

Теория строк (слов). Недостаточная выразительность рекурсивных функций и арифметики для теории алгоритмов.

Д. Л. Гуринович

25 апреля 2020 года

УДК 510.51

Содержание

	Аннотация	1
1	Эвристические соображения о недостаточности арифметики для целей теории алгоритмов	3
2	Теория строк	6
	0. Общая информация и свод результатов	6
	I. Разбиение строк	13
	II. Алфавит строк	20
	III. Концы строк, соединение строк	26
	IV. Равенство строк	35
	V. Сравнение, поиск, вставка	43
	VI. Алгоритмы, не сводимые к (частично) рекурсивным функциям и арифметике	54
	VII. Критерий локального изменения данных	61
	VIII. Критерий локального извлечения данных	70
3	Числа в теории строк	74
4	Приложение. Логика теорий первого порядка с равенством	78
	Список литературы	84

Аннотация

С момента формулировки гипотезы $NP \neq P$ (1971 г.) прошло без малого полвека, со времени создания первой ЭВМ (1941 г) – почти 80 лет. Но математической теории (первого

порядка) в том смысле, как понимал это Гильберт и другие логики первой половины 20 века – такой теории для изучения алгоритмов не было создано до сих пор. Понять, что нужной теории нет – было очень неожиданным открытием для меня.

В частности, (частично) рекурсивные функции, на которых строится представимость алгоритмов в арифметике, не могут работать с входными данными частично – а легко построить алгоритм, который завершает свою работу, прочитав первый символ во входных данных, переданных ему для работы. А частично рекурсивная функция читает входные данные до конца. С арифметикой та же проблема.

Разница же между временем работы алгоритмов, читающих один символ из «входа» с временем работы «представляющих» их функций, читающих всё «слово» во входных данных – может быть сколь угодно велика. Эта разница может превышать не только полиномиальные, но и любые другие ограничения относительно времени работы исходного алгоритма. И такая «представимость» – не годится для теории алгоритмов.

Это и многое другое выяснилось, когда я разбирался с вопросом о $NP \neq P$ и отвечал в дискуссиях на вопросы о времени работы алгоритма, о работе алгоритма с частью входных данных, о слишком длинных сертификатах в проверках принадлежности слов языкам класса NP и т.д. Был и вопрос, например, «почему вы не используете Гёделевы номера?».

Стала понятна первоочередная необходимость глубокой подготовки фундамента для теории алгоритмов, а не попытка построения доказательства для $NP \neq P$ без теоретического фундамента.

Поэтому я отложил вопрос о доказательстве $NP \neq P$ как менее приоритетный с точки зрения математики и занялся построением теории строк (слов). А это такая теории первого порядка, которая необходима для решения вопросов теории алгоритмов.

В процессе построения теории строк выяснились некоторые интересные парадоксы и появилась возможность доказать, в частности, недостаточную выразительность арифметики для решения вопросов теории алгоритмов. И удалось это сделать как раз на базе построенной теории строк.

Доказательство недостаточной выразительности арифметики для решения вопросов теории алгоритмов дано в данной статье в разделе 2. «Теория строк», подразделе VI. «Алгоритмы, не сводимые к (частично) рекурсивным функциям и арифметике».

И если читатель представляет себе, что такое строки (на уровне программиста), если он знаком с логикой, арифметикой и имеет поверхностное представление о рекурсивных функциях – то он в состоянии сразу прочитать раздел 2, подраздел VI настоящей статьи. И это будет хорошая мотивация для изучения теории строк, потому что арифметика для теории алгоритмов – не достаточно выразительна, как там доказано.

А помимо этого, для решения задач уровня $NP \neq P$ необходимо формализовать, что такое «сложность вычислений». Для кого/чего имеет место быть эта сложность. И, почему этому кому-то/чему-то «сложно» решать некоторые задачи. По сути – это тоже вопрос об алгоритмах, решающих вопросы теории алгоритмов. Но этот вопрос будет отложен до статьи 5 или дальше – если считать эту статью первой.

