигнатуры Ω в области автоматического доказательства теорем рассматривают множество H_Ω всех термов, которые можно построить из констант и функциональных символов сигнатуры Ω. Множество H_Ω называют эрбрановским универсумом [2]. При этом термы эрбрановского универсума могут содержать предметные переменные, которые в таком контексте рассматриваются как константы.

С процессом построения дерева поиска $T_0 \subseteq T_1 \subseteq T_2 \ldots$ связан процесс расширения исходного языка. А именно, когда в процедуре построения дерева поиска у листа S_n порождаются дочерние вершины, происходит обогащение сигнатуры Ω_n языка. В результате формируется новая сигнатура Ω_{n+1}, получаемая из Ω_n добавлением новых свободных предметных переменных, входящих в Ω_{n+1} как константы. При этом в том случае, когда получаемая из исходной сигнатуры Ω сигнатура Ω_0 не содержит констант, мы искусственно добавляем в Ω_0 новую предметную переменную, играющую роль константы. В результате в общем случае в дереве поиска получаем бесконечную иерархию языков и универсумов

$$\begin{array}{rc} & T_0 \subseteq T_1 \subseteq \ldots \subseteq T_n \subseteq T_{n+1} \subseteq \ldots \\ \Omega \subseteq & \Omega_0 \subseteq \Omega_1 \subseteq \ldots \subseteq \Omega_n \subseteq \Omega_{n+1} \subseteq \ldots \\ & H_0 \subseteq H_1 \subseteq \ldots \subseteq H_n \subseteq H_{n+1} \subseteq \ldots \end{array}$$

Необходимо отметить, что сопряженные тому или иному варианту исчисления предикатов продукционные системы формируют бесконечно ветвящиеся деревья поиска типа И/ИЛИ. В прикладной программе такие деревья приводят к комбинаторному взрыву. Чтобы сделать деревья поиска конечно ветвящимися используются метапеременные [3]. При этом под метапеременной понимается синтаксическая переменная, принимающая в качестве своих значений термы языка первого порядка. Мы не будем формулировать метод метапеременных в общем виде для широкого класса продукционных систем. В построенной теории ниже формулируются две конкретные продукционные системы с метапеременными, которые решают тот же объем дедуктивных задач, что и сопряженная логико-математическому исчислению продукционная система.

Традиционно методы автоматического доказательства теорем подразделяются на две категории: это алгоритмы установления выводимости и алгоритмы поиска вывода. Например, метод резолюций [2] — это алгоритм установления выводимости, а процедура [3] — это алгоритм поиска вывода. Каждый алгоритм поиска вывода есть алгоритм установления выводимости, но не наоборот. Так резолюционный вывод нельзя полиномиальным способом преобразовать в натуральный вывод. Используемые в методе резолюций метапеременные можно переименовать в рамках отдельного дизъюнкта — это локальные метапеременные. В процедуре Правица [2] и процедуре [3] метапеременные глобальные. Их значения вычисляются в ходе вычислительного процесса.

Использование метапеременных приводит к изменению языка. Теперь выражения языка — это псевдоформулы и псевдотермы, которые могут быть обычными формулами и термами, но могут и содержать метапеременные, тем самым отличаясь от последних. Полученные в результате применения правила декомпозиции дочерние задачи, кроме глобальных метапеременных родительской задачи, могут содержать новые глобальные метапеременные, которым приписывается в качестве диапазона значений тот эрбрановский универсум, который построен к текущему моменту. В поисковом дереве с метапеременными у вершин вместо оценок формируются конечные множества допустимых подстановок, которые указывают на какие термы должны быть заменены глобальные метапеременные. Метод вычисления допустимых подстановок в разных продукционных системах с метапеременными определяется по разному. Мы в этой статье используем метод [3]. Точные определения связанных с этим методом понятий будут даны в разделе 2. Процесс построения дерева поиска завершается, когда набор допустимых подстановок у корня становится непустым. Соответствующее допустимой подстановке $\theta = \begin{pmatrix} X_1 & \dots & X_r \\ t_1 & \dots & t_r \end{pmatrix}$ деривационное поддерево Θ поискового дерева таково, что $\Theta\theta$ есть вывод в логико-математическом исчислении. При этом если диапазоном значений метапеременной X_i является H_n, то $t_i \in H_n$.

Использование целеориентированного поиска позволяет некоторым разумным образом оптимизировать порядок применения правил декомпозиции [5]. Тем не менее применение кванторных правил приводит к появлению катастрофически огромного числа вариантов при переборе всех возможных альтернатив [8]. По нашему мнению дальнейшей оптимизации такого перебора можно достичь за счет использования скулемовских нормальных форм определенного вида. Предлагаемая в статье скулемовская нормальная форма позволяет элиминировать некоторые кванторы и тем самым сделать поиск более эффективным. При этом используется такая форма частичной скулемизации, которая позволяет не терять связь с искомым выводом в односукцедентном исчислении секвенций.

2. Односукцедентное исчисление секвенций

В этом разделе мы, следуя [5], опишем односукцедентное секвенциальное классическое исчисление предикатов, для которого в статье решается задача поиска вывода. Для этого здесь же будет определена сопряженная этому исчислению продукционная система. Исчисление секвенций выгодно отличается от системы натуральной дедукции [1] с точки зрения формулировки методов поиска вывода и их обоснования. При этом существует эффективный алгоритм получения натурального вывода из односукцедентного секвенциального вывода.

Основные определения и обозначения, связанные с исчислением секвенций, можно найти в статье [5]. В этой статье определены три односукцедентных секвенциальных исчисления предикатов: минимальное $\mathcal{S}_1$, интуиционистское $\mathcal{S}_2$ и классическое $\mathcal{S}_3$. Необходимо обратить внимание на то, что в статье [5] при определении правил вывода ($\supset\rightarrow$), ($\& \rightarrow$), ($\vee \rightarrow$), ($\forall \rightarrow$) и ($\exists \rightarrow$) допущена досадная оплошность. В этих правилах формула G может быть не только атомом, дизъюнкцией или $\perp$, но также формулой вида $\exists x\, C(x)$. Кроме этого, мы здесь используем другую, но эквивалентную, формулировку правила вывода ($\supset\rightarrow$), которая оказалась более удобной для изложения этой теории.

Для формулировки исчисления секвенций $\mathcal{S}_3$ прежде всего мы должны фиксировать некоторый логико-математический язык Ω, формулы которого выражают суждения и отношения рассматриваемой области математики. Мы предполагаем, что сложные формулы языка Ω строятся из атомарных формул этого языка при помощи логических связок $\&$, $\vee$, $\supset$, логических констант $\perp$, $\top$ и кванторов $\forall$, $\exists$. Отрицание $\neg A$ есть по определению формула $A \supset \perp$. Вхождения предметных переменных в формулу обычным образом делятся на свободные и связанные. Переменные, входящие в формулу свободно, называются ее параметрами. Через $A(x/t)$ обозначается результат подстановки терма t вместо всех свободных вхождений переменной x в формулу A. При этом предполагается, что производится переименование связанных переменных формулы A, если параметры t попадают в область действия кванторов A. Вместо $A(x/t)$ мы будем писать просто $A(t)$, когда контекст x очевиден.

Секвенцией [5] называется упорядоченная пара вида $\Gamma \rightarrow A$, где Γ есть конечное, в частности быть может пустое, множество формул языка Ω, а A есть формула этого языка. При этом Γ называется антецедентом, а формула A — сукцедентом секвенции. Теоретико-множественные операции над конечными множествами формул рассматриваются обычным образом. Например, объединение $\Gamma \cup \Delta$ конечных множеств формул Γ и Δ состоит из всех формул из Γ и Δ. Как обычно, мы пишем $\Gamma\Delta$ вместо $\Gamma \cup \Delta$. Так что $\Gamma\Delta$ и $\Delta\Gamma$ обозначают одно и то же.

Исчисление $\mathcal{S}_3$ содержит аксиомы следующих двух видов: $\Gamma C \to C$ и $\Delta \to \top$, где C — атомарная формула языка Ω или $\bot$. Первые десять правил вывода делятся на две группы и вводят логические связки слева и справа. Последнее правило вывода называется правилом противоречия. Правила вывода изображаются в виде двухэтажной конструкции. Над чертой пишутся секвенции, являющиеся посылками правила вывода, которые отделяются точкой с запятой. Под чертой пишется его заключение, которое является выводимой секвенцией, если посылки выводимы. Рядом с чертой мы указываем символическое обозначение правила:

$$\frac{(A \supset B)\,\Gamma \to A; \quad B\,(A \supset B)\,\Gamma \to G}{(A \supset B)\,\Gamma \to G}\ (\supset\to), \qquad \frac{\Gamma\, A \to B}{\Gamma \to A \supset B}\ (\to\supset),$$

$$\frac{A\,B\,(A\,\&\,B)\,\Gamma \to G}{(A\,\&\,B)\,\Gamma \to G}\ (\&\to), \qquad \frac{\Gamma \to A; \quad \Gamma \to B}{\Gamma \to A\,\&\,B}\ (\to\&),$$

$$\frac{A\,(A \vee B)\,\Gamma \to G; \quad B\,(A \vee B)\,\Gamma \to G}{(A \vee B)\,\Gamma \to G}\ (\vee\to), \qquad \frac{\Gamma \to A_i}{\Gamma \to A_1 \vee A_2}\ (\to\vee_i),$$

$$\frac{A(x/t)\,(\forall x\, A(x))\,\Gamma \to G}{(\forall x\, A(x))\,\Gamma \to G}\ (\forall\to), \qquad \frac{\Gamma \to A(x/y)}{\Gamma \to \forall x\, A(x)}\ (\to\forall),$$

$$\frac{A(x/y)\,(\exists x\, A(x))\,\Gamma \to G}{(\exists x\, A(x))\,\Gamma \to G}\ (\exists\to), \qquad \frac{\Gamma \to A(x/t)}{\Gamma \to \exists x\, A(x)}\ (\to\exists),$$

$$\frac{(\neg G)\,\Gamma \to \bot}{\Gamma \to G}\ (\bot_{\mathbf{c}}).$$

Здесь в правилах G — атом, дизъюнкция, $\bot$ или формула вида $\exists z\, C(z)$ языка Ω, A, B — любые формулы языка Ω, а Γ — любое конечное множество формул языка Ω. В кванторных правилах терм t и переменная y допустимы для x в $A(x)$. В кванторных правилах $(\to\forall)$ и $(\exists\to)$ переменная y не входит свободно в нижнюю секвенцию. Описанные условия называются ограничениями кванторных правил.

Выводом секвенции S в исчислении $\mathcal{S}_3$ называется конечная последовательность секвенций $S_1, \ldots, S_k$, в которой $S = S_k$ и каждая S_j либо является аксиомой исчисления $\mathcal{S}_3$, либо получается в результате применения некоторого правила вывода исчисления $\mathcal{S}_3$ из некоторых предшествующих секвенций в этой последовательности. Вывод в $\mathcal{S}_3$ удобно оформлять в виде дерева, у которого к каждому узлу приписана секвенция. При этом аксиомы приписаны к листьям этого дерева, а отношение между родительским узлом и дочерними узлами соответствует отношению между секвенцией-заключением и секвенциями-посылками в результате применения некоторого правила вывода. Высота вывода есть по определению количество секвенций в самой длинной ветви. Мы не будем делать различия между формулами и секвенциями, отличающимися лишь переименованием связанных переменных. Таким образом, если имеется вывод с нижней секвенцией S, то этот же вывод считается автоматически и выводом любой секвенции, полученной из S переименованием связанных переменых.

После сделанных определений теперь точным образом опишем сопряженную исчислению $\mathcal{S}_3$ продукционную систему $\mathcal{P}_3$ с метапеременными. В ней выражениями языка являются псевдотермы и псевдоформулы. *Псевдотермы* какого-либо языка Ω определяются индуктивно. Прежде всего, предметные константы языка Ω, предметные переменные и метапеременные считаются псевдотермами. Если f есть n-местный функциональный символ языка Ω и $t_1, \ldots, t_n$ суть псевдотермы языка Ω, то цепочка вида $f(t_1, \ldots, t_n)$ есть псевдотерм. Аналогичное определение *псевдоформул* языка Ω выглядит следующим образом. Прежде всего, $\perp$ и $\top$ суть псевдоформулы. Если P есть n-местный предикатный символ и $t_1, \ldots, t_n$ суть псевдотермы языка Ω, то цепочка $P(t_1, \ldots, t_n)$ есть псевдоформула. Если A и B суть псевдоформулы, то $(A \,\&\, B)$, $(A \vee B)$, $(A \supset B)$ суть псевдоформулы. Если A есть псевдоформула и x есть предметная переменная, то $(\forall x\, A)$ и $(\exists x\, A)$ суть псевдоформулы.

Задачами этой системы являются упорядоченные пары вида (Γ, A), где Γ есть конечное множество псевдоформул и A есть псевдоформула. Примитивными задачами являются задачи следующих двух видов: $(E\Gamma, E)$, $(\Delta, \top)$, где E — атом или $\perp$. Содержащая метапеременные X_1, $\ldots$, X_k дедуктивная задача π *примитивизируема*, если существует корректная подстановка $\theta = \begin{pmatrix} X_1 & \ldots & X_k \\ t_1 & \ldots & t_k \end{pmatrix}$ вместо метапеременных X_1, $\ldots$, X_k такая, что дедуктивная задача $\pi\theta$ является примитивной. При этом *корректность* θ означает, что если диапазоном значений метапеременной X_i является H_n, то t_i есть псевдотерм языка Ω_n. Корректная подстановка θ называется *основной*, если $t_i \in H_n$.

Для унифицируемых выражений E_1 и E_2 через $\mathbf{MGU}(E_1, E_2)$ ниже обозначается их наиболее общий унификатор [3], т. е. это такой унификатор θ выражений E_1 и E_2, что для каждого их унификатора λ выполняется $\lambda \leq \theta$. При этом $\lambda \leq \theta$ означает существование подстановки μ

такой, что $\lambda = \theta \circ \mu$. Через $\theta \circ \mu$ обозначается композиция подстановок [2].

Для вычисления допустимых подстановок у задач, имеющих в дереве поиска дочерние задачи, требуется понятие комбинации подстановок [2], которое определяется следующим образом. Пусть

$$\theta_1 = \begin{pmatrix} X_{11}, & \dots & X_{1n_1} \\ t_{11}, & \dots & t_{1n_1} \end{pmatrix},$$
$$\vdots$$
$$\theta_r = \begin{pmatrix} X_{r1}, & \dots & X_{rn_r} \\ t_{r1}, & \dots & t_{rn_r} \end{pmatrix}$$

суть подстановки ($r \geq 2$). Исходя из $\theta_1, \dots, \theta_r$, сформируем два выражения

$$E_1 = (X_{11}, \dots, X_{1n_1}, \dots, X_{r1}, \dots, X_{rn_r}),$$

$$E_2 = (t_{11}, \dots, t_{1n_1}, \dots, t_{r1}, \dots, t_{rn_r}).$$

Тогда подстановки $\theta_1, \dots, \theta_r$ называются *совместимыми*, если выражения E_1 и E_2 унифицируемы. При этом подстановка $\mathbf{MGU}(E_1, E_2)$ называется *комбинацией* подстановок $\theta_1, \dots, \theta_r$.

Формулируемые ниже правила 1 – 12 определяют правила декомпозиции $\mathcal{P}_3$.

1. Пусть E и E' — унифицируемые атомы или $\perp$. Пусть подстановка θ есть $\mathbf{MGU}(E', E)$. Тогда считается разрешимой задача $(E\,\Gamma,\, E')$ и подстановка θ считается ее допустимой подстановкой.

2. Пусть псевдоформула C есть атом, дизъюнкция, $\perp$ или имеет вид $\exists x\, D(x)$. Тогда если $((A \supset B)\Gamma,\, A)$ и $(B\,(A \supset B)\,\Gamma,\, C)$ суть разрешимые задачи, у которых существуют совместимые допустимые подстановки θ_1 и θ_2, то также считается разрешимой задача $((A \supset B)\Gamma,\, C)$ и ее допустимой подстановкой считается комбинация θ_1 и θ_2.

3. Если разрешима задача $(A\,\Gamma,\, B)$ и θ — ее допустимая подстановка, то считается разрешимой задача $(\Gamma,\, A \supset B)$ и θ считается ее допустимой подстановкой.

4. Пусть псевдоформула C есть атом, дизъюнкция, $\perp$ или имеет вид $\exists x\, D(x)$. Тогда если разрешима задача $(AB(A \,\&\, B)\Gamma,\, C)$ и θ — ее допустимая подстановка, то задача $((A \,\&\, B)\Gamma,\, C)$ также считается разрешимой и θ считается ее допустимой подстановкой.

5. Если $(\Gamma,\, A)$ и $(\Gamma,\, B)$ суть две разрешимые задачи, у которых существуют совместимые допустимые подстановки θ_1 и θ_2, то тогда разреши-

ма задача $(\Gamma,\ A\&B)$ и ее допустимой подстановкой является комбинация θ_1 и θ_2.

6. Пусть псевдоформула C есть атом, дизъюнкция, $\perp$ или имеет вид $\exists x\, D(x)$. Тогда если $(A\,(A\vee B)\Gamma,\ C)$ и $(B\,(A\vee B)\Gamma,\ C)$ суть две разрешимые дочерние задачи, у которых существуют совместимые допустимые подстановки θ_1 и θ_2, то разрешима также задача $((A\vee B)\Gamma,\ C)$ и ее допустимой подстановкой считается комбинация θ_1 и θ_2.

7. Если для $i=1$ или $i=2$ разрешима задача $(\Gamma,\ A_i)$ и θ — ее допустимая подстановка, то задача $(\Gamma,\ A_1\vee A_2)$ также считается разрешимой и θ считается ее допустимой подстановкой.

8. Пусть псевдоформула C есть атом, дизъюнкция, $\perp$ или имеет вид $\exists x\, D(x)$. Тогда задача $((\forall x\, A(x))\,\Gamma,\ C)$ может быть сведена к новой задаче $(A(X)\,(\forall x\, A(x))\,\Gamma,\ C)$, где X — новая глобальная метапеременная в поисковом дереве. При этом если дочерняя задача $(A(X)\,(\forall x\, A(x))\,\Gamma,\ C)$ разрешима и θ — ее допустимая подстановка, то считается разрешимой родительская задача $((\forall x\, A(x))\,\Gamma,\ C)$ и θ считается ее допустимой подстановкой.

9. Если разрешима задача $(\Gamma,\ A(y))$, где псевдоформулы из Γ не содержат свободно переменной y, а θ — ее допустимая подстановка, то разрешима задача $(\Gamma,\ \forall x\, A(x))$ и θ — ее допустимая подстановка.

10. Пусть псевдоформула C есть атом, дизъюнкция, $\perp$ или имеет вид $\exists x\, D(x)$. Тогда если дочерняя задача $(A(y)\,(\exists x\, A(x))\,\Gamma,\ C)$ разрешима, где псевдоформулы из Γ не содержат свободно переменной y, а θ — ее допустимая подстановка, то задача $((\exists x\, A(x))\,\Gamma,\ C)$ разрешима и θ — ее допустимая подстановка.

11. Задача $(\Gamma,\ \exists x\, A(x))$ может быть сведена к задаче $(\Gamma,\ A(X))$, где X — новая глобальная метапеременная в поисковом дереве. При этом если дочерняя задача $(\Gamma,\ A(X))$ разрешима и θ — ее допустимая подстановка, то считается разрешимой родительская задача $(\Gamma,\ \exists x\, A(x))$ и θ считается ее допустимой подстановкой.

12. Пусть псевдоформула C есть атом, дизъюнкция или имеет вид $\exists x\, D(x)$. Тогда если разрешима задача $((\neg C)\,\Gamma,\ \perp)$ и θ — ее допустимая подстановка, то задача $(\Gamma,\ C)$ также считается разрешимой и θ считается ее допустимой подстановкой.

Формулировка продукционной системы $\mathcal{P}_3$ завершена. Продукционная система $\mathcal{P}_3$ задает алгоритм поиска решения дедуктивной задачи не уточняя стратегию построения дерева поиска типа И/ИЛИ. Зафиксировав конкретную стратегию мы получим некоторую реализацию так сформулированного алгоритма. Стратегия построения дерева поиска заключается в методике выбора листа для формирования у него, во-первых, множества всех примитивизирующих этот лист подстановок, во-вторых, всех дочерних связок вершин, соответствующих правилам декомпозиции. Затем каждая примитивизирующая подстановка “поднимается” по ветви, соединяющей лист с корнем, с помощью операции комбинации [3]. Тем самым в ходе процесса построения дерева поиска у его вершин формируются наборы допустимых подстановок. Указанная продукционная система служит эталоном для установления корректности и полноты продукционной системы со скулемизацией, которая будет сформулирована в следующем разделе. В качестве символических названий сформулированных правил декомпозиции 2 – 12 будем соответственно использовать следующие обозначения: $(\supset\to)$, $(\to\supset)$, $(\&\to)$, $(\to\&)$, $(\vee\to)$, $(\to\vee)$, $(\forall\to)$, $(\to\forall)$, $(\exists\to)$, $(\to\exists)$, $(\perp_{\mathbf{c}})$.

