ОРДЕНА ЛЕНИНА
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
АКАДЕМИИ НАУК СССР

С.А. Романенко, А.В. Климов, В.Ф. Турчин.

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СИНТАКСИЧЕСКОГО ОТОЖДЕСТВЛЕНИЯ В ЯЗЫКЕ РЕФАЛ

Препринт № 13 за 1973г

Москва

С.А.Романенко, Ан.В.Климов, В.Ф.Турчин

Теоретические основы синтаксического отождествления в языке РЕФАЛ

I. ВВЕДЕНИЕ

В работах [1,2,5] описаны синтаксис и семантика языка рефал. Описание семантики дано в виде описания абстрактной рефал-машины, которая способна выполнять алгоритмы, записанные на языке рефал.

При описании рефал-машины основное внимание уделялось тому, чтобы она была принципиально проста. При этом сознательно игнорировался тот факт, что во многих случаях абстрактная рефал-машина практически неэффективна. Однако, когда дело доходит до практического использования рефала в качестве языка программирования, повышение эффективности рефал-машины приобретает решающее значение. Этим вопросам посвящены работы [3,4,6] и настоящая работа.

Принцип, который положен в основу этих работ заключается в том, что абстрактная рефал-машина заменяется на другую, "практическую" рефал-машину. Практическая рефал-машина во многих случаях сложней, чем абстрактная, однако работает эффективнее. Конечно, такая замена правомерна только в том случае, если практическая рефал-машина всегда дает тот же самый результат, что и абстрактная.

Таким образом, проблема "улучшения" рефал-машины сводится к доказательству эквивалентности тех или иных алгоритмов и выяснению условий при которых эта эквивалентность имеет место.

Работа рефал-машины распадается на шаги. Выполнение каждого шага заключается в последовательном выполнении синтаксического отождествления и замены. Наиболее сложный процесс—синтаксическое отождествление.

Настоящая работа посвящена анализу процесса синтаксического отождествления и выяснению условий, при которых этот процесс

может быть ускорен.

Следует отметить, что алгоритм синтаксического отождествления описан в работах [1,2,5] по-разному. В работе [2] описан алгоритм отождествления, который не эквивалентен алгоритмам, описанным в [1] и [5] . Алгоритмы отождествления, описанные в [1] и в [5] эквивалентны, хотя и выглядят по-разному. Их эквивалентность следует из результатов настоящей работы.

Рефал-интерпретатор (см. [3]) реализует алгоритм отождествления, описанный в [2] . Рефал-компилятор (см. [4,6]) реализует алгоритм отождествления, в соответствии с [1] и [5] .

В настоящей работе, с целью облегчения теоретического анализа рассматривается обобщенный алгоритм отождествления, точнее класс алгоритмов отождествления, по отношению к которым алгоритмы из [1] и [5] выступают как частный случай. Тем самым заодно выявляется естественность введения в базисный рефал некоторых дополнительных конструкций.

Развит новый подход к понятию алгоритма синтаксического отождествления: сначала определяется исчисление, порождающее всевозможные результаты отождествления, а затем на множестве выводов вводится разрешимое отношение порядка и таким образом определяется алгоритм обхода дерева выводов, который и вычисляет нужный вариант отождествления.

Полученные результаты не распространяются непосредственно на полный рефал (как он описан в [5]). Однако предполагается, что реализация полного рефала, сводится к построению компилятора с полного на базисный рефал (в [4] и [6]) описывается компилятор с базисного рефала на машинно-ориентированный язык сборки). Поэтому основной машинной реализации рефала является эффективная реализация базисного рефала.

2. ВЫРАЖЕНИЯ

2.1. Замечание. В настоящей работе использованы только те понятия рефала, которые непосредственно связаны с синтаксическим отождествлением, поэтому все необходимые понятия будут описаны заново. Определенное здесь понятие выражения является обобщением частного случая выражения рефала.

2.2. Синтаксис выражений.

<выражение> ::= <объектное выражение>
| <типовое выражение>

<объектное выражение> ::= <пусто>
| <объектный терм>
<объектное выражение>

<объектный терм> ::=
<символ> | (<объектное выражение>)

<типовое выражение> ::= <пусто>
| <типовой терм>
<типовое выражение>

<типовой терм> ::=
<переменная>
| (<типовое выражение>)

<скобка> ::= <левая скобка>
| <правая скобка>

<левая скобка> ::= (

<правая скобка> ::=)

<пусто> ::=

<терм> ::= <объектный терм>
| <типовой терм>

2.3. Переменные и символы.

Легко заметить, что синтаксис выражений, описанный в п.2.2 неполон. Он не содержит описания понятий < символ > и < переменная > . Для дальнейшего изложения синтаксис символов и переменных совершенно безразличен и мы его уточнять не будем.

Поскольку, используя конечный набор знаков можно построить не более чем счетное множество объектов, мы будем полагать, что множество символов не более чем счетно. В таком случае его можно перенумеровать. k-й символ обозначим через Z_k . Множество переменных будем считать тоже не более чем счетным. k -ую переменную будем обозначать через x_k .

Таким образом, имеется счетное множество символов

$$Z_1, Z_2, Z_3, \ldots$$

и счетное множество переменных

$$x_1, x_2, x_3, \ldots$$

Следует подчеркнуть, что Z_k и x_k представляют собой не сами символы и переменные, а только их мета-обозначения.

Введем еще одно мета-обозначение. Поскольку выражение может быть пустым, а во многих случаях изображать пустое выражение пустотой неудобно, мы введем для него обозначение $\varnothing$.

Здесь же отметим, что в этой работе знак $\varnothing$ употребляется и для других целей, в частности для обозначения пустого множества.

Теперь можно привести примеры объектных и типовых выражений.

Объектные выражения:

$$\varnothing,\ (\,)((\,)),\ Z_{99},\ (Z_1)Z_2Z_1$$

Типовые выражения:

$$\varnothing,\ (\,),\ x_{11},\ (x_1x_2)((x_3)x_3)$$

2.4. Нуль-термы и нуль-длина

Обозначим через $\mathcal{E}$ произвольное выражение. Очевидно, что всякое $\mathcal{E}$ можно однозначным образом представить в виде

$$\mathcal{T}_1\ \mathcal{T}_2 \ldots \mathcal{T}_n$$

где через $\mathcal{T}_k$ обозначен некоторый терм.

2.4.1. Термы, в соединение которых разлагается выражение $\mathcal{E}$, назовем нуль-термами этого выражения.

2.4.2. Назовем нуль-длиной выражения $\mathcal{E}$ количество нуль-термов, составляющих это выражение.

2.4.3. Нуль-длину выражения $\mathcal{E}$ мы будем обозначать через $\mathcal{L}_0[\mathcal{E}]$. Очевидно, что $\mathcal{L}_0[\mathcal{E}]=0$ тогда и только тогда, когда $\mathcal{E}$ – пустое выражение. Если же $\mathcal{L}_0[\mathcal{E}]=1$, то выражение $\mathcal{E}$ является термом.

2.5. Подходящие выражения и классификация переменных

Будем предполагать, что каждой переменной $\mathcal{X}_k$ поставлено в соответствие некоторое, вообще говоря, счетное множество объектных выражений $\mathcal{A}_k$. Каждое объектное выражение $\mathcal{A}$, такое, что $\mathcal{A}\in\mathcal{A}_k$ мы будем называть подходящим к переменной $\mathcal{X}_k$, а само множество $\mathcal{A}_k$ – множеством подходящих выражений для $\mathcal{X}_k$.

Как будет видно в дальнейшем, содержательно множество подходящих выражений – это множество тех значений, которые может принимать переменная. В настоящей работе нас не будет интересовать каким именно способом заданы множества $\mathcal{A}_k$ хотя, конечно, будет предполагаться, что они разрешимы.

Выделим теперь некоторые важные частные случаи множеств $\mathcal{A}_k$ и в соответствии с этим выделим некоторые важные классы переменных.

2.5.1. Константы. Переменная $\mathcal{X}_k$ является константой, если

множество $\mathcal{A}_k$ состоит из одного - единственного объектного выражения. Наиболее важный пример константы - это переменная, для которой $\mathcal{A}_k$ состоит из одного-единственного символа $\mathcal{Z}_m$. В дальнейшем мы будем использовать метаобозначение $\mathcal{Z}_m$ для переменной $\mathcal{X}_k$, если $\mathcal{A}_k = \{ \mathcal{Z}_m \}$ наряду с обозначением $\mathcal{X}_k$ Как будет ясно в дальнейшем, это не может привести к недоразумениям.

2.5.2. Твердые переменные. Переменная $\mathcal{X}_k$ является твердой, если все выражения, составляющие $\mathcal{A}_k$, имеют одинаковую нуль-длину. Иначе

$$a_1 \in \mathcal{A}_k \;\; и \;\; a_2 \in \mathcal{A}_k \Rightarrow \mathcal{L}_0[a_1] = \mathcal{L}_0[a_2]$$

Переменная-константа дает пример твердой переменной.

2.5.3. e - переменные. Переменная $\mathcal{X}_k$ является e -переменной, если $\mathcal{A}_k$ совпадает с множеством всех объектных выражений. Для e -переменной $\mathcal{X}_k$ мы будем употреблять мета-обозначение e_k наряду с обозначением $\mathcal{X}_k$.

Эта классификация, конечно, не исчерпывает всех возможностей. Множества $\mathcal{A}_k$ могут быть и более сложными. В базисном рефале, однако, до сих пор употреблялись только константы, твердые переменные и e -переменные.

3. СИНТАКСИЧЕСКОЕ ОТОЖДЕСТВЛЕНИЕ.

3.1. Определение отождествимости

Пусть $\mathcal{E}_t$ некоторое типовое выражение, а $\mathcal{E}_0$ некоторое объектное выражение. Будем говорить, что $\mathcal{E}_0$ отождествимо с $\mathcal{E}_t$, если все переменные, входящие в $\mathcal{E}_t$, можно заменить такими подходящими к ним объектными выражениями, что в результате выражение $\mathcal{E}_t$ совпадет с $\mathcal{E}_0$. Причем, если какая-то переменная входит в $\mathcal{E}_t$ несколько раз, то все её вхождения должны заменяться на одно и то же подходящее выражение.

3.2. Варианты и подстановки

3.2.1. Назовем вариантом произвольное множество упорядоченных пар вида (x, a), где x - произвольная переменная, а a - произвольное объектное выражение, подходящее к переменной x. Другими словами, если $x = x_k$, то $a \in \mathcal{A}_k$

3.2.2. Вариант ϑ называется противоречивым, если существуют две пары $(x', a') \in \vartheta$ и $(x'', a'') \in \vartheta$, такие, что $x' = x''$, но в то же время $a' \neq a''$

3.2.3. Вариант ϑ, который не является противоречивым, мы будем называть непротиворечивым.

3.2.4. Будем говорить, что переменная x входит в вариант ϑ, если существует пара $(x, a) \in \vartheta$. Будем говорить, что вариант ϑ содержит переменную x, если переменная x входит в ϑ.

3.2.5. Определим операцию подстановки. Пусть ϑ -непротиворечивый вариант и пусть все переменные, входящие в типовое выражение $\mathcal{E}_t$, входят и в вариант ϑ. Тогда через $Q_0(\vartheta, \mathcal{E}_t)$ обозначим объектное выражение, которое получается из $\mathcal{E}_t$, если каждую переменную x, входящую в $\mathcal{E}_t$, заменить на объектное выражение a, такое, что $(x, a) \in \vartheta$

Ясно, что определенный таким образом результат подстановки всегда существует и единственен.

3.2.6. Назовем вариант ϑ минимальным для $\mathcal{E}_t$, если он содержит только те переменные, которые входят в $\mathcal{E}_t$ и непротиворечив.

3.2.7. Очевидно, что любой непротиворечивый вариант ϑ, удовлетворяющий условиям п.3.2.5 представим в виде $\vartheta = \vartheta_0 \cup \vartheta_1$, где ϑ_0 - минимальный для $\mathcal{E}_t$, а $\vartheta_0 \cap \vartheta_1 = \emptyset$. В этом случае

$$Q_0(\vartheta_0 \cup \vartheta_1, \mathcal{E}_t) = Q_0(\vartheta_0, \mathcal{E}_t)$$

3.3. Варианты отождествления

3.3.1. Вариант ϑ называется вариантом отождествления для типового выражения ξ_t и объектного выражения ξ_o, если подстановка $Q_0(\vartheta, \xi_t)$ определена и $Q_0(\vartheta, \xi_t) = \xi_o$

3.3.2. Ясно, что, если для ξ_t и ξ_o существует хоть один вариант отождествления, то существует счетное множество вариантов отождествления для ξ_t и ξ_o согласно п.3.2.6. Однако, наибольший интерес представляют минимальные варианты отождествления. Обозначим через $W_0(\xi_t, \xi_o)$ множество всех минимальных вариантов отождествления для ξ_t и ξ_o. Другими словами

$$W_0(\xi_t, \xi_o) = \{\vartheta : \vartheta \text{ - минимальный для } \xi_t \text{ и } Q_0(\vartheta, \xi_t) = \xi_o\}$$

Теорема 3.1. Множество $W_0(\xi_t, \xi_o)$ конечно для любых ξ_t и ξ_o.

Доказательство. Сначала заметим, что равенство $Q_0(\vartheta, \xi_t) = \xi_o$ возможно только, если $Q_0(\vartheta, \xi_t)$ содержит те и только те символы, что и ξ_o. Однако, множество символов, входящих в ξ_o, конечно. Если хоть одна пара $(x, a) \in \vartheta$ такова, что в a входит символ $\mathcal{Z}$, не входящий в ξ_o, то равенство $Q_0(\vartheta, \xi_t) = \xi_o$ невозможно. Поэтому каждая пара (x, a) содержит только символы, входящие в ξ_o. Далее, если ξ_o содержит N символов и скобок, то и $Q_0(\vartheta, \xi_t)$ содержит N символов и скобок. Отсюда следует, что для любой пары $(x, a) \in \vartheta$ длина a - не более чем N. Таким образом, для каждой пары $(x, a) \in \vartheta$ выражение a имеет длину не более N и может содержать символы из конечного множества. Теорема доказана.

3.3.3. Алгоритмом отождествления называется правило, которое позволяет каждой паре $(\mathfrak{E}_t, \mathfrak{E}_o)$ поставить в соответствие один и только один вариант $v \in W_o(\mathfrak{E}_t, \mathfrak{E}_o)$, при условии, что

$$W_o(\mathfrak{E}_t, \mathfrak{E}_o) \neq \emptyset$$

Другими словами, алгоритм отождествления – это функция $T_o(\mathfrak{E}_t, \mathfrak{E}_o)$, которая определена на всех парах $(\mathfrak{E}_t, \mathfrak{E}_o)$, для которых $W_o(\mathfrak{E}_t, \mathfrak{E}_o) \neq \emptyset$ и $T_o(\mathfrak{E}_t, \mathfrak{E}_o) \in W_o(\mathfrak{E}_t, \mathfrak{E}_o)$, если $W_o(\mathfrak{E}_t, \mathfrak{E}_o) \neq \emptyset$.

Это определение в дальнейшем будет сначала обобщено, а затем конкретизировано.

4. СИСТЕМЫ И ОПЕРАЦИИ НАД НИМИ.

4.1. Определение системы.

Всякое конечное, вполне упорядоченное множество мы будем называть системой. Пусть система A состоит из элементов $a_1, a_2, \ldots, a_n$, причем a_i предшествует a_j в том и только в том случае, когда $i < j$. В этом случае мы будем изображать систему A в виде $(a_1, a_2, \ldots, a_n)$. Таким образом, по определению $A = (a_1, a_2, \ldots, a_n)$.

4.2. Равенство систем

Будем говорить, что система A и система B равны, если они состоят из одних и тех же элементов и если для них задано одно и то же отношение порядка. Другими словами, если $A = (a_1, \ldots, a_n)$ и $B = (b_1, \ldots, b_m)$, то $A = B$ в том и только в том случае, если $a_1 = b_1, \ldots, a_n = b_m$.

4.3. Конкатенация систем.

4.3.1. Пусть даны две системы $A = (a_1, \ldots, a_n)$ и $B = (b_1, \ldots b_m)$. Конкатенацией систем A и B называется система

$$(a_1, \dots, a_n, b_1, \dots, b_m)$$

Знак операции конкатенации мы будем опускать (подобно тому, как это обычно делается для умножения).

Таким образом, по определению имеем:

$$AB = (a_1, \dots, a_n)(b_1, \dots, b_m) = (a_1, \dots, a_n, b_1, \dots, b_m)$$

4.3.2. Круглые скобки использованы нами для обозначения систем. Однако нам понадобятся скобки и для обозначения порядка выполнения операций. Чтобы не возникало двусмысленностей, порядок выполнения операций мы будем в дальнейшем обозначать только квадратными скобками.

4.3.3. Дальше нам часто придется использовать пустую систему, которая не содержит ни одного элемента. В наших обозначениях она записывается как $()$, однако это не всегда удобно, поэтому для пустой системы будем использовать обозначение $\emptyset$ наряду с $()$.