Вопрос изучения теории средствами самой теории – именно то, что было сделано в первой половине 20 века логиками при реализации программы Гильберта. А программа Гильбер-

та требовала тщательной формализации теории для того, чтобы применить методы теории (арифметики и логики в том случае) к ней самой. Вот для логической формализации теории алгоритмов и написана эта статья.

Данная статья является самой объёмной и содержательной в теоретическом отношении среди всех статей (остальные в черновиках пока) цикла. Остальные статьи (кроме второй) в основном получают свои результаты как логические выводы на базе данной статьи и их размер значительно меньше размера данной статьи. Впрочем, в процессе их подготовки к публикации что-то, может быть, и изменится.

1 Эвристические соображения о недостаточности арифметики для целей теории алгоритмов

Если бы в рамках арифметики мы могли доказать какое-то свойство или провести какое-то действие, характерное именно для позиционного представления чисел, а не для стандартной интерпретации, то это означало бы, что арифметика не годится для стандартной интерпретации. Но именно для стандартной интерпретации она и создана в первую очередь.

А переходя от текстов к гёделевым номерам – мы переходим от объектов более высокого уровня к объектам уровня существенно проще – и эти «упрощенные» объекты не позволяют исследовать свойства «слов», необходимых для теории алгоритмов.

Теперь немного разъясню сказанное в 2х предыдущих абзацах.

Стандартная интерпретация – это представление чисел в виде «счётных палочек», где число 10 выглядит примерно так:

1111111111

Записано, кстати, при помощи 11 (Одиннадцати) единиц, потому что для нуля тоже необходимо какое-то видимое изображение.

В основе всех чисел лежит именно стандартная интерпретация. Арифметика является теорией для чисел в любых формах – в том числе и в стандартной интерпретации. И арифметика описывает те свойства чисел, которые общие для всех их представлений. Ничего от специфического представления (от позиционного представления, например) в арифметике – нет. Иначе она не годилась бы для стандартной интерпретации. Мы – можем записать в позиционном десятичном представлении число 10 таким образом:

10

Но наши условности – как это представление соответствует вышеприведённому представлению 1111111111 – не являются частью арифметики, и не могут являться частью этой общей теории о натуральных числах. А причина изложена в первом абзаце данного раздела.

Казалось бы, мы можем представить число как сумму степеней 10. Например, число 2361 можно записано так:

$$2 \times 10^3 + 6 \times 10^1 + 1 + 3 \times 10^2$$

Но и это тоже аналог записи позиционным образом. Потому что в стандартной интерпретации

мы имеем дело с «кучками счётных палочек».

Да, мы можем записывать числа в виде $2 \times 10^2 + 3 \times 10^3 + 6 \times 10^1 + 7 \times 10^4$ даже при помощи определяемых внутри теории Пеано функций – есть там (можно определить) и функция возведения в степень, а как записано 10 в таком случае – пусть в виде счётных палочек – это не меняет в принципе степень сжатости данного представления. Вот как число

$$2 \times 10^2 + 3 \times 10^3 + 6 \times 10^1 + 7 \times 10^4$$

тогда будет выглядеть (цифра n обозначена строкой из $n + 1$ символов «1»):

$$111 \times 1111111111^{111} + 1111 \times 11111111111^{1111} + 1111111 \times 111111111111^{11} + 11111111 \times 1111111111111^{11111}$$

И можно доказать, что такое представление (десятичное, например) существует и единственное для каждого числа, что у него есть аналоги умножения, сложения, деления, вычитания «в столбик», но проблема в том, что сами объекты арифметики – это числа, не отличимые от чисел в стандартной интерпретации и в формулах они представлены и передаются именно так.

Если у нас определена в арифметике функция $f(i)$, то значение переменной i не может передаваться в виде представления через сумму степеней 10 с коэффициентами. У нас нет на уровне объекта (числа) внутри арифметики способа его передачи в готовом «сжатом» виде.

Грубо говоря, в рамках арифметики у нас нет «инвентаризационных ведомостей», несущих сжатую информацию о количестве товара «на складе». Мы можем передавать только склад с его товарами, а пересчёт получатель должен делать каждый раз заново. В случае с товарами это, кстати, оправдано, но не в случае операций с расчётами, когда число нам нужно получить сразу в готовом («сжатом») виде – без его проверок и пересчётов.