Заметим, что если θ является допустимой подстановкой разрешимой задачи, то ее допустимой по определению считается также всякая подстановка $\lambda \leq \theta$. Если π есть разрешимая задача и θ — ее допустимая подстановка, то индукцией по высоте деривационного дерева устанавливается, что всякая подстановка $\lambda \leq \theta$ является допустимой подстановкой всех потомков этой задачи π в деривационном дереве. Это позволяет по любой допустимой подстановке θ однозначно находить соответствующее деривационное поддерево Θ, которое считается решением задачи. Такое поддерево Θ получается в результате рекурсивного перебора всех альтернативных связок дочерних подзадач и выбора на каждом шаге той связки из них, которая содержит в качестве комбинации подстановку θ.

Деривационное дерево Θ позволяет решать задачу поиска вывода в следующем смысле. Если деривационное поддерево Θ в дереве поиска соответствует допустимой подстановке θ, то индукцией по высоте Θ устанавливается, что $\theta \geq \lambda$ для любой основной подстановки λ такой, что $\Theta\lambda$ есть вывод в исчислении секвенций $\mathcal{S}_3$ [3]. Следствием последнего утверждения является теорема о корректности и полноте для сформулированной продукционной системы $\mathcal{P}_3$ по отношению к исчислению $\mathcal{S}_3$, т. е. секвенция $\Gamma \to A$ выводима в $\mathcal{S}_3$ тогда и только тогда, когда задача $(\Gamma,\ A)$ разрешима в $\mathcal{P}_3$. Указанная система декомпозиции дедуктивных задач часто ставится во главу угла при разработке компьютерных систем автоматизации дедукции [3], [8], [9]. Недостатки $\mathcal{P}_3$ были отмечены во введении и первом разделе.

3. Продукционная система дедуктивных задач со скулемизацией

При решении задач установления выводимости или поиска вывода с целью повышения эффективности этого процесса часто применяют те или иные инвариантные преобразования формул. Преобразование, которое здесь рассматривается, называется *скулемизацией* [2], [10]. Скулемизация позволяет упростить алгоритмическую часть процедуры установления выводимости. Имея в виду контекст метода метапеременных, мы будем заниматься преобразованием псевдоформул. При этом мы будем использовать такой новый вариант частичной скулемизации, который позволяет не терять связь с искомым односукцедентным секвенциальным выводом, и тем самым с натуральным выводом. Это позволяет использовать скулемизацию в рамках процедуры поиска натурального вывода [1].

Положительные и *отрицательные вхождения* псевдоформулы в псевдоформулу определим индукцией по числу логических связок. Вхождение псевдоформулы в себя считается положительным. Если псевдоформула B положительно (отрицательно) входит в псевдоформулу A, то B положительно (отрицательно) входит в псевдоформулы вида: $A \mathbin{\&} A'$, $A' \mathbin{\&} A$, $\forall x\, A$, $A \vee A'$, $A' \vee A$, $\exists x\, A$, $A' \supset A$; и отрицательно (положительно) входит в псевдоформулы вида: $A \supset A'$.

Вхождение псевдоформулы C в псевдоформулу D называется *строго положительным*, если оно является положительным и не находится в антецеденте некоторой импликации в D. Вхождение C в D будем называть *строго положительным аналитическим*, если оно является строго положительным и не находится в области действия некоторой дизъюнкции в D.

Под *знаком вхождения логической связки* в указанную псевдоформулу будем понимать знак того вхождения псевдоформулы в указанную псевдоформулу, главной связкой которого оно является. Под *знаком вхождения предикатного символа* будем понимать знак соответствующего вхождения атома. Будем говорить, что псевдоформула A обладает *свойством чистоты переменных*, если, во-первых, все ее связанные переменные отличны от свободных и, во-вторых, любые два различные вхождения кванторных приставок связывают различные переменные.

Пусть A — произвольная псевдоформула со свойством чистоты переменных и $\exists x$ — некоторая строго положительная аналитическая экзистенциальная приставка в A. *Размерностью* переменной x в A будем называть число различных кванторов всеобщности, управляющих переменной x. Пусть k — размерность переменной x в A и f — k-местный функциональный символ, не встречающийся в A. Список всех связанных кванторами всеобщности переменных в порядке вхождения в A, управляющих переменной x в A, запишем в виде $z_1, \ldots, z_k$. *Результатом применения* к паре $\langle A, x\rangle$ *скулемовского метода элиминации квантора существования* будем называть псевдоформулу $A^- \begin{pmatrix} x \\ f(z_1, \ldots, z_k) \end{pmatrix}$, где A^- обозначает результат вычеркивания из A единственного вхождения кванторной приставки $\exists x$. Эту псевдоформулу будем обозначать $\mathbf{S}(A, x)$.

Скулемовской нормальной формой псевдоформулы A будем называть псевдоформулу, обозначаемую $\mathbf{Sk}(A)$, которая определяется индукцией по числу логических связок:

- $\mathbf{Sk}(B\&C) = \mathbf{Sk}(B) \,\&\, \mathbf{Sk}(C)$, $\mathbf{Sk}(B \supset C) = B \supset \mathbf{Sk}(C)$, $\mathbf{Sk}(B \vee C) = B \vee C$, $\mathbf{Sk}(\top) = \top$, $\mathbf{Sk}(\bot) = \bot$.

- Если A имеет вид $\exists x\, B$, то $\mathbf{Sk}(A) = B \begin{pmatrix} x \\ c \end{pmatrix}$.

- Если A имеет вид $\forall z\, B$, то процесс вычисления $\mathbf{Sk}(A)$ разбивается на два этапа. На первом этапе после элиминации строго положительных аналитических кванторов существования и введения скулемовских функций получаем псевдоформулу

$$A' = \mathbf{S}(\ldots(\mathbf{S}(\mathbf{S}(A, x_1), x_2), \ldots), x_l).$$

  На втором этапе после вычеркивания из A' строго положительных аналитических кванторов всеобщности получаем A''. Наконец, после введения новых локальных метапеременных $Z_1, \ldots, Z_m$ получаем искомый результат

$$\mathbf{Sk}(A) = A'' \begin{pmatrix} z_1 & \ldots & z_m \\ Z_1 & \ldots & Z_m \end{pmatrix}.$$

Например, скулемовской нормальной формой формулы

$$\forall x \forall y\, ((\exists z\, (P(x, z) \,\&\, P(z, y))) \supset \exists u\, Q(x, u, y))$$

является псевдоформула

$$(\exists z\, (P(X, z) \,\&\, P(z, Y))) \supset Q(X, f(X, Y), Y),$$

где f — новая скулемовская функция, а X и Y — новые локальные метапеременные.

Пусть D есть строго положительная аналитическая подпсевдоформула псевдоформулы C и D есть атом, дизъюнкция, $\bot$ или $\top$. Пусть D находится в области действия последовательности кванторов

$$\forall z_1 \ldots \forall z_{n_1} \exists v_1 \forall z_{n_1+1} \ldots \forall z_{n_2} \exists v_2 \forall z_{n_2+1} \ldots \forall z_{n_k} \exists v_k \forall z_{n_k+1} \ldots \forall z_{n_k+m}. \tag{1}$$

Пусть в процессе скулемизации при элиминации кванторов $\exists v_1$, …, $\exists v_k$ вводятся новые скулемовские функции $\varphi_1^{n_1}$, …, $\varphi_k^{n_k}$, а при элиминации кванторов $\forall z_1$, …, $\forall z_{n_k+m}$ вводятся локальные метапеременные Z_1, …, Z_{n_k+m}. Тогда *протоколом скулемизации C относительно D* назовем последовательность псевдоформул P_1, …, P_q, которая формируется из частей псевдоформулы C и моделирует процесс скулемизации. Точнее говоря, для псевдоформул P_1, …, P_q имеем $P_1 = C$, $P_q = D\begin{pmatrix} z_1 & \ldots & z_{n_k+m} & v_1 & \ldots & v_k \\ Z_1 & \ldots & Z_{n_k+m} & \varphi_1^{n_1}(Z_1, \ldots, Z_{n_1}) & \ldots & \varphi_k^{n_k}(Z_1, \ldots, Z_{n_k}) \end{pmatrix}$ и

- если $P_j = \forall z_i\, B(z_i)$, то $P_{j+1} = B(Z_i)$;
- если $P_j = \exists v_i\, B(v_i)$, то $P_{j+1} = B(\varphi_i^{n_i}(Z_1, \ldots, Z_{n_i}))$;
- если $P_j = B_1 \supset B_2$, то $P_{j+1} = B_2$;
- если $P_j = B_1 \,\&\, B_2$ и D входит в B_i, то $P_{j+1} = B_i$.

Отношением подобия будем называть рефлексивно транзитивное замыкание отношения между псевдоформулами, когда одна из них получается из другой в результате замены подпсевдоформулы вида $(M \,\&\, N)$ на $(N \,\&\, M)$.

С каждым строго положительным аналитическим вхождением D в C связана подобная C псевдоформула вида

$$\begin{array}{l} Q_{11}x_{11} \ldots Q_{1r_1}x_{1r_1}(C_1\, \lambda_1 \\ \quad Q_{21}x_{21} \ldots Q_{2r_2}x_{2r_2}(C_2\, \lambda_2\, (\ldots \\ \quad\quad \ldots \\ \quad\quad Q_{s1}x_{s1} \ldots Q_{sr_s}x_{sr_s}(C_s\, \lambda_s\, Q_{s+1,1}x_{s+1,1} \ldots Q_{s+1,r_{s+1}}x_{s+1,r_{s+1}}\, D)) \ldots), \end{array} \tag{2}$$

у которой цепочка кванторов

$$Q_{11}x_{11}\, \ldots\, Q_{sr_s}x_{sr_s}\, Q_{s+1,1}x_{s+1,1}\, \ldots\, Q_{s+1,r_{s+1}}x_{s+1,r_{s+1}} \tag{3}$$

совпадает с (1), $\lambda_i \in \{\&, \supset\}$ и протокол скулемизации относительно D совпадает с протоколом скулемизации C относительно D. Будем называть (2) *нормальной формой C относительно D*.

Пусть D есть строго положительная аналитическая подпсевдоформула псевдоформулы C и D есть атом, дизъюнкция, $\bot$ или $\top$. Тогда *пренексной формой* C *относительно* D назовем псевдоформулу, получаемую из C в результате вынесения в начало псевдоформулы всех кванторов, которые управляют D в C.

Очевидно, что если нормальная форма C относительно D имеет вид (2) и C' есть пренексная форма C относительно D, то нормальная форма C' относительно D имеет вид

$$Q_{11}x_{11}\,\ldots\,Q_{s+1,r_{s+1}}x_{s+1,r_{s+1}}\,(C_1\,\lambda_1\,(C_2\,\lambda_2\,(\ldots(C_s\,\lambda_s\,D))\ldots). \tag{4}$$

Псевдоформулу (4) будем называть *пренексной нормальной формой* C *относительно* D. Эту псевдоформулу будем обозначать $\mathbf{PNF}(C,\,D)$.

Пусть $F=\mathbf{Sk}(C)$ и нормальная форма C отностительно D имеет вид (2). Тогда результат скулемизации псевдоформулы (2) имеет вид

$$F_1\,\lambda_1\,(F_2\,\lambda_2\,(\ldots(F_s\,\lambda_s\,G))\ldots), \tag{5}$$

где $G=D\zeta$, $F_i=\tilde{C}_i\zeta$,

$$\zeta=\left(\begin{array}{ccccccc} z_1 & \ldots & z_{n_k+m} & v_1 & \ldots & v_k \\ Z_1 & \ldots & Z_{n_k+m} & \varphi_1^{n_1}(Z_1,\,\ldots,\,Z_{n_1}) & \ldots & \varphi_k^{n_k}(Z_1,\,\ldots,\,Z_{n_k}) \end{array}\right),$$

$$\tilde{C}_i=\begin{cases} C_i, & \text{если } \lambda_i=\supset; \\ \mathbf{Sk}(C_i), & \text{если } \lambda_i=\&. \end{cases}$$

Будем называть (5) *нормальной формой* F *относительно* G. Псевдоформулу (5) будем обозначать $\mathbf{NF}(F,\,G)$. *Импликантой* нормальной формы F относительно G назовем кратную конъюнкцию

$$F_{i_1}\,\&\,F_{i_2}\,\&\,\ldots\,\&\,F_{i_n},$$

где $\lambda_{i_1}=\lambda_{i_2}=\ldots=\lambda_{i_n}=$ “$\supset$” — все импликации среди $\lambda_i\in\{\&,\,\supset\}$.

Дедуктивной задачей назовем пару вида $(\mathbf{Sk}(\Gamma)\,?\,G)$, где Γ — конечное множество псевдоформул, которое называется *антецедентом*, а G — псевдоформула, которая называется *сукцедентом* этой дедуктивной задачи. При этом псевдоформулы из Γ и псевдоформула G не содержат локальных метапеременных, но, возможно, содержат глобальные метапеременные.

Сформулируем теперь продукционную систему $\mathcal{F}_3$ с метапеременными и со скулемизацией, в рамках которой будут решаться дедуктивные задачи. *Примитивные дедуктивные задачи* этой системы суть произвольные задачи двух видов

$$\Gamma \mathbin{?} \top \quad \text{и} \quad \left(\&_{i=1}^{i=m} A_i\right) \Gamma \mathbin{?} G,$$

где G совпадает с некоторой A_i и при этом G — атом или $\perp$.

Чтобы решать дедуктивные задачи теперь определим *правила сведения задач к подзадачам* продукционной системы $\mathcal{F}_3$. Правила построены весьма естественно и удаляют логические связки слева и справа. Правила декомпозиции изображаются в виде двухэтажной конструкции. Под чертой мы пишем родительскую задачу. Над чертой пишем ее дочерние подзадачи. Рядом с чертой мы указываем символическое обозначение правила.

1. Если $\Gamma \mathbin{?} A_1$ и $\Gamma \mathbin{?} A_2$ суть две разрешимые задачи, у которых существуют совместимые допустимые подстановки θ_1 и θ_2, то тогда разрешима задача $\Gamma \mathbin{?} A_1 \mathbin{\&} A_2$ и комбинация подстановок θ_1 и θ_2 считается ее допустимой подстановкой. Символически сформулированное правило декомпозиции записывается в виде

$$\frac{\Gamma \mathbin{?} A_1; \quad \Gamma \mathbin{?} A_2}{\Gamma \mathbin{?} A_1 \mathbin{\&} A_2}\ (?\&)$$

и понимается так: задача вида $\Gamma \mathbin{?} A_1 \mathbin{\&} A_2$ сводится к двум задачам $\Gamma \mathbin{?} A_1$ и $\Gamma \mathbin{?} A_2$.

2. Если разрешима задача $\mathbf{Sk}(A)\, \Gamma \mathbin{?} B$ и θ — ее допустимая подстановка, то считается разрешимой задача $\Gamma \mathbin{?} A \supset B$ и θ считается ее допустимой подстановкой. То есть, задача $\Gamma \mathbin{?} A \supset B$ сводится к новой задаче $\mathbf{Sk}(A)\, \Gamma \mathbin{?} B$:

$$\frac{\mathbf{Sk}(A)\, \Gamma \mathbin{?} B}{\Gamma \mathbin{?} A \supset B}\ (?\supset).$$

3. Если разрешима хотя бы одна из задач $\Gamma \mathbin{?} A_i$, где $i = 1$ или $i = 2$, а θ — ее допустимая подстановка, то считается разрешимой задача $\Gamma \mathbin{?} A_1 \vee A_2$ и θ считается ее допустимой подстановкой. То есть, задача $\Gamma \mathbin{?} A_1 \vee A_2$ может быть сведена к подзадаче $\Gamma \mathbin{?} A_i$ (для $i = 1$ или 2):

$$\frac{\Gamma \mathbin{?} A_i}{\Gamma \mathbin{?} A_1 \vee A_2}\ (?\vee_i).$$

4. Если разрешима задача $\Gamma \mathbin{?} A(y)$, где псевдоформулы из Γ не содержат свободно переменной y, а θ — ее допустимая подстановка, то считается разрешимой задача $\Gamma \mathbin{?} \forall x\, A(x)$ и θ считается ее допустимой подстановкой. Кратко говоря, задача $\Gamma \mathbin{?} \forall x\, A(x)$ сводится подзадаче $\Gamma \mathbin{?} A(y)$,

где y — новая свободная предметная переменная в поисковом дереве:

$$\frac{\Gamma\,?\,A(y)}{\Gamma\,?\,\forall x\,A(x)}\ (?\forall).$$

5. Если разрешима задача $\Gamma\,?\,A(X)$, где X — новая глобальная метапеременная в поисковом дереве, а θ — ее допустимая подстановка, то считается разрешимой задача $\Gamma\,?\,\exists x\,A(x)$ и θ считается ее допустимой подстановкой. То есть, задача $\Gamma\,?\,\exists x\,A(x)$ может быть сведена к дочерней задаче $\Gamma\,?\,A(X)$:

$$\frac{\Gamma\,?\,A(X)}{\Gamma\,?\,\exists x\,A(x)}\ (?\exists).$$

6. Пусть G и G' — унифицируемые атомы или $\perp$, $\mathbf{NF}(A,\,G')$ имеет вид $\&_{i=1}^{i=m}A_i$ и $A_m = G'$. Пусть подстановка η замещает все локальные метапеременные в A новыми глобальными и $\theta = \mathbf{MGU}(G,\,G'\eta)$. Тогда считается разрешимой задача $A\,\Gamma\,?\,G$ и подстановка θ считается ее допустимой подстановкой. Обозначив через B_i псевдоформулу $\&_{j=i+1}^{j=m}A_j$ мы можем записать указанное правило в виде

$$\frac{(\&_{i=1}^{i=m}A_i)\eta\,(A_1\eta)\,(B_1\eta)\,\ldots\,(A_{m-1}\eta)\,(B_{m-1}\eta)\,A\,\Gamma\,?\,G}{A\,\Gamma\,?\,G}\ (\&?).$$

7. Пусть G и G' — унифицируемые атомы или $\perp$ и G' есть строго позитивная аналитическая подпсевдоформула псевдоформулы A. Пусть подстановка η замещает все локальные метапеременные в A новыми глобальными, $\sigma = \mathbf{MGU}(G,\,G'\eta)$ и $\xi = \eta \circ \sigma$. Пусть нормальная форма A относительно G' имеет вид

$$A_1\,\lambda_1\,(A_2\,\lambda_2\,(\ldots(A_n\,\lambda_n\,G'))\ldots),$$

где $\lambda_i \in \{\&,\,\supset\}$. Пусть $A_{i_1}\,\&\,A_{i_2}\,\&\,\ldots\,\&\,A_{i_k}$ — импликанта этой нормальной формы. Для каждого $i = 1,\,\ldots,\,n$ пусть B_i обозначает псевдоформулу

$$A_{i+1}\,\lambda_{i+1}\,(A_{i+2}\,\lambda_{i+2}\,(\ldots(A_n\,\lambda_n\,G))\ldots)$$

и для каждого $j = 1,\,\ldots,\,k$ пусть Δ_j обозначает множество, задаваемое равенствами

$$\Delta_1 = \{\mathbf{NF}(A,\,G'),\,A_1,\,B_1,\,A_2,\,B_2,\,\ldots,\,A_{i_1-1},\,B_{i_1-1}\},$$

$$\Delta_j = \Delta_{j-1} \cup \{B_{i_{(j-1)}},\,\ldots,\,A_{i_j-1},\,B_{i_j-1}\}.$$

Тогда если $\Delta_1\xi\, A\Gamma\,?\,A_{i_1}\xi$, …, $\Delta_k\xi\, A\Gamma\,?\,A_{i_k}\xi$ образуют k разрешимых задач, у которых существуют соответственно допустимые подстановки θ_1, …, θ_k такие, что совместимы подстановки σ, θ_1, …, θ_k, то считается разрешимой также задача $A\,\Gamma\,?\,G$ и комбинация σ, θ_1, …, θ_k считается ее допустимой подстановкой. Символически сформулированное правило сведения задач к подзадачам записывается следующим образом:

$$\frac{\Delta_1\xi\, A\Gamma\,?\,A_{i_1}\xi; \quad \Delta_2\xi\, A\Gamma\,?\,A_{i_2}\xi; \quad \ldots; \quad \Delta_k\xi\, A\Gamma\,?\,A_{i_k}\xi}{A\,\Gamma\,?\,G}\ (\supset ?).$$