4.3.4. Очевидно, что для любой системы A имеет место свойство:

$$\emptyset A = A \emptyset = A$$

Кроме того, операция конкатенации ассоциативна, т.е. для любых систем A, B и C имеем

$$[AB]C = A[BC] = ABC$$

4.3.5. Пусть нам даны n систем $A_1, \dots, A_n$. Тогда для конкатенации $A_1 A_2 \dots A_n$ введем обозначение $\overset{n}{\underset{i=1}{C}} A_i$. Если $n = 0$, будем по определению считать, что $\overset{n}{\underset{i=1}{C}} A_i = \emptyset$. Таким образом

$$A_1 A_2 \dots A_n = \overset{n}{\underset{i=1}{C}} A_i$$

4.4. Операция $\circledast$

4.4.1. Пусть ϑ и ω – произвольные варианты, а V, V_1 и V_2 – произвольные системы, составленные из вариантов. Тогда бинарная операция $\circledast$ над системами вариантов аксиоматически определяется соотношениями:

I. $\varnothing \circledast V = V \circledast \varnothing = \varnothing$

II. $(\vartheta) \circledast (\omega) = (\vartheta \cup \omega)$

III. $(\vartheta) \circledast [(\omega) V] = [(\vartheta) \circledast (\omega)] [(\vartheta) \circledast V]$

IV. $[(\vartheta) V_1] \circledast V_2 = [(\vartheta) \circledast V_2] [V_1 \circledast V_2]$

4.4.2. Можно получить явное выражение для $V_x \circledast V_y$. Действительно, если $V_x = \overset{n}{\underset{i=1}{C}} (\vartheta_i)$; $V_y = \overset{m}{\underset{j=1}{C}} (\omega_j)$; то

$$V_x \circledast V_y = [\overset{n}{\underset{i=1}{C}} (\vartheta_i)] \circledast [\overset{m}{\underset{j=1}{C}} (\omega_j)] =$$

$$= \overset{n}{\underset{i=1}{C}} [(\vartheta_i) \circledast [\overset{m}{\underset{j=1}{C}} (\omega_j)]] = \overset{n}{\underset{i=1}{C}} \overset{m}{\underset{j=1}{C}} [(\vartheta_i) \circledast (\omega_j)] =$$

$$= \overset{n}{\underset{i=1}{C}} \overset{m}{\underset{j=1}{C}} (\vartheta_i \cup \omega_j).$$

4.4.3. Из ассоциативности конкатенации следует ассоциативность операции $\circledast$. Действительно, пусть

$V_x = \overset{n}{\underset{i=1}{C}} (\vartheta_i)$; $V_y = \overset{m}{\underset{j=1}{C}} (\omega_j)$; $V_z = \overset{k}{\underset{k=1}{C}} (u_k)$;

Тогда

$$[V_x \circledast V_y] \circledast V_z = V_x \circledast [V_y \circledast V_z] = V_x \circledast V_y \circledast V_z =$$

$$= \overset{n}{\underset{i=1}{C}} \overset{m}{\underset{j=1}{C}} \overset{k}{\underset{k=1}{C}} (\vartheta_i \cup \omega_j \cup u_k).$$

4.4.4. Нетрудно проверить справедливость свойств:

$$(\varnothing) \circledast V = V \circledast (\varnothing) = V$$

$$(\vartheta) \circledast [V_1 V_2] = [(\vartheta) \circledast V_1] [(\vartheta) \circledast V_2]$$

$$[V_1 V_2] \circledast V = [V_1 \circledast V] [V_2 \circledast V]$$

4.5. Операция фильтрации.

4.5.1. Введем унарную операцию фильтрации $F[V]$, которая определена на множестве всех систем вариантов. Пусть ϑ - произвольный вариант, а V - произвольная система вариантов. Тогда F определяется соотношениями:

I. $F[\varnothing] = \varnothing$

II. $F[(\vartheta)V] = F[(\vartheta)]\,F[V]$

III. $F[(\vartheta)] = \begin{cases} (\vartheta), & \text{если } \vartheta \text{ - непротиворечивый,} \\ \varnothing, & \text{если } \vartheta \text{ - противоречивый.} \end{cases}$

4.5.2. Легко убедиться, что выполняются следующие свойства:

$F[V_1 V_2] = F[V_1]\,F[V_2]$

$F[F[V]] = F[V]$

Если $F[(\vartheta)] = \varnothing$, то и $F[(\vartheta \cup \mathcal{W})] = \varnothing$

Если $V_x = \overset{n}{\underset{i=1}{\mathfrak{C}}}(\vartheta_i)$; $V_y = \overset{m}{\underset{j=1}{\mathfrak{C}}}(\mathcal{W}_j)$; то

$$F[V_x \circledast V_y] = \overset{n}{\underset{i=1}{\mathfrak{C}}}\ \overset{m}{\underset{j=1}{\mathfrak{C}}} F[(\vartheta_i \cup \mathcal{W}_j)]$$

4.5.3. Пусть ϑ и $\mathcal{W}$ - произвольные варианты. Тогда имеет место свойство:

$$F[F[(\vartheta)] \circledast (\mathcal{W})] = F[(\vartheta) \circledast F[(\mathcal{W})]] = F[(\vartheta \cup \mathcal{W})]$$

Доказательство проводится разбором случаев. Сначала покажем, что $F[F[(\vartheta)] \circledast (\mathcal{W})] = F[(\vartheta \cup \mathcal{W})]$. Поскольку всегда либо $F[(\vartheta)] = \varnothing$, либо $F[(\vartheta)] = (\vartheta)$ достаточно разобрать два случая.

Пусть $F[(\vartheta)] = \emptyset$. Тогда

$$F[F[(\vartheta)] \circledast (\omega)] = F[\emptyset \circledast (\omega)] = F[\emptyset] = \emptyset$$

С другой стороны $F[(\vartheta)] = \emptyset \Rightarrow F[(\vartheta \cup \omega)] = \emptyset$. Таким образом, в этом случае $F[F[(\vartheta)] \circledast (\omega)] = F[(\vartheta \cup \omega)] = \emptyset$.

Пусть $F[(\vartheta)] = (\vartheta)$. Тогда

$$F[F[(\vartheta)] \circledast (\omega)] = F[(\vartheta) \circledast (\omega)] = F[(\vartheta \cup \omega)]$$

Таким образом и в этом случае равенство выполняется.

Точно так же доказывается соотношение $F[(\vartheta) \circledast F[(\omega)]] = F[(\vartheta \cup \omega)]$.

4.6. Операция *

4.6.1. Пусть V_x и V_y произвольные системы вариантов. Тогда, по определению, имеет место соотношение

$$V_x * V_y = F[V_x \circledast V_y] .$$

4.6.2. Из свойств операции фильтрации и определения операции $\circledast$ сразу же получаем свойства:

I. $\emptyset * V = V * \emptyset = \emptyset$

II. $(\vartheta) * (\omega) = F[(\vartheta \cup \omega)]$

III. $(\vartheta) * [V_x V_y] = [(\vartheta) * V_1][(\vartheta) * V_2]$

IV. $[V_1 V_2] * V = [V_1 * V][V_2 * V]$

4.6.3. Из ассоциативности конкатенации и из свойства, доказанного в п.4,5.3 следует ассоциативность операции $*$. Действительно, пусть $V_x = \overset{n}{\underset{i=1}{C}}(\vartheta_i)$; $V_y = \overset{m}{\underset{j=1}{C}}(\omega_j)$; $V_z = \overset{\ell}{\underset{k=1}{C}}(u_k)$; Тогда

$$[V_x * V_y] * V_z = F[F[V_x \circledast V_y] \circledast V_z] =$$

$$= \overset{\ell}{\underset{k=1}{C}} \overset{n}{\underset{i=1}{C}} \overset{m}{\underset{j=1}{C}} F[F[(\vartheta_i \cup \omega_j)] \circledast (u_k)] =$$

$$= \overset{\ell}{\underset{k=1}{C}} \overset{n}{\underset{i=1}{C}} \overset{m}{\underset{j=1}{C}} F[(\vartheta_i \cup \omega_j \cup u_k)] =$$

$$= F[V_x \circledast V_y \circledast V_z]$$

Точно так же доказывается

$$V_x * [V_y * V_z] = F[V_x \circledast V_y \circledast V_z]$$

откуда следует ассоциативность операции $*$.

4.7. Операция inf

Эта операция определена на множестве непустых систем и представляет собой "взятие нижней грани" упорядоченного множества.

Пусть $V = \overset{n}{\underset{i=1}{C}}(\vartheta_i)$. Тогда по определению

$$inf V = \begin{cases} \vartheta_1 \text{ , если } n \geqslant 1 \\ \text{не определен, если } n = 0 \end{cases}$$

4.8 Операция $\times$

Бинарная операция $\times$ (не путать с декартовым произведением множеств) определена на множестве пар систем. Обе системы, к которым применяется $\times$, должны содержать одинаковое число элементов. При этих условиях $\times$ определяется из соотношений:

I. $\varnothing \times \varnothing = \varnothing$

II. $[(\vartheta) V_1] \times [(w) V_2] = ((\vartheta, w))[V_1 \times V_2]$

Из этих соотношений можно получить и явное выражение для результата операции $\times$. Пусть $A = \overset{n}{\underset{i=1}{C}}(a_i)$; $B = \overset{n}{\underset{i=1}{C}}(b_i)$; тогда

$$A \times B = \overset{n}{\underset{i=1}{C}}((a_i, b_i))$$

Т.е. результат - система, составленная из упорядоченных пар (a_i, b_i) :

$$((a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n))$$

5. ГЕНЕРАТОРЫ ПОРЯДКА.

5.1. Определение

5.1.1. Пусть a - произвольное объектное выражение, а $\mathcal{E}$ - произвольное конечное множество объектных выражений. Пусть E некоторое множество пар вида $(a, \mathcal{E})$, обладающее следующими свойствами:

I. $(a, \mathcal{E}) \in E \Rightarrow a \in \mathcal{E}$

II. $(a, \mathcal{E}) \in E$ и $a' \in \mathcal{E} \Rightarrow (a', \mathcal{E}) \in E$

III. $(a, \mathcal{E}) \in E$ и $a \in \mathcal{E}' \cup \mathcal{E}' \subset \mathcal{E} \Rightarrow (a, \mathcal{E}') \in E$

5.1.2. Генератором порядка на E называется функция $g(a,\mathcal{E})$, определенная на множестве пар $(a,\mathcal{E}) \in E$ и принимающая значения в множестве натуральных чисел, такая, что если $\mathcal{E}$ фиксировано и содержит n различных объектных выражений, то $g(a, \mathcal{E})$ устанавливает взаимно-однозначное соответствие между $a \in \mathcal{E}$ и элементами множества $\{1, 2, \ldots, n\}$.

5.2. Устойчивость

5.2.1. Генератор порядка $g(a, \mathcal{E})$ **устойчив**, если для любых $(a_1, \mathcal{E}) \in E$ и $(a_2, \mathcal{E}) \in E$ имеет место:

$$\forall \mathcal{E}' \; \mathcal{E}' \subset \mathcal{E} \; и \; g(a_1, \mathcal{E}) < g(a_2, \mathcal{E}) \Rightarrow g(a_1, \mathcal{E}') < g(a_2, \mathcal{E}')$$

Содержательно такое определение означает, что при "сжатии" множества $\mathcal{E}$ отношение порядка, которое задает $g(a,\mathcal{E})$, сохраняется.

5.2.2. Докажем тривиальную теорему.

Теорема 5.I. Если $(a_1,\varepsilon)\in E$ и $(a_2,\varepsilon)\in E$ и $g(a,\varepsilon)$ генератор порядка, устойчивый на E, то

$$\varepsilon'\subset\varepsilon \;и\; g(a_1,\varepsilon')<g(a_2,\varepsilon') \Rightarrow$$
$$\Rightarrow g(a_1,\varepsilon)<g(a_2,\varepsilon)$$

Доказательство. Проводится от противного. Предположим, что $\varepsilon'\subset\varepsilon, g(a_1,\varepsilon')<g(a_2,\varepsilon')$, но $g(a_1,\varepsilon)>g(a_2,\varepsilon)$. Из определения $g(a,\varepsilon)$ и $g(a_1,\varepsilon')<g(a_2,\varepsilon')$ следует, что $a_1\neq a_2$. Из $g(a_1,\varepsilon)>g(a_2,\varepsilon)$ и устойчивости $g(a,\varepsilon)$ следует, что $g(a_1,\varepsilon')>g(a_2,\varepsilon')$, что противоречит условию теоремы.

5.3. Δ - генераторы и R - генераторы порядка

5.3.I. Пусть множество E удовлетворяет следующему условию: если $(a,\varepsilon)\in E$, то все объектные выражения, входящие в ε имеют различную нуль-длину. Т.е. $a_1\neq a_2$ и

$$a_1\in\varepsilon \;и\; a_2\in\varepsilon \Rightarrow \Delta_0[a_1]\neq\Delta_0[a_2]$$

5.3.2. Пусть E удовлетворяет условию, сформулированному в п.5.3.I. Тогда назовем Δ -генератором, такой генератор порядка $g(a,\varepsilon)$, определенный на E, что для любых $(a_1,\varepsilon)\in E$ и $(a_2,\varepsilon)\in E$ имеет место

$$g(a_1,\varepsilon)<g(a_2,\varepsilon)\Leftrightarrow\Delta_0[a_1]<\Delta_0[a_2]$$

5.3.3. Пусть E удовлетворяет условию, сформулированному в п.5.3.I. Тогда назовем R - генератором такой генератор порядка $g(a,\varepsilon)$, определенный на E, что для любых $(a_1,\varepsilon)\in E$ и $(a_2,\varepsilon)\in E$ имеет место

$$g(a_1,\varepsilon)<g(a_2,\varepsilon)\Leftrightarrow\Delta_0[a_1]>\Delta_0[a_2]$$

5.3.4. Теорема 5.2. $\mathcal{L}$-генератор и $\mathcal{R}$ - генератор устойчивы. Доказательство тривиально.

5.4. Пример неустойчивого генератора порядка

Из определения генератора порядка видно, что он нумерует выражения, составляющие множество $\mathcal{E}$ (если, конечно, определен для этого множества). Так, например, $\mathcal{L}$ - генератор присваивает номер I самому короткому (в смысле нуль-длины) выражению из $\mathcal{E}$, затем номер 2 самому короткому из оставшихся и т.д.

Можно рассмотреть генератор порядка, который присваивает номер I самому короткому из выражений $\mathcal{E}$, затем номер 2 самому длинному из оставшихся, затем номер 3 самому короткому из оставшихся, затем - номер 4 самому длинному из еще непронумерованных и т.д. Этот генератор порядка неустойчив.

6. ОТОЖДЕСТВЛЕНИЕ СИСТЕМ.

6.I. Системы дыр и образов

6.I.I. Пусть нам даны n типовых выражений $\mathcal{E}_{t1}, \mathcal{E}_{t2}, \ldots, \mathcal{E}_{tn}$ и n объектных выражений $\mathcal{E}_{o1}, \mathcal{E}_{o2}, \ldots, \mathcal{E}_{on}$. Рассмотрим две системы

$$\Delta = (\mathcal{E}_{t1}, \ldots, \mathcal{E}_{tn}) = \overset{n}{\underset{i=1}{C}} (\mathcal{E}_{ti})$$

$$\Omega = (\mathcal{E}_{o1}, \ldots, \mathcal{E}_{on}) = \overset{n}{\underset{i=1}{C}} (\mathcal{E}_{oi})$$

Каждое из выражений $\mathcal{E}_{ti}$ мы будем называть дырой, а каждое выражение $\mathcal{E}_{oi}$ мы будем называть образом дыры $\mathcal{E}_{ti}$. Δ будем называть системой дыр, а Ω - системой образов.

Кроме того, будем рассматривать систему

$$\Lambda = \Delta \times \Omega = \overset{n}{\underset{i=1}{C}} ((\mathcal{E}_{ti}, \mathcal{E}_{oi}))$$

6.I.2. Будем говорить, что переменная x входит в систему дыр Δ, если переменная x входит хотя бы в одну дыру, принадлежащую Δ.

6.I.3. Обобщим понятие подстановки, данное в п.3.2.5. Пусть ϑ -непротиворечивый вариант и пусть все переменные, входящие в систему дыр Δ, входят и в вариант ϑ. Тогда определим операцию подстановки для варианта ϑ и системы дыр Δ следующими соотношениями:

I. $Q(\vartheta, \varnothing) = \varnothing$

II. $Q(\vartheta, (\xi_t)) = (Q_0(\vartheta, \xi_t))$

III. $Q(\vartheta, (\xi_t)\Delta) = Q(\vartheta, (\xi_t))\, Q(\vartheta, \Delta)$

Здесь ξ_t - произвольное типовое выражение, Δ произвольная системы дыр, а $Q_0(\vartheta, \xi_t)$ операция подстановки, определенная в п.3.2.5.

6.I.4. Пусть ρ - произвольный непротиворечивый вариант. Будем говорить, что ρ является частичным вариантом для непротиворечивого варианта ϑ, если $\rho \subset \vartheta$.

6.I.5. Пусть ρ - произвольный непротиворечивый вариант, а Δ - произвольная система дыр. Назовем вариант ϑ минимальным для ρ и Δ, если ϑ содержит только те переменные которые входят в ρ или в Δ, и при этом непротиворечив.

6.2. Обобщение понятия варианта отождествления

6.2.I. Вариант ϑ называется вариантом отождествления для системы дыр Δ, системы образов Ω и непротиворечивого варианта ρ, если подстановка $Q(\vartheta, \Delta)$ определена, $Q(\vartheta, \Delta) = \Omega$ и $\rho \subset \vartheta$.

6.2.2. Пусть $\Lambda = \Delta \times \Omega$. Тогда обозначим через $W(\rho, \Lambda)$ множество всех вариантов отождествления для Δ, Ω и ρ, кото-

рне минимальны для ρ и Δ. Другими словами:

$$W(\rho,\Lambda)=\{\vartheta:\vartheta\text{- минимальный для }\rho,\ \Delta\cup\Omega,\ Q(\vartheta,\Delta)=\Omega\cup\rho\subset\vartheta\}$$

6.2.3. Из определения $W(\rho,\Lambda)$ немедленно следует, что ρ является частичным вариантом для любого $\vartheta\in W(\rho,\Lambda)$. Поэтому в дальнейшем мы будем кратко называть ρ "частичным вариантом", подразумевая, что он является частичным для всех $\vartheta\in W(\rho,\Delta)$ вне зависимости от Λ.

6.2.4. Будем говорить, что вариант ϑ минимален относительно Δ, если он содержит только те переменные, которые входят в Δ. ϑ минимален для $\Lambda=\Delta\times\Omega$, если он минимален для Δ.

6.2.5. Пусть ρ минимален для Λ. Тогда любой $\vartheta\in W(\rho,\Delta)$ минимален для Λ. Кроме того, тогда всегда выполнено соотношение

$$W(\rho,\Lambda)\subset W(\emptyset,\Lambda)$$

В общем случае любой частичный вариант ρ можно представить в виде $\rho=\rho_0\cup\rho_1$, где ρ_0 - минимален для Λ и $\rho_0\cap\rho_1=\emptyset$. При этом, очевидно, что

$$\vartheta\in W(\rho_0,\Lambda)\Leftrightarrow\vartheta\cup\rho_1\in W(\rho_0\cup\rho_1,\Lambda)$$

Поэтому $W(\rho_0,\Lambda)$ и $W(\rho,\Lambda)$ содержат одинаковое количество вариантов отождествления.