А если мы запишем десятичное представление числа i в виде другого числа j , которое является гёделевым номером десятичного представления, но записано тоже «кучей счётных палочек», то это число j не будет, в общем случае, меньше числа i и ничего не «сожмёт», вообще говоря.

Сказанное выше создаёт впечатление (правильное, как мы убедимся в этой статье), что гёделевы номера не годятся в качестве «слов» для теории алгоритмов. Гёделевы номера имеют свойства «куч счётных палочек» и не более, потому что формализованы в рамках арифметики.

Да, «кучи счётных палочек» можно поставить во взаимно-однозначное соответствие текстам, но вот что касается операций по поиску, перестановке и соединению текстов, то в рамках арифметики нет никакой возможности представить их иначе, чем логическими формулами с использованием функций следования (добавления единицы), сложения и умножения. Притом эти операции понимаются тут как предельно примитивные – пригодные даже для стандартной интерпретации.

Грубо говоря, чтобы перейти от текстов «мама» и «мыла раму» к тексту «мама мыла раму» – придётся сопоставлять эту операцию множеству операций умножений, сложений, следований, которые будут соединены посредством некоторой формулы логики первого порядка. А размеры «куч счётных палочек», соответствующих этим текстам, будут вне рамок каких-либо полиномиальных ограничений относительно размера строк вроде «мама мыла раму», получающихся в результате конкатенации.

Поэтому теория строк нам необходима уже по той причине, что иначе нет инструмента для

формализации понятия «слово» и свойств этих «слов», делающих возможными обычные операции с этими объектами.

Поэтому приступим к построению теории строк (слов), а уже на базе построенной теории строк будет доказана (вместо приведённых сейчас эвристических соображений) недостаточная выразительность арифметики (и гёделевых номеров) для решения задач теории алгоритмов.

2 Теория строк

Общая информация и свод результатов

Математики называют объекты теории алгоритмов термином «слово», программисты используют для того же термин «строка». Далее буду использовать термин программистов, потому что аксиомы построены именно в соответствии с общеизвестной практикой программирования.

Для предметных переменных типа строка буду использовать все переменные, кроме i, j, k, l, m, n – которые оставим для натуральных чисел. И кроме переменной q , которую оставим для числовых значений и пустой строки \ominus . Впрочем, натуральные числа тоже являются строками особого вида, и форма представления чисел внутри теории строк будет рассмотрена в разделе 3. Но обратное (что всякая строка является числом) не верно.

Увеличение числа i на 1 буду обозначать так:

i' – такое обозначение увеличения числа на 1 является аксиоматизированной операцией в арифметике (и в её кратком варианте в виде теории Пеано), поэтому сохраним тут это стандартное обозначение.

Кроме того, вместо специального обозначения для арифметического вычитания от числа i числа j (результат которого равен $i - j$, когда $i > j$, и нулю в противном случае) буду использовать обозначение:

$$\max(i, j) - j$$

Результат приведенной формулы совпадает с результатом арифметического вычитания и более привычен людям, особенно если они программисты, например.

Логические выводы очевидных формул из теории Пеано строить не буду – их можно посмотреть в учебниках логики, и они очевидны для большинства читателей сами по себе. Но в той части, которая касается вывода из собственных аксиом теории строк, буду давать доказательства с подробностью, равной или близкой логическому выводу. Потому что для новой теории надо аккуратно убедиться в достаточности её аксиом для получения необходимых теорем.

Данный раздел будет разбит на подразделы, в каждом из которых будут изложены аксиомы теории строк, независимые от предыдущих подразделов. Начиная с V подраздела новых аксиом не будет, а будут только определения, теоремы и метатеоремы.

Сразу дам сводку всех результатов данного раздела в данном его подразделе. Это удобно для того, чтобы проверять доказательства – сразу видеть все предыдущие утверждения теории, на которые опираются доказательства. Разумеется, не все смотрят доказательства – тогда им вполне хватит этого подраздела вместо того, чтобы читать все остальные подразделы данного раздела.