8. Пусть G есть атом, дизъюнкция, $\perp$ или псевдоформула вида $\exists x\, B$, а также $D_1 \vee D_2$ есть строго позитивная аналитическая подпсевдоформула псевдоформулы A. Пусть нормальная форма A относительно $D_1 \vee D_2$ имеет вид

$$A_1\,\lambda_1\,(A_2\,\lambda_2\,(\ldots(A_n\,\lambda_n\,(D_1 \vee D_2))\ldots),$$

где $\lambda_i \in \{\&, \supset\}$. Пусть $A_{i_1}\,\&\,A_{i_2}\,\&\,\ldots\,\&\,A_{i_k}$ — импликанта этой нормальной формы. Пусть подстановка η замещает все локальные метапеременные в A новыми глобальными. Для каждого $i = 1, \ldots, n$ пусть B_i обозначает псевдоформулу

$$A_{i+1}\,\lambda_{i+1}\,(A_{i+2}\,\lambda_{i+2}\,(\ldots(A_n\,\lambda_n\,(D_1 \vee D_2))\ldots)$$

и пусть множества Δ и Δ_j, где $j = 1, \ldots, k$, задаются равенствами

$$\Delta_1 = \{\mathbf{NF}(A,\, D_1 \vee D_2),\, A_1,\, B_1,\, A_2,\, B_2,\, \ldots,\, A_{i_1-1},\, B_{i_1-1}\},$$

$$\Delta_j = \Delta_{j-1} \cup \{B_{i_{(j-1)}},\, \ldots,\, A_{i_j-1},\, B_{i_j-1}\},$$
$$\Delta = \Delta_k \cup \{B_{i_k},\, \ldots,\, A_n,\, B_n\}, \quad B_n = D_1 \vee D_2.$$

Тогда если $(\Delta_1\eta\, A\Gamma\,?\,A_{i_1}\eta)$, …, $(\Delta_k\eta\, A\Gamma\,?\,A_{i_k}\eta)$, $(\mathbf{Sk}(D_1\eta)\,(\Delta\eta)\,A\Gamma\,?\,G)$, $(\mathbf{Sk}(D_2\eta)\,(\Delta\eta)\,A\Gamma\,?\,G)$ образуют $k+2$ разрешимые задачи, у которых существуют совместимые допустимые подстановки θ_1, …, θ_k, θ_{k+1}, θ_{k+2}, то считается разрешимой также задача $A\,\Gamma\,?\,G$ и ее допустимой подстановкой считается комбинация подстановок θ_1, …, θ_k, θ_{k+1}, θ_{k+2}. Таким образом, задача $A\Gamma\,?\,G$ сводится к следующим $k+2$ подзадачам:

$$\frac{\begin{array}{c}\Delta_1\eta\, A\Gamma\,?\,A_{i_1}\eta; \qquad \ldots; \qquad \Delta_k\eta\, A\Gamma\,?\,A_{i_k}\eta; \\ \mathbf{Sk}(D_1\eta)\,(\Delta\eta)\,A\Gamma\,?\,G; \quad \mathbf{Sk}(D_2\eta)\,(\Delta\eta)\,A\Gamma\,?\,G\end{array}}{A\,\Gamma\,?\,G}\ (\vee ?).$$

9. Наконец, мы имеем правило противоречия $(\perp_{\mathbf{c}})$: если G — атом, дизъюнкция или псевдоформула вида $\exists x\, B$, разрешима задача $(\neg G)\,\Gamma\,?\,\perp$

и θ — ее допустимая подстановка, то разрешима задача $\Gamma\,?\,G$ и θ — ее допустимая подстановка. То есть, задача $\Gamma\,?\,G$ сводится к подзадаче $(\neg G)\,\Gamma\,?\,\bot$:

$$\frac{(\neg G)\,\Gamma\,?\,\bot}{\Gamma\,?\,G}\;(\bot_{\mathbf{c}}).$$

Формулировка продукционной системы $\mathcal{F}_3$ завершена. Будем ее называть *продукционной системой эвристического поиска вывода*. Продукционная система $\mathcal{F}_3$ задает алгоритм поиска решения задачи с точностью до стратегии построения дерева поиска типа И/ИЛИ. Зафиксировав конкретную стратегию мы получим некоторую реализацию так сформулированного алгоритма. Например, алгоритм [1] использует стратегию перебора в глубину.

Заметим, что если θ является допустимой подстановкой разрешимой задачи, то ее допустимой считается также всякая подстановка $\lambda \leq \theta$. Если π есть разрешимая задача и θ — ее допустимая подстановка, то индукцией по высоте деривационного дерева устанавливается, что всякая подстановка $\lambda \leq \theta$ является допустимой подстановкой всех потомков этой задачи π в деривационном дереве.

Очевидно, что рассматриваемые в заданном дереве поиска Σ нормальные формы псевдоформул зависят от дерева Σ: $\mathbf{Sk}_\Sigma(A)$, $\mathbf{NF}_\Sigma(A,\,B)$, $\mathbf{PNF}_\Sigma(A,\,B)$. Тем не менее мы будем в этих обозначениях часто опускать индекс Σ, когда подразумеваемый контекст Σ очевиден.

В следующих двух разделах доказываются теоремы о корректности и полноте для сформулированной продукционной системы $\mathcal{F}_3$ по отношению к исчислению $\mathcal{S}_3$. Тем самым устанавливается взаимосвязь между поиском решения задач в $\mathcal{F}_3$ и поиском вывода в классическом исчислении секвенций $\mathcal{S}_3$. Для этого там устанавливается эквивалентность $\mathcal{F}_3$ и $\mathcal{P}_3$ с точки зрения объема решаемых задач. Использование скулемизации в $\mathcal{F}_3$ приводит к элиминации строго положительных аналитических кванторов в антецедентах дедуктивных задачах. Это позволяет избежать комбинаторного взрыва при применении кванторных правил, возникающего в $\mathcal{P}_3$.

4. Теорема о корректности

Доказываемая в этом разделе теорема о корректности предоставляет эффективный способ извлечения логического вывода из деривационного поддерева поискового дерева разрешимой дедуктивной задачи. Указанная теорема легко доказывается для пропозициональной части продукционной системы эвристического поиска $\mathcal{F}_3$. Распространение этого доказательства на всю $\mathcal{F}_3$ связано с определенными трудностями, которые, тем не менее, удается успешно преодолеть. В доказательстве теоремы о корректности будут использованы два предложения, которые формулируются ниже.

Предложение 4.1. *Пусть D есть строго положительная аналитическая подпсевдоформула псевдоформулы C и C' есть пренексная форма C относительно D или подобная C псевдоформула. Тогда если в $\mathcal{P}_3$ разрешима задача $(C'\, C\, \Gamma,\ A)$ и θ — ее допустимая подстановка, то задача $(C\, \Gamma,\ A)$ также является разрешимой и θ является ее допустимой подстановкой.*

Доказательство может быть получено тривиальной индукцией по высоте деривационного дерева. Это утверждение, а также следующее очевидное следствие, нам потребуются ниже в доказательстве предложения 4.2.

Следствие. *Пусть D есть строго положительная аналитическая подпсевдоформула псевдоформулы C. Тогда если в $\mathcal{P}_3$ разрешима задача $(\mathbf{PNF}(C,\ D)\, C\, \Gamma,\ A)$ и θ — ее допустимая подстановка, то задача $(C\, \Gamma,\ A)$ также является разрешимой и θ является ее допустимой подстановкой.*

Предложение 4.1 и его следствие, по сути, представляют собой формулировки допустимых правил декомпозиции, которые мы будем соответственно символически обозначать (ПФ$\rightarrow$) и (ПНФ$\rightarrow$). Через $\Theta|S$ будем обозначать поддерево дерева Θ, которое содержит узел S дерева Θ в качестве корня, а также все его потомки. На диаграммах фрагмент $\ddots\vdots\iddots$ будет обозначать часть дерева. Например, фигуры

$$\begin{array}{c} \\ \ddots\vdots\iddots \\ S \end{array} \qquad \text{и} \qquad \begin{array}{ccc} S_1 & S_2 & S_3 \\ & \ddots\vdots\iddots & \\ & S & \end{array}$$

обозначают соответственно некоторое дерево с корнем S и некоторое дерево с корнем S и его потомками S_1, S_2, S_3.

Предложение 4.2. *Если задача $(\mathbf{Sk}(\Gamma)\,?\,A)$ разрешима в продукционной системе $\mathcal{F}_3$, то задача $(\Gamma,\ A)$ разрешима в продукционной системе $\mathcal{P}_3$.*

Доказательство. Пусть $\theta = \begin{pmatrix} X_1 & \ldots & X_r \\ t_1 & \ldots & t_r \end{pmatrix}$ — подстановка из набора допустимых подстановок задачи $\pi_0 = (\mathbf{Sk}(\Gamma)\,?\,A)$. Мы предполагаем, что подстановка θ является основной. В противном случае мы можем заменить "нерешенные" метапеременные на константы из H_0. Пусть Θ — соответствующее этой подстановке θ деривационное дерево. Индукцией по построению деривационного дерева Θ покажем, как в рамках $\mathcal{P}_3$ построить допустимую для задачи $(\Gamma,\, A)$ подстановку $\theta' = \begin{pmatrix} X_1 & \dots & X_r & \dots \\ t'_1 & \dots & t'_r & \dots \end{pmatrix}$ и соответствующее ей деривационное дерево Θ'. Разобъем процесс построения дерева Θ на этапы. Будем считать, что для формирования дерева Θ_{n+1} из дерева Θ_n на текущем этапе n выбирается некоторый лист построенного поддерева $\Theta_n \subseteq \Theta$ и формируются все его дочерние вершины. Пусть процесс расширения языка при построении дерева Θ имеет вид

$$\begin{array}{rl} & \Theta_0 \subseteq \Theta_1 \subseteq \ldots \subseteq \Theta_n \subseteq \Theta_{n+1} \subseteq \ldots \subseteq \Theta_h = \Theta, \\ \Omega \subseteq & \Omega_0 \subseteq \Omega_1 \subseteq \ldots \subseteq \Omega_n \subseteq \Omega_{n+1} \subseteq \ldots \subseteq \Omega_h, \\ & H_0 \subseteq H_1 \subseteq \ldots \subseteq H_n \subseteq H_{n+1} \subseteq \ldots \subseteq H_h. \end{array} \tag{6}$$

Прежде всего введем следующие обозначения. Обозначим через S_n множество всех скулемовских символов и предметных переменных, появившихся в задаче π_n в результате декомпозиции ее родительской задачи. Обозначим через $T_n \subseteq H_n$ множество всех термов $t \in H_n$ вида $f^m(\tau_1,\, \ldots,\, \tau_m)$, где f^m — некоторый скулемовский символ, которые находятся в дереве $(\Theta|\pi_n)\theta$ и содержат символы из S_n.

Покажем, как для каждого дерева Θ_n можно построить подстановки θ'_n, μ_n и дерево Θ'_n такие, что корнем дерева Θ'_n является задача $\pi'_0 = (\Gamma,\, A)$, дочерние вершины получаются из родительских по одному из допустимых правил декомпозиции системы $\mathcal{P}_3$, а листья дерева Θ'_n либо примитивизируются подстановкой θ'_n, либо соответствуют нераскрытым листьям $\pi_n = (\mathbf{Sk}(\Gamma_n)?A_n)$ дерева Θ_n и являются задачами π'_n вида $\pi'_n = (\Delta_n\Gamma'_n,\, A'_n)$. При этом роль подстановок θ'_n и μ_n в обозначаемом штрихом формируемом соответствии заключается в том, что на каждом шаге n θ'_n составляет часть искомой θ' и содержит свободные предметные переменные, вводимые для устранения скулемовских функций, а μ_n заменяет эти предметные переменные на термы со скулемовскими функциями. Для этого формируемое соответствие будет таким, чтобы соблюдались следующие свойства:

- если псевдовыражение E' входит в π'_n и соответствует псевдовыражению E, то $E\theta = E'(\theta'_n \circ \mu_n)$; в частности, $A_n\theta = A'_n(\theta'_n \circ \mu_n)$ и $\Gamma_n\theta = \Gamma'_n(\theta'_n \circ \mu_n)$;

- каждому терму $t \in H_n$ вида $f^m(\tau_1, \ldots, \tau_m)$ из дерева $(\Theta|\pi_n)\theta$ такому, что $t \notin T_n$, сопоставляется некоторый терм t' без скулемовских функций такой, что $t = t'\mu_n$.

Введем несколько новых обозначений. Для множества термов T_n определим подстановки δ_n и ν_n, которые будем называть *сопряженными* листу $\pi'_n = (\Delta_n\Gamma'_n, A'_n)$ дерева Θ'_n. Для этого рассмотрим последовательность псевдоформул $E_1, \ldots, E_l$ такую, что для каждого терма $t = f(\tau_1, \ldots, \tau_N)$ из T_n в этой последовательности содержится соответствующий этому терму фрагмент $P_1, \ldots, P_q$, *составленный согласно протоколу скулемизации*. Последнее означает следующее. Пусть скулемовский символ f появился в дереве поиска в результате скулемизации псевдоформулы $C \in \Gamma_n$. Пусть $\mathbf{PNF}(C, D)$ имеет вид (4). Пусть кванторная приставка (3) псевдоформулы C имеет вид (1). При этом будем считать, что символы $\varphi_1^{n_1}, \ldots, \varphi_k^{n_k}$ вводятся при скулемизации C с помощью элиминации кванторов $\exists v_1, \ldots, \exists v_k$ из (1) и f есть один из символов $\varphi_1^{n_1}, \ldots, \varphi_k^{n_k}$. Введем новые предметные переменные $y_1, \ldots, y_k$ и новые глобальные метапеременные $Y_1, \ldots, Y_{n_k}$. Составим цепочку псевдоформул $P_1, \ldots, P_q$ такую, что $P_1 = \mathbf{PNF}(C', D')$,

$$P_q = \forall z_{n_k+1} \ldots \forall z_{n_k+m}(C'_1\,\lambda_1\,(C'_2\,\lambda_2\,(\ldots(C'_s\,\lambda_s\,D'))\ldots)\gamma,$$

$$\gamma = \begin{pmatrix} z_1 & \ldots & z_{n_k} & v_1 & \ldots & v_k \\ Y_1 & \ldots & Y_{n_k} & y_1 & \ldots & y_k \end{pmatrix};$$

- если $P_j = \forall z_i\, B(z_i)$, то $P_{j+1} = B(Y_i)$;
- если $P_j = \exists v_i\, B(v_i)$, то $P_{j+1} = B(y_i)$.

Пусть термы $\tau'_1, \ldots, \tau'_{n_k}$ соответствуют термам $\tau_1, \ldots, \tau_{n_k} \in H_n$, т. е. $\tau_i = \tau'_i\mu_n$. *Сопряженными терму* $t = f(\tau_1, \ldots, \tau_N) \in T_n$ мы будем называть следующие подстановки:

$$\begin{pmatrix} Y_1 & Y_2 & \ldots & Y_{n_1} & \ldots & Y_{n_k} \\ \tau'_1 & \tau'_2 & \ldots & \tau'_{n_1} & \ldots & \tau'_{n_k} \end{pmatrix}, \tag{7}$$

$$\begin{pmatrix} y_1 & \ldots & y_k \\ \varphi_1^{n_1}(\tau_1, \ldots, \tau_{n_1}) & \ldots & \varphi_k^{n_k}(\tau_1, \ldots, \tau_{n_k}) \end{pmatrix}. \tag{8}$$

Объединив все вычисленные для каждого терма из T_n такие сопряженные им подстановки (7) мы и получим *сопряженную задаче* π'_n подстановку δ_n. Аналогично, объединив все подстановки (8) мы получаем подстановку ν_n.

Для так определенных подстановок δ_n и ν_n, сопряженных задаче $\pi'_n = (\Delta_n \Gamma'_n,\, A'_n)$, определим последовательность конечных множеств псевдоформул $\Delta_n^1 \subseteq \Delta_n^2 \subseteq \ldots \subseteq \Delta_n^l$ следующим образом: $\Delta_n^j = \Delta_n \cup \{E_1, \ldots, E_j\}$, $j = 1, \ldots, l$. Следствием индуктивного предположения о задаче π'_n будет то, что задачи

$$(\Delta_n \Gamma'_n,\, A'_n),\ (\Delta_n^1 \Gamma'_n,\, A'_n),\ \ldots,\ (\Delta_n^l \Gamma'_n,\, A'_n) \tag{9}$$

образуют последовательность, в которой каждая следующая задача является результатом применения одного из правил декомпозиции (ПНФ$\to$), ($\forall \to$) или ($\exists \to$) к предыдущей. При этом δ_n и ν_n были построены так, чтобы цепочка задач (9) обладала следующими свойствами:

- если псевдовыражение E' входит в $(\Delta_n^l \Gamma'_n,\, A'_n)$ и соответствует псевдовыражению E, то $E\theta = E'((\theta'_n \cup \delta_n) \circ (\mu_n \cup \nu_n))$; в частности, $A_n\theta = A'_n((\theta'_n \cup \delta_n) \circ (\mu_n \cup \nu_n))$ и $\Gamma_n\theta = \Gamma'_n((\theta'_n \cup \delta_n) \circ (\mu_n \cup \nu_n))$;

- каждому терму $t \in H_n$ из дерева $(\Theta|\pi_n)\theta$ сопоставляется некоторый терм t' без скулемовских функций такой, что $t = t'(\mu_n \cup \nu_n)$.

В начале процесса построения дерева Θ' положим $\Delta_0 = \emptyset$, $\Gamma'_0 = \Gamma$, $A'_0 = A$, $\pi'_0 = (\Delta_0 \Gamma'_0,\, A'_0)$, $\Theta'_0 = \{\pi'_0\}$, $\theta'_0 = \varepsilon$ и $\mu_0 = \varepsilon$. Очевидно, что дерево Θ'_0 по отношению к Θ_0 обладает требуемым свойством.

Покажем как получить дерево Θ'_{n+1} из дерева Θ'_n в зависимости от правила декомпозиции, которое используется при формировании дерева Θ_{n+1} из дерева Θ_n. Всего имеется 10 правил, которые мы рассмотрим в следующей последовательности: ($?\supset$), ($?\top$), ($?\vee_i$), ($?\exists$), ($?\&$), ($?\forall$), ($\perp_{\mathbf{c}}$), ($\vee?$), ($\&?$), ($\supset?$).

1. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции ($?\supset$). Это значит, что $A_n = (B_1 \supset B_2)$, $A_n\theta = (B_1 \supset B_2)\theta = (B_1\theta) \supset (B_2\theta) = (B'_1(\theta'_n \circ \mu_n)) \supset (B'_2(\theta'_n \circ \mu_n)) = (B'_1 \supset B'_2)(\theta'_n \circ \mu_n) = A'_n(\theta'_n \circ \mu_n)$, $A'_n = (B'_1 \supset B'_2)$, $\Gamma_{n+1} = \Gamma_n \cup \{B_1\}$, $A_{n+1} = B_2$ и язык Ω_{n+1} является расширением языка Ω_n за счет скулемизации B_1. Пусть подстановки δ_n и ν_n сопряжены листу π'_n. Пусть $(\Delta_n \Gamma'_n,\, B'_1 \supset B'_2)$, $(\Delta_n^1 \Gamma'_n,\, B'_1 \supset B'_2)$, $\ldots$, $(\Delta_n^l \Gamma'_n,\, B'_1 \supset B'_2)$ есть построенная так, как указано выше (9), соответствующая подстановкам δ_n и ν_n цепочка задач, в которой каждая следующая задача является результатом применения одного из правил декомпозиции (ПНФ$\to$), ($\forall \to$) или ($\exists \to$) к предыдущей. Положим $\theta'_{n+1} = \theta'_n \cup \delta_n$, $\mu_{n+1} = \mu_n \cup \nu_n$, $\Gamma'_{n+1} = \Gamma'_n \cup \{B'_1\}$, $A'_{n+1} = B'_2$ и $\Delta_{n+1} = \Delta_n^l$. Дерево Θ'_{n+1} сформируем из дерева Θ'_n с помощью применения сначала правил из списка (ПНФ$\to$), ($\forall \to$), ($\exists \to$), а

затем правила ($\to\supset$):

$$\cfrac{(B'_1\,\Delta^l_n\Gamma'_n,\,B'_2)}{(\Delta^l_n\Gamma'_n,\,B'_1\supset B'_2)}$$
$$\vdots$$
$$\cfrac{(\Delta^1_n\Gamma'_n,\,B'_1\supset B'_2)}{(\Delta_n\Gamma'_n,\,B'_1\supset B'_2)}$$
$$\ddots\vdots\cdot^{\cdot^{\cdot}}$$
$$(\Gamma,\,A)$$

В результате для листа $(\mathbf{Sk}(\Gamma_{n+1})\,?\,A_{n+1})$ дерева Θ_{n+1} в дереве Θ'_{n+1} сформирован лист требуемого вида $(\Delta_{n+1}\Gamma'_{n+1},\,A'_{n+1})$ такой, что $A_{n+1}\theta = A'_{n+1}(\theta'_{n+1}\circ\mu_{n+1})$ и $\Gamma_{n+1}\theta = \Gamma'_{n+1}(\theta'_{n+1}\circ\mu_{n+1})$. Таким образом, дерево Θ'_{n+1} по отношению к Θ_{n+1} снова обладает требуемым свойством.