6.2.6. Теорема 6.I. Множество $W(\rho,\Lambda)$ конечно для любых ρ и Λ.

Доказательство. Представим ρ в виде $\rho=\rho_0\cup\rho_1$, где ρ_0 минимальный для Λ и $\rho_0\cap\rho_1=\emptyset$. В силу замечаний предыдущего пункта

$$W(\rho,\Lambda)=W(\rho_0\cup\rho_1,\Lambda)\subset W(\rho_1,\Lambda)$$

Однако, $W(\rho_1,\Lambda)$ содержит столько же элементов, сколько

и $W(\varphi, \mathcal{A})$. Поэтому достаточно показать конечность $W(\varphi, \mathcal{A})$. Это можно сделать дословно повторяя рассуждения теоремы 3.I.

6.3. Обобщение понятия алгоритма отождествления

6.3.I. Обобщим понятие алгоритма отождествления, данное в 3.3.3. Будем называть алгоритмом отождествления правило, которое каждой совокупности $(\rho, \mathcal{A})$ ставит в соответствие один и только один вариант $\vartheta \in W(\rho, \mathcal{A})$ при условии, что $W(\rho, \mathcal{A}) \neq \emptyset$.

Другими словами, алгоритм отождествления - это функция $T(\rho, \mathcal{A})$, которая определена для всех $(\rho, \mathcal{A})$, для которых $W(\rho, \mathcal{A}) \neq \emptyset$, причем $T(\rho, \mathcal{A}) \in W(\rho, \mathcal{A})$

6.3.2. Очевидно, что определение алгоритма отождествления, данное в п.6.3.I включает в себя как частный случай определение, данное в п.3.3.3, поскольку

$$T_0(\mathcal{E}_t, \mathcal{E}_0) = T(\varphi, ((\mathcal{E}_t, \mathcal{E}_0)))$$

6.4. Упорядочивающие алгоритмы отождествления

6.4.I. Обозначим через $V(\rho, \mathcal{A})$ функцию, которая определена для каждой совокупности $(\rho, \mathcal{A})$ и значением которой для каждой совокупности $(\rho, \mathcal{A})$ является система, составленная из вариантов отождествления минимальных для ρ и $\mathcal{A}$. Другими словами $V(\rho, \mathcal{A}) = (\vartheta_1, \vartheta_2, \dots, \vartheta_n)$

и при этом

$$W(\rho, \mathcal{A}) = \{\vartheta_1, \vartheta_2, \dots, \vartheta_n\}$$

Таким образом, если $W(\rho, \mathcal{A})$ представляет собой неупорядоченное множество вариантов отождествления, то $V(\rho, \mathcal{A})$ - это уже некоторым образом упорядоченное множество, хотя и состоящее из тех же вариантов.

6.4.2. Будем говорить, что задан упорядочивающий алгоритм отождествления, если задана функция $V(\rho, \Lambda)$, а функция $T(\rho, \Lambda)$ определена равенством

$$T(\rho, \Lambda) = \inf V(\rho, \Lambda)$$

Очевидно, что $T(\rho, \Lambda)$ при этом определена тогда и только тогда, когда $W(\rho, \Lambda) \neq \emptyset$, поскольку

$$W(\rho, \Lambda) = \emptyset \iff V(\rho, \Lambda) = \emptyset .$$

В дальнейшем мы будем рассматривать только упорядочивающие алгоритмы отождествления, поэтому вместо "упорядочивающий алгоритм" будем говорить просто "алгоритм".

6.4.3. Будем говорить, что два алгоритма отождествления эквивалентны, если они задают одну и ту же функцию $V(\rho, \Lambda)$.

7. ПРОЕКТИРУЮЩИЕ АЛГОРИТМЫ ОТОЖДЕСТВЛЕНИЯ.

7.1. Слабая выводимость.

7.1.1. Назовем совокупность (ρ, Λ) точкой. Предположим, что задана совокупность правил вывода, таких что каждое правило вывода ставит в соответствие каждой точке (ρ, Λ), вообще говоря, несколько других точек. Если точке (ρ, Λ) ставится в соответствие точка (ρ', Λ') по одному из правил вывода, будем говорить, что точка (ρ', Λ') является непосредственным следствием точки (ρ, Λ).

7.1.2. Будем говорить, что точка (ρ', Λ') является слабым непосредственным следствием точки (ρ, Λ), если она является непосредственным следствием (ρ, Λ) по одному из правил вывода, перечисленных в этом пункте. Слабую непосредственную выводимость (ρ', Λ') из (ρ, Λ) будем обозначать через

$$(\rho, \Lambda) \xrightarrow{o} (\rho', \Lambda')$$

Опишем теперь правила вывода. Пусть ξ_t, ξ_t', ξ_t'' – произвольные типовые выражения, ξ_o, ξ_o', ξ_o'' – произвольные объектные

выражения. Пусть $\mathcal{X}$ - произвольная переменная, а $\mathcal{E}_{\mathcal{X}}$ - произвольное объектное выражение, которое является подходящим к переменной $\mathcal{X}$. Заданы следующие правила вывода:

PS. Ликвидация дыры

$$(\rho, \Lambda_x((\phi,\phi))\Lambda_y) \to (\rho, \Lambda_x\Lambda_y)$$

Правило PS заключается в том, что из Λ можно удалить произвольную пару (ϕ,ϕ), состоящую из пустой дыры и пустого образа.

PBL. Проектирование скобок левое

$$(\rho, \Lambda_x(((\mathcal{E}_t')\mathcal{E}_t'', (\mathcal{E}_o')\mathcal{E}_o''))\Lambda_y) \to$$
$$\to (\rho, \Lambda_x((\mathcal{E}_t', \mathcal{E}_o'), (\mathcal{E}_t'', \mathcal{E}_o''))\Lambda_y)$$

Правило сводится к тому, что любую пару из Λ, которая имеет вид $((\mathcal{E}_t')\mathcal{E}_t'', (\mathcal{E}_o')\mathcal{E}_o'')$ можно заменить на последовательность двух пар $(\mathcal{E}_t', \mathcal{E}_o')$ и $(\mathcal{E}_t'', \mathcal{E}_o'')$.

Здесь используется одно и то же обозначение для скобок, входящих в выражения и для скобок, изображающих системы. Мы не стали вводить для них различные обозначения, так как из контекста всегда будет ясно, о каких скобках идет речь.

PBR. Проектирование скобок правое

$$(\rho, \Lambda_x((\mathcal{E}_t'(\mathcal{E}_t''), \mathcal{E}_o'(\mathcal{E}_o'')))\Lambda_y) \to$$
$$\to (\rho, \Lambda_x((\mathcal{E}_t', \mathcal{E}_o'), (\mathcal{E}_t'', \mathcal{E}_o''))\Lambda_y)$$

PXC. Проектирование закрытого вхождения переменной

$$(\rho, \Lambda_x((\mathcal{X}, \mathcal{E}_{\mathcal{X}}))\Lambda_y) \to$$
$$\to (\rho \cup \{(\mathcal{X}, \mathcal{E}_{\mathcal{X}})\}, \Lambda_x((\phi,\phi))\Lambda_y)$$

Вхождение переменной называется закрытым, если оно принадлежит дыре, состоящей только из этой переменной. Правило PXC применимо только при условии, что $\rho \cup \{(x, \xi_x)\}$ непротиворечив.

PXL. Проектирование вхождения переменной левое

$$(\rho, \Lambda_x((x\,\xi_t, \xi_x\,\xi_0))\Lambda_y) \xrightarrow{0} \xrightarrow{0} (\rho \cup \{(x, \xi_x)\}, \Lambda_x((\xi_t, \xi_0))\Lambda_y)$$

Правило PXL применимо при условии, что $\rho \cup \{(x, \xi_x)\}$ непротиворечив.

PXR. Проектирование вхождения переменной правое

$$(\rho, \Lambda_x((\xi_t\,x, \xi_0\,\xi_x))\Lambda_y) \xrightarrow{0} \xrightarrow{0} (\rho \cup \{(x, \xi_x)\}, \Lambda_x((\xi_t, \xi_0))\Lambda_y)$$

Правило PXR применимо при условии, что $\rho \cup \{(x, \xi_x)\}$ непротиворечив.

Будем считать, что других правил вывода, кроме PS, PBL, PBR, PXC, PXL и PXR - нет.

7.I.3. Будем говорить, что точка (ρ'', Λ'') слабо выводится из точки (ρ', Λ'), если существует такая последовательность точек $(\rho_0, \Lambda_0), (\rho_1, \Lambda_1), \ldots, (\rho_n, \Lambda_n)$, что $(\rho', \Lambda') = (\rho_0, \Lambda_0)$, $(\rho'', \Lambda'') = (\rho_n, \Lambda_n)$ и для любого $i = 1, 2, \ldots, n$ имеет место $(\rho_{i-1}, \Lambda_{i-1}) \xrightarrow{0} (\rho_i, \Lambda_i)$. Если (ρ'', Λ'') слабо выводится из (ρ', Λ') будем писать $(\rho', \Lambda') \xrightarrow{*} (\rho'', \Lambda'')$, а последовательность точек $(\rho_0, \Lambda_0), \ldots, (\rho_n, \Lambda_n)$ будем называть слабым выводом (ρ'', Λ'') из (ρ', Λ').

7.I.4. Точку вида $(\rho, \emptyset)$ назовем концевой точкой, а всякий вывод вида $(\rho, \Lambda) \xrightarrow{*} (\vartheta, \emptyset)$ назовем концевым выводом.

7.I.5. Точку $(\rho, \mathcal{A})$ назовем слабо тупиковой точкой, если она не является концевой точкой и в то же время из нее нельзя получить ни одного слабого следствия. Вывод, оканчивающийся слабо тупиковой точкой, назовем слабо тупиковым.

7.2. Ранг $\mathcal{A}$

7.2.I. Каждой системе дыр Δ поставим в соответствие натуральное число $N[\Delta]$ в соответствии с соотношениями:

I. $N[\phi] = 0$

II. $N[(\phi)] = 1$

III. $N[(\mathfrak{C}_t)\Delta] = N[(\mathfrak{C}_t)] + N[\Delta]$

IУ. $N[(\alpha\, \mathfrak{C}_t)] = 1 + N[(\mathfrak{C}_t)]$

У. $N[((\mathfrak{C}'_t)\,\mathfrak{C}''_t)] = 1 + N[(\mathfrak{C}'_t)] + N[(\mathfrak{C}''_t)]$

Очевидно, что соотношения I–У однозначно определяют $N[\Delta]$. Назовем число $N[\Delta]$ рангом системы дыр Δ.

7.2.2. Пусть Δ – система дыр, а Ω – система образов. Пусть $\mathcal{A} = \Delta \times \Omega$. Тогда рангом системы $\mathcal{A}$ назовем ранг системы Δ. Таким образом ранг $\mathcal{A}$ не зависит от Ω. Обозначим $N[\mathcal{A}] = N[\Delta]$.

7.2.3. Нетрудно доказать следующие теоремы.

Теорема 7.I. Если $(\rho, \mathcal{A}) \xrightarrow{0} (\rho', \mathcal{A}')$, то ранг $\mathcal{A}'$ на единицу меньше ранга $\mathcal{A}$.

Теорема 7.2. Пусть задан концевой вывод $(\rho, \mathcal{A}) \xrightarrow{*} (\vartheta, \phi)$ Тогда длина этого вывода равна рангу $\mathcal{A}$.

Следствие. Все концевые выводы из точки $(\rho, \mathcal{A})$ имеют одинаковую длину.

7.3. Вывод вариантов

7.3.1. Будем говорить, что вариант ϑ слабо выводится из точки (ρ, Λ), если существует концевой вывод $(\rho, \Lambda) \xrightarrow{*} \xrightarrow{*} (\vartheta, \emptyset)$, а сам этот вывод будем называть выводом варианта ϑ.

7.3.2. Индукцией по рангу Λ легко показать справедливость следующей теоремы:

Теорема 7.3. Если вариант ϑ слабо выводится из точки (ρ, Λ), то $\vartheta \in W(\rho, \Lambda)$ и наоборот, для любого $\vartheta \in W(\rho, \Lambda)$ существует слабый вывод $(\rho, \Lambda) \xrightarrow{*} (\vartheta, \emptyset)$.

Интересно отметить, что теорема 7.3 остается справедливой, даже если мы не будем пользоваться правилами PBR, PXC, PXR. В этом смысле набор правил, перечисленных в п.7.1.2 избыточен. Более того, в оставшихся правилах PS, PBL, PXL можно положить $\Lambda_x = \emptyset$, и теорема 7.3 все равно сохраняет силу.

7.4. Сильная выводимость

7.4.1. Пусть $\Lambda_1 = \Delta_1 \times \mathfrak{N}_1$, $\Lambda_2 = \Delta_2 \times \mathfrak{N}_2$. Будем говорить, что точка (ρ_1, Λ_1) параллельна точке (ρ_2, Λ_2), если $\Delta_1 = \Delta_2$. Будем записывать параллельность точек (ρ_1, Λ_1) и (ρ_2, Λ_2) в виде $(\rho_1, \Lambda_1) \| (\rho_2, \Lambda_2)$. Будем говорить, что шаг $(\rho_1, \Lambda_1) \xrightarrow{\circ} (\rho_1', \Lambda_1')$ параллелен шагу $(\rho_2, \Lambda_2) \xrightarrow{\circ} (\rho_2', \Lambda_2')$, если

$$(\rho_1, \Lambda_1) \| (\rho_2, \Lambda_2) \quad \text{и} \quad (\rho_1', \Lambda_1') \| (\rho_2', \Lambda_2')$$

7.4.2. Введем теперь наряду с понятием слабой выводимости понятие сильной выводимости. Будем считать, что каждой точке (ρ, Λ) ставится в соответствие вообще говоря несколько других точек, которые мы будем называть сильными непосредственными следствиями этой точки. Если (ρ', Λ') - сильное непосредственное следствие точки (ρ, Λ) - будем писать

$$(\rho, \Lambda) \longrightarrow (\rho', \Lambda').$$

7.4.3. Будем говорить, что точка (ρ'', Λ'') сильно выводится из точки (ρ', Λ'), если существует такая последовательность точек $(\rho_0, \Lambda_0), \dots, (\rho_n, \Lambda_n)$, что $(\rho', \Lambda') = (\rho_0, \Lambda_0)$, $(\rho'', \Lambda'') = (\rho_n, \Lambda_n)$ и для любого $i = 1, 2, \dots, n$ $(\rho_{i-1}, \Lambda_{i-1}) \to (\rho_i, \Lambda_i)$. Если (ρ', Λ') сильно выводится из (ρ, Λ), будем писать $(\rho, \Lambda) \xrightarrow{\bullet} (\rho', \Lambda')$.

7.4.4. Будем считать, что отношение сильной выводимости удовлетворяет требованиям:

I. $(\rho, \Lambda) \to (\rho', \Lambda') \Rightarrow$
$\Rightarrow (\rho, \Lambda) \xrightarrow{\circ} (\rho', \Lambda')$

II. $\left.\begin{array}{l}(\rho, \Lambda) \to (\rho_1, \Lambda_1) \\ (\rho, \Lambda) \to (\rho_2, \Lambda_2)\end{array}\right\} \Rightarrow$
$\Rightarrow (\rho_1, \Lambda_1) \parallel (\rho_2, \Lambda_2)$

III. $\vartheta \in W(\rho, \Lambda) \Rightarrow$
Существует вывод: $(\rho, \Lambda) \xrightarrow{\bullet} (\vartheta, \emptyset)$

7.4.5. Из свойств, перечисленных в п.7.4.4 можно сделать следующие выводы. Свойство I означает, что сильная выводимость – всегда частный случай слабой выводимости.

Из свойства II следует, что все сильные непосредственные следствия из (ρ, Λ) всегда можно получить, пользуясь только одним правилом вывода и применяя это правило только к одной дыре. Из свойства III следует, что все $\vartheta \in W(\rho, \Lambda)$ можно получить, пользуясь только сильной выводимостью. Используя свойства I, II, III и проводя индукцию по рангу Λ, можно доказать следующую теорему:

Теорема 7.4. Любой сильный концевой вывод из (ρ, Λ) является выводом некоторого $\vartheta \in W(\rho, \Lambda)$. Для любого $\vartheta \in W(\rho, \Lambda)$ существует один и только один сильный вывод этого варианта из (ρ, Λ).

7.4.6. В дальнейшем нам придется иметь дело в основном с сильной выводимостью. Поэтому мы будем вместо "сильная выводимость" говорить просто "выводимость", а вместо "сильное следствие" просто "следствие" и т.д., особо оговаривая случаи, когда будет идти речь о слабой выводимости.

7.5. Упорядочение выводов

7.5.I. Назовем точку (ρ, Λ) тупиковой, если она не является концевой и если нет ни одного следствия из этой точки. Вывод, который оканчивается тупиковой точкой, назовем тупиковым.

7.5.2. Назовем точку однозначной, если существует ровно одно следствие из этой точки.

7.5.3. Назовем точку неоднозначной, если для нее существует больше, чем одно следствие.

7.5.4. Предположим теперь, что каждой переменной x, имеющей множество подходящих выражений $\mathfrak{A}_x$, поставлен в соответствие генератор порядка $g_x(a, \mathcal{E})$, определенный на множестве E пар $(a, \mathcal{E})$, таких, что $\mathcal{E}$ удовлетворяет хотя бы одному из условий:

I. $\mathcal{E}_1 \mathcal{E}_2 \in \mathfrak{A}_x$, $\mathcal{E}_1 \in \mathfrak{A}_x$, $\mathcal{E}_1 \mathcal{E}_2 \in \mathcal{E} \Rightarrow \mathcal{E}_1 \in \mathcal{E}$

II. $\mathcal{E}_1 \mathcal{E}_2 \in \mathfrak{A}_x$, $\mathcal{E}_2 \in \mathfrak{A}_x$, $\mathcal{E}_1 \mathcal{E}_2 \in \mathcal{E} \Rightarrow \mathcal{E}_2 \in \mathcal{E}$

Такие генераторы порядка существуют. Например, $\mathcal{L}$ - генератор и $\mathcal{R}$ - генератор обладают таким свойством для любого $\mathfrak{A}_x$.