Однако подраздел VI «Алгоритмы, не сводимые к (частично) рекурсивным функциям и арифметике» прочитать было бы полезно – там обоснована невозможность обойтись арифметикой для решения вопросов теории алгоритмов. И это является причиной для построения теории строк.

Но аксиомы Пеано – которые тоже являются частью теории строк, как и вся арифметика – записаны в следующем разделе. Единственная оговорка – аксиома индукции. Она тоже будет дана в следующем разделе, но будет использоваться и в этом. Однако схема доказательства по индукции

отлично известна и без написания аксиомы, но нужно учитывать, что мы распространим аксиому индукции и на те функции, в которых помимо числовых аргументов имеются ещё и строковые, но при этом не числовые.

Итак, ключевые результаты построения теории строк – кроме аксиом арифметики и аксиомы индукции, расширенной на формулы теории строк, следующие:

Аксиомы, определения и метатеоремы (теоремы о теории)

I. Разбиение строк

Аксиома о делении строки на начальную и конечную части в произвольной позиции i :

$$(1) a = \text{beg}(a, i) \cdot \text{end}(a, i')$$

Аксиомы пустых строк:

$$(2.1) \text{beg}(a, 0) = \ominus$$

$$(2.2) \text{end}(a, 0) = \ominus$$

(2.3) $\text{end}(\text{beg}(a, i), i') = \ominus$ - если мы взяли в качестве новой строки начало старой строки до i -го символа включительно, то после этого символа в новой строке нет символов.

Альтернативное обозначение: $a.\text{beg}(i).\text{end}(i') = \ominus$

(2.4) $\text{beg}(\ominus, q) = \ominus$, начало пустой строки – пустая строка, независимо от того, является второй аргумент числом или нет.

(2.5) $\text{end}(\ominus, q) = \ominus$, то же самое можно сказать про конец пустой строки.

(2.6) $i \neq \ominus$, никакое число (включая ноль) не является пустой строкой.

Аксиомы уменьшения вложенности:

$$(3.1) \text{beg}(a, i) = \text{beg}(\text{beg}(a, i), i)$$

Альтернативное обозначение: $a.\text{beg}(i) = a.\text{beg}(i).\text{beg}(i)$

$$(3.2) \text{beg}(a, i) = \text{beg}(\text{beg}(a, i), i')$$

Альтернативное обозначение: $a.\text{beg}(i) = a.\text{beg}(i).\text{beg}(i')$

$$(3.3) \text{beg}(a, i) = \text{beg}(\text{beg}(a, i'), i)$$

Альтернативное обозначение: $a.\text{beg}(i) = a.\text{beg}(i').\text{beg}(i)$

$$(3.4) a = \text{end}(a, 0')$$

$$(3.5) i \neq 0 \Rightarrow \text{end}(a, i') = \text{end}(\text{end}(a, i), 0'')$$

Альтернативное обозначение: $i \neq 0 \Rightarrow a.\text{end}(i') = a.\text{end}(i).\text{end}(0'')$

Определение функции $\text{str}()$ и аксиома перестановки в композиции $\text{beg}()$ с $\text{end}()$:

$$(4.1) \text{str}(a, i, j) = \text{beg}(\text{end}(a, i), j)$$

Альтернативное обозначение: $a.\text{str}(i, j) = a.\text{end}(i).\text{end}(j)$

$$(4.2) \text{end}(\text{beg}(a, i), i) = \text{beg}(\text{end}(a, i), 1)$$

Альтернативное обозначение: $a.\text{beg}(i).\text{end}(i) = a.\text{end}(i).\text{beg}(1)$

II. Алфавит строк

Аксиомы об «атомах», из которых образованы строки:

$$(5.1) \forall a \forall i \exists j \text{str}(a, i, 1) = \text{Chr}(j)$$

(5.2) Аксиома об «алфавите»:

$$(\forall i > l_{\text{ChrLim}} : \text{Chr}(i) = \ominus)$$

$$\wedge(\forall i \leq l_{ChrLim} : Chr(i) \neq \ominus \wedge Chr(i) = str(Chr(i), 1, 1))$$

$$\wedge(\forall i \leq l_{ChrLim} \forall j \leq l_{ChrLim} : Chr(i) = Chr(j) \Rightarrow i = j)$$

(м.5.1) Аксиома алфавита (5.2) независима от предыдущих аксиом, так как массивы произвольных натуральных чисел удовлетворяют предыдущим аксиомам, но не удовлетворяют аксиоме (5.2).