2. Пусть лист $\pi_n = (\mathbf{Sk}(\Gamma_n)\,?\,A_n)$ дерева Θ_n является примитивной задачей, у которой $A_n = \top$. Это значит, что $A'_n(\theta'_n\circ\mu_n) = A_n\theta = \top\theta = \top$ и $A'_n = \top$. В этом случае $\Theta_{n+1} = \Theta_n$. Положим $\Theta'_{n+1} = \Theta'_n$, $\theta'_{n+1} = \theta'_n$ и $\mu_{n+1} = \mu_n$. При этом дерево Θ'_{n+1} вновь обладает требуемым свойством и имеет вид:

$$(\Delta_n\Gamma'_n,\,\top)$$
$$\ddots\vdots\cdot^{\cdot^{\cdot}}$$
$$(\Gamma,\,A)$$

3. Пусть Θ_{n+1} получается из Θ_n в результате применения продукции $(?\vee_i)$. Следовательно имеем $A_n = (B_1\vee B_2)$, $A'_n = (B'_1\vee B'_2)$. Пусть подстановкам δ_n и ν_n соответствует цепочка задач $(\Delta^1_n\Gamma'_n,\,B'_1\vee B'_2)$, …, $(\Delta^l_n\Gamma'_n,\,B'_1\vee B'_2)$. Положим $\theta'_{n+1} = \theta'_n\cup\delta_n$, $\mu_{n+1} = \mu_n\cup\nu_n$ и $A'_{n+1} = B'_i$. Дерево Θ'_{n+1} сформируем из дерева Θ'_n с помощью применения правил (ПНФ$\to$), $(\forall\to)$, $(\exists\to)$ и $(\to\vee_i)$:

$$\cfrac{(\Delta^l_n\Gamma'_n,\,B'_i)}{(\Delta^l_n\Gamma'_n,\,B'_1\vee B'_2)}$$
$$\vdots$$
$$\cfrac{(\Delta^1_n\Gamma'_n,\,B'_1\vee B'_2)}{(\Delta_n\Gamma'_n,\,B'_1\vee B'_2)}$$
$$\ddots\vdots\cdot^{\cdot^{\cdot}}$$
$$(\Gamma,\,A)$$

В результате сформированный в Θ'_{n+1} лист имеет вид $(\Delta_{n+1}\Gamma'_{n+1},\,A'_{n+1})$. Что соответствует новому листу $(\mathbf{Sk}(\Gamma_{n+1})?A_{n+1})$ в дереве Θ_{n+1}.

4. Пусть Θ_{n+1} получается из Θ_n с помощью применения правила декомпозиции $(?\exists)$. Это значит, что $A_n = (\exists x\,B(x))$, $A'_n = (\exists x\,B'(x))$,

$\Gamma_{n+1} = \Gamma_n$, $A_{n+1} = B(X)$, $X = X_s$, $X_s \in \mathbf{dom}\,\theta$ для некоторого $s \in \{1, \ldots, r\}$, $t_s \in H_n$, $t_s \in (\Theta|\pi_n)\theta$. Пусть терм t'_s соответствует терму t_s, т. е. $t'_s(\mu_n \cup \nu_n) = t_s$. Тогда положим $\theta'_{n+1} = \theta'_n \cup \delta_n \cup \begin{pmatrix} X_s \\ t'_s \end{pmatrix}$, $\mu_{n+1} = \mu_n \cup \nu_n$ и $A'_{n+1} = B'(X_s)$. Отсюда имеем $B'(X_s)\,\theta'_{n+1} = B'(t'_s)\,\theta'_n$ и $A_{n+1}\theta = A'_{n+1}(\theta'_{n+1} \circ \mu_{n+1})$. Таким образом, искомое дерево Θ'_{n+1} получается из дерева Θ'_n в результате применения сначала правил из списка (ПНФ$\to$), ($\forall \to$), ($\exists \to$), а затем правила ($\to \exists$):

$$\cfrac{(\Delta^l_n\Gamma'_n,\, B'(X_s))}{(\Delta^l_n\Gamma'_n,\, \exists x\, B'(x))}$$
$$\vdots$$
$$\cfrac{(\Delta^1_n\Gamma'_n,\, \exists x\, B'(x))}{(\Delta_n\Gamma'_n,\, \exists x\, B'(x))}$$
$$\ddots\vdots\cdot^{\cdot^{\cdot}}$$
$$(\Gamma,\, A)$$

5. Пусть Θ_{n+1} получается из Θ_n по правилу (?&). Это значит, что $A_n = (B_1 \,\&\, B_2)$, $A'_n = (B'_1 \,\&\, B'_2)$. Положим $\theta'_{n+1} = \theta'_n \cup \delta_n$ и $\mu_{n+1} = \mu_n \cup \nu_n$. Тогда искомое дерево Θ'_{n+1} получается из Θ'_n в результате применения правил (ПНФ$\to$), ($\forall \to$), ($\exists \to$) и ($\to$ &):

$$\cfrac{(\Delta^l_n\Gamma'_n,\, B'_1) \qquad (\Delta^l_n\Gamma'_n,\, B'_2)}{(\Delta^l_n\Gamma'_n,\, B'_1 \,\&\, B'_2)}$$
$$\vdots$$
$$\cfrac{(\Delta^1_n\Gamma'_n,\, B'_1 \,\&\, B'_2)}{(\Delta_n\Gamma'_n,\, B'_1 \,\&\, B'_2)}$$
$$\ddots\vdots\cdot^{\cdot^{\cdot}}$$
$$(\Gamma,\, A)$$

В результате каждый сформированный в Θ'_{n+1} новый лист имеет вид $(\Delta_{n+1}\Gamma'_{n+1},\, A'_{n+1})$, где $A'_{n+1} = B'_1$ или $A'_{n+1} = B'_2$. Что соответствует каждому новому листу $(\mathbf{Sk}(\Gamma_{n+1})?A_{n+1})$ дерева Θ_{n+1}.

6. Пусть Θ_{n+1} получается из Θ_n с помощью применения правила декомпозиции (?$\forall$). Это значит, что $A_n = (\forall x\, B(x))$, $A_{n+1} = B(a)$, $\Omega_{n+1} = \Omega_n \cup \{a\}$, $\Gamma_{n+1} = \Gamma_n$. Положим $\theta'_{n+1} = \theta'_n \cup \delta_n$, $\mu_{n+1} = \mu_n \cup \nu_n$ и $A'_{n+1} = B'(a)$. Дерево Θ'_{n+1} получим из дерева Θ'_n как результат нескольких применений сначала правил из списка (ПНФ$\to$), ($\forall \to$), ($\exists \to$), а

затем правила $(\to \forall)$:

$$\dfrac{\dfrac{(\Delta_n^l \Gamma_n',\, B'(a))}{(\Delta_n^l \Gamma_n',\, \forall x\, B'(x))}}{\begin{array}{c}\vdots\\ \dfrac{(\Delta_n^1 \Gamma_n',\, \forall x\, B'(x))}{(\Delta_n \Gamma_n',\, \forall x\, B'(x))}\\ \cdot\cdot\vdots\cdot\cdot \\ (\Gamma,\, A)\end{array}}$$

7. Пусть Θ_{n+1} получается из Θ_n по правилу $(\perp_{\mathbf{c}})$. Следовательно имеем $\Gamma_{n+1} = \Gamma_n \cup \{\neg A_n\}$, $A_{n+1} = \perp$, $\Gamma_n\theta = \Gamma_n'(\theta_n' \circ \mu_n)$ и $A_n\theta = A_n'(\theta_n' \circ \mu_n)$. Положим $\theta_{n+1}' = \theta_n' \cup \delta_n$, $\mu_{n+1} = \mu_n \cup \nu_n$, $\Gamma_{n+1}' = \Gamma_n' \cup \{\neg A_n'\}$ и $A_{n+1}' = \perp$. Дерево Θ_{n+1}' сформируем из дерева Θ_n' с помощью применения сначала правил (ПНФ$\to$), $(\forall \to)$, $(\exists \to)$, а затем правила $(\perp_{\mathbf{c}})$:

$$\begin{array}{c}\dfrac{((\neg A_n')\, \Delta_n^l \Gamma_n',\, \perp)}{(\Delta_n^l \Gamma_n',\, A_n')}\\ \vdots\\ \dfrac{(\Delta_n^1 \Gamma_n',\, A_n')}{(\Delta_n \Gamma_n',\, A_n')}\\ \cdot\cdot\vdots\cdot\cdot \\ (\Gamma,\, A)\end{array}$$

В результате для листа $(\mathbf{Sk}(\Gamma_{n+1})?A_{n+1})$ дерева Θ_{n+1} в дереве Θ_{n+1}' сформирован лист требуемого вида $(\Delta_{n+1}\Gamma_{n+1}',\, A_{n+1}')$ такой, что $\Gamma_{n+1}\theta = \Gamma_{n+1}'(\theta_{n+1}' \circ \mu_{n+1})$ и $A_{n+1}\theta = \perp = A_{n+1}'(\theta_{n+1}' \circ \mu_{n+1})$. Тем самым рассмотрение случая $(\perp_{\mathbf{c}})$ завершено.

Рассмотрение последних трех случаев $(\vee?)$, $(\&?)$ и $(\supset?)$ предварим определением новых понятий. Пусть F есть псевдоформула из левой части задачи, нормальная форма $\mathbf{NF}(F,\, G)$ которой используется в указанных правилах декомпозиции. Пусть $F = \mathbf{Sk}(C)$ и соответствующая (5) нормальная форма псевдоформулы C имеет вид (2). При этом будем считать, что при скулемизации C вводятся k скулемовских функций $\varphi_1^{n_1}, \ldots, \varphi_k^{n_k}$ с помощью элиминации кванторов $\exists v_1, \ldots, \exists v_k$. Список всех строго положительных аналитических связанных кванторами всеобщности переменных в порядке вхождения в C запишем в виде $z_1, \ldots, z_m$. Пусть $Y_1, \ldots, Y_m$ суть глобальные метапеременные подстановки η, которые вводятся вместо локальных метапеременных $Z_1, \ldots, Z_m$ псевдоформулы F. Возьмем часть $\begin{pmatrix} Y_1 & Y_2 & \ldots & Y_m \\ \tau_1 & \tau_2 & \ldots & \tau_m \end{pmatrix}$ подстановки θ. Пусть термы $\tau_1', \ldots,$

τ'_m соответствуют термам $\tau_1, \ldots, \tau_m \in H_n$, т. е. $\tau_i = \tau'_i(\mu_n \cup \nu_n)$. Положим

$$\tilde{\delta}_n = \delta_n \cup \begin{pmatrix} Y_1 & Y_2 & \ldots & Y_m \\ \tau'_1 & \tau'_2 & \ldots & \tau'_m \end{pmatrix}.$$

Введем новые свободные предметные переменные $y_1, \ldots, y_k$ и положим

$$\tilde{\nu}_n = \nu_n \cup \begin{pmatrix} y_1 & \ldots & y_k \\ \varphi_1^{n_1}(\tau_1, \ldots, \tau_{n_1}) & \ldots & \varphi_k^{n_k}(\tau_1, \ldots, \tau_{n_k}) \end{pmatrix}.$$

Составим список $D'_1, \ldots, D'_N$ всех строго положительных аналитических вхождений атомов, дизъюнкций, $\bot$ и $\top$ в $\mathbf{PNF}(C', D')$ такой, что $D'_1 = D'$. Сформируем список псевдоформул $P_1, \ldots, P_N$ такой, что $P_1 = \mathbf{PNF}(C', D'_1)$ и каждая следующая P_{j+1} в этом списке есть пренексная форма предыдущей P_j относительно D'_{j+1}. Достроим этот список, добавив новые псевдоформулы $P_{N+1}, \ldots, P_q$ такие, что P_q не содержит строго положительных аналитических кванторов и

- если $P_j = \forall z_i\, B(z_i)$, то $P_{j+1} = B(Y_i)$;
- если $P_j = \exists v_i\, B(v_i)$, то $P_{j+1} = B(y_i)$.

Обозначим $\tilde{C} = \mathbf{PNF}(C, D)$ и $\tilde{F} = \mathbf{NF}(F, G)\eta$. Псевдоформулы $P_1, \ldots, P_q$ в частности включают псевдоформулы $\tilde{C}' = P_1$ и $\tilde{F}' = P_q$, для которых выполняются равенства

$$\begin{array}{rcl} \tilde{C}\theta &=& \tilde{C}'((\theta'_n \cup \tilde{\delta}_n) \circ (\mu_n \cup \tilde{\nu}_n)), \\ \tilde{F}\theta &=& \tilde{F}'((\theta'_n \cup \tilde{\delta}_n) \circ (\mu_n \cup \tilde{\nu}_n)). \end{array} \tag{10}$$

Для сформированных псевдоформул $P_1, P_2, \ldots, P_q$ построим соответствующие им множества $\Delta_n^{l+1}, \Delta_n^{l+2}, \ldots, \Delta_n^{l'}$, где $l' = l + q$, согласно следующим равенствам:

$$\Delta_n^{l+1} = \Delta_n^l \cup \{P_1\},\ \Delta_n^{l+2} = \Delta_n^l \cup \{P_1, P_2\},\ \ldots,\ \Delta_n^{l'} = \Delta_n^l \cup \{P_1, \ldots, P_q\}.$$

Следствием индуктивного предположения о задаче π'_n будет то, что задачи

$$\begin{array}{c} (\Delta_n\Gamma'_n, A'_n),\ (\Delta_n^1\Gamma'_n, A'_n),\ \ldots,\ (\Delta_n^l\Gamma'_n, A'_n), \\ (\Delta_n^{l+1}\Gamma'_n, A'_n),\ \ldots,\ (\Delta_n^{l'}\Gamma'_n, A'_n) \end{array} \tag{11}$$

образуют последовательность, в которой каждая следующая задача является результатом применения одного из правил декомпозиции (ПНФ$\to$), (ПФ$\to$), ($\forall\ \to$) или ($\exists\ \to$) к предыдущей. При этом цепочка задач (11) обладает следующими свойствами:

- если псевдовыражение E' входит в $(\Delta_{\tilde{n}}^{l'}\Gamma'_n,\, A'_n)$ и соответствует псевдовыражению E, то $E\theta = E'((\theta'_n \cup \tilde{\delta}_n) \circ (\mu_n \cup \tilde{\nu}_n))$; в частности, $A_n\theta = A'_n((\theta'_n \cup \tilde{\delta}_n) \circ (\mu_n \cup \tilde{\nu}_n))$ и $\Gamma_n\theta = \Gamma'_n((\theta'_n \cup \tilde{\delta}_n) \circ (\mu_n \cup \tilde{\nu}_n))$;

- каждому терму $t \in H_n$ из дерева $(\Theta|\pi_n)\theta$ сопоставляется некоторый терм t' без скулемовских функций такой, что $t = t'(\mu_n \cup \tilde{\nu}_n)$.

Эти свойства указанных задач (11) нам потребуются ниже при рассмотрении тех случаев, когда дерево Θ_{n+1} формируется из дерева Θ_n с помощью правил декомпозиции $(\vee?)$, $(\&?)$ и $(\supset?)$.

8. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции $(\vee?)$. Тогда существует псевдоформула $F \in \mathbf{Sk}(\Gamma_n)$ с нормальной формой вида

$$F_1\,\lambda_1\,(F_2\,\lambda_2\,(\ldots(F_s\,\lambda_s\,(G_1 \vee G_2))\ldots)$$

такая, что $p+2$ задачи

$$((\Sigma_1\eta)\,\mathbf{Sk}(\Gamma_n)\,?\,F_{i_1}\eta),\ \ldots,\ ((\Sigma_p\eta)\,\mathbf{Sk}(\Gamma_n)\,?\,F_{i_p}\eta),$$

$$(\mathbf{Sk}(G_1\eta)\,(\Sigma\eta)\,\mathbf{Sk}(\Gamma_n)\,?\,A_n), \quad (\mathbf{Sk}(G_2\eta)\,(\Sigma\eta)\,\mathbf{Sk}(\Gamma_n)\,?\,A_n)$$

образуют у задачи π_n все дочерние вершины в дереве Θ. При этом будем считать, что $F_{i_1} \,\&\, F_{i_2} \,\&\, \ldots \,\&\, F_{i_p}$ — импликанта этой нормальной формы, подстановка η замещает все локальные метапеременные в $\mathbf{NF}(F,\, G_1 \vee G_2)$ новыми глобальными, а также

$$R_i = F_{i+1}\,\lambda_{i+1}\,(F_{i+2}\,\lambda_{i+2}\,(\ldots(F_s\,\lambda_s\,(G_1 \vee G_2))\ldots), \quad i = 1,\,\ldots,\,s,$$

$$\Sigma_1 = \{\mathbf{NF}(F,\, G_1 \vee G_2),\, F_1,\, R_1,\, F_2,\, R_2,\, \ldots,\, F_{i_1-1},\, R_{i_1-1}\},$$

$$\Sigma_j = \Sigma_{j-1} \cup \{R_{i_{(j-1)}},\, \ldots,\, F_{i_j-1},\, R_{i_j-1}\}, \quad j = 2,\,\ldots,\,p,$$

$$\Sigma = \Sigma_p \cup \{R_{i_p},\, \ldots,\, F_s,\, R_s\}, \quad R_s = G_1 \vee G_2.$$

Введем обозначения. Пусть $F = \mathbf{Sk}(C)$ и соответствующая (5) нормальная форма псевдоформулы C имеет вид (2). Пусть кванторная приставка (3) псевдоформулы $\mathbf{PNF}(C,\, D_1 \vee D_2)$ имеет вид (1). Пусть $Y_1, \ldots, Y_m$ суть глобальные метапеременные подстановки η. Пусть

$$(\Delta_n\Gamma'_n,\, A'_n),\ (\Delta_n^1\Gamma'_n,\, A'_n),\ \ldots,\ (\Delta_n^l\Gamma'_n,\, A'_n),\ (\Delta_n^{l+1}\Gamma'_n,\, A'_n),\ \ldots,\ (\Delta_n^{l'}\Gamma'_n,\, A'_n)$$

есть построенная так, как указано выше (11), соответствующая подстановкам $\tilde{\delta}_n$ и $\tilde{\nu}_n$, цепочка задач, в которой каждая следующая задача является результатом применения одного из правил декомпозиции (ПНФ$\to$), (ПФ$\to$), ($\forall \to$) или ($\exists \to$) к предыдущей. Положим

$$\theta'_{n+1} = \theta'_n \cup \tilde{\delta}_n = \theta'_n \cup \delta_n \cup \begin{pmatrix} Y_1 & Y_2 & \dots & Y_m \\ \tau'_1 & \tau'_2 & \dots & \tau'_m \end{pmatrix},$$

$$\mu_{n+1} = \mu_n \cup \tilde{\nu}_n, \qquad \Delta_{n+1} = \Delta_n^{l'}.$$

Введем обозначения: $\tilde{C} = \mathbf{PNF}(C,\, D_1 \vee D_2)$, $\tilde{F} = \mathbf{NF}(F,\, G_1 \vee G_2)\eta$, $\tilde{F}_i = F_i\eta$, $\tilde{R}_i = R_i\eta$, $\tilde{\Sigma}_j = \Sigma_j\eta$ и $\tilde{\Sigma} = \Sigma\eta$. Пусть псевдоформулы $\tilde{F}'$, $\tilde{F}'_1$, $\tilde{R}'_1$, $\dots$, $\tilde{F}'_s$, $\tilde{R}'_s$ соответствуют псевдоформулам $\tilde{F}$, $\tilde{F}_1$, $\tilde{R}_1$, $\dots$, $\tilde{F}_s$, $\tilde{R}_s$. Введем следующие обозначения:

$$\tilde{\Sigma}'_1 = \{\tilde{F}',\, \tilde{F}'_1,\, \tilde{R}'_1,\, \tilde{F}'_2,\, \tilde{R}'_2,\, \dots,\, \tilde{F}'_{i_1-1},\, \tilde{R}'_{i_1-1}\},$$

$$\tilde{\Sigma}'_j = \tilde{\Sigma}'_{j-1} \cup \{\tilde{R}'_{i_{(j-1)}},\, \dots,\, \tilde{F}'_{i_j-1},\, \tilde{R}'_{i_j-1}\}, \quad j = 2,\, \dots,\, p,$$

$$\tilde{\Sigma}' = \tilde{\Sigma}'_p \cup \{\tilde{R}'_{i_p},\, \dots,\, \tilde{F}'_s,\, \tilde{R}'_s\}, \quad \tilde{R}'_s = \tilde{G}'_1 \vee \tilde{G}'_2.$$