7.5.5. Теперь рассмотрим произвольную точку (ρ, Λ). Упорядочим все следствия из этой точки следующим образом.

Если точка концевая, тупиковая или однозначная, то упорядочивать нечего, ибо имеется не более чем одно следствие из (ρ, Λ). Предположим теперь, что точка неоднозначная. Тогда согласно п.7.4.5 все следствия из (ρ, Λ) получаются либо с помощью правила $P\times L$, либо с помощью правила $P\times R$. Поэтому любое следствие из (ρ, Λ) имеет вид: $(\rho \cup \{(x, \xi_x)\}, \Lambda')$. Обозначим

$$\mathcal{E}(\rho, \Lambda) = \{\xi_x : (\rho, \Lambda) \to (\rho \cup \{(x, \xi_x)\}, \Lambda')\}.$$

Рассмотрим теперь два различных следствия из (ρ, Λ)

$$(\rho, \Lambda) \to (\rho \cup \{(x, \xi_x^{(1)})\}, \Lambda'^{(1)})$$

$$(\rho, \Lambda) \to (\rho \cup \{(x, \xi_x^{(2)})\}, \Lambda'^{(2)})$$

Будем считать, что следствие $(\rho \cup \{(x, \xi_x^{(1)})\}, \Lambda'^{(1)})$ левее, чем следствие $(\rho \cup \{(x, \xi_x^{(2)})\}, \Lambda'^{(2)})$, тогда и только тогда, когда

$$g_x(\xi_x^{(1)}, \mathcal{E}(\rho, \Lambda)) < g_x(\xi_x^{(2)}, \mathcal{E}(\rho, \Lambda))$$

7.5.6. Таким образом, все следствия из точки (ρ, Λ) можно перенумеровать, в соответствии с введенным в п.7.5.5. отношением порядка. j -е следствие из (ρ, Λ) обозначим через $(\rho'^{(j)}, \Lambda'^{(j)})$. Ясно, что по определению $j = g_x(\xi_x^{(j)}, \mathcal{E}(\rho, \Lambda))$ и

$$\rho'^{(j)} = \rho \cup \{(x, \xi_x^{(j)})\}$$

7.5.7. Теперь упорядочим все выводы из фиксированной точки (ρ_0, Λ_0). Будем говорить, что вывод $(\rho_0, \Lambda_0) \overset{\cdot}{\to} (\rho_1, \Lambda_1)$ левее, чем вывод $(\rho_0, \Lambda_0) \overset{\cdot}{\to} (\rho_2, \Lambda_2)$, если

$$(\rho_0, \Lambda_0) \overset{\cdot}{\to} (\rho, \Lambda) \to (\rho'^{(j_1)}, \Lambda'^{(j_1)}) \overset{\cdot}{\to} (\rho_1, \Lambda_1)$$

$$(\rho_0, \Lambda_0) \xrightarrow{\cdot} (\rho, \Lambda) \to (\rho'^{(j_2)}, \Lambda'^{(j_2)}) \xrightarrow{\cdot} (\rho_2, \Lambda_2)$$

и при этом $j_1 < j_2$

7.5.8. Пользуясь определением п.7.5.7 мы теперь можем упорядочить все концевые вывод^{ы} из (ρ, Λ), а значит и все $\vartheta \in W(\rho, \Lambda)$

7.6. Определение проектирующего алгоритма отождествления

7.6.I. Согласно п.6.4.2, для того, чтобы определить упорядочивающий алгоритм отождествления, достаточно задать способ вычисления $V(\rho, \Lambda)$. Мы определим $V(\rho, \Lambda)$ явными рекурсивными соотношениями.

I. $V(\rho, \emptyset) = (\rho)$

II. $V(\rho, \Lambda) = \overset{\ell}{\underset{j=1}{C}} V(\rho'^{(j)}, \Lambda'^{(j)})$

если (ρ, Λ) не концевая точка, из этой точки выводится ровно ℓ непосредственных следствий, и $(\rho'^{(j)}, \Lambda'^{(j)})$ есть j-е непосредственное следствие из (ρ, Λ).

Случай II формально включает в себя случаи тупиковой $(\ell = 0)$ однозначной $(\ell = 1)$ и неоднозначной $(\ell > 1)$ точки.

7.6.2. Алгоритм отождествления, определенный п.7.6.I. мы будем называть проектирующим алгоритмом отождествления.

7.6.3. Очевидно, что если $\vartheta_1 \in W(\rho, \Lambda)$, $\vartheta_2 \in W(\rho, \Lambda)$ и при этом ϑ_1 предшествует ϑ_2 в $V(\rho, \Lambda)$, то вывод $(\rho, \Lambda) \xrightarrow{\cdot} (\vartheta_1, \emptyset)$ левее, чем вывод $(\rho, \Lambda) \xrightarrow{\cdot} (\vartheta_2, \emptyset)$ и наоборот (при условии, что $V(\rho, \Lambda)$ определено согласно п.7.6.I). Этим мы будем иногда пользоваться в дальнейшем.

7.7. Замечания

Предыдущими пунктами введены два понятия выводимости: сильной и слабой. Можно, конечно, было сразу определить понятие сильной выводимости, однако следующие соображения могут "оправдать" слабую выводимость. Ясно, что любой проектирующий алгоритм отождествления полностью задается установленным для него отношением сильной выводимости и тем, какие генераторы порядка сопоставлены переменным. Однако, всегда сильная выводимость оказывается частным случаем слабой выводимости. Тем самым слабая выводимость характеризует особенности, общие для всех мыслимых проектирующих алгоритмов отождествления. Между тем, выбор того или иного отношения сильной выводимости во многом произволен. В следующих разделах формулируются некоторые условия, при которых проектирующие алгоритмы отождествления являются эквивалентными. При этом оказывается, что в известных пределах отношение сильной выводимости можно менять произвольно, не меняя $V(\rho, \mathcal{A})$. Ясно, что в этом случае выбор того или иного отношения сильной выводимости во многом является делом соглашения.

8. УСТОЙЧИВОСТЬ

8.1. Определение устойчивости

Будем говорить, что алгоритм отождествления устойчив, если для любой точки $(\rho, \mathcal{A})$ выполнено равенство

$$V(\rho, \mathcal{A}) = (\rho) * V(\emptyset, \mathcal{A})$$

Наша ближайшая задача - найти достаточные условия, при которых проектирующий алгоритм отождествления устойчив.

8.2. Принцип параллельности

8.2.1. Пусть ρ - некоторый вариант. Обозначим через $X(\rho)$ множество тех переменных, которые входят в ρ. Иначе

$$X(\rho) = \{x : \exists a\ (x,a) \in \rho\}$$

8.2.2. Будем говорить, что алгоритм отождествления удовлетворяет принципу параллельности, если выполнены два условия:

I. Пусть $(\rho_1, \Lambda_1) \to (\rho_1', \Lambda_1')$ и $(\rho_2, \Lambda_2) \to (\rho_2', \Lambda_2')$ Тогда, если $X(\rho_1) = X(\rho_2)$ и $(\rho_1, \Lambda_1) \| (\rho_2, \Lambda_2)$, то и $(\rho_1', \Lambda_1') \| (\rho_2', \Lambda_2')$.

II. Пусть δ такой вариант, который содержит только такие переменные, которые не входят в Λ. Тогда

$$(\rho, \Lambda) \to (\rho', \Lambda') \Leftrightarrow (\rho \cup \delta, \Lambda) \to (\rho' \cup \delta, \Lambda')$$

8.2.3. В п.7.4.I было дано определение параллельности двух шагов. Определим теперь параллельность выводов. Два вывода $(\rho_1, \Lambda_1) \overset{*}{\to} (\rho_1', \Lambda_1')$ и $(\rho_2, \Lambda_2) \overset{*}{\to} (\rho_2', \Lambda_2')$ параллельны, если они имеют одинаковую длину n и если для любого $k \le n$ k-й шаг первого вывода параллелен k-му шагу второго вывода. Отношение параллельности выводов рефлексивно, симметрично и транзитивно.

8.2.4. Если выполняется принцип параллельности, то все выводы одинаковой длины из фиксированной точки (ρ, Λ) - параллельны. В частности, все концевые выводы из точки (ρ, Λ) параллельны.

8.2.5. Пусть δ содержит только такие переменные, которые не входят в Λ. Тогда из принципа параллельности следует, что если $(\rho, \Lambda) \overset{*}{\to} (\rho', \Lambda')$, то и $(\rho \cup \delta, \Lambda) \overset{*}{\to} (\rho' \cup \delta, \Lambda')$ и наоборот. Причем эти выводы параллельны между собой.

8.3. Вхождения переменных

8.3.I. Рассмотрим произвольную систему дыр Δ. Ясно, что в общем случае в Δ может входить несколько различных переменных, причем каждая переменная может входить в Δ несколько раз. Перенумеруем слева направо все вхождения всех переменных в Δ

(не обращая внимания, к какой из переменных относится каждое вхождение). Обозначим k-е по счету вхождение через y_k. Число k назовем геометрическим номером вхождения y_k.

8.3.2. Пусть y - некоторое вхождение некоторой переменной x, и пусть переменной x поставлен в соответствие генератор порядка $g_x(a,\varepsilon)$. Тогда будем говорить, что генератор порядка $g_x(a,\varepsilon)$ поставлен в соответствие вхождению y. Генератор порядка, поставленный в соответствие вхождению y_k, в дальнейшем будем обозначать через $h_k(a,\varepsilon)$.

8.3.3. Рассмотрим произвольную точку (ρ_0, Λ_0), где $\Lambda_0 = \Delta_0 \times \Omega_0$, и все концевые выводы из этой точки: $(\rho_0, \Lambda_0) \dot{\to} \dot{\to} (\vartheta, \varphi)$. Пусть имеется n вхождений переменных в Δ_0: $y_1, y_2, \ldots, y_n$. Будем говорить, что вхождение y_k принимает значение b_k для варианта ϑ, если y_k является вхождением переменной x и $(x, b_k) \in \vartheta$. Ясно, что для каждого $\vartheta \in W(\rho_0, \Lambda_0)$ каждое вхождение принимает какое-то значение. И наоборот, если известны значения всех вхождений, по ним однозначно можно восстановить вариант ϑ.

8.3.4. Пусть $b_1, b_2, \ldots, b_n$ значения вхождений $y_1, y_2, \ldots, y_n$ Составим систему $(b_1, b_2, \ldots, b_n)$. Как было отмечено в предыдущем пункте, если (ρ_0, Λ_0) фиксирована, то существует взаимно-однозначное соответствие между вариантами $\vartheta \in W(\rho_0, \Lambda_0)$ и системами значений $(b_1, b_2, \ldots, b_n)$. Обозначим $\beta = (b_1, \ldots, b_n)$ Пусть $V(\rho_0, \Lambda_0) = (\vartheta_1, \ldots, \vartheta_m)$. Тогда обозначим $B(\rho_0, \Lambda_0) = (\beta_1, \ldots, \beta_m)$, где β_k соответствует варианту ϑ_k. Таким образом можно заменить упорядочение вариантов отождествления, упорядочением систем значений вхождений.

8.3.5. Назовем точку (ρ, Λ) точкой проектирования переменной x, если все следствия из (ρ, Λ) могут быть получены посредством одного из правил PXC, PXL, PXR и любое следствие имеет вид $(\rho \cup \{(x, \xi_x)\}, \Lambda')$.

8.3.6. Предположим теперь, что выполняется принцип параллельности и рассмотрим все концевые выводы из фиксированной точки (ρ_0, Λ_0). Любой концевой вывод имеет вид $(\rho_0, \Lambda_0), (\rho_1, \Lambda_1), \ldots, (\rho_m, \Lambda_m)$, где $(\rho_m, \Lambda_m) = (\vartheta, \emptyset)$. Причем, если $\Lambda_i = \Delta_i \times \Omega_i$, то последовательность Δ_i одинакова для всех концевых выводов. Теперь выберем из последовательности точек $(\rho_0, \Lambda_0), \ldots, (\rho_m, \Lambda_m)$ подпоследовательность точек, являющихся точками проектирования какой-либо переменной: $(\rho_{i_1}, \Lambda_{i_1})$, $(\rho_{i_2}, \Lambda_{i_2}), \ldots, (\rho_{i_n}, \Lambda_{i_n})$, где n равно числу вхождений переменных в Δ_0. В силу принципа параллельности, подпоследовательность индексов i_ℓ одинакова для всех концевых выводов. Поэтому обозначим $(\rho_{i_\ell}, \Lambda_{i_\ell}) = (\rho^\ell, \Lambda^\ell)$.

8.3.7. Теперь введем понятие проекционного номера вхождения. Будем говорить, что вхождение y имеет проекционный номер ℓ, если для точки $(\rho^\ell, \Lambda^\ell)$ вхождение y входит в Λ^ℓ, но в то же время, если $(\rho^\ell, \Lambda^\ell) \to (\rho', \Lambda')$, то y не входит в Λ'. Такое определение корректно, если выполняется принцип параллельности.

8.3.8. Таким образом, каждому вхождению ставятся во взаимно однозначное соответствие геометрический номер этого вхождения и его проекционный номер. Но это означает, что между проекционными и геометрическими номерами существует взаимно однозначное соответствие. Обозначим через $p(\ell)$ тот геометрический номер, который соответствует проекционному номеру ℓ. Теперь мы можем выписать все вхождения в порядке возрастания их проекционных номеров:

$$y_{p(1)}, y_{p(2)}, \ldots, y_{p(n)}$$

8.3.9. Теперь мы можем явно задать отношение порядка для всех $\beta \in B(\rho_0, \Lambda_0)$. Действительно, если (ρ, Λ) - неоднозначная точка, то она обязательно является точкой проектирования переменной. Определим в каждой точке проектирования переменной множество $\mathcal{E}(\rho, \Lambda) = \{\mathcal{E}_x : (\rho, \Lambda) \to (\rho \cup \{(x, \mathcal{E}_x)\}, \Lambda')\}$
Если вариант ϑ' предшествует варианту ϑ'', то вывод $(\rho_0, \Lambda_0) \dot{\to} (\vartheta', \emptyset)$ левее, чем вывод $(\rho_0, \Lambda_0) \dot{\to} (\vartheta'', \emptyset)$
Но это значит, что существует такая точка $(\rho^\ell, \Lambda^\ell)$, что

$$(\rho_0, \Lambda_0) \dot{\to} (\rho^\ell, \Lambda^\ell)$$
$$(\rho^\ell, \Lambda^\ell) \to (\rho^\ell \cup \{(x, \mathcal{E}'_x)\}, \Lambda'^\ell) \dot{\to} (\vartheta', \emptyset)$$
$$(\rho^\ell, \Lambda^\ell) \to (\rho^\ell \cup \{(x, \mathcal{E}''_x)\}, \Lambda''^\ell) \dot{\to} (\vartheta'', \emptyset)$$

и при этом $q_x(\mathcal{E}'_x, \mathcal{E}(\rho^\ell, \Lambda^\ell)) < q_x(\mathcal{E}''_x, \mathcal{E}(\rho^\ell, \Lambda^\ell))$. Пусть при этом варианту ϑ' соответствует $(\beta'_1, \beta'_2, \ldots, \beta'_n) \in B(\rho_0, \Lambda_0)$ а варианту ϑ'' соответствует $(\beta''_1, \beta''_2, \ldots, \beta''_n) \in B(\rho_0, \Lambda_0)$. Очевидно, что $\beta'_{p(1)} = \beta''_{p(1)}, \ldots, \beta'_{p(\ell-1)} = \beta''_{p(\ell-1)}$, но $\beta'_{p(\ell)} \neq \beta''_{p(\ell)}$
Причем $h_{p(\ell)}(\beta'_{p(\ell)}, \mathcal{E}(\rho^\ell, \Lambda^\ell)) < h_{p(\ell)}(\beta''_{p(\ell)}, \mathcal{E}(\rho^\ell, \Lambda^\ell))$. Кроме того, очевидно, что точка $(\rho^\ell, \Lambda^\ell)$ однозначно определяется точкой (ρ_0, Λ_0) и значениями $\beta_{p(1)}, \beta_{p(2)}, \ldots, \beta_{p(\ell-1)}$, поэтому можно рассматривать $\mathcal{E}(\rho^\ell, \Lambda^\ell)$ как функцию от ρ_0, Λ_0 и $\beta_{p(1)}, \ldots, \beta_{p(\ell-1)}$. Обозначим

$$\mathcal{E}_{p(\ell)}(\rho_0, \Lambda_0, \beta_{p(1)}, \ldots, \beta_{p(\ell-1)}) = \mathcal{E}(\rho^\ell, \Lambda^\ell)$$

8.3.10. Таким образом, $\beta' = (\beta'_1, \ldots, \beta'_n)$ предшествует $\beta'' = (\beta''_1, \ldots, \beta''_n)$ в том и только том случае, если существует такое ℓ, что $1 \leq \ell \leq n$, $\beta_{p(1)} = \beta''_{p(2)}, \ldots, \beta_{p(\ell-1)} = \beta''_{p(\ell-1)}$ но $\beta'_{p(\ell)} \neq \beta''_{p(\ell)}$ и

$$h_{p(\ell)}(\beta'_{p(\ell)}, \mathcal{E}_{p(\ell)}(\rho_0, \Lambda_0, \beta_{p(1)}, \ldots, \beta_{p(\ell-1)})) <$$
$$< h_{p(\ell)}(\beta''_{p(\ell)}, \mathcal{E}_{p(\ell)}(\rho_0, \Lambda_0, \beta_{p(1)}, \ldots, \beta_{p(\ell-1)}))$$

8.4. Левосторонние алгоритмы отождествления

8.4.1. Назовем вхождение y_k неоднозначным, если существует такая точка проектирования вхождения переменной y_k $(\rho^\ell, \lambda^\ell)$, что $k = p(\ell)$, $(\rho_0, \lambda_0) \to (\rho^\ell, \lambda^\ell)$ и при этом точка $(\rho^\ell, \lambda^\ell)$ неоднозначная. Если вхождение не является неоднозначным, будем называть его однозначным.