(5.3) Определение функции $Ach(u)$, обратной к функции $Chr(i)$:

$$(Ach(u) = l'_{ChrLim} \wedge u = \ominus)$$

$$\vee(Ach(u) = i \wedge u \neq \ominus \wedge Chr(i) = u)$$

$$\vee(Ach(u) = \ominus \wedge \forall i Chr(i) \neq u)$$

(5.4) Определение функции $Comp(Chr(i), Chr(j))$, сравнивающей произвольные элементы алфавита $Chr(i)$ и $Chr(j)$:

$$(Comp(Chr(i), Chr(j)) = 0 \wedge i > l_{ChrLim} \wedge j > l_{ChrLim})$$

$$\vee(Comp(Chr(i), Chr(j)) = 0 \wedge i \leq l_{ChrLim} \wedge j \leq l_{ChrLim} \wedge i = j)$$

$$\vee(Comp(Chr(i), Chr(j)) = 1 \wedge i \leq l_{ChrLim} \wedge j \leq l_{ChrLim} \wedge i < j)$$

$$\vee(Comp(Chr(i), Chr(j)) = 1 \wedge i > l_{ChrLim} \wedge j \leq l_{ChrLim})$$

$$\vee(Comp(Chr(i), Chr(j)) = 2 \wedge i \leq l_{ChrLim} \wedge j \leq l_{ChrLim} \wedge i > j)$$

$$\vee(Comp(Chr(i), Chr(j)) = 2 \wedge i \leq l_{ChrLim} \wedge j > l_{ChrLim})$$

III. Концы строк, соединение строк

Аксиома о конце строки, для случая, когда у строки есть конец:

$$(6.1) (i \neq 0 \wedge str(a, i, 1) = \ominus) \Rightarrow str(a, i', 1) = \ominus$$

(м.6.1) Аксиома (6.1) независима от предыдущих аксиом, так как случай, когда пустая строка может чередоваться с другими элементами алфавита внутри «строки», соответствует всем аксиомам перед аксиомой (6.1), а аксиоме (6.1) – не соответствует.

(6.2) Определение для $len(u)$:

$$(len(u) = 0 \wedge str(u, 1, 1) = \ominus)$$

$$\vee(len(u) = i \wedge str(u, i, 1) \neq \ominus \wedge str(u, i', 1) = \ominus)$$

$$\vee(len(u) = \ominus \wedge \forall i str(u, i, 1) \neq \ominus)$$

(7.1) Аксиома о результате конкатенации при конечной 1-й строке:

$$len(a) \neq \ominus$$

$$\Rightarrow (i \leq len(a) \Rightarrow str(a \cdot b, i, 1) = str(a, i, 1)) \wedge (i > len(a) \Rightarrow str(a \cdot b, i, 1) = str(b, i - len(a), 1))$$

(7.2) Аксиома о результате конкатенации при бесконечной 1-й строке:

$$len(a) = \ominus \Rightarrow a = a \cdot b$$

(м.7.1) Аксиомы (7.1) и (7.2) независимы от предыдущих аксиом, так как случай упорядоченных массивов символов соответствует всем аксиомам перед аксиомами (7.1) и (7.2), но не соответствует аксиомам (7.1) и (7.2).