Следствием (10) являются следующие равенства:

$$\tilde{C}\theta \;=\; \tilde{C}'(\theta'_{n+1} \circ \mu_{n+1}), \qquad \tilde{F}\theta \;=\; \tilde{F}'(\theta'_{n+1} \circ \mu_{n+1}),$$

$$\tilde{F}_i\theta = \tilde{F}'_i(\theta'_{n+1} \circ \mu_{n+1}), \quad \tilde{R}_i\theta = \tilde{R}'_i(\theta'_{n+1} \circ \mu_{n+1}), \quad i = 1,\, \dots,\, s,$$

$$\tilde{\Sigma}_j\theta = \tilde{\Sigma}'_j(\theta'_{n+1} \circ \mu_{n+1}), \quad \tilde{\Sigma}\theta = \tilde{\Sigma}'(\theta'_{n+1} \circ \mu_{n+1}), \quad j = 1,\, \dots,\, p.$$

Дерево Θ'_{n+1} сформируем из дерева Θ'_n как результат нескольких применений сначала правил из списка (ПНФ$\to$), (ПФ$\to$), ($\forall \to$), ($\exists \to$), затем из списка ($\supset\to$), ($\& \to$), и в конце концов ($\vee \to$):

$$
\begin{array}{c}
\dfrac{(\tilde{G}'_1\,\tilde{\Sigma}'\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)\qquad(\tilde{G}'_2\,\tilde{\Sigma}'\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}{((\tilde{G}'_1\vee\tilde{G}'_2)\,\tilde{\Sigma}'\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}\\
\ddots\\
\nwarrow\\
\dfrac{(\tilde{\Sigma}'_p\,\Delta_n^{l'}\,\Gamma'_n,\,\tilde{F}'_{i_p})\qquad(\tilde{R}'_{i_p}\,\tilde{\Sigma}'_p\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}{(\tilde{\Sigma}'_p\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}\\
\ddots\\
\nwarrow\\
\dfrac{(\tilde{\Sigma}'_1\,\Delta_n^{l'}\,\Gamma'_n,\,\tilde{F}'_{i_1})\qquad(\tilde{R}'_{i_1}\,\tilde{\Sigma}'_1\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}{(\tilde{\Sigma}'_1\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}\\
\vdots\\
\dfrac{\dfrac{(\tilde{R}'_2\,\tilde{F}'_2\,\tilde{R}'_1\,\tilde{F}'_1\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}{(\tilde{R}'_1\,\tilde{F}'_1\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}}{(\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}\\
\vdots\\
\dfrac{(\Delta_n^1\,\Gamma'_n,\,A'_n)}{(\Delta_n\,\Gamma'_n,\,A'_n)}\\
\ddots\vdots\iddots\\
(\Gamma,\,A)
\end{array}
$$

В результате в дереве Θ'_{n+1} сформированы $p+2$ новых листа. При этом для каждого нового листа $(\mathbf{Sk}(\Gamma_{n+1})\,?\,A_{n+1})$ дерева Θ_{n+1} в дереве Θ'_{n+1} сформирован новый лист требуемого вида $(\Delta_{n+1}\Gamma'_{n+1},\,A'_{n+1})$, где $\Delta_{n+1}=\Delta_n^{l'}$.

9. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции (&?). В этом случае подстановка θ примитивизирует задачу $\pi_n=(\mathbf{Sk}(\Gamma_n)\,?\,A_n)$. Это означает наличие псевдоформулы $F=\mathbf{Sk}(C)\in\mathbf{Sk}(\Gamma_n)$ с нормальной формой вида

$$F_1\,\&\,(F_2\,\&\,(\ldots(F_s\,\&\,G)\ldots)$$

такой, что подстановка θ является унификатором A_n и $G\eta$. При этом η — подстановка, которая заменяет все локальные метапеременные в $\mathbf{NF}(F,\,G)$ на новые глобальные. Введем следующие обозначения:

$$R_i=F_{i+1}\,\&\,(F_{i+2}\,\&\,(\ldots(F_s\,\&\,G)\ldots),\quad i=1,\,\ldots,\,s;\qquad R_s=G.$$

Пусть

$$(\Delta_n\Gamma'_n,\,A'_n),\ (\Delta_n^1\Gamma'_n,\,A'_n),\ \ldots,\ (\Delta_n^l\Gamma'_n,\,A'_n),\ (\Delta_n^{l+1}\Gamma'_n,\,A'_n),\ \ldots,\ (\Delta_n^{l'}\Gamma'_n,\,A'_n)$$

есть построенная так, как указано выше (11), соответствующая подстановкам $\tilde{\delta}_n$ и $\tilde{\nu}_n$, цепочка задач, в которой каждая следующая задача является результатом применения одного из правил декомпозиции (ПНФ$\rightarrow$), (ПФ$\rightarrow$), ($\forall\rightarrow$) или ($\exists\rightarrow$) к предыдущей. Положим

$$\theta'_{n+1} = \theta'_n \cup \tilde{\delta}_n = \theta'_n \cup \delta_n \cup \begin{pmatrix} Y_1 & Y_2 & \dots & Y_m \\ \tau'_1 & \tau'_2 & \dots & \tau'_m \end{pmatrix},$$

$$\mu_{n+1} = \mu_n \cup \tilde{\nu}_n.$$

Введем обозначения: $\tilde{C} = \mathbf{PNF}(C, D)$, $\tilde{F} = \mathbf{NF}(F, G)\eta$, $\tilde{F}_i = F_i\eta$, $\tilde{R}_i = R_i\eta$, $\tilde{\Sigma}_j = \Sigma_j\eta$, $\tilde{\Sigma} = \Sigma\eta$ и $\tilde{G} = G\eta$. Пусть псевдоформулы $\tilde{F}'$, $\tilde{F}'_1$, $\tilde{R}'_1$, $\dots$, $\tilde{F}'_s$, $\tilde{R}'_s$ соответствуют псевдоформулам $\tilde{F}$, $\tilde{F}_1$, $\tilde{R}_1$, $\dots$, $\tilde{F}_s$, $\tilde{R}_s$. Следствием (10) являются следующие равенства:

$$\tilde{C}\theta = \tilde{C}'(\theta'_{n+1} \circ \mu_{n+1}), \qquad \tilde{F}\theta = \tilde{F}'(\theta'_{n+1} \circ \mu_{n+1}),$$

$$\tilde{F}_i\theta = \tilde{F}'_i(\theta'_{n+1} \circ \mu_{n+1}), \quad \tilde{R}_i\theta = \tilde{R}'_i(\theta'_{n+1} \circ \mu_{n+1}), \quad i = 1, \dots, s.$$

Таким образом имеем

$$A'_n(\theta'_{n+1}\circ\mu_{n+1}) = A'_n(\theta'_n\circ\mu_n) = A_n\theta = (G\eta)\theta = \tilde{G}\theta = \tilde{R}_s\theta = \tilde{R}'_s(\theta'_{n+1}\circ\mu_{n+1}),$$

$$A'_n\theta'_{n+1} = \tilde{R}'_s\theta'_{n+1} = \tilde{G}'\theta'_{n+1}.$$

Дерево Θ'_{n+1} сформируем из дерева Θ'_n как результат нескольких применений сначала правил из списка (ПНФ$\rightarrow$), (ПФ$\rightarrow$), ($\forall\rightarrow$), ($\exists\rightarrow$), а затем ($\&\rightarrow$):

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}(\tilde{R}'_s\,\tilde{F}'_s\,\dots\,\tilde{R}'_2\,\tilde{F}'_2\,\tilde{R}'_1\,\tilde{F}'_1\,\Delta_n^{l'}\,\Gamma'_n,\ A'_n)\\ \vdots \\ (\tilde{R}'_2\,\tilde{F}'_2\,\tilde{R}'_1\,\tilde{F}'_1\,\Delta_n^{l'}\,\Gamma'_n,\ A'_n)\end{array}}{(\tilde{R}'_1\,\tilde{F}'_1\,\Delta_n^{l'}\,\Gamma'_n,\ A'_n)}}{\begin{array}{c}(\Delta_n^{l'}\,\Gamma'_n,\ A'_n)\\ \vdots \\ (\Delta_n^{1}\,\Gamma'_n,\ A'_n)\end{array}}}{\begin{array}{c}(\Delta_n\,\Gamma'_n,\ A'_n)\\ \ddots\vdots\cdot^{\cdot^{\cdot}} \\ (\Gamma,\ A)\end{array}}$$

В результате в дереве Θ'_{n+1} сформирован лист, который примитивизируется подстановкой θ'_{n+1}.

10. Пусть Θ_{n+1} получается из Θ_n по правилу $(\supset ?)$. Тогда существует псевдоформула $F \in \mathbf{Sk}(\Gamma_n)$ с нормальной формой вида

$$F_1 \, \lambda_1 \, (F_2 \, \lambda_2 \, (\ldots (F_s \, \lambda_s \, G) \ldots)$$

такой, что p задач

$$((\Sigma_1 \xi) \, \mathbf{Sk}(\Gamma_n) \, ? \, F_{i_1} \xi), \ \ldots, \ ((\Sigma_p \xi) \, \mathbf{Sk}(\Gamma_n) \, ? \, F_{i_p} \xi)$$

образуют у задачи π_n все дочерние вершины в дереве Θ. При этом псевдоформула $F_{i_1} \,\&\, F_{i_2} \,\&\, \ldots \,\&\, F_{i_p}$ есть импликанта этой нормальной формы, подстановка η замещает все локальные метапеременные в F новыми глобальными, $\xi = \eta \circ \sigma$, $\sigma = \mathbf{MGU}(A_n, \, G\eta)$, подстановка θ совместима с σ и также является унификатором A_n и $G\eta$. Кроме того, имеем

$$R_i = F_{i+1} \, \lambda_{i+1} \, (F_{i+2} \, \lambda_{i+2} \, (\ldots (F_s \, \lambda_s \, G) \ldots), \quad i = 1, \, \ldots, \, s,$$

$$\Sigma_1 = \{\mathbf{NF}(F, \, G), \, F_1, \, R_1, \, F_2, \, R_2, \, \ldots, \, F_{i_1 - 1}, \, R_{i_1 - 1}\},$$

$$\Sigma_j = \Sigma_{j-1} \cup \{R_{i_{(j-1)}}, \, \ldots, \, F_{i_j - 1}, \, R_{i_j - 1}\}, \quad j = 2, \, \ldots, \, p,$$

$$\Sigma = \Sigma_p \cup \{R_{i_p}, \, \ldots, \, F_s, \, R_s\}, \quad R_s = G.$$

Введем обозначения. Пусть $F = \mathbf{Sk}(C)$ и соответствующая (5) нормальная форма псевдоформулы C имеет вид (2). Пусть кванторная приставка (3) псевдоформулы $\mathbf{PNF}(C, \, D)$ имеет вид (1). Пусть $Y_1, \, \ldots, \, Y_m$ суть глобальные метапеременные подстановки η. Пусть

$$(\Delta_n \Gamma'_n, \, A'_n), \, (\Delta_n^1 \Gamma'_n, \, A'_n), \, \ldots, \, (\Delta_n^l \Gamma'_n, \, A'_n), \, (\Delta_n^{l+1} \Gamma'_n, \, A'_n), \, \ldots, \, (\Delta_n^{l'} \Gamma'_n, \, A'_n)$$

есть построенная так, как указано выше (11), соответствующая подстановкам $\tilde{\delta}_n$ и $\tilde{\nu}_n$, цепочка задач, в которой каждая следующая задача является результатом применения одного из правил декомпозиции (ПНФ$\to$), (ПФ$\to$), $(\forall \to)$ или $(\exists \to)$ к предыдущей. Положим

$$\theta'_{n+1} = \theta'_n \cup \tilde{\delta}_n = \theta'_n \cup \delta_n \cup \begin{pmatrix} Y_1 & Y_2 & \ldots & Y_m \\ \tau'_1 & \tau'_2 & \ldots & \tau'_m \end{pmatrix},$$

$$\mu_{n+1} = \mu_n \cup \tilde{\nu}_n, \quad \Delta_{n+1} = \Delta_n^{l'}.$$

Как было установлено ранее (10), для псевдоформул $\tilde{C} = \mathbf{PNF}(C, \, D)$ и $\tilde{F} = \mathbf{NF}(F, \, G)\eta$ выполняются равенства

$$\tilde{C}\theta \; = \; \tilde{C}'(\theta'_{n+1} \circ \mu_{n+1}), \quad \tilde{F}\theta \; = \; \tilde{F}'(\theta'_{n+1} \circ \mu_{n+1}).$$

Кроме того, имеем следующие равенства: $\theta = \sigma \circ \theta$,

$$(\mathbf{NF}(F, G)\xi)\theta = (\mathbf{NF}(F, G)(\eta \circ \sigma))\theta = (\mathbf{NF}(F, G)\eta)(\sigma \circ \theta)$$
$$= (\mathbf{NF}(F, G)\eta)\theta = \tilde{F}\theta.$$

Введем обозначения: $\tilde{F}_i = F_i\xi$, $\tilde{R}_i = R_i\xi$, $\tilde{\Sigma}_j = \Sigma_j\xi$, $\tilde{\Sigma} = \Sigma\xi$, $\tilde{G} = G\xi$, $\tilde{F}_0 = \mathbf{NF}(F, G)\xi$. Пусть псевдоформулы $\tilde{F}'_0$, $\tilde{F}'_1$, $\tilde{R}'_1$, $\ldots$, $\tilde{F}'_s$, $\tilde{R}'_s$ соответствуют псевдоформулам $\tilde{F}_0$, $\tilde{F}_1$, $\tilde{R}_1$, $\ldots$, $\tilde{F}_s$, $\tilde{R}_s$. Введем следующие обозначения

$$\tilde{\Sigma}'_1 = \{\tilde{F}'_0, \tilde{F}'_1, \tilde{R}'_1, \tilde{F}'_2, \tilde{R}'_2, \ldots, \tilde{F}'_{i_1-1}, \tilde{R}'_{i_1-1}\},$$

$$\tilde{\Sigma}'_j = \tilde{\Sigma}'_{j-1} \cup \{\tilde{R}'_{i_{(j-1)}}, \ldots, \tilde{F}'_{i_j-1}, \tilde{R}'_{i_j-1}\}, \quad j = 2, \ldots, p,$$

$$\tilde{\Sigma}' = \tilde{\Sigma}'_p \cup \{\tilde{R}'_{i_p}, \ldots, \tilde{F}'_s, \tilde{R}'_s\}, \quad \tilde{R}'_s = \tilde{G}'.$$

Для перечисленных псевдовыражений выполняются следующие равенства

$$\tilde{F}_i\theta = \tilde{F}'_i(\theta'_{n+1} \circ \mu_{n+1}), \quad \tilde{R}_i\theta = \tilde{R}'_i(\theta'_{n+1} \circ \mu_{n+1}), \quad i = 1, \ldots, s,$$

$$\tilde{\Sigma}_j\theta = \tilde{\Sigma}'_j(\theta'_{n+1} \circ \mu_{n+1}), \quad \tilde{\Sigma}\theta = \tilde{\Sigma}'(\theta'_{n+1} \circ \mu_{n+1}), \quad j = 1, \ldots, p.$$

Таким образом имеем

$$A'_n(\theta'_{n+1} \circ \mu_{n+1}) = A'_n(\theta'_n \circ \mu_n) = A_n\theta = (G\eta)\theta = (G\xi)\theta$$
$$= \tilde{G}\theta = \tilde{G}'(\theta'_{n+1} \circ \mu_{n+1}),$$

$$A'_n\theta'_{n+1} = \tilde{R}'_s\theta'_{n+1} = \tilde{G}'\theta'_{n+1}.$$

Дерево Θ'_{n+1} сформируем из дерева Θ'_n как результат нескольких применений сначала правил из списка (ПНФ$\rightarrow$), (ПФ$\rightarrow$), ($\forall \rightarrow$), ($\exists \rightarrow$), а затем из списка ($\supset\rightarrow$), ($\& \rightarrow$):

$$
\begin{array}{c}
(\tilde{G}'\,\tilde{\Sigma}'\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)\\
\ddots\\
\nwarrow\\
\dfrac{(\tilde{\Sigma}'_p\,\Delta_n^{l'}\,\Gamma'_n,\,\tilde{F}'_{i_p})\qquad(\tilde{R}'_{i_p}\,\tilde{\Sigma}'_p\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}{(\tilde{\Sigma}'_p\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}\\
\ddots\\
\nwarrow\\
\dfrac{(\tilde{\Sigma}'_1\,\Delta_n^{l'}\,\Gamma'_n,\,\tilde{F}'_{i_1})\qquad(\tilde{R}'_{i_1}\,\tilde{\Sigma}'_1\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}{(\tilde{\Sigma}'_1\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}\\
\vdots\\
\dfrac{\dfrac{(\tilde{R}'_2\,\tilde{F}'_2\,\tilde{R}'_1\,\tilde{F}'_1\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}{(\tilde{R}'_1\,\tilde{F}'_1\,\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}}{(\Delta_n^{l'}\,\Gamma'_n,\,A'_n)}\\
\vdots\\
\dfrac{(\Delta_n^{1}\,\Gamma'_n,\,A'_n)}{(\Delta_n\,\Gamma'_n,\,A'_n)}\\
\ddots\vdots\iddots\\
(\Gamma,\,A)
\end{array}
$$

В результате в дереве Θ'_{n+1} сформирован $p+1$ лист, один из которых примитивизируется подстановкой θ'_{n+1}, а остальные p листьев соответствуют новым p листьям дерева Θ_{n+1} и имеют требуемый вид. Тем самым рассмотрение случая $(\supset ?)$ завершено.

Для построения Θ_{n+1}, а затем θ'_{n+1} и Θ'_{n+1}, в соответствии с применяемой стратегией мы выбирали лист $\pi_n = (\mathbf{Sk}(\Gamma_n)?A_n) \in \Theta_n$, у которого порождали дочерние разрешимые вершины, соответствующие подстановке θ и обеспечивающие разрешимость вершины π_n. В результате была сформирована подстановка θ'_{n+1} вида $\theta'_{n+1} = \theta'_n \cup \delta_n \cup \beta$, а также было сформировано дерево Θ'_{n+1}, которое получается из дерева Θ'_n в результате замены листа $\pi'_n = (\Delta_n\Gamma'_n,\, A'_n)$, соответствующего листу π_n дерева Θ_n, на некоторое новое поддерево с корнем π'_n. При этом каждая вершина ρ полученного дерева Θ'_{n+1} обладает следующим свойством: во-первых, если ρ является листом дерева Θ'_{n+1}, который не соответствует нераскрытому листу дерева Θ_{n+1}, то $\rho\theta'_{n+1}$ является примитивной задачей; во-вторых, если θ'_{n+1} является допустимой для вершин $\rho_1, \ldots, \rho_k$, всех дочерних у вершины ρ в дереве Θ'_{n+1}, то θ'_{n+1} является также допустимой и для ρ. На заключительном шаге этого процесса мы получаем дерево $\Theta' = \Theta'_h$ и подстановку $\theta' = \theta'_h$ такие, что θ' является допустимой подстановкой задачи (Γ, A) и Θ' является соответствующим этой подстановке θ' деривационным деревом. Предложение 4.2 доказано. $\square$

Теорема о корректности. *Если задача* $(\mathbf{Sk}(\Gamma) \mathbin{?} A)$ *разрешима в* $\mathcal{F}_3$*, то секвенция* $\Gamma \to A$ *выводима в* $\mathcal{S}_3$*.*

Доказательство. Пусть задача $(\mathbf{Sk}(\Gamma) \mathbin{?} A)$ разрешима в $\mathcal{F}_3$. По предложению 4.2 отсюда задача (Γ, A) разрешима в $\mathcal{P}_3$. По теореме о корректности $\mathcal{P}_3$ тогда секвенция $\Gamma \to A$ выводима в $\mathcal{S}_3$. Что и требовалось доказать. $\square$

Доказательство корректности содержит в себе эффективный алгоритм получения вывода в односукцедентном исчислении секвенций из деривационного поддерева поискового дерева. В то же время существует эффективный алгоритм получения натурального вывода из односукцедентного секвенциального вывода. Все это вместе дает обоснование корректности сформулированной в виде продукционной системы процедуры поиска натурального вывода. Полнота этого метода поиска вывода рассматривается в следующем разделе.