8.4.2. Будем называть левосторонними такие алгоритмы отождествления, для которых каждое неоднозначное вхождение проектируется после того, как все вхождения с меньшими, чем у него геометрическими номерами уже спроектированы. Другими словами:

$y_{p(\ell)}$ - неоднозначное и $p(\ell') < p(\ell) \Rightarrow \ell' < \ell$

8.4.3. Теорема 8.1. Пусть $(b_1, \dots, b_m) \in B(\rho_0, \lambda_0)$ и пусть $y_{p(\ell)}$ - однозначное вхождение. Тогда $b_{p(\ell)}$ однозначно определяется значениями неоднозначных вхождений $b_{p(\ell')}$, для которых $\ell' < \ell$.

Доказательство. Во-первых, заметим, что $b_{p(\ell)} \in \mathcal{E}_{p(\ell)}(\rho_0, \lambda_0, b_{p(1)}, \dots b_{p(\ell-1)})$. А если $y_{p(\ell)}$ - однозначное вхождение, то $\mathcal{E}_{p(\ell)}$ состоит только из одного элемента. Поэтому $b_{p(\ell)}$ можно рассматривать как функцию от $\rho_0, \lambda_0, b_{p(1)}, \dots, b_{p(\ell-1)}$. Т.е. $b_{p(\ell)} = b_{p(\ell)}(\rho_0, \lambda_0, b_{p(1)}, \dots, b_{p(\ell-1)})$. Теперь докажем теорему индукцией по количеству однозначных вхождений, имеющих проекционные номера меньшие, чем ℓ.

Действительно, если все $y_{p(1)}, \dots, y_{p(\ell-1)}$ - неоднозначные, то утверждение теоремы выполнено. Предположим, что среди $y_{p(1)}, \dots, y_{p(\ell-1)}$ есть однозначные вхождения. Тогда, по индуктивному предположению значение каждого из этих вхождений - функция от значений неоднозначных вхождений с меньшими проекционными номерами. Поэтому

и $\mathcal{B}_{p(\ell)}$ можно рассматривать как функцию от значений только неоднозначных входжений с меньшими проекционными номерами.

8.4.4. Теорема 8.2. Пусть $(\mathcal{B}_1,\dots,\mathcal{B}_n)\in B(\rho_0,\Lambda_0)$. Тогда $\mathcal{E}_{p(\ell)}$ однозначно определяется значениями неоднозначных входжений $\mathcal{B}_{p(\ell')}$, для которых $\ell'<\ell$.

Показательство. Действительно, $\mathcal{E}_{p(\ell)}=\mathcal{E}_{p(\ell)}(\rho_0,\Lambda_0,\mathcal{B}_{p(1)},\dots,\mathcal{B}_{p(\ell-1)})$ Причем среди входжений $y_{p(1)},\dots,y_{p(\ell-1)}$ есть как однозначные, так и неоднозначные. Однако, по теореме 8.1. мы можем рассматривать значения однодначных входжений как функции от значений неоднозначных входжений с меньшими, чем у них проектионными номерами. Отсюда следует утверждение теоремы.

8.4.5. Теорема 8.3. Пусть $(\mathcal{B}_1,\dots,\mathcal{B}_n)\in B(\rho_0,\Lambda_0)$ и пусть алгоритм отождествления левосторонний. Тогда $\mathcal{E}_{p(\ell)}$ однозначно определяется значениями тех входжений $y_{p(\ell')}$, для которых $p(\ell')<p(\ell)$

Показательство. Рассмотрим все неоднозначные входжения, проекционные номера которых меньше, чем ℓ. Обозначим их проекционные номера $\ell_1,\ell_2,\dots,\ell_m$. Таким образом, для всех $i=1,2,\dots,m$ $y_{p(\ell_i)}$ – неоднозначное и $\ell_i<\ell$.

По теореме 8.2. мы имеем

$$\mathcal{E}_{p(\ell)}=\mathcal{E}_{p(\ell)}(\rho_0,\Lambda_0,\mathcal{B}_{p(\ell_1)},\dots,\mathcal{B}_{p(\ell_m)}).$$

Покажем, что для $i=1,2,\dots,m$ имеет место $p(\ell_i)<p(\ell)$ В самом деле, если бы хотя бы для одного i имело место $p(\ell_i)>p(\ell)$, то из левосторонности алгоритма отождествления следовало бы, что $\ell_i>\ell$, что противоречит $\ell_i<\ell$ для всех $i=1,2,\dots,m$. Значит $p(\ell_i)<p(\ell)$ для всех $i=1,2,\dots,m$

8.4.6. Из теоремы 8.3. получаем следующие следствия:

Следствие 8.3.1. $\mathcal{E}_k$ однозначно определяется значениями вхождений $\mathcal{B}_{k'}$, для которых $k' < k$. Т.е.

$$\mathcal{E}_k = \mathcal{E}_k(\rho_0, \Lambda_0, \mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{k-1})$$

Следствие 8.3.2. Если $\mathcal{Y}_k$ - однозначное вхождение, то его значение $\mathcal{B}_k$ однозначно определяется значениями вхождений $\mathcal{B}_{k'}$, для которых $k' < k$. Т.е.

$$\mathcal{B}_k = \mathcal{B}_k(\rho_0, \Lambda_0, \mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{k-1})$$

8.5. Эквивалентность и устойчивость левосторонних алгоритмов отождествления.

8.5.1. Пусть (ρ, Λ) - точка проектирования переменной. Обозначим через $\mathcal{E}^0(\rho, \Lambda)$ множество объектных выражений

$$\mathcal{E}^0(\rho, \Lambda) = \{\mathcal{E}_x : (\rho, \Lambda) \to (\rho \cup \{(x, \mathcal{E}_x)\}, \Lambda') \\ \text{и } \exists \vartheta (\rho \cup \{(x, \mathcal{E}_x)\}, \Lambda') \to (\vartheta, \emptyset)\}.$$

Здесь по сравнению с $\mathcal{E}(\rho, \Lambda)$ наложено дополнительное ограничение: рассматриваются только те следствия из (ρ, Λ), из которых выходит хотя бы один концевой вывод. Ясно, что всегда

$$\mathcal{E}^0(\rho, \Lambda) \subset \mathcal{E}(\rho, \Lambda).$$

8.5.2. Теперь мы можем дословно повторить рассуждения предыдущих теорем, заменив везде $\mathcal{E}(\rho^\ell, \Lambda^\ell)$ на $\mathcal{E}^0(\rho^\ell, \Lambda^\ell)$ и обозначив $\mathcal{E}^0(\rho^\ell, \Lambda^\ell) = \mathcal{E}^0_{p(\ell)}(\rho_0, \Lambda_0, \mathcal{B}_{p(0)}, \ldots, \mathcal{B}_{p(\ell)})$ и доказав следующую теорему:

Теорема 8.4. Пусть $(\mathcal{B}_1, \ldots, \mathcal{B}_n) \in B(\rho_0, \Lambda_0)$ и пусть алгоритм отождествления левосторонний. Тогда $\mathcal{E}^0_{p(\ell)}$ однозначно определяется значениями тех вхождений $\mathcal{Y}_{p(\ell')}$, для которых $p(\ell') < p(\ell)$

Следствие 8.4.1. $\mathcal{E}^0_k$ однозначно определяется значениями вхождений $\mathcal{B}_{k'}$, для которых $k' < k$. Т.е.

$$\mathcal{E}^0_k = \mathcal{E}^0_k(\rho_0, \Lambda_0, \mathcal{B}_1, \ldots, \mathcal{B}_{k-1})$$

Следствие 8.4.2. Множества $\mathcal{E}^0_k(\rho_0, \lambda_0, b_1, \dots, b_{k-1})$ одинаковы для всех левосторонних алгоритмов отождествления.

$$\mathcal{E}^0_k(\rho_0, \lambda_0, b_1, \dots, b_{k-1}) = \{b_k : (b_1, \dots, b_k, \dots, b_m) \in B(\rho_0, \lambda_0)\}.$$

8.5.3. Теорема 8.5. Пусть $\beta' \in B(\rho_0, \lambda_0)$ и $\beta'' \in B(\rho_0, \lambda_0)$, где $\beta' = (b'_1, \dots, b'_m)$ и $\beta'' = (b''_1, b''_2, \dots, b''_m)$. Тогда, если алгоритм отождествления левосторонний и если все $h_k(b, \varepsilon)$, устойчивы, то для того, чтобы β' предшествовала β'', необходимо и достаточно, чтобы существовало такое k, что: $b'_1 = b''_1$, $b'_2 = b''_2, \dots, b'_{k-1} = b''_{k-1}$ и в то же время $b'_k \neq b''_k$ и

$$h_k(b'_k, \mathcal{E}^0_k(\rho_0, \lambda_0, b'_1, \dots, b'_{k-1})) < h_k(b''_k, \mathcal{E}^0_k(\rho_0, \lambda_0, b'_1, \dots, b^*_{k-1}))$$

Доказательство. Сначала заметим, что в силу устойчивости генераторов порядка $h_k(b, \varepsilon)$ неравенство

$$h_k(b'_k, \mathcal{E}^0_k(\rho_0, \lambda_0, b'_1, \dots, b'_{k-1})) < h_k(b''_k, \mathcal{E}^0_k(\rho_0, \lambda_0, b'_1, \dots, b'_{k-1}))$$

выполняется, если только выполнено

$$h_k(b'_k, \mathcal{E}_k(\rho_0, \lambda_0, b'_1, \dots, b'_{k-1})) < h_k(b''_k, \mathcal{E}_k(\rho_0, \lambda_0, b'_1, \dots, b'_k))$$

Теперь докажем необходимость. Действительно, если β' предшествует β'', то согласно п.8.3.10 существует ℓ, такое, что $b'_{p(1)} = b''_{p(1)}, \dots, b'_{p(\ell-1)} = b''_{p(\ell-1)}$ и $h_{p(\ell)}(b'_{p(\ell)}, \mathcal{E}_{p(\ell)}) < h_{p(\ell)}(b''_{p(\ell)}, \mathcal{E}_{p(\ell)})$

Пусть $p(\ell) = k$. Из $b'_k \neq b''_k$ следует, что γ_k – неоднозначное вхождение. Пусть ℓ' – такое, что $\ell' < \ell$, $p(\ell') = k'$ и $\gamma_{k'}$ – неоднозначное вхождение. Тогда из

левосторонности алгоритма отождествления следует, что $p(\ell') < p(\ell)$. Отсюда получаем, что если $k' < k$ и $y_{k'}$ – неоднозначное вхождение, то $b'_{k'} = b''_{k'}$. Теперь вспомним, что для любого однозначного $y_{k'}$ имеем $b_{k'} = b_{k'}(\rho_0, \lambda_0, b_{k_1}, \ldots, b_{k_m})$, где $k_1, \ldots, k_m$ – геометрические номера неоднозначных вхождений с меньшими геометрическими номерами. Но для всех k_i $b'_{k_i} = b''_{k_i}$, как это было только что показано. Значит $b'_{k'} = b''_{k'}$. Таким образом для любого $k' < k$ имеем $b'_{k'} = b''_{k'}$. Необходимость доказана.

Докажем достаточность. Пусть $b'_1 = b''_1, \ldots, b'_{k-1} = b''_{k-1}$ и $h_k(b'_k, \varepsilon_k) < h_k(b''_k, \varepsilon_k)$. Пусть ℓ такое число, что $k = p(\ell)$. Тогда, из левосторонности алгоритма имеем: $y_{p(\ell')}$ – неоднозначное и $\ell' < \ell \Rightarrow p(\ell') < p(\ell)$. Отсюда следует, что если $y_{p(\ell')}$ – неоднозначное, то $b'_{p(\ell')} = b''_{p(\ell')}$, поскольку $p(\ell') < k$. Покажем, что для любого $\ell' < \ell$ имеет место $b'_{p(\ell')} = b''_{p(\ell')}$. Действительно, если $\ell_1, \ldots, \ell_m$ – проекционные номера, неоднозначных вхождений, для которых $\ell_i < \ell'$ и $y_{p(\ell')}$ – однозначное вхождение, для которого $\ell' < \ell$, то $b_{p(\ell')} = b_{p(\ell')}(\rho_0, \lambda_0, b_{p(\ell_1)}, \ldots, b_{p(\ell_m)})$. Но $b'_{p(\ell_i)} = b''_{p(\ell_i)}$, как было только что показано, отсюда и $b'_{p(\ell')} = b''_{p(\ell')}$. Таким образом $b'_{p(1)} = b''_{p(1)}, \ldots, b'_{p(\ell-1)} = b''_{p(\ell-1)}$,

$$h_{p(\ell)}(b'_{p(\ell)}, \varepsilon_{p(\ell)}) < h_{p(\ell)}(b''_{p(\ell)}, \varepsilon_{p(\ell)})$$

Достаточность доказана. Теорема доказана.

8.5.4. Теорема 8.6. Два левосторонних алгоритма отождествления эквивалентны, если одним и тем же переменным они ставят в соответствие одни и те же устойчивые генераторы порядка.

Доказательство. Согласно следствию 8.4.2 множества $\varepsilon^0_k(\rho_0, \lambda_0, b_1, \ldots, b_{k-1})$ не зависят от алгоритма отождествления, поскольку $W(\rho_0, \lambda_0)$ одинаково для всех

алгоритмов отождествления, а значит и $B(\rho_0, \Lambda_0)$ составлено из одних и тех же элементов для любого алгоритма отождествления. Теперь воспользуемся теоремой 8.5. В ней доказано, что для любого левостороннего алгоритма отождествления упорядочение по проекционным номерам можно заменить на упорядочение по геометрическим номерам. А геометрические номера от алгоритма отождествления не зависят. Теорема доказана.

8.5.5. Теорема 8.7. Любой левосторонний алгоритм отождествления, сопоставляющий переменным устойчивые генераторы порядка - устойчив.

Доказательство. Нужно показать, что для любых ρ и Λ

$$V(\rho, \Lambda) = (\rho) * V(\phi, \Lambda)$$

Сначала заметим, что ρ представим в виде $\rho = \rho_0 \cup \rho_1$, где ρ_0 - минимальный для Λ и $\rho_1 \cap \rho_0 = \phi$. Из принципа параллельности (утверждение П) следует, что каждому выводу $(\rho_0 \cup \rho_1, \Lambda) \dashrightarrow (\rho_1 \cup \vartheta, \phi)$ можно поставить в соответствие параллельный вывод $(\rho_0, \Lambda) \dashrightarrow (\vartheta, \phi)$. Причем отношение порядка при этом сохраняется. Поэтому $V(\rho_0 \cup \rho_1, \Lambda) = (\rho_1) * V(\rho_0, \Lambda)$.

Далее, из $W(\rho_0, \Lambda) \subset W(\phi, \Lambda)$ и следствия 8.5.2 вытекает, что $\mathcal{E}_k^0(\rho_0, \Lambda_0, \beta_1, \ldots, \beta_{k-1}) \subset \mathcal{E}_k^0(\phi, \Lambda, \beta_1, \ldots, \beta_{k-1})$. В силу устойчивости генераторов порядка имеем:

$$h_k(\beta_k', \mathcal{E}_k^0(\phi, \Lambda_0, \beta_1, \ldots, \beta_{k-1})) < h_k(\beta_k'', \mathcal{E}_k^0(\phi, \Lambda, \beta_1, \ldots, \beta_{k-1})) \Leftrightarrow$$

$$\Leftrightarrow h_k(\beta_k', \mathcal{E}_k^0(\rho_0, \Lambda, \beta_1, \ldots, \beta_{k-1})) < h_k(\beta_k'', \mathcal{E}_k^0(\rho_0, \Lambda, \beta_1, \ldots, \beta_{k-1}))$$

Отсюда по теореме 8.5 следует, что если $\beta' \in B(\rho_0, \Lambda)$ и $\beta'' \in B(\rho_0, \Lambda)$ и β' предшествует β'' в системе $B(\rho_0, \Lambda)$, то β' предшествует β'' и в $B(\phi, \Lambda)$ и наоборот. Домножение же $V(\phi, \Lambda)$ на (ρ_0) удаляет из $V(\phi, \Lambda)$ те варианты ϑ, для которых $F[\vartheta \cup \rho_0] = \phi$, однако не меняет порядка остающихся вариантов. Отсюда следует утверждение теоремы.

9. АДДИТИВНОСТЬ

9.1. Определение аддитивности.

9.1.1. Упорядочивающий алгоритм отождествления аддитивен в точке (ρ, Λ), если для любых Λ_x и Λ_y, таких, что $\Lambda_x \Lambda_y = \Lambda$ выполняется соотношение

$$V(\rho, \Lambda_x \Lambda_y) = V(\rho, \Lambda_x) * V(\rho, \Lambda_y)$$

9.1.2. Упорядочивающий алгоритм отождествления аддитивен, если он аддитивен в любой точке.

9.2. Принцип однородности.

9.2.1. Проектирующий алгоритм отождествления удовлетворяет принципу однородности, если для любых $\rho, \Lambda_x, \Lambda_y$ и Λ_z из того, что

$$(\rho, \Lambda_x \Lambda_y \Lambda_z) \to (\rho', \Lambda_x \Lambda'_y \Lambda_z)$$

следует, что

$$(\rho, \Lambda_y) \to (\rho', \Lambda'_y)$$

9.2.2. Если алгоритм отождествления однороден и удовлетворяет принципу параллельности, то из того, что $(\rho, \Lambda_x \Lambda_y \Lambda_z)$ есть точка проектирования переменной, вытекает, что и (ρ, Λ_y) есть точка проетирования переменной, причем $\mathcal{E}(\rho, \Lambda_x \Lambda_y \Lambda_z) = \mathcal{E}(\rho, \Lambda_y)$. Таким образом, если имеется ровно ℓ следствий

из $(\rho, \Lambda_x \Lambda_y \Lambda_z)$, то имеется ровно ℓ следствий из (ρ, Λ_y). При этом каждому следствию вида $(\rho', \Lambda_x \Lambda'_y \Lambda_z)$ можно сопоставить следствие вида (ρ', Λ'_y) из (ρ, Λ_y), причем при этом отношение порядка для следствий из (ρ, Λ_y) такое же, как и для соответствующих им следствий из $(\rho, \Lambda_x \Lambda_y \Lambda_z)$

9.2.3. Теперь из рассуждений предыдущего пункта и из п.7.6.I получаем, что если

$$V(\rho, \Lambda_x \Lambda_y \Lambda_z) = \sum_{j=1}^{\ell} V(\rho'^{(j)}, \Lambda_x \Lambda_y'^{(j)} \Lambda_z)$$

то и для $V(\rho, \Lambda_y)$ имеем

$$V(\rho, \Lambda_y) = \sum_{j=1}^{\ell} V(\rho'^{(j)}, \Lambda_y'^{(j)})$$

9.3. Признак аддитивности

Теорема 9.I. Если проектирующий алгоритм отождествления является левосторонним, устойчивым и однородным, он аддитивен.