IV. Равенство строк

Аксиома о равенстве строк:

$$(8.1) a = b \Leftrightarrow \forall i str(a, i, 1) = str(b, i, 1)$$

(м.8.1) Аксиома (8.1) независима от предыдущих аксиом, так как есть пример из ООП (объектно-ориентированного программирования), когда функции $\text{beg}()$ и $\text{end}()$ создают «детей» у «независимых» строк. И между «независимыми» строками и «детьми» есть разница помимо последовательности символов. Такие объекты ООП соответствуют всем аксиомам перед аксиомой (8.1), но не соответствует аксиоме (8.1).

$$(8.2) (x_1 = y_1 \wedge x_2 = y_2) \Rightarrow x_1 \cdot x_2 = y_1 \cdot y_2$$

$$(8.3) (x = y \wedge i = j) \Rightarrow \text{beg}(x, i) = \text{beg}(y, j)$$

$$(8.3) (x = y \wedge i = j) \Rightarrow \text{end}(x, i) = \text{end}(y, j)$$

$$(8.4) (i = j) \Rightarrow \text{Chr}(i) = \text{Chr}(j)$$

$$(8.5) x = x$$

$$(8.6) x = y \Rightarrow (z = x \Rightarrow z = y)$$

V. Сравнение, поиск, вставка

(9.1) Определение для $\text{CompIn}(a, b)$:

$$(\text{str}(a, \text{CompIn}(a, b), 1) \neq \text{str}(b, \text{CompIn}(a, b), 1) \wedge \forall i (i < \text{CompIn}(a, b) \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))) \vee (\text{CompIn}(a, b) = 0 \wedge a = b)$$

(9.2) Определение для $\text{Comp}(a, b)$:

$$\text{Comp}(a, b) = \text{Comp}(\text{str}(a, \text{CompIn}(a, b), 1), \text{str}(b, \text{CompIn}(a, b), 1))$$

(9.3) Определение для $\text{find}(in, \text{what}, 0)$:

$$(\text{what} \neq \ominus \wedge \text{len}(\text{what}) \neq \ominus \wedge \text{str}(in, \text{find}(in, \text{what}, 0), \text{len}(\text{what})) = \text{what})$$

$$\wedge \forall j (j < \text{find}(in, \text{what}, 0) \Rightarrow \text{str}(in, j, \text{len}(\text{what})) \neq \text{what})$$

$$\vee (\text{find}(in, \text{what}, 0) = 0 \wedge (\text{what} = \ominus \vee \text{len}(\text{what}) = \ominus \vee \forall j \text{ str}(in, j, \text{len}(\text{what})) \neq \text{what}))$$

(9.4) Определение для $\text{find}(in, \text{what}, i)$:

$$\text{find}(in, \text{what}, i) = \text{find}(\text{end}(in, i), \text{what}, 0)$$

(9.5) Определение для $if_0(a, b, c)$:

$$(if_0(a, b, c) = b \wedge a = 0) \vee (if_0(a, b, c) = c \wedge a \neq 0)$$

(9.6) Определение для $\text{Ins}(a, i, b)$:

$$\text{Ins}(a, i, b) = \text{beg}(a, \max(i, 1) - 1) \cdot b \cdot \text{end}(a, if_0(\min(\text{len}(\text{len}(b)), i), 0, \text{len}(b) + i))$$

(9.7) Определение для функции $\text{IsIns}(q_a, i, q_b)$:

$$(\text{IsIns}(q_a, i, q_b) = 1 \wedge i \neq 0 \wedge q_a \neq \ominus \wedge q_b \neq \ominus \wedge (q_b = 0 \vee q_a \geq i + q_b - 1))$$

$$\vee (\text{IsIns}(q_a, i, q_b) = 0 \wedge (i = 0 \vee q_a = \ominus \vee q_b = \ominus \vee (q_b \neq 0 \wedge q_a < i + q_b - 1)))$$

(9.8) Определение для функции $\text{IsStr}(q_a, i, j)$:

$$(\text{IsStr}(q_a, i, j) = 1 \wedge q_a \neq \ominus \wedge (i = 0 \vee j = 0 \vee q_a \geq i + j - 1))$$

$$\vee (\text{IsStr}(q_a, i, j) = 0 \wedge (q_a = \ominus \vee (i \neq 0 \wedge j \neq 0 \wedge q_a < i + j - 1)))$$

(9.9) Определение для $\text{ss}(i)$ и $\text{ss}_k(i)$:

$\text{len}(\text{ss}(i)) = i \wedge \forall j \leq i : \text{str}(\text{ss}(i), j, 1) = \text{Chr}(0)$. Функция $\text{ss}_k(i)$ определяется так же, но только вместо $\text{Chr}(0)$ используется $\text{Chr}(k)$, где $k \leq l_{\text{ChrLim}}$.