5. Теорема о полноте

Формулируемая в этом разделе теорема о полноте обеспечивает для всякой выводимой в исчислении $\mathcal{S}_3$ секвенции вычисление ее логического вывода в рамках продукционной системы $\mathcal{F}_3$. Будем обозначать через $\overline{\mathcal{P}}_3$ продукционную систему, которая получается из $\mathcal{P}_3$ в результате обобщения правил декомпозиции системы $\mathcal{P}_3$ на произвольную псевдоформулу C в сукцеденте. Нам удобно установить полноту $\mathcal{F}_3$ в $\overline{\mathcal{P}}_3$. Это влечет полноту $\mathcal{F}_3$ в $\mathcal{P}_3$. Прежде всего мы должны установить справедливость нескольких формулируемых ниже предложений, в которых устанавливается допустимость в $\mathcal{F}_3$ правил декомпозиции $\overline{\mathcal{P}}_3$.

Предложение 5.1. *Если в* $\mathcal{F}_3$ *разрешима задача* $\mathbf{Sk}(A)\,\mathbf{Sk}(B)\,\Gamma?G$ *и* θ *— ее допустимая подстановка, то задача* $\mathbf{Sk}(A \mathbin{\&} B)\,\Gamma?G$ *также является разрешимой и* θ *является ее допустимой подстановкой.*

Доказательство. Утверждение докажем индукцией по высоте деривационного дерева разрешимой задачи $\mathbf{Sk}(A)\,\mathbf{Sk}(B)\,\Gamma?G$. Во всех случаях строения этого деривационного дерева рассуждения имеют подобный характер. В связи с этим рассмотрим лишь три случая. Остальные подобны рассмотренным.

1. Рассмотрим случай, когда задача $\mathbf{Sk}(A)\,\mathbf{Sk}(B)\,\Gamma?G$ является примитивизируемой и θ — ее примитивизирующая подстановка. Будем считать, что $\theta = \mathbf{MGU}(G,\, G'\eta)$, где $G' = C_m$, $\mathbf{NF}(C,\, G') = \&_{i=1}^{i=m} C_i$ и подстановка η замещает все локальные метапеременные в C на новые глобальные. Тогда $C \in \Gamma$, $C = \mathbf{Sk}(A)$ или $C = \mathbf{Sk}(B)$. Если $C \in \Gamma$, то задача $\mathbf{Sk}(A \,\&\, B)\,\Gamma?G$ также примитивизируется подстановкой θ. Если же $C = \mathbf{Sk}(A)$, то $\mathbf{Sk}(A \,\&\, B) = \mathbf{Sk}(A) \,\&\, \mathbf{Sk}(B)$, $\mathbf{NF}(\mathbf{Sk}(A \,\&\, B),\, G') = \&_{i=0}^{i=m} C_i = C_0 \,\&\, \mathbf{NF}(C,\, G')$, где $C_0 = \mathbf{Sk}(B)$. Таким образом подстановка θ будет также примитивизировать задачу $\mathbf{Sk}(A \,\&\, B)\Gamma?G$. Аналогично рассматривается случай $C = \mathbf{Sk}(B)$.

2. Рассмотрим случай, когда задача $\mathbf{Sk}(A)\,\mathbf{Sk}(B)\,\Gamma?G$ получается в деривационном дереве по правилу $(\supset?)$ из подзадач

$$\mathbf{Sk}(A)\,\mathbf{Sk}(B)\,(\Delta_1\xi)\,\Gamma\,?\,C_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(A)\,\mathbf{Sk}(B)\,(\Delta_k\xi)\,\Gamma\,?\,C_{i_k}\xi,$$

где $C_1\lambda_1(C_2\lambda_2(\ldots(C_n\lambda_n G'))\ldots)$ — нормальная форма псевдоформулы C, $C_{i_1} \,\&\, C_{i_2} \,\&\, \ldots,\, C_{i_k}$ — ее импликанта, $\xi = \eta \circ \sigma$ и θ — комбинация σ, θ_1, $\ldots$, θ_k. Здесь следует рассмотреть несколько подслучаев.

(а) Случай $C \in \Gamma$. По индуктивному предположению тогда разрешимы задачи

$$\mathbf{Sk}(A \,\&\, B)\,(\Delta_1\xi)\,\Gamma\,?\,C_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(A \,\&\, B)\,(\Delta_k\xi)\,\Gamma\,?\,C_{i_k}\xi \qquad (12)$$

и θ_1, $\ldots$, θ_k являются их допустимыми подстановками. По правилу $(\supset ?)$ тогда разрешима задача $\mathbf{Sk}(A \,\&\, B)\,\Gamma?G$ и θ является ее допустимой подстановкой.

(б) Теперь пусть $C = \mathbf{Sk}(B)$. По индуктивному предположению тогда разрешимы задачи (12) и θ_1, $\ldots$, θ_k суть их допустимые подстановки. Тогда разрешимы задачи

$$\mathbf{Sk}(A \,\&\, B)\,(\tilde{\Delta}_1\xi)\,\Gamma\,?\,C_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(A \,\&\, B)\,(\tilde{\Delta}_k\xi)\,\Gamma\,?\,C_{i_k}\xi,$$

где $\tilde{\Delta}_i = \Delta_i \cup \{\tilde{C},\, C_0\}$, $\tilde{C} = \mathbf{NF}(\mathbf{Sk}(A\&B),\, G')$, $C_0 = \mathbf{Sk}(A)$. По правилу $(\supset?)$ тогда также разрешима задача $\mathbf{Sk}(A \,\&\, B)\,\Gamma?G$, у которой θ является допустимой подстановкой.

(в) Случай $C = \mathbf{Sk}(A)$ рассматривается аналогично случаю (б).

3. Рассмотрим случай, когда $G = C_1 \vee C_2$ и задача $\mathbf{Sk}(A)\,\mathbf{Sk}(B)\,\Gamma?G$ получается в деривационном дереве по правилу $(?\vee_i)$ из $\mathbf{Sk}(A)\,\mathbf{Sk}(B)\,\Gamma?C_i$.

По индуктивному предположению тогда должна быть разрешима задача $\mathbf{Sk}(A \mathbin{\&} B)\,\Gamma?C_i$, у которой θ будет допустимой подстановкой. По правилу $(?\vee_i)$ тогда разрешима задача $\mathbf{Sk}(A \mathbin{\&} B)\,\Gamma?G$ и θ является ее допустимой подстановкой. $\square$

Предложение 5.2. *Если задача $\mathbf{Sk}(A)\,\mathbf{Sk}(B)\,\mathbf{Sk}(A \mathbin{\&} B)\,\Gamma?G$ разрешима в $\mathcal{F}_3$ и θ — ее допустимая подстановка, то задача $\mathbf{Sk}(A \mathbin{\&} B)\,\Gamma?G$ также является разрешимой и θ является ее допустимой подстановкой.*

Доказательство. Пусть разрешима задача $\mathbf{Sk}(A)\,\mathbf{Sk}(B)\,\mathbf{Sk}(A \mathbin{\&} B)\,\Gamma?G$ и θ — ее допустимая подстановка. По предложению 5.1 тогда должна быть также разрешима задача $\mathbf{Sk}(A \mathbin{\&} B)\,\mathbf{Sk}(A \mathbin{\&} B)\,\Gamma?G$ и θ будет ее допустимой подстановкой. Что и требовалось доказать. $\square$

Предложение 5.3. *Если $\mathbf{Sk}(A \supset B)\Gamma?A$ и $\mathbf{Sk}(B)\,\mathbf{Sk}(A \supset B)\,\Gamma?G$ суть разрешимые задачи в $\mathcal{F}_3$, у которых существуют совместимые допустимые подстановки θ_1 и θ_2, то задача $\mathbf{Sk}(A \supset B)\Gamma?G$ также является разрешимой и комбинация θ_1 и θ_2 является ее допустимой подстановкой.*

Доказательство. Утверждение докажем индукцией по высоте деривационного дерева разрешимой задачи $\mathbf{Sk}(B)\,\mathbf{Sk}(A \supset B)\,\Gamma?G$. Разбор случаев имеет шаблонный характер. Рассмотрим только два случая строения деривационного дерева.

1. Рассмотрим случай, когда задача $\mathbf{Sk}(B)\,\mathbf{Sk}(A \supset B)\,\Gamma?G$ является примитивизируемой и θ_2 — ее примитивизирующая подстановка. Будем считать, что $\theta_2 = \mathbf{MGU}(G, G'\eta)$, где $G' = C_m$, $\mathbf{NF}(C, G') = \&_{i=1}^{i=m} C_i$ и подстановка η замещает все локальные метапеременные в C на новые глобальные. Тогда $C \in \Gamma$ или $C = \mathbf{Sk}(B)$. Если $C \in \Gamma$, то задача $\mathbf{Sk}(A \supset B)\,\Gamma?G$ также примитивизируется подстановкой θ_2 и, следовательно, примитивизируется комбинацией подстановок θ_1 и θ_2. Если же $C = \mathbf{Sk}(B)$, то $\mathbf{Sk}(A \supset B) = A \supset \mathbf{Sk}(B) = A \supset C$, $\mathbf{NF}(A \supset C, G') = A \supset \&_{i=1}^{i=m} C_i$, $A\eta = A$, $(A \supset \&_{i=1}^{i=m} C_i)\eta = A \supset \&_{i=1}^{i=m} (C_i\eta)$ и комбинация подстановок θ_1 и θ_2 будет не только допустимой подстановкой для задачи $\mathbf{Sk}(A \supset B)\Gamma?A\theta_2$, но по правилу $(\supset?)$ будет также допустимой подстановкой для задачи $\mathbf{Sk}(A \supset B)\Gamma?G$.

2. Рассмотрим случай, когда задача $\mathbf{Sk}(B)\,\mathbf{Sk}(A \supset B)\,\Gamma?G$ получается в деривационном дереве по правилу $(\supset?)$ из подзадач

$$\mathbf{Sk}(B)\,\mathbf{Sk}(A \supset B)\,(\Delta_1\xi)\,\Gamma\,?\,C_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(B)\,\mathbf{Sk}(A \supset B)\,(\Delta_k\xi)\,\Gamma\,?\,C_{i_k}\xi,$$

где $C_{i_1} \,\&\, C_{i_2} \,\&\, \ldots,\, C_{i_k}$ — импликанта $\mathbf{NF}(C,\, G')$, $\xi = \eta \circ \sigma$ и θ_2 — комбинация подстановок $\sigma,\, \beta_1,\, \ldots,\, \beta_k$. Здесь возникают два варианта.

(а) Случай $C \in \Gamma$ или $C = \mathbf{Sk}(A \supset B)$. По индуктивному предположению тогда разрешимы задачи

$$\mathbf{Sk}(A \supset B)\,(\Delta_1\xi)\,\Gamma\,?\,C_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(A \supset B)\,(\Delta_k\xi)\,\Gamma\,?\,C_{i_k}\xi. \qquad (13)$$

По правилу ($\supset$?) тогда разрешима задача $\mathbf{Sk}(A \supset B)\,\Gamma?G$ и комбинация подстановок $\theta_1,\, \sigma,\, \beta_1,\, \ldots,\, \beta_k$, которая совпадает с комбинацией θ_1 и θ_2, является ее допустимой подстановкой.

(б) Случай $C = \mathbf{Sk}(B)$. По индуктивному предположению тогда разрешимы задачи (13). Отметим, что $\mathbf{Sk}(A \supset B) = A \supset \mathbf{Sk}(B) = A \supset C$, $A\eta = A$, $A\xi = A\sigma$, а также комбинация θ_1 и θ_2 является допустимой подстановкой для $\mathbf{Sk}(A \supset B)\,\Gamma?A\xi$. Тогда разрешимы задачи

$$(A \supset \mathbf{Sk}(B))\,\Gamma?A\xi,\ (A \supset \mathbf{Sk}(B))\,(\tilde{\Delta}_1\xi)\Gamma?C_{i_1}\xi,\ \ldots,\ (A \supset \mathbf{Sk}(B))\,(\tilde{\Delta}_k\xi)\Gamma?C_{i_k}\xi,$$

где $\tilde{\Delta}_i = \Delta_i \cup \{\tilde{C},\, C_0\}$, $\tilde{C} = \mathbf{NF}(A \supset \mathbf{Sk}(B),\, G')$, $C_0 = A$. По правилу ($\supset$?) тогда разрешима задача $\mathbf{Sk}(A \supset B)\,\Gamma?G$, у которой комбинация θ_1 и θ_2 является ее допустимой подстановкой. □

Предложение 5.4. *Если задача* $\mathbf{Sk}(A(X))\,\mathbf{Sk}(\forall x\, A(x))\,\Gamma?G$ *разрешима в* $\mathcal{F}_3$ *и* θ *— ее допустимая подстановка, где* X *— новая глобальная метапеременная, то разрешима также задача* $\mathbf{Sk}(\forall x\, A(x))\,\Gamma?G$ *и* θ *является ее допустимой подстановкой.*

Доказательство. Утверждение докажем индукцией по высоте деривационного дерева разрешимой задачи $\mathbf{Sk}(A(X))\,\mathbf{Sk}(\forall x\, A(x))\,\Gamma?G$. Мы анализируем только два случая строения этого деривационного дерева. Остальные подобны рассмотренным.

1. Рассмотрим случай, когда задача $\mathbf{Sk}(A(X))\,\mathbf{Sk}(\forall x\, A(x))\,\Gamma?G$ является примитивизируемой и θ — ее примитивизирующая подстановка. Будем считать, что $\theta = \mathbf{MGU}(G,\, G'\eta)$, где $G' = C_m$, $\mathbf{NF}(C,\, G') = \&_{i=1}^{i=m} C_i$ и подстановка η замещает все локальные метапеременные в C на новые глобальные. Тогда $C \in \Gamma$, $C = \mathbf{Sk}(A(X))$ или $C = \mathbf{Sk}(\forall x\, A(x))$. Если $C \in \Gamma$ или $C = \mathbf{Sk}(\forall x\, A(x))$, то задача $\mathbf{Sk}(\forall x\, A(x))\,\Gamma?G$ также примитивизируется подстановкой θ. Теперь пусть $C = \mathbf{Sk}(A(X))$ и при скулемизации $\forall x\, A(x)$ вместо предметной переменной x вводится локальная метапеременная Z. Положим $\eta_1 = \eta \circ \begin{pmatrix} Z \\ X \end{pmatrix}$. Подстановка η_1 замещает все локальные метапеременные в псевдоформуле $C\begin{pmatrix} X \\ Z \end{pmatrix}$, кото-

рая совпадает с $\mathbf{Sk}(\forall x\, A(x))$, на новые глобальные и $C\eta = C\begin{pmatrix} X \\ Z \end{pmatrix}\eta_1$. Таким образом подстановка θ будет также примитивизировать задачу $\mathbf{Sk}(\forall x\, A(x))\Gamma?G$.

2. Рассмотрим случай, когда задача $\mathbf{Sk}(A(X))\,\mathbf{Sk}(\forall x\, A(x))\,\Gamma?G$ получается в деривационном дереве по правилу $(\supset ?)$ из подзадач

$$\mathbf{Sk}(A(X))\,\mathbf{Sk}(\forall x\, A(x))\,(\Delta_1\xi)\,\Gamma\,?\,C_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(A(X))\,\mathbf{Sk}(\forall x\, A(x))\,(\Delta_k\xi)\,\Gamma\,?\,C_{i_k}\xi,$$

где $C_{i_1}\,\&\,C_{i_2}\,\&\,\ldots,\,C_{i_k}$ — импликанта нормальной формы псевдоформулы C, $\xi = \eta\circ\sigma$ и θ — комбинация подстановок σ, β_1, $\ldots$, β_k. Здесь следует рассмотреть два подслучая.

(а) Случай $C \in \Gamma$ или $C = \mathbf{Sk}(\forall x\, A(x))$. По индуктивному предположению тогда разрешимы задачи

$$\mathbf{Sk}(\forall x\, A(x))\,(\Delta_1\xi)\,\Gamma\,?\,C_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(\forall x\, A(x))\,(\Delta_k\xi)\,\Gamma\,?\,C_{i_k}\xi \tag{14}$$

и соответственно β_1, $\ldots$, β_k являются их допустимыми подстановками. По правилу $(\supset ?)$ тогда разрешима задача $\mathbf{Sk}(\forall x\, A(x))\,\Gamma?G$ и θ является ее допустимой подстановкой.

(б) Пусть $C = \mathbf{Sk}(A(X))$ и при скулемизации $\forall x\, A(x)$ вместо предметной переменной x вводится локальная метапеременная Z. Положим $\eta_1 = \eta\circ\begin{pmatrix} Z \\ X \end{pmatrix}$. Тогда подстановка η_1 замещает все локальные метапеременные в псевдоформуле $C\begin{pmatrix} X \\ Z \end{pmatrix}$, которая совпадает с $\mathbf{Sk}(\forall x\, A(x))$, на новые глобальные. Тогда ξ есть $\mathbf{MGU}\left(G,\, G'\begin{pmatrix} X \\ Z \end{pmatrix}\eta_1\right)$. По индуктивному предположению тогда разрешимы задачи (14) и соответственно β_1, $\ldots$, β_k являются их допустимыми подстановками. По правилу $(\supset ?)$ тогда из разрешимости этих задач следует разрешимость задачи $\mathbf{Sk}(\forall x\, A(x))\,\Gamma?G$, у которой θ является допустимой подстановкой. □

Предложение 5.5. *Если задача* $\mathbf{Sk}(A(y))\,\mathbf{Sk}(\exists x\, A(x))\,\Gamma?G$ *разрешима в* $\mathcal{F}_3$ *и* θ *является ее допустимой подстановкой, где псевдоформулы* $\Gamma\cup\{(\exists x\, A(x)),\, G\}$ *не содержат свободно переменной* y, *то задача* $\mathbf{Sk}(\exists x\, A(x))\,\Gamma?G$ *также является разрешимой и* θ *является ее допустимой подстановкой.*

Доказательство. Утверждение докажем индукцией по высоте деривационного дерева разрешимой задачи $\mathbf{Sk}(A(y))\,\mathbf{Sk}(\exists x\, A(x))\,\Gamma?G$. Во всех случаях рассуждения носят шаблонный характер. Рассмотрим лишь два случаи строения деривационного дерева. Остальные аналогичны рассмотренным.

1. Рассмотрим случай, когда задача $\mathbf{Sk}(A(y))\,\mathbf{Sk}(\exists x\, A(x))\,\Gamma ?G$ является примитивизируемой и θ — ее примитивизирующая подстановка. Будем считать, что $\theta = \mathbf{MGU}(G,\, G'\eta)$, где $G' = C_m$, $\mathbf{NF}(C,\, G') = \&_{i=1}^{i=m} C_i$ и подстановка η замещает все локальные метапеременные в C на новые глобальные. Тогда $C \in \Gamma$, $C = \mathbf{Sk}(A(y))$ или $C = \mathbf{Sk}(\exists x\, A(x))$. Если $C \in \Gamma$ или $C = \mathbf{Sk}(\exists x\, A(x))$, то задача $\mathbf{Sk}(\exists x\, A(x))\,\Gamma ?G$ также примитивизируется подстановкой θ. Теперь пусть $C = \mathbf{Sk}(A(y))$. Тогда при скулемизации $\exists x\, A(x)$ для элиминации кванторной приставки $\exists x$ возьмем в качестве скулемовского символа y. Таким образом псевдоформула C будет совпадать с $\mathbf{Sk}(\exists x\, A(x))$ и подстановка θ будет также примитивизировать задачу $\mathbf{Sk}(\exists x\, A(x))\Gamma ?G$.

2. Рассмотрим случай, когда задача $\mathbf{Sk}(A(y))\,\mathbf{Sk}(\exists x\, A(x))\,\Gamma ?G$ получается в деривационном дереве по правилу $(\supset ?)$ из подзадач

$$\mathbf{Sk}(A(y))\,\mathbf{Sk}(\exists x\, A(x))\,(\Delta_1\xi)\,\Gamma\,?\,C_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(A(y))\,\mathbf{Sk}(\exists x\, A(x))\,(\Delta_k\xi)\,\Gamma\,?\,C_{i_k}\xi,$$

где $C_{i_1} \,\&\, C_{i_2} \,\&\, \ldots,\, C_{i_k}$ — импликанта нормальной формы псевдоформулы C, $\xi = \eta \circ \sigma$ и θ — комбинация подстановок σ, $\beta_1,\, \ldots,\, \beta_k$. Здесь существует два возможных варианта.