Доказательство. Проводится индукцией по рангу Λ для точек (ρ, Λ).

Базис индукции. Сначала покажем, что алгоритм аддитивен во всех точках (ρ, Λ), для которых ранг Λ равен нулю. В этом случае $\Lambda = \emptyset$ и для любых $\Lambda_x \Lambda_y = \Lambda$ имеем $\Lambda_x = \Lambda_y = \emptyset$. Поскольку $V(\rho, \emptyset) = (\rho)$, имеем

$$V(\rho, \emptyset) * V(\rho, \emptyset) = (\rho) * (\rho) = (\rho) = V(\rho, \emptyset)$$

Таким образом аддитивность выполнена тривиально.

Индукционный шаг. Теперь рассмотрим множество точек (ρ, Λ) для которых ранг Λ равен $m+1$. Предположим, что аддитивность алгоритма для всех точек ранга m уже доказана. Покажем, что из этого следует аддитивность алгоритма в любой точке ранга $m+1$.

Действительно, пусть (ρ, Λ) имеет ранг m. Если

(ρ, Λ) тупиковая точка, имеем $V(\rho, \Lambda) = \phi$ и аддитивность выполняется тривиально, поскольку из $V(\rho, \Lambda_x) * V(\rho, \Lambda_y) \neq \phi \Rightarrow V(\rho, \Lambda_x \Lambda_y) \neq \phi$. Предположим теперь, что (ρ, Λ) не тупиковая.

Тогда $V(\rho, \Lambda) = \overset{\ell}{\underset{j=1}{C}} V(\rho'^{(j)}, \Lambda'^{(j)})$. Из устойчивости следует $V(\rho'^{(j)}, \Lambda'^{(j)}) = (\delta^{(j)}) * V(\rho, \Lambda'^{(j)})$ где $\delta^{(j)}$ находим из условия $\rho'^{(j)} = \rho \cup \delta^{(j)}$, $\rho \cap \delta^{(j)} = \phi$

Пусть теперь $\Lambda_x \Lambda_y = \Lambda$. Получаем:

$$V(\rho, \Lambda_x \Lambda_y) = \overset{\ell}{\underset{j=1}{C}} [(\delta^{(j)}) * V(\rho, \Lambda_x'^{(j)} \Lambda_y'^{(j)})]$$

Ранг любой из точек $(\rho, \Lambda_x'^{(j)} \Lambda_y'^{(j)})$ есть m. поэтому, по индуктивному предположению

$$V(\rho, \Lambda_x'^{(j)} \Lambda_y'^{(j)}) = V(\rho, \Lambda_x'^{(j)}) * V(\rho, \Lambda_y'^{(j)})$$

Таким образом, окончательно получаем:

$$V(\rho, \Lambda_x \Lambda_y) = \overset{\ell}{\underset{j=1}{C}} [(\delta^{(j)}) * V(\rho, \Lambda_x'^{(j)}) * V(\rho, \Lambda_y'^{(j)})]$$

Теперь рассмотрим три случая.

<u>Случай 1.</u> $\Lambda_x = \phi$ или $\Lambda_y = \phi$. В этом случае аддитивность выполнена тривиально. Дальше будем предполагать, что $\Lambda_x \neq \phi$ и $\Lambda_y \neq \phi$

<u>Случай 2.</u> $\Lambda_y = \Lambda_y'^{(j)}$. При этом

$$V(\rho, \Lambda_x \Lambda_y) = \overset{\ell}{\underset{j=1}{C}} [(\delta^{(j)}) * V(\rho, \Lambda_x'^{(j)}) * V(\rho, \Lambda_y)]$$

Используем дистрибутивность справа $\overset{\ell}{\underset{j=1}{C}} [V_i * V] = [\overset{\ell}{\underset{j=1}{C}} V_i] * V$

Получим, что

$$V(\rho, \Lambda_x \Lambda_y) = \overset{\ell}{\underset{j=1}{C}} [(\delta^{(j)}) * V(\rho, \Lambda_x'^{(j)})] * V(\rho, \Lambda_y)$$

Теперь воспользуемся принципом однородности. Согласно п.9.2.3 имеем $V(\rho, \Lambda_x) = \overset{\ell}{\underset{j=1}{C}} [(\delta^{(j)}) * V(\rho, \Lambda_x'^{(j)})]$

Отсюда $V(\rho, \Lambda_x \Lambda_y) = V(\rho, \Lambda_x) * V(\rho, \Lambda_y)$. В случае 2/ аддитивность выполняется.

Случай 3. $\Lambda_x = \Lambda'^{(j)}_x$. В этом случае из левосторонности алгоритма сразу же следует, что точка (ρ, Λ) - однозначная. Но тогда $\ell = 1$ и мы имеем:

$$V(\rho, \Lambda_x \Lambda_y) = (\delta^{(1)}) * V(\rho, \Lambda_x) * V(\rho, \Lambda'^{(1)}_y) =$$
$$= V(\rho, \Lambda_x) * [(\delta^{(1)}) * V(\rho, \Lambda'^{(1)}_y)]$$

Из принципа однородности

$$V(\rho, \Lambda_y) = (\delta^{(1)}) * V(\rho, \Lambda'^{(1)}_y)$$

Отсюда

$$V(\rho, \Lambda_x \Lambda_y) = V(\rho, \Lambda_x) * V(\rho, \Lambda_y)$$

В случае 3 аддитивность выполнена.

Поскольку каждый шаг вывода преобразует только одну дыру, разобранные три случая исчерпывают все возможности. Индукционный шаг закончен. Теорема доказана.

IO. ОРТОГОНАЛЬНОСТЬ И ЗАВИСИМОСТЬ ДЫР.

IO.I. Ортогональность дыр

IO.I.I. Пусть ρ - некоторый вариант. Через $X(\rho)$ обозначим множество тех переменных, которые входят в ρ. Будем говорить, что типовые выражения ξ'_t и ξ''_t ортогональны для варианта ρ, если для любой переменной x из того, что она входит сразу и в ξ'_t и в ξ''_t следует, что $x \in X(\rho)$

IO.I.2. Будем обозначать ортогональность ξ'_t и ξ''_t для ρ через $\xi'_t \perp_\rho \xi''_t$. Отношение ортогональности симметрично, но не рефлексивно и не транзитивно.

IO.I.3. Пусть даны две системы дыр Δ_x и Δ_y и вариант ρ. Будем говорить, что Δ_x и Δ_y ортогональны для ρ, если для любой переменной x, входящей и в Δ_x и в Δ_y одновременно, имеет место $x \in X(\rho)$. Ортогональность Δ_x и Δ_y будем обозначать через $\Delta_x \perp_\rho \Delta_y$.

10.1.4. Можно доказать теорему:

Теорема 10.1. (Критерий ортогональности).

$$\Delta_x \perp_\rho \Delta_y \Leftrightarrow [\mathcal{E}_t' \in \Delta_x \text{ и } \mathcal{E}_t'' \in \Delta_y \Rightarrow \Rightarrow \mathcal{E}_t' \perp_\rho \mathcal{E}_t'']$$

10.2. Зависимость дыр.

10.2.1. Будем говорить, что два типовых выражения $\mathcal{E}_t'$ и $\mathcal{E}_t''$ непосредственно зависимы для варианта ρ, если они не ортогональны для ρ. Непосредственную зависимость для $\mathcal{E}_t'$ и $\mathcal{E}_t''$ обозначим через $\mathcal{E}_t' \underset{\rho}{\models\!\!\dashv} \mathcal{E}_t''$.

10.2.2. Пусть задана некоторая система дыр Δ и вариант ρ. Пусть $\mathcal{E}_t' \in \Delta$ и $\mathcal{E}_t'' \in \Delta$. Будем говорить, что $\mathcal{E}_t'$ и $\mathcal{E}_t''$ зависимы на Δ для ρ, если существует такая последовательность дыр $\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_n$, что для $i = 1, 2, \dots, n$ имеет место $\mathcal{E}_i \in \Delta$ и для $i = 2, \dots, n$ имеет место $\mathcal{E}_{i-1} \underset{\rho}{\models\!\!\dashv} \mathcal{E}_i$, причем $\mathcal{E}_t' = \mathcal{E}_1$, $\mathcal{E}_t'' = \mathcal{E}_n$.

10.2.3. Введенное выше отношение зависимости для двух дыр $\mathcal{E}_t'$ и $\mathcal{E}_t''$ на Δ для ρ будем обозначать через $\mathcal{E}_t' \underset{\rho,\Delta}{\vdash\!\!\dashv} \mathcal{E}_t''$. Это отношение симметрично и транзитивно, но не рефлексивно.

10.2.4. Можно доказать теорему:

Теорема 10.2. Пусть даны вариант ρ и система дыр $\Delta = (\mathcal{E}_1, \dots, \mathcal{E}_n)$. Тогда множество всех $\mathcal{E}_i \in \Delta$ можно разбить на непересекающиеся классы, такие, что все дыры, входящие в один класс, попарно зависимы, а любые две дыры, принадлежащие разным классам, - ортогональны. Это разбиение единственно.

10.3. Ортогональность $\Lambda = \Delta \times \Omega$

Будем считать, что Λ_x ортогональна к Λ_y для варианта

ρ, если $\Lambda_x = \Delta_x \times \Omega_x$, $\Lambda_y = \Delta_y \times \Omega_y$ и при этом $\Delta_x \perp_\rho \Delta_y$. Соглашение вводим для внесения ясности.

II. РАЗЛОЖЕНИЕ $T(\rho, \Lambda)$ ПО ОРТОГОНАЛЬНЫМ КЛАССАМ ДЫР.

II.1. Замечание. В этом разделе мы будем предполагать, что рассматриваются аддитивные упорядочивающие алгоритмы отождествления. Это значит, что для любых ρ, Λ_x и Λ_y

$$V(\rho, \Lambda_x \Lambda_y) = V(\rho, \Lambda_x) * V(\rho, \Lambda_y)$$

а функция $T(\rho, \Lambda)$ по определению $T(\rho, \Lambda) = \inf V(\rho, \Lambda)$. Все равенства будем понимать в том смысле, что если (при каких-то значениях аргументов) не определена одна часть равенства, то не определена и другая часть равенства. А если обе части равенства определены, то они равны.

II.2. Теорема II.1. Пусть $\Lambda_x \perp_\rho \Lambda_y$. Тогда

$$T(\rho, \Lambda_x \Lambda_y) = T(\rho, \Lambda_x) \cup T(\rho, \Lambda_y)$$

Доказательство. В силу аддитивности $V(\rho, \Lambda_x \Lambda_y) = V(\rho, \Lambda_x) * V(\rho, \Lambda_y)$. Обозначим $V_x = V(\rho, \Lambda_x)$, $V_y = V(\rho, \Lambda_y)$. Из ортогональности $\Lambda_x \perp_\rho \Lambda_y$ следует, что $V_x * V_y = F[V_x \otimes V_y] = V_x \otimes V_y$, поскольку если $\vartheta \in V_x$ и $w \in V_y$, то $F[\vartheta \cup w] = \vartheta \cup w \in V(\rho, \Lambda_x \Lambda_y)$.

Отсюда получаем, что если $T(\rho, \Lambda_x \Lambda_y)$ не существует, то обязательно хотя бы один из $T(\rho, \Lambda_x)$ и $T(\rho, \Lambda_y)$ не существует. Предположим теперь, что $V(\rho, \Lambda_x \Lambda_y) \neq \emptyset$. Тогда $V_x = \overset{n}{\underset{i=1}{C}}(\vartheta_i)$; $V_y = \overset{m}{\underset{j=1}{C}}(w_j)$; $T(\rho, \Lambda_x \Lambda_y) = \inf [V_x \otimes V_y] = \vartheta_1 \cup w_1$. С другой стороны $T(\rho, \Lambda_x) = \inf V_x = \vartheta_1$ $T(\rho, \Lambda_y) = \inf V_y = w_1$. Отсюда следует

$$T(\rho, \Lambda_x) \cup T(\rho, \Lambda_y) = \vartheta_1 \cup w_1 = T(\rho, \Lambda_x \Lambda_y)$$

Теорема доказана.

II.3. Теорема II.2. Пусть $\Lambda_x \perp_\rho \Lambda_y$. Тогда

$$T(\rho, \Lambda_z \Lambda_x \Lambda_y) = T(\rho, \Lambda_z \Lambda_y \Lambda_x)$$

Доказательство. Обозначим $V_z = V(\rho, \Lambda_z)$; $V_y = V(\rho, \Lambda_y)$; $V_x = V(\rho, \Lambda_x)$. Тогда в силу аддитивности будем иметь $T(\rho, \Lambda_z \Lambda_x \Lambda_y) = \inf F[V_z \otimes V_x \otimes V_y]$

Аналогично $T(\rho, \Lambda_z \Lambda_y \Lambda_x) = \inf F[V_z \otimes V_y \otimes V_x]$

II.3.1. Пусть $V_z = \overset{\ell}{\underset{k=1}{\mathfrak{C}}} (u_k)$; $V_x = \overset{n}{\underset{i=1}{\mathfrak{C}}} (v_i)$; $V_y = \overset{m}{\underset{j=1}{\mathfrak{C}}} (w_j)$. Обозначим через $\Theta^{(1)}$ и $\Theta^{(2)}$ системы вариантов:

$$\Theta^{(1)} = V_z \otimes V_x \otimes V_y = \overset{\ell}{\underset{k=1}{\mathfrak{C}}} \overset{n}{\underset{i=1}{\mathfrak{C}}} \overset{m}{\underset{j=1}{\mathfrak{C}}} (u_k \cup v_i \cup w_j)$$

$$\Theta^{(2)} = V_z \otimes V_y \otimes V_x = \overset{\ell}{\underset{k=1}{\mathfrak{C}}} \overset{m}{\underset{j=1}{\mathfrak{C}}} \overset{n}{\underset{i=1}{\mathfrak{C}}} (u_k \cup w_j \cup v_i)$$

Ясно, что $\vartheta \in \Theta^{(1)} \Leftrightarrow \vartheta \in \Theta^{(2)}$. Отсюда следует, что $\vartheta \in F[\Theta^{(1)}] \Leftrightarrow \vartheta \in F[\Theta^{(2)}]$

II.3.2. Представим $\Theta^{(1)}$ в виде $\Theta^{(1)} = \overset{\ell}{\underset{k=1}{\mathfrak{C}}} \Theta_k^{(1)}$, где $\Theta_k^{(1)} = \overset{n}{\underset{i=1}{\mathfrak{C}}} \overset{m}{\underset{j=1}{\mathfrak{C}}} (u_k \cup v_i \cup w_j)$. Представим $\Theta^{(2)}$ в виде $\Theta^{(2)} = \overset{\ell}{\underset{k=1}{\mathfrak{C}}} \Theta_k^{(2)}$, где $\Theta_k^{(2)} = \overset{m}{\underset{j=1}{\mathfrak{C}}} \overset{n}{\underset{i=1}{\mathfrak{C}}} (u_k \cup w_j \cup v_i)$

Очевидно, что $\vartheta \in \Theta_k^{(1)} \Leftrightarrow \vartheta \in \Theta_k^{(2)}$, откуда

$$\vartheta \in F[\Theta_k^{(1)}] \Leftrightarrow \vartheta \in F[\Theta_k^{(2)}]$$

II.3.3. Теперь заметим, что $F[\Theta^{(1)}] = \emptyset \Leftrightarrow F[\Theta^{(2)}] = \emptyset$

Отсюда следует, что обе части равенства $T(\rho, \Lambda_z \Lambda_x \Lambda_y) = T(\rho, \Lambda_z \Lambda_y \Lambda_x)$ существуют или не существуют одновременно. Поэтому для доказательства теоремы осталось рассмотреть случай, когда обе части равенства существуют.

II.3.4. Пусть теперь

$$\inf F[\Theta^{(1)}] = u_z \cup v_p \cup w_q$$

$$\inf F[\Theta^{(2)}] = u_{z'} \cup v_{p'} \cup w_{q'}$$

Очевидно, что $\inf F[\Theta^{(1)}] \in F[\Theta_z^{(1)}]$ и $\inf F[\Theta^{(2)}] \in F[\Theta_{z'}^{(2)}]$, откуда следует $F[\Theta_z^{(1)}] \neq \emptyset$ и $F[\Theta_{z'}^{(2)}] \neq \emptyset$.

Покажем теперь, что $z' = z$. Доказательство проводит-

ся от противного. Допустим, что $\mathcal{Z}' < \mathcal{Z}$. Поскольку $\inf F[\theta^{(1)}] \in F[\theta_{\mathcal{Z}}^{(1)}]$ имеем: $F[\theta_k^{(1)}] = \emptyset$ при $k = 1, 2, \ldots, \mathcal{Z}-1$. Но если $\mathcal{Z}' < \mathcal{Z}$, то должно быть $F[\theta_{\mathcal{Z}'}^{(2)}] \neq \emptyset \Rightarrow F[\theta_{\mathcal{Z}'}^{(1)}] \neq \emptyset$. Получили противоречие. Значит, $\mathcal{Z}' < \mathcal{Z}$ невозможно. Точно так же доказывается невозможность $\mathcal{Z}' > \mathcal{Z}$. Таким образом $\mathcal{Z}' = \mathcal{Z}$

II.3.5. Из предыдущего пункта имеем:

$$\begin{cases} \inf F[\theta^{(1)}] = \inf [(u_r) * V_x * V_y] \\ \inf F[\theta^{(2)}] = \inf [(u_r) * V_y * V_x] \end{cases}$$

Теперь воспользуемся ортогональностью $\Lambda_x \perp_\rho \Lambda_y$. Из неё следует, что если $\vartheta \in V_x$, а $w \in V_y$, то $F[(\vartheta \cup w)] = (\vartheta \cup w)$. Поэтому

$$\left.\begin{array}{l} u_r \cup \vartheta_i \in (u_r) * V_x \\ u_r \cup w_j \in (u_r) * V_y \end{array}\right\} \Rightarrow$$

$$\Rightarrow u_r \cup \vartheta_i \cup w_j \in (u_r) * V_x * V_y$$. Этим свойством мы будем пользоваться в следующих пунктах.

II.3.6. Пусть числа p и q определяются равенствами:

$$\begin{cases} \inf [(u_r) * V_x] = u_r \cup \vartheta_p \\ \inf [(u_r) * V_y] = u_r \cup w_q \end{cases}$$

Покажем, что в этом случае

$$\inf [(u_r) * V_x * V_y] = u_r \cup \vartheta_p \cup w_q .$$

Отсюда, из симметрии в определении чисел p и q будет следовать, что $\inf [(u_r) * V_y * V_x] = u_r \cup w_q \cup \vartheta_p$ Из этого будет немедленно следовать $\inf F[\theta^{(1)}] = \inf F[\theta^{(2)}]$ и теорема будет доказана.

II.3.7. Проведем доказательство равенства $\inf [(u_r) * V_x * V_y] = u_r \cup \vartheta_p \cup w_q$ от противного. Пусть

$$\inf[(u_r)*V_x*V_y] = u_r \cup v_{p'} \cup w_{q'}$$

Обозначим $\varphi = (u_r) \otimes V_x \otimes V_y$. Представим φ в виде $\varphi = \sum_{i=1}^{n} \varphi_i$, где $\varphi_i = \sum_{j=1}^{m} (u_r \cup v_i \cup w_j)$

Предположим теперь, что $p' < p$. Тогда

$$\inf \varphi \in F[\varphi_{p'}] \Rightarrow F[\varphi_{p'}] \neq \emptyset$$

С другой стороны $\inf[(u_r)*V_x] = u_r \cup v_p \Rightarrow$ для всех $i = 1, 2, \ldots, p-1$ выполнено $F[(u_r \cup v_i)] = \emptyset$ Отсюда имеем $F[(u_r \cup v_i \cup w_j)] = \emptyset$ для всех $i = 1, 2, \ldots, p-1$ Поэтому $F[\varphi_i] = \emptyset$ для всех $i = 1, 2, \ldots, p-1$. Однако, если $p' < p$, то $F[\varphi_{p'}] \neq \emptyset$. Противоречие доказывает, что $p' < p$ невозможно.

Предположим, что $p' > p$. Тогда $\inf F[\varphi] \in F[\varphi_{p'}] \Rightarrow$ $\Rightarrow F[\varphi_i] = \emptyset$ для всех $i = 1, 2, \ldots, p'-1$. Однако, согласно п.II.3.5 имеем: $u_r \cup v_p \cup w_q \in F[\varphi_p]$, откуда $F[\varphi_p] \neq \emptyset$. Следовательно $p' > p$ невозможно. Доказано, что $p' = p$.

II.3.8. Теперь докажем, что $q' = q$. Действительно, доказано, что $\inf[\theta^{(1)}] = \inf[(u_r \cup v_p)*V_y]$. Обозначим $\psi = (u_r \cup v_p) \otimes V_y$. Представим ψ в виде $\psi = \sum_{j=1}^{m} \psi_j$ где $\psi_j = (u_r \cup v_p \cup w_j)$

Предположим теперь, что $q' < q$. Тогда $\inf F[\psi] \in$ $\in F[\psi_{q'}] \Rightarrow F[\psi_{q'}] \neq \emptyset$ С другой стороны, $\inf[(u_r)*V_y] = u_r \cup w_q$ влечет $F[(u_r \cup w_j)] = \emptyset$ для любого $j = 1, 2, \ldots, q-1$. Отсюда $F[\psi_j] = \emptyset$ для любого $j = 1, 2, \ldots, q-1$. Если $q' < q$, то и $F[\psi_{q'}] = \emptyset$, что противоречит $F[\psi_{q'}] \neq \emptyset$. Следовательно $q' < q$ невозможно.

Предположим теперь, что $q' > q$. Тогда $\inf F[\psi] \in F[\psi_{q'}] \Rightarrow F[\psi_j] = \emptyset$ для всех $j = 1, 2, \ldots, q'-1$. С другой стороны из $\inf [(u_r) * v_y] = u_r \cup w_q$ следует $F[(u_r \cup w_q)] = (u_r \cup w_q)$. Следовательно, выполнено $F[(u_r \cup w_q \cup v_p)] = (u_r \cup w_q \cup v_p)$. Таким образом $F[\psi_q] \neq \emptyset$. Однако из $q' > q$ и $F[\psi_j] = \emptyset$ для всех $j = 1, 2, \ldots, q'-1$ следует $F[\psi_q] = \emptyset$, что противоречит $F[\psi_q] \neq \emptyset$.

Таким образом, $q' > q$ невозможно. Доказано, что $q' = q$.

Отсюда, в силу п.II.3.6. следует утверждение теоремы. Теорема доказана.

II.4. Теорема II.3. Пусть $\Lambda_x = \overset{n}{\underset{i=1}{C}} \Lambda'_i$; $\Lambda_y = \overset{n}{\underset{i=1}{C}} \Lambda''_i$; $\Lambda_{xy} = \overset{n}{\underset{i=1}{C}} [\Lambda'_i \Lambda''_i]$ и пусть $\Lambda_x \perp_\rho \Lambda_y$. Тогда

$$T(\rho, \Lambda_{xy}) = T(\rho, \Lambda_x) \cup T(\rho, \Lambda_y)$$

Доказательство. Доказательство проводится индукцией по n

Базис индукции. Если $n = 1$, по теореме II.I. получаем

$$T(\rho, \Lambda_{xy}) = T(\rho, \Lambda'_1 \Lambda''_1) = T(\rho, \Lambda'_1) \cup T(\rho, \Lambda''_1) = T(\rho, \Lambda_x) \cup T(\rho, \Lambda_y)$$

Индукционный шаг. Пусть $n = m+1$. Предположим, что для $n = m$ теорема доказана. Тогда

$$T(\rho, \Lambda_{xy}) = T\left(\rho, \overset{m-1}{\underset{i=1}{C}} [\Lambda'_i \Lambda''_i] \Lambda'_m \Lambda''_m \Lambda'_{m+1} \Lambda''_{m+1}\right)$$

Отсюда, учитывая, что $\Lambda'_{m+1} \perp_\rho \Lambda''_{m+1}$ и $\Lambda''_m \Lambda''_{m+1} \perp_\rho \Lambda'_{m+1}$ и дважды применяя теорему II.2, получаем: $T(\rho, \Lambda_{xy}) =$

$$= T\left(\rho, \overset{m-1}{\underset{i=1}{C}} [\Lambda'_i \Lambda''_i] [\Lambda'_m \Lambda'_{m+1}] [\Lambda''_m \Lambda''_{m+1}]\right)$$

Теперь переобозначим

$$\Lambda'_m \; \Lambda'_{m+1} \to \Lambda'_m$$
$$\Lambda''_m \; \Lambda''_{m+1} \to \Lambda''_m$$

Тогда получим

$$T(\rho, \Lambda_{xy}) = T(\rho, \overset{m}{\underset{i=1}{C}} [\Lambda'_i \Lambda''_i])$$

Отсюда по индуктивному предположению получаем:

$$T(\rho, \Lambda_{xy}) = T(\rho, \overset{m}{\underset{i=1}{C}} \Lambda'_i) \cup T(\rho, \overset{m}{\underset{i=1}{C}} \Lambda''_i)$$

Отсюда, возвращаясь к прежним обозначениям, получим

$$T(\rho, \Lambda_{xy}) = T(\rho, \overset{m+1}{\underset{i=1}{C}} \Lambda'_i) \cup T(\rho, \overset{m+1}{\underset{i=1}{C}} \Lambda''_i) =$$
$$= T(\rho, \Lambda_x) \cup T(\rho, \Lambda_y)$$

Индукционный шаг закончен. Теорема доказана.

II.5. Следствие. Допустим, что нам нужно вычислить $T(\rho, \Lambda)$ Тогда по теореме I0.2 все дыры, входящие в Λ можно разбить на n непересекающихся классов, так, что любые две дыры, принадлежащие к различным классам, будут ортогональны. После этого n раз применяя теорему II.3, мы представим $T(\rho, \Lambda)$ в виде

$$T(\rho, \Lambda) = \bigcup_{i=1}^{n} T(\rho, \Lambda_i)$$

где $\Lambda_i \perp_\rho \Lambda_j$ если $i \neq j$.

I2. РАЗЛОЖЕНИЕ $T(\rho, \Lambda)$ ДЛЯ СИСТЕМЫ ОТКРЫТЫХ ДЫР.

I2.I. Твердые вхождения переменных и твердые термы.

I2.I.I. Рассмотрим некоторую точку (ρ, Λ) . Пусть y - это некоторое вхождение переменной x в систему Λ Вхождение y называется твердым вхождением, если $x \in X(\rho)$ или если x - твердая переменная.

I2.I.2. Рассмотрим некоторый типовой терм $\mathcal{T}$, который входит в $\mathcal{A}$. Будем говорить, что $\mathcal{T}$ - твердый терм, если он имеет вид $(\mathcal{E}_t)$, где $\mathcal{E}_t$ - типовое выражение или если $\mathcal{T}$ является твердым вхождением некоторой переменной.

I2.2. Открытые и закрытые дыры.

I2.2.I. Рассмотрим точку $(\mathcal{S}, \mathcal{A})$. Пусть $\mathcal{A} = \Delta \times \mathcal{R}$ и пусть $\mathcal{E}_t$ - некоторая дыра $\mathcal{E}_t \in \Delta$. Дыра $\mathcal{E}_t$ называется открытой, если она представима в виде $\mathcal{X}' \mathcal{E}_t' \mathcal{X}''$, где $\mathcal{E}_t'$ - типовое выражение, а $\mathcal{X}'$ и $\mathcal{X}''$ - вхождения некоторых переменных, не являющиеся твердыми вхождениями.

I2.2.2. Дыра, которая не является открытой, будет называться закрытой дырой. Ясно, что любая закрытая дыра либо пустая, либо состоит из одного нуль-терма, либо хотя бы один из ее крайних нуль-термов твердый.

I2.3. $\mathcal{L}$ - ПЕРЕМЕННЫЕ И $\mathcal{Z}$-ПЕРЕМЕННЫЕ

I2.3.I. Будем называть переменную $\mathcal{X}_k$ $\mathcal{L}$- переменной, если она не является твердой, и если ей сопоставлен $\mathcal{L}$- генератор порядка. Будем использовать для такой переменной мета-обозначение $\mathcal{L}_k$ наряду с обычным обозначением $\mathcal{X}_k$.

I2.3.2. Будем называть переменную $\mathcal{X}_k$ $\mathcal{Z}$- переменной, если она не является твердой, и если ей сопоставлен R-генератор порядка. Будем использовать для такой переменной мета-обозначение $\mathcal{Z}_k$ наряду с обычным обозначением $\mathcal{X}_k$.

I2.3.3. Ясно, что $\mathcal{L}$-переменная не может одновременно являться $\mathcal{Z}$-переменной и наоборот. Однако $\mathcal{L}$-переменная может быть одновременно и $\mathcal{E}$-переменной, равно как и $\mathcal{Z}$-переменная может в то же время являться $\mathcal{E}$-переменной.

I2.3.4. Если переменная x является одновременно и l-переменной и e-переменной, будем говорить, что она является le-переменной. Если же переменная x является одновременно и r-переменной и e-переменной, будем говорить, что она является re-переменной.

I2.4. Левая выводимость

I2.4.I. Введем понятие левой выводимости. Будем говорить, что точка (ρ', Λ') лево выводится из точки (ρ, Λ), если:

I. $(\rho, \Lambda) \dot{\to} (\rho', \Lambda')$

II. Любой вывод из точки (ρ, Λ), имеющий ту же длину, что и вывод $(\rho, \Lambda) \dot{\to} (\rho', \Lambda')$ и отличный от этого вывода, является более правым по сравнению с выводом $(\rho, \Lambda) \dot{\to} (\rho', \Lambda')$

I2.4.2. Если точка (ρ', Λ') лево выводится из (ρ, Λ) будем писать $(\rho, \Lambda) \xrightarrow{\cdot\alpha} (\rho', \Lambda')$. Очевидно, что левая выводимость есть частный случай сильной выводимости.

I2.4.3. Ясно, что если $(\rho, \Lambda) \xrightarrow{\cdot\alpha} (\vartheta, \phi)$, то $\vartheta = T(\rho, \Lambda)$. Т.е. тождественно выполнено

$$(\rho, \Lambda) \xrightarrow{\cdot\alpha} (T(\rho, \Lambda), \phi)$$

I2.4.3. Пусть теперь концевой вывод $(\rho, \Lambda) \dot{\to} (T(\rho, \Lambda), \phi)$ представим в виде $(\rho, \Lambda) \dot{\to} (\rho', \Lambda') \dot{\to} (T(\rho, \Lambda), \phi)$ Ясно, что при этом всегда заведомо $(\rho', \Lambda') \xrightarrow{\cdot\alpha} (T(\rho, \Lambda), \phi)$, однако совсем не обязательно выполнено

$$(\rho, \Lambda) \xrightarrow{\cdot\alpha} (\rho', \Lambda')$$

I2.4.4. Пусть теперь $(\rho, \Lambda) \xrightarrow{\cdot\alpha} (\rho', \Lambda')$ и $(\rho', \Lambda') \xrightarrow{\cdot\alpha} (\rho'', \Lambda'')$. Элементарные рассуждения показывают, что в этом случае будет иметь место

$$(\rho, \Lambda) \xrightarrow{\cdot\alpha} (\rho'', \Lambda'')$$

I2.5. Принцип приоритета

I2.5.I. Будем говорить, что проектирующий алгоритм отождествления удовлетворяет принципу приоритета, если в каждой точке (ρ, Λ) выполняются следующие условия:

I. Если (ρ, Λ) содержит хоть одну закрытую дыру, то точка (ρ, Λ) - однозначная, причем если все следствия из этой точки п получаются по правилу PXL или по правилу PXR, то точка (ρ, Λ) является точкой проектирования твердого вхождения переменной.

II. Если все дыры, входящие в (ρ, Λ) открытые, то все следствия из (ρ, Λ) получаются по правилу PXL, причем это правило применяется к самой левой дыре.

III. Пусть точка (ρ, Λ) имеет вид: $(\rho, ((\xi_1 x \xi_2, \omega))\Lambda)$ где x - вхождение переменной, не являющееся твердым вхождением. Тогда из выводимости

$$(\rho, ((\xi_1 x \xi_2, \omega))\Lambda) \rightarrow (\rho', \Lambda_1((\xi_1' x \xi_2, \omega'))\Lambda)$$

следует выводимость

$$(\rho, ((\xi_1 x, \omega))\Lambda) \rightarrow (\rho', \Lambda_1((\xi_1' x, \omega'))\Lambda)$$

I2.5.2. Если проектирующий алгоритм удовлетворяет принципу параллельности и принципу приоритета, то он заведомо левосторонний.

I2.6. Теоремы о разложении $T(\rho, \Lambda)$

В этом разделе будем предполагать, что рассматриваются проектирующие алгоритмы отождествления, которые удовлетворяют принципу параллельности, принципу однородности и принципу приоритета.

I2.6.I. Теорема I2.I. Рассмотрим точку (ρ, Λ), такую, что все дыры, входящие в Λ - открытые, и Λ имеет вид:

$\Lambda = ((l_x \xi_0 e_y \xi_1, \omega)) \Lambda_6$, где ξ_0 - типовое выражение, каждый нуль-терм которого либо твердый, либо является вхождением переменной l_x . Пусть e_y входит в Λ только один раз, и пусть $(l_x \xi_0) \perp_\rho (\xi_1) \Delta_6$. Тогда $T(\rho, \Lambda) = \rho \cup \delta \cup \vartheta \cup \{(e_y, \omega_y'')\}$, где δ, ϑ и ω_y'' находятся из соотношений

$$T(\rho, ((l_x \xi_0 e_y, \omega))) = \rho \cup \delta \cup \{(e_y, \omega'')\}$$
$$T(\rho, ((e_y \xi_1, \omega'')) \Lambda_6) = \rho \cup \vartheta \cup \{(e_y, \omega_y'')\}$$

Причем $T(\rho, \Lambda)$ определен только тогда, когда определены оба последних соотношения.

Доказательство. Очевидно, что $\rho \cup \delta \cup \vartheta \cup \{(e_y, \omega_y'')\} \in W(\rho, \Lambda)$ поэтому $T(\rho, \Lambda)$ заведомо существует, если два последних соотношения определены. Осталось показать, что если $T(\rho, \Lambda)$ существует, то и два последних соотношения определены и при этом

$$T(\rho, \Lambda) = \rho \cup \delta \cup \vartheta \cup \{(e_y, \omega_y'')\}$$

В самом деле, из принципа приоритета следует, что вывод $(\rho, \Lambda) \xrightarrow{\cdot\mathcal{L}} (T(\rho, \Lambda), \emptyset)$ представим в виде:

$$(\rho, ((l_x \xi_0 e_y \xi_1, \omega)) \Lambda_6) \xrightarrow{\cdot} (\rho \cup \delta, ((e_y \xi_1, \omega'')) \Lambda_6) \xrightarrow{\cdot\mathcal{L}} \xrightarrow{\cdot\mathcal{L}} (\rho \cup \delta \cup \vartheta \cup \{(e_y, \omega_y'')\}, \emptyset)$$

Здесь предполагается $\rho \cap \delta = \emptyset$. Далее, пусть δ_0 такой вариант, что $\rho \cap \delta_0 = \emptyset$ и

$$(\rho, ((l_x \xi_0 e_y \xi_1, \omega)) \Lambda_6) \xrightarrow{\cdot\mathcal{L}} \xrightarrow{\cdot\mathcal{L}} (\rho \cup \delta_0, ((e_y \xi_1, \omega_0'')) \Lambda_6)$$

Покажем, что $\delta_0 = \delta$. Доказательство проводится от противного. Предположим, что $\delta_0 \neq \delta$. Тогда из точки

$(\varsigma \cup \delta_0, ((e_y \xi_1, \omega'')) \Lambda_\ell)$ нет ни одного концевого вывода, так как иначе из нее в силу п.I2.4.4 выводился бы $\tau(\varsigma, \Lambda)$.