(a) Случай $C \in \Gamma$ или $C = \mathbf{Sk}(\exists x\, A(x))$. По индуктивному предположению тогда разрешимы задачи

$$\mathbf{Sk}(\exists x\, A(x))\,(\Delta_1\xi)\,\Gamma\,?\,C_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(\exists x\, A(x))\,(\Delta_k\xi)\,\Gamma\,?\,C_{i_k}\xi \tag{15}$$

и соответственно $\beta_1,\, \ldots,\, \beta_k$ являются их допустимыми подстановками. По правилу $(\supset ?)$ тогда разрешима задача $\mathbf{Sk}(\exists x\, A(x))\,\Gamma ?G$ и θ является ее допустимой подстановкой.

(б) Теперь пусть $C = \mathbf{Sk}(A(y))$. Тогда при скулемизации $\exists x\, A(x)$ для элиминации кванторной приставки $\exists x$ в качестве скулемовского символа возьмем y. Таким образом псевдоформула C будет совпадать с $\mathbf{Sk}(\exists x\, A(x))$. Тогда разрешимы задачи (15) и соответственно $\beta_1,\, \ldots,\, \beta_k$ являются их допустимыми подстановками. По правилу $(\supset ?)$ тогда также разрешима задача $\mathbf{Sk}(\exists x\, A(x))\,\Gamma ?G$, у которой θ является допустимой подстановкой. □

Предложение 5.6. *Если* $\mathbf{Sk}(A)\,\mathbf{Sk}(A \vee B)\,\Gamma ?C$ *и* $\mathbf{Sk}(B)\,\mathbf{Sk}(A \vee B)\,\Gamma ?C$ *суть две разрешимые в* $\mathcal{F}_3$ *задачи, у которых существуют совместимые допустимые подстановки* θ_1 *и* θ_2, *то разрешима также задача*

$\mathbf{Sk}(A \vee B)\,\Gamma ?C$ *и комбинация* θ_1 *и* θ_2 *является ее допустимой подстановкой.*

Доказательство. Пусть $\mathbf{Sk}(A)\,\mathbf{Sk}(A \vee B)\,\Gamma ?C$ и $\mathbf{Sk}(B)\,\mathbf{Sk}(A \vee B)\,\Gamma ?C$ суть две разрешимые в $\mathcal{F}_3$ задачи, у которых существуют совместимые допустимые подстановки θ_1 и θ_2 соответственно. Прежде всего имеем $\mathbf{Sk}(A \vee B) = (A \vee B)$. Возьмем $\eta = \varepsilon$. Тогда $(A \vee B)\eta = (A \vee B)$ и можно считать, что подстановка η заменяет все локальные метапеременные в $A \vee B$, которых нет, на новые глобальные. Тогда разрешимы задачи $\mathbf{Sk}(A\eta)\,(A \vee B)\,\Gamma ?C$ и $\mathbf{Sk}(B\eta)\,(A \vee B)\,\Gamma ?C$. По правилу $(\vee ?)$ тогда разрешима задача $(A \vee B)\,\Gamma ?C$ и комбинация θ_1 и θ_2 является ее допустимой подстановкой. Что и требовалось доказать. □

Предложение 5.7. *Если задача* $\mathbf{Sk}(\neg(A \,\&\, B))\,\Gamma\,?\,C$ *разрешима в* $\mathcal{F}_3$ *и* θ *— ее допустимая подстановка, то разрешимы также обе задачи* $\mathbf{Sk}(\neg A)\,\Gamma\,?\,C$*,* $\Gamma\,\mathbf{Sk}(\neg B) ?C$ *и* θ *— их допустимая подстановка.*

Доказательство. Утверждение докажем индукцией по высоте деривационного дерева разрешимой задачи $\mathbf{Sk}(\neg(A \,\&\, B))\,\Gamma\,?\,C$. Рассмотрим лишь два случая строения этого деривационного дерева. Остальные подобны рассмотренным.

1. Если $\mathbf{Sk}(\neg(A \,\&\, B))\,\Gamma\,?\,C$ примитивизируется подстановкой θ, то и $\mathbf{Sk}(\neg A)\,\Gamma\,?\,C$ должна примитивизироваться подстановкой θ.

2. Рассмотрим случай, когда задача $\mathbf{Sk}(\neg(A \,\&\, B))\,\Gamma\,?\,C$ разбивается на подзадачи в деривационном дереве по правилу $(\supset ?)$. Здесь два варианта.

(а) Пусть $C = \bot$ и $\mathbf{Sk}(\neg(A \,\&\, B))\,\Gamma\,?\,C$ сводится к задаче $\mathbf{Sk}((A \,\&\, B) \supset \bot)\,\Gamma\,?\,A \,\&\, B$. Тогда дочерняя для нее задача $\mathbf{Sk}(\neg(A \,\&\, B))\,\Gamma\,?\,A$ также разрешима и θ — ее допустимая подстановка. По индуктивному предположению тогда разрешима задача $\mathbf{Sk}(\neg A)\,\Gamma\,?\,A$. По правилу $(\supset ?)$ тогда разрешима задача $\mathbf{Sk}(\neg A)\,\Gamma\,?\,\bot$ и θ — ее допустимая подстановка.

(б) Пусть задача $\mathbf{Sk}(\neg(A \,\&\, B))\,\Gamma\,?\,C$ разбивается в деривационном дереве по правилу $(\supset ?)$ на подзадачи

$$\mathbf{Sk}(\neg(A \,\&\, B))\,(\Delta_1\xi)\,\Gamma\,?\,F_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(\neg(A \,\&\, B))\,(\Delta_k\xi)\,\Gamma\,?\,F_{i_k}\xi,$$

где $F_{i_1} \,\&\, F_{i_2} \,\&\, \ldots,\ F_{i_k}$ — импликанта нормальной формы псевдоформулы $F \in \Gamma$, $\xi = \eta \circ \sigma$ и θ — комбинация подстановок σ, β_1, $\ldots$, β_k. По индуктивному предположению тогда разрешимы задачи

$$\mathbf{Sk}(\neg A)\,(\Delta_1\xi)\,\Gamma\,?\,F_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(\neg A)\,(\Delta_k\xi)\,\Gamma\,?\,F_{i_k}\xi$$

и соответственно $\beta_1, \ldots, \beta_k$ являются их допустимыми подстановками. По правилу ($\supset$?) тогда разрешима задача $\mathbf{Sk}(\neg A)\,\Gamma\,?\,C$ и θ является ее допустимой подстановкой. $\square$

Предложение 5.8. *Если задача* $\mathbf{Sk}(\neg(A \supset B))\,\Gamma\,?\,C$ *разрешима в* $\mathcal{F}_3$ *и* θ *— ее допустимая подстановка, то разрешима также задача* $\mathbf{Sk}(A)\,\mathbf{Sk}(\neg B)\,\Gamma\,?\,C$ *и* θ *— ее допустимая подстановка.*

Доказательство. Утверждение докажем индукцией по высоте деривационного дерева разрешимой задачи $\mathbf{Sk}(\neg(A \supset B))\,\Gamma\,?\,C$. Рассмотрим два случая строения этого деривационного дерева. Остальные случаи рассматриваются аналогично.

1. Если задача $\mathbf{Sk}(\neg(A \supset B))\,\Gamma\,?\,C$ примитивизируется подстановкой θ, то и задача $\mathbf{Sk}(A)\,\mathbf{Sk}(\neg B)\,\Gamma\,?\,C$ должна примитивизироваться подстановкой θ.

2. Рассмотрим случай, когда задача $\mathbf{Sk}(\neg(A \supset B))\,\Gamma\,?\,C$ разбивается на подзадачи в деривационном дереве по правилу ($\supset$?). Рассмотрим два возможных подслучая.

(а) Пусть $C = \perp$ и $\mathbf{Sk}(\neg(A \supset B))\,\Gamma\,?\,C$ сводится к задаче $\mathbf{Sk}((A \supset B) \supset \perp)\,\Gamma\,?\,A \supset B$. Тогда должна быть разрешима дочерняя для нее задача $\mathbf{Sk}(A)\,\mathbf{Sk}(\neg(A \supset B))\,\Gamma\,?\,B$ и θ должна быть ее допустимой подстановкой. По индуктивному предположению тогда также разрешима задача $\mathbf{Sk}(A)\,\mathbf{Sk}(A)\,\mathbf{Sk}(B \supset \perp)\,\Gamma\,?\,B$. По правилу ($\supset$?) тогда разрешима задача $\mathbf{Sk}(A)\,\mathbf{Sk}(\neg B)\,\Gamma\,?\perp$ и θ — ее допустимая подстановка.

(б) Пусть задача $\mathbf{Sk}(\neg(A \supset B))\,\Gamma\,?\,C$ разбивается в деривационном дереве по правилу ($\supset$?) на подзадачи

$$\mathbf{Sk}(\neg(A \supset B))\,(\Delta_1\xi)\,\Gamma\,?\,F_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(\neg(A \supset B))\,(\Delta_k\xi)\,\Gamma\,?\,F_{i_k}\xi,$$

где $F_{i_1} \,\&\, F_{i_2} \,\&\, \ldots,\ F_{i_k}$ — импликанта нормальной формы псевдоформулы $F \in \Gamma$, $\xi = \eta \circ \sigma$ и θ — комбинация подстановок σ, $\beta_1, \ldots, \beta_k$. По индуктивному предположению тогда разрешимы задачи

$$\mathbf{Sk}(A)\,\mathbf{Sk}(\neg B)\,(\Delta_1\xi)\,\Gamma\,?\,F_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(A)\,\mathbf{Sk}(\neg B)\,(\Delta_k\xi)\,\Gamma\,?\,F_{i_k}\xi$$

и соответственно $\beta_1, \ldots, \beta_k$ являются их допустимыми подстановками. По правилу ($\supset$?) тогда разрешима задача $\mathbf{Sk}(A)\,\mathbf{Sk}(\neg B)\,\Gamma\,?\,C$ и θ является ее допустимой подстановкой. $\square$

Предложение 5.9. *Если задача* $\mathbf{Sk}(\neg\forall x\, A(x))\,\Gamma\,?\,C$ *разрешима в* $\mathcal{F}_3$ *и* θ *— ее допустимая подстановка, где псевдоформулы* $\Gamma \cup \{(\forall x\, A(x))\}$

не содержат свободно переменной y, то задача $\mathbf{Sk}(\neg\, A(y))\, \Gamma\,?\, C$ *также разрешима и* θ *— ее допустимая подстановка.*

Доказательство. Утверждение докажем индукцией по высоте деривационного дерева разрешимой задачи $\mathbf{Sk}(\neg\forall x\, A(x))\, \Gamma\,?\, C$. Рассмотрим два случая строения деривационного дерева. Остальные случаи рассматриваются подобным образом.

1. Если задача $\mathbf{Sk}(\neg\forall x\, A(x))\, \Gamma\,?\, C$ примитивизируется подстановкой θ, то и задача $\mathbf{Sk}(\neg\, A(y))\, \Gamma\,?\, C$ должна примитивизироваться подстановкой θ.

2. Рассмотрим случай, когда задача $\mathbf{Sk}(\neg\forall x\, A(x))\, \Gamma\,?\, C$ разбивается на подзадачи в деривационном дереве по правилу $(\supset ?)$. Здесь могут быть два варианта.

(а) Пусть $C = \perp$ и $\mathbf{Sk}(\neg\forall x\, A(x))\, \Gamma\,?\, C$ в деривационном дереве сводится к задаче $\mathbf{Sk}((\forall x\, A(x)) \supset \perp)\, \Gamma\,?\, \forall x\, A(x)$. Тогда разрешима дочерняя для нее задача $\mathbf{Sk}(\neg\forall x\, A(x))\, \Gamma\,?\, A(y)$ и θ — ее допустимая подстановка. По индуктивному предположению тогда должна быть разрешима задача $\mathbf{Sk}(\neg\, A(y))\, \Gamma\,?\, A(y)$. По правилу $(\supset ?)$ тогда разрешима задача $\mathbf{Sk}(\neg\, A(y))\, \Gamma\,?\perp$ и θ — ее допустимая подстановка.

(б) Пусть задача $\mathbf{Sk}(\neg\forall x\, A(x))\, \Gamma\,?\, C$ разбивается в деривационном дереве по правилу $(\supset ?)$ на подзадачи

$$\mathbf{Sk}(\neg\forall x\, A(x))\, (\Delta_1\xi)\, \Gamma\,?\, F_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(\neg\forall x\, A(x))\, (\Delta_k\xi)\, \Gamma\,?\, F_{i_k}\xi,$$

где $F_{i_1} \,\&\, F_{i_2} \,\&\, \ldots,\, F_{i_k}$ — импликанта нормальной формы псевдоформулы $F \in \Gamma$, $\xi = \eta \circ \sigma$ и θ — комбинация подстановок σ, β_1, $\ldots$, β_k. По индуктивному предположению тогда разрешимы задачи

$$\mathbf{Sk}(\neg\, A(y))\, (\Delta_1\xi)\, \Gamma\,?\, F_{i_1}\xi,\ \ldots,\ \mathbf{Sk}(\neg\, A(y))\, (\Delta_k\xi)\, \Gamma\,?\, F_{i_k}\xi$$

и соответственно β_1, $\ldots$, β_k являются их допустимыми подстановками. По правилу $(\supset ?)$ тогда разрешима задача $\mathbf{Sk}(\neg\, A(y))\, \Gamma\,?\, C$ и θ является ее допустимой подстановкой. □

Предложение 5.10. *Если в* $\mathcal{F}_3$ *разрешима задача* $\mathbf{Sk}(\neg C)\, \Gamma\,?\perp$ *и* θ *— ее допустимая подстановка, то разрешима задача* $\Gamma ? C$ *и* θ *— ее допустимая подстановка.*

Доказательство. Утверждение докажем индукцией по количеству логических связок и кванторов в C.

1. Если C — атом, дизъюнкция или псевдоформула вида $\exists x\, D(x)$, то утверждение просто совпадает с формулировкой правила декомпозиции $(\perp_{\mathbf{c}})$. Рассмотрим остальные случаи индукции.

2. Пусть $C = D \,\&\, E$ и разрешима задача $\mathbf{Sk}(\neg(D \,\&\, E))\,\Gamma\,?\,\perp$. По предложению 5.7 тогда должны быть разрешимы задачи $\mathbf{Sk}(\neg D)\,\Gamma\,?\,\perp$ и $\mathbf{Sk}(\neg E)\,\Gamma\,?\,\perp$. По предположению индукции отсюда должны быть разрешимы задачи $\Gamma\,?\,D$ и $\Gamma\,?\,E$. Наконец по правилу $(?\&)$ тогда разрешима задача $\Gamma\,?\,C$.

3. Пусть $C = D \supset E$ и разрешима задача $\mathbf{Sk}(\neg(D \supset E))\,\Gamma\,?\,\perp$. По предложению 5.8 тогда разрешима задача $\mathbf{Sk}(D)\,\mathbf{Sk}(\neg E)\,\Gamma\,?\,\perp$. По индуктивному предположению отсюда разрешима задача $\mathbf{Sk}(D)\,\Gamma\,?\,E$. В результате по правилу $(?\supset)$ тогда разрешима задача $\Gamma\,?\,C$.

4. Пусть $C = \forall x\, D(x)$ и разрешима задача $\mathbf{Sk}(\neg\forall x\, D(x))\,\Gamma\,?\,\perp$. По предложению 5.9 тогда должна быть разрешима задача $\mathbf{Sk}(\neg\, D(y))\,\Gamma\,?\,\perp$. По предположению индукции отсюда должна быть разрешима задача $\Gamma\,?\,D(y)$. Наконец по правилу $(?\forall)$ тогда разрешима задача $\Gamma\,?\,C$. $\square$

Будем обозначать через $\overline{\mathcal{F}}_3$ продукционную систему, которая получается из $\mathcal{F}_3$ в результате добавления правил декомпозиции, сформулированных в предложениях 5.2, 5.3, 5.4, 5.5, 5.6, 5.10. Очевидно, что $\overline{\mathcal{F}}_3$ и $\mathcal{F}_3$ эквивалентны с точки зрения объема решаемых задач.

Предложение 5.11. *Если задача* $(\Gamma,\ A)$ *разрешима в продукционной системе* $\overline{\mathcal{P}}_3$*, то задача* $(\mathbf{Sk}(\Gamma)\,?\,A)$ *разрешима в продукционной системе* $\overline{\mathcal{F}}_3$.

Доказательство. Пусть θ — подстановка из набора допустимых подстановок задачи $\pi_0 = (\Gamma\,?\,A)$. Мы предполагаем, что подстановка θ является основной. В противном случае мы можем заменить "нерешенные" метапеременные на константы из H_0. Пусть Θ — соответствующее этой подстановке θ деривационное дерево. Пусть процесс расширения языка при построении дерева Θ имеет вид

$$\begin{array}{rl} & \Theta_0 \subseteq \Theta_1 \subseteq \ldots \subseteq \Theta_n \subseteq \Theta_{n+1} \subseteq \ldots \subseteq \Theta_h = \Theta, \\ \Omega \subseteq & \Omega_0 \subseteq \Omega_1 \subseteq \ldots \subseteq \Omega_n \subseteq \Omega_{n+1} \subseteq \ldots \subseteq \Omega_h, \\ & H_0 \subseteq H_1 \subseteq \ldots \subseteq H_n \subseteq H_{n+1} \subseteq \ldots \subseteq H_h. \end{array} \tag{16}$$

Индукцией по построению деривационного дерева Θ покажем, как для каждого дерева Θ_n можно построить дерево Θ'_n такое, что корнем дерева Θ'_n является задача $\pi'_0 = (\mathbf{Sk}(\Gamma)\,?\,A)$, дочерние вершины получаются из родительских по одному из правил декомпозиции системы $\overline{\mathcal{F}}_3$, а листья дерева Θ'_n либо примитивизируются подстановкой θ, либо соответствуют нераскрытым листьям $\pi_n = (\Gamma_n,\, A_n)$ дерева Θ_n и являются задачами π'_n вида $\pi'_n = (\Gamma'_n \mathbin{?} A'_n)$. При этом обозначаемое штрихом формируемое соответствие будет обладать следующим свойством: $A'_n = A_n$ и $\Gamma'_n = \mathbf{Sk}(\Gamma_n)$.

В начале процесса построения дерева Θ' положим $\Gamma'_0 = \mathbf{Sk}(\Gamma)$, $A'_0 = A$, $\pi'_0 = (\Gamma'_0 \mathbin{?} A'_0)$, $\Theta'_0 = \{\pi'_0\}$. Очевидно, что дерево Θ'_0 обладает требуемым свойством.

Покажем как получить дерево Θ'_{n+1} из дерева Θ'_n в зависимости от правила декомпозиции, которое используется при формировании дерева Θ_{n+1} из дерева Θ_n.

1. Если $A_n = \top$, то $\Theta_{n+1} = \Theta_n$. В этом случае листу $\pi_n = (\Gamma_n,\, A_n)$ соответствует лист $\pi'_n = (\Gamma'_n \mathbin{?} \top)$, где $\Gamma'_n = \mathbf{Sk}(\Gamma_n)$, который также является примитивной задачей. При этом задача $\pi'_n = (\Gamma'_n \mathbin{?} \top)$ примитивизируется любой подстановкой, в том числе θ. Положим $\Theta'_{n+1} = \Theta'_n$. При этом дерево Θ'_{n+1} вновь обладает требуемым свойством и имеет вид:

$$\begin{array}{c}(\Gamma'_n \mathbin{?} \top)\\ \ddots\vdots\cdot^{\cdot^{\cdot}}\\ (\Gamma'_0 \mathbin{?} A'_0)\end{array}$$

2. Пусть A_n — атомарная псевдоформула или $\bot$, лист $\pi_n = (\Gamma_n,\, A_n)$ примитивизируется подстановкой θ, $A_n\theta = E\theta$, $E \in \Gamma_n$ и лист $\pi'_n = (\Gamma'_n \mathbin{?} A'_n)$ соответствует листу π_n, где $A'_n = A_n$ и $\Gamma'_n = \mathbf{Sk}(\Gamma_n)$. Тогда $E' = \mathbf{Sk}(E) = E$. Отсюда задача $\pi'_n = (\Gamma'_n \mathbin{?} A'_n)$ примитивизируется подстановкой θ. Положим $\Theta'_{n+1} = \Theta'_n$. При этом дерево Θ'_{n+1} вновь обладает требуемым свойством и имеет вид:

$$\begin{array}{c}(\Gamma'_n \mathbin{?} A_n)\\ \ddots\vdots\cdot^{\cdot^{\cdot}}\\ (\Gamma'_0 \mathbin{?} A'_0)\end{array}$$

3. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции $(\rightarrow \vee_i)$. Следовательно имеем $A_n = (B_1 \vee B_2)$, $A'_n = A_n$. Положим $A'_{n+1} = A_{n+1} = B_i$ и $\Gamma'_{n+1} = \Gamma'_n$. Дерево Θ'_{n+1} сформируем из дерева Θ'_n с помощью применения правила $(?\vee_i)$:

$$\begin{array}{c}\dfrac{(\Gamma'_n \mathbin{?} B_i)}{(\Gamma'_n \mathbin{?} B_1 \vee B_2)}\\ \ddots\vdots\cdot^{\cdot^{\cdot}}\\ (\Gamma'_0 \mathbin{?} A'_0)\end{array}$$

При этом если θ является допустимой подстановкой для $(\Gamma'_n \mathbin{?} B_i)$, то θ является допустимой подстановкой для задачи $(\Gamma'_n \mathbin{?} B_1 \vee B_2)$.

4. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции $(\to \exists)$. Это значит, что $A'_n = A_n = (\exists x\, B(x))$. Положим $A'_{n+1} = A_{n+1} = B(X)$ и $\Gamma'_{n+1} = \Gamma'_n$. Дерево Θ'_{n+1} сформируем из дерева Θ'_n с помощью применения правила $(?\exists)$:

$$\begin{array}{c}\dfrac{(\Gamma'_n \mathbin{?} B(X))}{(\Gamma'_n \mathbin{?} \exists x\, B(x))}\\ \ddots\vdots\cdot^{\cdot^{\cdot}}\\ (\Gamma'_0 \mathbin{?} A'_0)\end{array}$$

Очевидно, что если θ является допустимой подстановкой для задачи $(\Gamma'_n \mathbin{?} B(X))$, то θ является допустимой подстановкой для $(\Gamma'_n \mathbin{?} \exists x\, B(x))$.

5. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции $(\to \&)$. Это значит, что $A'_n = A_n = (B_1 \,\&\, B_2)$. Дерево Θ'_{n+1} сформируем из дерева Θ'_n с помощью применения правила декомпозиции $(?\&)$:

$$\begin{array}{c}\dfrac{(\Gamma'_n \mathbin{?} B_1) \qquad (\Gamma'_n \mathbin{?} B_2)}{(\Gamma'_n \mathbin{?} B_1 \,\&\, B_2)}\\ \ddots\vdots\cdot^{\cdot^{\cdot}}\\ (\Gamma'_0 \mathbin{?} A'_0)\end{array}$$

При этом если θ является допустимой подстановкой для обеих задач $(\Gamma'_n \mathbin{?} B_1)$ и $(\Gamma'_n \mathbin{?} B_2)$, то θ — допустимая подстановка для $(\Gamma'_n \mathbin{?} B_1 \,\&\, B_2)$.

6. Пусть Θ_{n+1} получается из Θ_n в результате применения продукции $(\to\supset)$. Это значит, что $A'_n = A_n = (B_1 \supset B_2)$. Положим $A'_{n+1} = A_{n+1} = B_2$, $B'_1 = \mathbf{Sk}(B_1)$ и $\Gamma'_{n+1} = \Gamma'_n \cup \{B'_1\} = \mathbf{Sk}(\Gamma_{n+1})$. Дерево Θ'_{n+1} сформируем из дерева Θ'_n с помощью применения правила декомпозиции $(? \supset)$:

$$\begin{array}{c}\dfrac{(\mathbf{Sk}(B_1)\, \Gamma'_n \mathbin{?} B_2)}{(\Gamma'_n \mathbin{?} B_1 \supset B_2)}\\ \ddots\vdots\cdot^{\cdot^{\cdot}}\\ (\Gamma'_0 \mathbin{?} A'_0)\end{array}$$

Очевидно, что если θ является допустимой подстановкой для задачи $(B'_1\, \Gamma'_n \mathbin{?} B_2)$, то θ является допустимой подстановкой для $(\Gamma'_n \mathbin{?} B_1 \supset B_2)$.

7. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции $(\to \forall)$. Это значит, что $A'_n = A_n = (\forall x\, B(x))$. Положим

$A'_{n+1} = A_{n+1} = B(y)$ и $\Gamma'_{n+1} = \Gamma'_n$. Дерево Θ'_{n+1} сформируем из дерева Θ'_n с помощью применения правила (?∀):

$$\cfrac{(\Gamma'_n \,?\, B(y))}{\cfrac{(\Gamma'_n \,?\, \forall x\, B(x))}{\begin{array}{c}\ddots\vdots\cdot^{\cdot^{\cdot}}\\ (\Gamma'_0 \,?\, A'_0)\end{array}}}$$

Очевидно, что если θ является допустимой подстановкой для задачи $(\Gamma'_n \,?\, B(y))$, то θ является допустимой подстановкой для $(\Gamma'_n \,?\, \forall x\, B(x))$.

8. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции $(\& \to)$. Это значит, что $A'_n = A_n$, $C = (B_1 \,\&\, B_2) \in \Gamma_n$, $\Gamma'_n = \mathbf{Sk}(\Gamma_n)$, $B'_i = \mathbf{Sk}(B_i)$, $C' = \mathbf{Sk}(C) = \mathbf{Sk}(B_1 \,\&\, B_2) = \mathbf{Sk}(B_1) \,\&\, \mathbf{Sk}(B_2) = (B'_1 \,\&\, B'_2) \in \Gamma'_n$, $\Gamma_{n+1} = \Gamma_n \cup \{B_1,\, B_2\}$. Заметим, что результатом применения к задаче $(\mathbf{Sk}(B_1 \,\&\, B_2)\, \Gamma'_n \,?\, A'_n)$ сформулированного в предложении 5.2 правила декомпозиции будет $(\mathbf{Sk}(B_1)\, \mathbf{Sk}(B_2)\, \mathbf{Sk}(B_1 \,\&\, B_2)\, \Gamma'_n \,?\, A'_n)$. Положим $\Gamma'_{n+1} = \Gamma'_n \cup \{B'_1,\, B'_2\}$, $A'_{n+1} = A'_n = A_n$. Дерево Θ'_{n+1} сформируем из дерева Θ'_n с помощью применения этого правила декомпозиции:

$$\cfrac{(\mathbf{Sk}(B_1)\, \mathbf{Sk}(B_2)\, \mathbf{Sk}(B_1 \,\&\, B_2)\, \Gamma'_n \,?\, A_n)}{\cfrac{(\mathbf{Sk}(B_1 \,\&\, B_2)\, \Gamma'_n \,?\, A_n)}{\begin{array}{c}\ddots\vdots\cdot^{\cdot^{\cdot}}\\ (\Gamma'_0 \,?\, A'_0)\end{array}}}$$

По предложению 5.2 заключаем, что если θ является допустимой подстановкой для задачи $(B'_1\, B'_2\, \Gamma'_n \,?\, A_n)$, то θ является допустимой подстановкой также и для задачи $(\Gamma'_n \,?\, A_n)$.

9. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции $(\supset\to)$. Это значит, что $A'_n = A_n$, $C = (B_1 \supset B_2) \in \Gamma_n$, $\Gamma'_n = \mathbf{Sk}(\Gamma_n)$, $B'_1 = B_1$, $B'_2 = \mathbf{Sk}(B_2)$, $C' = \mathbf{Sk}(C) = \mathbf{Sk}(B_1 \supset B_2) = B_1 \supset \mathbf{Sk}(B_2) = (B'_1 \supset B'_2) \in \Gamma'_n$. Тогда задачи $(\mathbf{Sk}(B_1 \supset B_2)\, \Gamma'_n \,?\, B_1)$ и $(\mathbf{Sk}(B_2)\, \mathbf{Sk}(B_1 \supset B_2)\, \Gamma'_n \,?\, A'_n)$ являются дочерними для родительской задачи $(\mathbf{Sk}(B_1 \supset B_2)\, \Gamma'_n \,?\, A'_n)$, если к последней применить правило декомпозиции, формулировка которого содержится в предложении 5.3. Дерево Θ'_{n+1} сформируем из дерева Θ'_n с помощью применения этого правила декомпозиции:

$$\cfrac{(\mathbf{Sk}(B_1 \supset B_2)\, \Gamma'_n \,?\, B_1) \qquad (\mathbf{Sk}(B_2)\, \mathbf{Sk}(B_1 \supset B_2)\, \Gamma'_n \,?\, A_n)}{\cfrac{(\mathbf{Sk}(B_1 \supset B_2)\, \Gamma'_n \,?\, A_n)}{\begin{array}{c}\ddots\vdots\cdot^{\cdot^{\cdot}}\\ (\Gamma'_0 \,?\, A'_0)\end{array}}}$$

Из предложения 5.3 следует, что если θ является допустимой подстановкой для задач $(B'_2\,\Gamma'_n\,?\,A_n)$ и $(\Gamma'_n\,?\,B_1)$, то θ является допустимой подстановкой также и для задачи $(\Gamma'_n\,?\,A_n)$.

10. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции $(\forall\ \rightarrow)$. Это значит, что $A'_n = A_n$, $C = (\forall x\, B(x)) \in \Gamma_n$, $\Gamma'_n = \mathbf{Sk}(\Gamma_n)$, $C' = \mathbf{Sk}(C) = \mathbf{Sk}(\forall x\, B(x)) \in \Gamma'_n$, $\Gamma_{n+1} = \Gamma_n \cup \{B(X)\}$. Тогда задача $(\mathbf{Sk}(B(X))\,\mathbf{Sk}(\forall x\, B(x))\,\Gamma'_n\,?\,A'_n)$ является результатом применения к задаче $(\mathbf{Sk}(\forall x\, B(x))\,\Gamma'_n\,?\,A'_n)$ сформулированного в предложении 5.4 правила декомпозиции. Положим $\Gamma'_{n+1} = \Gamma'_n \cup \{\mathbf{Sk}(B(X))\}$. Дерево Θ'_{n+1} сформируем из дерева Θ'_n с помощью применения этого правила декомпозиции:

$$\cfrac{(\mathbf{Sk}(B(X))\,\mathbf{Sk}(\forall x\, B(x))\,\Gamma'_n\,?\,A_n)}{\begin{array}{c}(\mathbf{Sk}(\forall x\, B(x))\,\Gamma'_n\,?\,A_n)\\ \ddots\vdots\cdot^{\cdot^{\cdot}}\\ (\Gamma'_0\,?\,A'_0)\end{array}}$$

По предложению 5.4 заключаем, что если θ является допустимой подстановкой для задачи $(\mathbf{Sk}(B(X))\,\Gamma'_n\,?\,A_n)$, то θ является допустимой подстановкой также и для задачи $(\Gamma'_n\,?\,A_n)$.

11. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции $(\exists\ \rightarrow)$. Это значит, что $A'_n = A_n$, $C = (\exists x\, B(x)) \in \Gamma_n$, $\Gamma'_n = \mathbf{Sk}(\Gamma_n)$, $C' = \mathbf{Sk}(C) = \mathbf{Sk}(\exists x\, B(x)) \in \Gamma'_n$, $\Gamma_{n+1} = \Gamma_n \cup \{B(y)\}$. Тогда задача $(\mathbf{Sk}(B(y))\,\mathbf{Sk}(\exists x\, B(x))\,\Gamma'_n\,?\,A'_n)$ является результатом применения к задаче $(\mathbf{Sk}(\exists x\, B(x))\,\Gamma'_n\,?\,A'_n)$ сформулированного в предложении 5.5 правила декомпозиции. Положим $\Gamma'_{n+1} = \Gamma'_n \cup \{\mathbf{Sk}(B(y))\}$. Дерево Θ'_{n+1} построим из дерева Θ'_n, применив указанное правило декомпозиции:

$$\cfrac{(\mathbf{Sk}(B(y))\,\mathbf{Sk}(\exists x\, B(x))\,\Gamma'_n\,?\,A_n)}{\begin{array}{c}(\mathbf{Sk}(\exists x\, B(x))\,\Gamma'_n\,?\,A_n)\\ \ddots\vdots\cdot^{\cdot^{\cdot}}\\ (\Gamma'_0\,?\,A'_0)\end{array}}$$

Из предложения 5.5 следует, что если θ является допустимой подстановкой для задачи $(\mathbf{Sk}(B(y))\,\Gamma'_n\,?\,A_n)$, то θ является допустимой подстановкой также и для задачи $(\Gamma'_n\,?\,A_n)$.

12. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции $(\vee \rightarrow)$. Это значит, что $A'_n = A_n$, $C = (B_1 \vee B_2) \in \Gamma_n$, $\Gamma'_n = \mathbf{Sk}(\Gamma_n)$, $B'_i = \mathbf{Sk}(B_i)$, $C' = \mathbf{Sk}(C) = \mathbf{Sk}(B_1 \vee B_2) = (B_1 \vee B_2) = C \in \Gamma'_n$. Тогда задачи $(\mathbf{Sk}(B_1)\,\mathbf{Sk}(B_1 \vee B_2)\,\Gamma'_n\,?\,A'_n)$ и $(\mathbf{Sk}(B_2)\,\mathbf{Sk}(B_1 \vee B_2)\,\Gamma'_n\,?\,A'_n)$ являются дочерними для задачи $(\mathbf{Sk}(B_1 \vee B_2)\,\Gamma'_n\,?\,A'_n)$ после применения сформулированного в предложении 5.6 правила декомпозиции. Дерево

Θ'_{n+1} сформируем из дерева Θ'_n с помощью применения указанного правила декомпозиции:

$$\dfrac{(\mathbf{Sk}(B_1)\,\mathbf{Sk}(B_1 \vee B_2)\,\Gamma'_n \mathrel{?} A_n) \qquad (\mathbf{Sk}(B_2)\,\mathbf{Sk}(B_1 \vee B_2)\,\Gamma'_n \mathrel{?} A_n)}{(\mathbf{Sk}(B_1 \vee B_2)\,\Gamma'_n \mathrel{?} A_n)}$$
$$\ddots \vdots \cdot^{\cdot^{\cdot}}$$
$$(\Gamma'_0 \mathrel{?} A'_0)$$

Из предложения 5.6 следует, что если θ является допустимой подстановкой для задач $(B'_1\,\Gamma'_n \mathrel{?} A_n)$ и $(B'_2\,\Gamma'_n \mathrel{?} A_n)$, то θ является допустимой подстановкой также и для задачи $(\Gamma'_n \mathrel{?} A_n)$.

13. Пусть Θ_{n+1} получается из Θ_n в результате применения правила декомпозиции $(\bot_{\mathbf{c}})$. Это значит, что $A'_n = A_n$, $B = (\neg A_n) \in \Gamma_n$, $\Gamma'_n = \mathbf{Sk}(\Gamma_n)$, $B' = \mathbf{Sk}(B) = \mathbf{Sk}(\neg A_n) = (\neg A_n) \in \Gamma'_n$, $\Gamma_{n+1} = \Gamma_n \cup \{(\neg A_n)\}$, $A_{n+1} = \bot$. Тогда задача $((\neg A_n)\,\Gamma'_n \mathrel{?} \bot)$ является результатом применения к задаче $(\Gamma'_n \mathrel{?} A'_n)$ сформулированного в предложении 5.10 правила декомпозиции. Положим $\Gamma'_{n+1} = \Gamma'_n \cup \{(\neg A_n)\}$, $A'_{n+1} = \bot$. Дерево Θ'_{n+1} сформируем из дерева Θ'_n с помощью применения этого правила декомпозиции:

$$\dfrac{((\neg A_n)\,\Gamma'_n \mathrel{?} \bot)}{(\Gamma'_n \mathrel{?} A_n)}$$
$$\ddots \vdots \cdot^{\cdot^{\cdot}}$$
$$(\Gamma'_0 \mathrel{?} A'_0)$$

По предложению 5.10 заключаем, что если θ является допустимой подстановкой для задачи $((\neg A_n)\,\Gamma'_n \mathrel{?} \bot)$, то θ является допустимой подстановкой также и для задачи $(\Gamma'_n \mathrel{?} A_n)$. Тем самым рассмотрение случая $(\bot_{\mathbf{c}})$ завершено.

Для построения Θ_{n+1}, а затем Θ'_{n+1}, в соответствии с применяемой стратегией мы выбирали лист $\pi_n = (\Gamma_n, A_n) \in \Theta_n$, у которого порождали дочерние разрешимые вершины, соответствующие подстановке θ и обеспечивающие разрешимость вершины π_n. В результате было сформировано дерево Θ'_{n+1}, которое получается из дерева Θ'_n в результате замены листа $\pi'_n = (\Gamma'_n \mathrel{?} A'_n)$, соответствующего листу π_n дерева Θ_n, на некоторое новое поддерево с корнем π'_n. При этом каждая вершина ρ полученного дерева Θ'_{n+1} обладает следующим свойством: во-первых, если ρ является листом дерева Θ'_{n+1}, который не соответствует нераскрытому листу дерева Θ_{n+1}, то $\rho\theta$ является примитивной задачей; во-вторых, если θ является допустимой для вершин $\rho_1, \ldots, \rho_k$, всех дочерних у вершины ρ в дереве Θ'_{n+1}, то θ является также допустимой и для ρ. На заключительном шаге этого процесса мы получим дерево $\Theta' = \Theta'_h$ такое, что θ является допустимой подстановкой задачи $(\mathbf{Sk}(\Gamma)\,?\,A)$ и Θ' является соответствующим этой подстановке θ деривационным деревом в $\overline{\mathcal{F}}_3$. Предложение 5.11 доказано. □

Теорема о полноте. *Если секвенция $\Gamma \to A$ выводима в исчислении $\mathcal{S}_3$, то задача $(\mathbf{Sk}(\Gamma)\,?\,A)$ разрешима в $\mathcal{F}_3$.*

Доказательство. Пусть секвенция $\Gamma \to A$ выводима в исчислении $\mathcal{S}_3$. По теореме о полноте $\mathcal{P}_3$ тогда задача $(\Gamma,\ A)$ разрешима в $\mathcal{P}_3$. Продукционные системы $\overline{\mathcal{P}}_3$ и $\overline{\mathcal{F}}_3$ получаются соответственно из $\mathcal{P}_3$ и $\mathcal{F}_3$ за счет добавления некоторых, причем допустимых, правил декомпозиции. Это значит, что задача $(\Gamma,\ A)$ разрешима и в $\overline{\mathcal{P}}_3$. По предложению 5.11 тогда задача $(\mathbf{Sk}(\Gamma)\,?\,A)$ разрешима в $\overline{\mathcal{F}}_3$, а значит и в $\mathcal{F}_3$. Что и требовалось доказать. □

Автор выражает искреннюю признательность проф. В.Ю. Попову за советы, замечания и обсуждение вопросов, которые возникали при построении этой теории.

Список литературы

[1] Вторушин Ю.И., "О поиске вывода в системе натуральной дедукции логики предикатов", *Интеллектуальные системы*, **13**:3 (2009), 263–288.

[2] Чень Ч., Ли Р., *Математическая логика и автоматическое доказательство теорем*, «Наука», Москва, 1983, 360 с.

[3] Конев Б.Ю., Жебелеан Т., "Метод подъема решений для работы с метапеременными в системе TH∃OREM∀", *Записки научных семинаров ПОМИ*, **293** (2002), 94–117

[4] Sieg W., Byrnes J., "Normal natural deduction proofs (in classical logic)", *Studia Logica*, **60**:1 (1998), 67–106

[5] Okhotnikov O., "A new sequent calculus for automated proof search", *Applied Mathematical Sciences*, **8**:100 (2014), 4977–4984

[6] Большакова Е.И., Мальковский М.Г., Пильщиков В.Н., *Искусственный интеллект: методы и алгоритмы эвристического поиска*, МГУ, Москва, 2002, 81 с.

[7] Маслов С.Ю., *Теория дедуктивных систем и ее применения*, «Наука», Москва, 1986, 136 с.

[8] Lyaletski A., "Admissibility, compatibility, and deducibility in first-order sequent logics", *Computer Science Journal of Moldova*, **23**:3 (2015), 289–303

[9] Degtyarev A., Voronkov A., "Kanger's choices in automated reasoning", *Collected Papers of Stig Kanger with Essays on His Life and Work*, 2001, 53–68

[10] Минц Г.Е., "Теорема Эрбрана", *Математическая теория логического вывода*, 1967, 311–350.

About proof-search in classical natural deduction calculus using partial skolemization

Okhotnikov O.A.

Automated proof search in single-conclusion sequential variant of classical predicate calculus is considered. In this algorithm, metavariables and partial skolemization are used. Theorems of soundness and completeness for the considered algorithm are proved.

Keywords: automated theorem proving, mechanical proof search, first order language, predicate calculus, sequent calculus, production system, artificial intelligence.