Далее, при тех ограничениях, которые наложены на ξ_0, имеем $\Lambda_0[Q_0(\varsigma \cup \delta_0, l_x \xi_0)] \leqslant \Lambda_0[Q_0(\varsigma \cup \delta, l_x \xi_0)]$ поэтому $\Lambda_0[\omega''] \leqslant \Lambda_0[\omega_0'']$. Следовательно ω_0'' можно представить в виде $\omega_0'' = \omega' \circ \omega''$. Докажем теперь, что имеет место

$$(\varsigma \cup \delta_0, ((e_y \xi_1, \omega_0'')) \Lambda_\ell) \dashrightarrow$$
$$\dashrightarrow (\varsigma \cup \delta_0 \cup \vartheta \cup \{(e_y, \omega' \circ \omega_y'')\}, \phi)$$

Это будет противоречить тому, что из $(\varsigma \cup \delta_0, ((e_y \xi_1, \omega_0'')) \Lambda_\ell)$ нет ни одного концевого вывода, а отсюда будет следовать, что $\delta = \delta_0$.

Действительно, вывод $(\varsigma \cup \delta, ((e_y \xi_1, \omega'')) \Lambda_\ell) \overset{\chi}{\dashrightarrow}$ $\overset{\chi}{\dashrightarrow} (\varsigma \cup \delta \cup \vartheta \cup \{(e_y, \omega_y'')\}, \phi)$ заведомо представляется в виде

$$(\varsigma \cup \delta, ((e_y \xi_1, \omega'')) \Lambda_\ell) \dashrightarrow$$
$$\dashrightarrow (\varsigma \cup \delta \cup \{(e_y, \omega_y'')\}, ((\xi_1, \omega_1'')) \Lambda_\ell) \overset{\chi}{\dashrightarrow}$$
$$\overset{\chi}{\dashrightarrow} (\varsigma \cup \delta \cup \vartheta \cup \{(e_y, \omega_y'')\}, \phi)$$

Однако, в силу того, что e_y e- переменная и входит в Λ только один раз, из выводимости

$$(\varsigma \cup \delta, ((e_y \xi_1, \omega'')) \Lambda_\ell) \dashrightarrow$$
$$\dashrightarrow (\varsigma \cup \delta \cup \{(e_y, \omega_y'')\}, ((\xi_1, \omega_1'')) \Lambda_\ell)$$

вытекает выводимость

$$(\varsigma \cup \delta, ((e_y \xi_1, \omega' \circ \omega'')) \Lambda_\ell) \dashrightarrow$$
$$\dashrightarrow (\varsigma \cup \delta \cup \{(e_y, \omega' \circ \omega_y'')\}, ((\xi_1, \omega_1'')) \Lambda_\ell).$$

Теперь заметим, что из ортогональности $(e_x \xi_0) \perp_0 (\xi_1) \Delta_\ell$ вытекает, что в δ и δ_0 входят только те переменные, которые

не входят в ξ_1 и Δ_β. Поэтому в силу принципа параллельности можно заменить δ на δ_0 и получить:

$$(\rho \cup \delta_0, ((e_y \xi_1, \omega' \omega'')) \Lambda_\beta) \to$$
$$\to (\rho \cup \delta_0 \cup \{(e_y, \omega' \omega_y'')\}, ((\xi_1, \omega_1'')) \Lambda_\beta).$$

Точно так же из выводимости

$$(\rho \cup \delta \cup \{(e_y, \omega_y'')\}, ((\xi_1, \omega_1'')) \Lambda_\beta) \dot{\to}$$
$$\dot{\to} (\rho \cup \delta \cup \vartheta \cup \{(e_y, \omega_y'')\}, \phi)$$

следует выводимость

$$(\rho \cup \delta_0 \cup \{(e_y, \omega' \omega_y'')\}, ((\xi_1, \omega_1'')) \Lambda_\beta) \dot{\to}$$
$$\dot{\to} (\rho \cup \delta_0 \cup \vartheta \cup \{(e_y, \omega' \omega_y'')\}, \phi)$$

Но из этих двух выводимостей непосредственно следует:

$$(\rho \cup \delta_0, ((e_y \xi_1, \omega_0'')) \Lambda_\beta) \dot{\to}$$
$$\dot{\to} (\rho \cup \delta_0 \cup \vartheta \cup \{(e_y, \omega' \omega_y'')\}, \phi)$$

Следовательно, существует хотя бы один концевой вывод из $(\rho \cup \delta_0, ((e_y \xi_1, \omega_0'')) \Lambda_\beta)$. Из противоречия следует $\delta = \delta_0$.

Теперь докажем соотношение $T(\rho, ((l_x \xi_0 e_y, \omega))) = \rho \cup \delta \cup \{(e_y, \omega'')\}$. Действительно, из $\delta = \delta_0$ следует, что

$$(\rho, ((l_x \xi_0 e_y \xi_1, \omega)) \Lambda_\beta) \dot{\to}$$
$$\dot{\to} (\rho \cup \delta, ((e_y \xi_1, \omega'')) \Lambda_\beta)$$

Однако в силу принципа однородности отсюда следует:

$$(\rho, ((l_x \xi_0 e_y \xi_1, \omega))) \dot{\to}$$
$$\dot{\to} (\rho \cup \delta, ((e_y \xi_1, \omega'')))$$

Кроме того, в силу принципа приоритета:

$$(\rho, ((\ell_x \mathcal{E}_0 e_y, \omega))) \xrightarrow{\cdot\mathcal{L}}$$
$$\xrightarrow{\cdot\mathcal{L}} (\rho \cup \delta, ((e_y, \omega'')))$$

Отсюда следует

$$T(\rho, ((\ell_x \mathcal{E}_0 e_y, \omega))) = \rho \cup \delta \cup \{(e_y, \omega'')\}$$

Теперь докажем, что

$$T(\rho, ((e_y \mathcal{E}_1, \omega'')) \Lambda_\ell) = \rho \cup \vartheta \cup \{(e_y, \omega_y'')\}$$

В самом деле, из выводимости

$$(\rho \cup \delta, ((e_y \mathcal{E}_1, \omega'')) \Lambda_\ell) \xrightarrow{\cdot\mathcal{L}}$$
$$\xrightarrow{\cdot\mathcal{L}} (\rho \cup \delta \cup \vartheta \cup \{(e_y, \omega_y'')\}, \phi)$$

в силу принципа параллельности следует выводимость

$$(\rho, ((e_y \mathcal{E}_1, \omega'')) \Lambda_\ell) \xrightarrow{\cdot\mathcal{L}} (\rho \cup \vartheta \cup \{(e_y, \omega_y'')\}, \phi)$$

Отсюда следует доказываемое соотношение. Теорема доказана.

I2.6.2. Теорема I2.2. Рассмотрим точку (ρ, Λ) такую, что все дыры, входящие в Λ – открытые, и Λ имеет вид: $\Lambda = ((\mathcal{E}_1 \tau_x \mathcal{E}_0, \omega)) \Lambda_\ell$, где τ_x – переменная, являющаяся τe – переменной. Пусть τ_x входит в Λ только один раз, пусть $(\mathcal{E}_0) \perp_\rho (\mathcal{E}_1) \Delta_\ell$ и при этом каждый нуль-терм $\mathcal{E}_1$, не являющийся твердым, представляет собой вхождение переменной, которой сопоставлен устойчивый генератор порядка. Тогда:

$$T(\rho, \Lambda) = \rho \cup \delta \cup \vartheta \cup \{(\tau_x, \omega_x')\}$$

где δ, ϑ и ω_x' находятся из соотношений:

$$T(\rho, ((\tau_x \mathcal{E}_0, \omega))) = \rho \cup \delta \cup \{(\tau_x, \omega')\}$$
$$T(\rho, ((\mathcal{E}_1 \tau_x, \omega')) \Lambda_\ell) = \rho \cup \vartheta \cup \{(\tau_x, \omega_x')\}$$

Причем $T(\rho, \Lambda)$ определен только тогда когда определены оба последних соотношения.

Показательство. Очевидно, что $\rho \cup \delta \cup \vartheta \cup \{(z_x, \omega'_x)\} \in W(\rho, \mathcal{A})$, поэтому $T(\rho, \mathcal{A})$ заведомо существует, если два последних соотношения определены. Осталось показать, что если $T(\rho, \mathcal{A})$ существует, то и два последних соотношения определены и при этом

$$T(\rho, \mathcal{A}) = \rho \cup \delta \cup \vartheta \cup \{(z_x, \omega'_x)\}$$

Действительно, вывод $(\rho, \mathcal{A}) \overset{\mathcal{X}}{\rightarrow} (T(\rho, \mathcal{A}), \emptyset)$ представим в виде:

$$(\rho, ((\mathfrak{E}_1 z_x \mathfrak{E}_0, \omega)) \mathcal{A}_6) \rightarrow$$
$$\rightarrow (\rho \cup \rho_1, ((z_x \mathfrak{E}_0, \omega'_x \omega'')) \mathcal{A}'_6) \rightarrow$$
$$\rightarrow (\rho \cup \rho_1 \cup \mathfrak{V}, \mathcal{A}'_6) \overset{\mathcal{X}}{\rightarrow} (\rho \cup \rho_1 \cup \mathfrak{V} \cup \rho_2, \emptyset)$$

Здесь предполагается, что ρ, ρ_1, ρ_2 и $\mathfrak{V}$ попарно не пересекаются.

Пусть теперь $\mathfrak{V}_0$ такой вариант, что $\mathfrak{V}_0$ не пересекается с ρ, ρ_1 и ρ_2 и в то же время

$$(\rho \cup \rho_1, ((z_x \mathfrak{E}_0, \omega'_x \omega'')) \mathcal{A}'_6) \overset{\mathcal{X}}{\rightarrow}$$
$$\overset{\mathcal{X}}{\rightarrow} (\rho \cup \rho_1 \cup \mathfrak{V}_0, \mathcal{A}'_6)$$

Покажем, что $\mathfrak{V} = \mathfrak{V}_0$. В самом деле, если $\mathfrak{V} \neq \mathfrak{V}_0$, то из точки $(\rho \cup \rho_1 \cup \mathfrak{V}_0, \mathcal{A}'_6)$ не должно быть ни одного концевого вывода. Однако из принципа параллельности и ортогональности $(z_x \mathfrak{E}_0) \perp_\rho \mathcal{A}'_6$ вытекает, что выводимость

$$(\rho \cup \rho_1 \cup \mathfrak{V}, \mathcal{A}'_6) \rightarrow (\rho \cup \rho_1 \cup \mathfrak{V} \cup \rho_2, \emptyset)$$

влечет выводимость

$$(\rho \cup \rho_1 \cup \mathfrak{V}_0, \mathcal{A}'_6) \rightarrow (\rho \cup \rho_1 \cup \mathfrak{V}_0 \cup \rho_2, \emptyset)$$

Значит из $(\rho \cup \rho_1 \cup \mathfrak{V}_0, \mathcal{A}'_6)$ есть хотя бы один концевой вывод. Из противоречия следует $\mathfrak{V} = \mathfrak{V}_0$. Таким образом:

$$(\rho \cup \rho_1, ((z_x \mathfrak{E}_0, \omega'_x \omega'')) \mathcal{A}'_6) \overset{\mathcal{X}}{\rightarrow}$$
$$\overset{\mathcal{X}}{\rightarrow} (\rho \cup \rho_1 \cup \mathfrak{V}, \mathcal{A}'_6)$$

Обозначим теперь $\mathfrak{V} = \delta \cup \{(z_x, \omega'_x)\}$ где $\delta \cap \{(z_x, \omega'_x)\} = \emptyset$. Покажем, что δ можно найти из

соотношения

$$T(\rho, ((\tau_x \xi_0, \omega))) = \rho \cup \delta \cup \{(\tau_x, \omega')\}$$

Действительно, используя принцип однородности из выводимости

$$(\rho \cup \rho_1, ((\tau_x \xi_0, \omega_x' \omega'')) \Lambda_\theta') \xrightarrow{\cdot \chi}$$
$$\xrightarrow{\cdot \chi} (\rho \cup \rho_1 \cup \{(\tau_x, \omega_x')\} \cup \delta, \Lambda_\theta')$$

получим выводимость

$$(\rho \cup \rho_1, ((\tau_x \xi_0, \omega_x' \omega''))) \xrightarrow{\cdot \chi}$$
$$\xrightarrow{\cdot \chi} (\rho \cup \rho_1 \cup \{(\tau_x, \omega_x')\} \cup \delta, \emptyset)$$

Теперь воспользуемся принципом параллельности и получим выводимость

$$(\rho, ((\tau_x \xi_0, \omega_x' \omega''))) \xrightarrow{\cdot \chi}$$
$$\xrightarrow{\cdot \chi} (\rho \cup \{(\tau_x, \omega_x')\} \cup \delta, \emptyset)$$

Теперь используем то, что τ_x - суть τe - переменная.

Отсюда получаем выводимость:

$$(\rho, ((\tau_x \xi_0, \omega_1' \omega_x' \omega''))) \xrightarrow{\cdot \chi}$$
$$\xrightarrow{\cdot \chi} (\rho \cup \{(\tau_x, \omega_1' \omega_x')\} \cup \delta, \emptyset)$$

где $\omega = \omega' \omega''$ и $\omega' = \omega_1' \omega_x'$

Отсюда непосредственно получаем:

$$T(\rho, ((\tau_x \xi_0, \omega))) = \rho \cup \{(\tau_x, \omega')\} \cup \delta$$

Теперь нужно доказать соотношение: $T(\rho, ((\xi_1 \tau_x, \omega')) \Lambda_\theta) =$ $= \rho \cup \vartheta \cup \{(\tau_x, \omega_x')\}$, где $\vartheta = \rho_1 \cup \rho_2$. Для этого заметим, что δ, а следовательно и ω'' от ρ_1 не зависят. Поэтому концевой вывод $(\rho, \Lambda) \xrightarrow{\cdot \chi} (T(\rho, \Lambda), \emptyset)$ принадлежит к множеству выводов вида:

$$(\rho, ((\xi_1 \tau_x \xi_0, \omega' \omega'')) \Lambda_\theta) \xrightarrow{\cdot}$$
$$\xrightarrow{\cdot} (\rho \cup \rho_1, ((\tau_x \xi_0, \omega_x' \omega'')) \Lambda_\theta') \longrightarrow$$

$$\to (\rho \cup \rho_1 \cup \{(z_x, \omega'_x)\}, ((\xi_0, \omega'')) \mathcal{N}'_\ell) \to$$
$$\to (\rho \cup \rho_1 \cup \{(z_x, \omega'_x)\} \cup \delta, \mathcal{N}'_\ell) \xrightarrow{\cdot \mathcal{L}}$$
$$\xrightarrow{\cdot \mathcal{L}} (\rho \cup \rho_1 \cup \{(z_x, \omega'_x)\} \cup \delta \cup \rho_2, \emptyset)$$

где δ и ω'' одинаковы для всех выводов. Однако из принципов параллельности, однородности и приоритета следует, что каждому выводу из этого множества можно взаимно-однозначно сопоставить вывод:

$$(\rho, ((\xi_1 z_x, \omega')) \mathcal{N}_\ell) \xrightarrow{\cdot}$$
$$\xrightarrow{\cdot} (\rho \cup \rho_1, ((z_x, \omega'_x)) \mathcal{N}'_\ell) \to$$
$$\to (\rho \cup \rho_1 \cup \{(z_x, \omega'_x)\}, ((\emptyset, \emptyset)) \mathcal{N}'_\ell) \to$$
$$\to (\rho \cup \rho_1 \cup \{(z_x, \omega'_x)\}, \mathcal{N}'_\ell) \xrightarrow{\cdot \mathcal{L}}$$
$$\xrightarrow{\cdot \mathcal{L}} (\rho \cup \rho_1 \cup \{(z_x, \omega'_x)\} \cup \rho_2, \emptyset)$$

Причем, из устойчивости генераторов порядка следует, что при этом отображении отношение порядка на множестве выводов сохраняется. Отсюда непосредственно следует

$$T(\rho, ((\xi_1 z_x, \omega')) \mathcal{N}_\ell) = \rho \cup \vartheta \cup \{(z_x, \omega'_x)\}$$

где $\vartheta = \rho_1 \cup \rho_2$

Теорема доказана.

I3. ЛИТЕРАТУРА

I. В.Ф.Турчин, Метаалгоритмический язык, "Кибернетика", № 4, I968.

2. В.Ф.Турчин, Алгоритмический язык рекурсивных функций (РЕФАЛ), Препринт ИПМ АН СССР, I968.

3. С.Н.Флоренцев, В.Ю.Слонин, В.Ф.Турчин, Эффективный интерпретатор для языка РЕФАЛ, Препринт ИПМ АН СССР, I969.

4. С.А.Романенко, В.Ф.Турчин, Рефал-компилятор. Труды 2-й всесоюзной конференции по программированию. Новосибирск, I970.

5. В.Ф.Турчин, Программирование на языке РЕФАЛ, Препринт ИПМ АН СССР, I97I.

6. А.В.Климов, С.А.Романенко, В.Ф.Турчин. Компилятор с языка РЕФАЛ (I. Общие принципы построения рефал-компилятора, II. Неформальное описание языка сборки для базисного рефала), ИПМ АН СССР, I972.

С О Д Е Р Ж А Н И Е

№ Т-00679 от " 8 " 02 1973 г. Заказ № 1592 Тираж 200 экз.

Ордена Ленина институт прикладной математики
Москва, Миусская пл., 4