VI. Алгоритмы, не сводимые к (частично) рекурсивным функциям и арифметике

(м.9.1) В теории строк с односимвольным алфавитом ($l_{\text{ChrLim}} = 0$) невозможно в общем случае представить работу алгоритмов со строками многосимвольного алфавита таким образом, чтобы

размер логического вывода для результата работы такого алгоритма был в полиномиальных (или любых иных) пределах относительно времени работы этого алгоритма.

(м.9.2) В арифметике невозможно в общем случае представить работу алгоритмов со строками многосимвольного алфавита таким образом, чтобы размер логического вывода для результата работы такого алгоритма был в полиномиальных (или любых иных) пределах относительно времени работы этого алгоритма.

Теоремы, леммы и следствия

I. Разбиение строк

$$(т.3.1) \text{beg}(a, \min(i, j)) = \text{beg}(\text{beg}(a, i), j)$$

Альтернативное обозначение: $a.\text{beg}(\min(i, j)) = a.\text{beg}(i).\text{beg}(j)$

$$(т.3.2) (i \neq 0 \wedge j \neq 0) \Rightarrow \text{end}(a, i + j - 1) = \text{end}(\text{end}(a, i), j)$$

Альтернативное обозначение: $(i \neq 0 \wedge j \neq 0) \Rightarrow a.\text{end}(i + j - 1) = a.\text{end}(i).\text{end}(j)$

$$(т.3.3) i < k \Rightarrow \text{end}(\text{beg}(a, i), k) = \ominus$$

Альтернативное обозначение: $i < k \Rightarrow a.\text{beg}(i).\text{end}(k) = \ominus$

Определение функции $\text{str}()$ и аксиома перестановки в композиции $\text{beg}()$ с $\text{end}()$:

$$(т.4.1) (i \neq 0 \wedge j \neq 0) \Rightarrow \text{beg}(\text{end}(a, i), j) = \text{end}(\text{beg}(a, i + j - 1), i)$$

Альтернативное обозначение: $(i \neq 0 \wedge j \neq 0) \Rightarrow a.\text{end}(i).\text{beg}(j) = a.\text{beg}(i + j - 1).\text{end}(i)$

$$(т.4.2) (i \neq 0 \wedge j \neq 0) \Rightarrow \text{str}(a, i, j) = \text{end}(\text{beg}(a, i + j - 1), i)$$

Альтернативное обозначение: $(i \neq 0 \wedge j \neq 0) \Rightarrow a.\text{str}(i, j) = a.\text{beg}(i + j - 1).\text{end}(i)$

$$(т.4.3) \text{str}(a, i, j') = \text{str}(a, i, j) \cdot \text{str}(a, i + j, 1)$$

$$(т.4.4) j < k \Rightarrow \text{str}(\text{str}(a, i, j), k, 1) = \ominus$$

Альтернативное обозначение: $j < k \Rightarrow a.\text{str}(i, j).\text{str}(k, 1) = \ominus$

$$(т.4.5) 1 < k \Rightarrow \text{str}(\text{str}(a, i, 1), k, 1) = \ominus$$

Альтернативное обозначение: $1 < k \Rightarrow a.\text{str}(i, 1).\text{str}(k, 1) = \ominus$

$$(т.4.6) \text{str}(a, 1, j) = \text{beg}(a, j)$$

$$(т.4.7) j \leq k \Rightarrow \text{str}(\text{str}(a, 1, j), 1, k) = \text{str}(a, 1, j)$$

Альтернативное обозначение: $j \leq k \Rightarrow a.\text{str}(1, j).\text{str}(1, k) = a.\text{str}(1, j)$

$$(т.4.8) \text{end}(\text{str}(a, 1, j), k) = \text{str}(\text{str}(a, 1, j), k, j), \text{ притом вместо одного данного } \text{end}() \text{ может быть}$$

композиция из подобных $\text{end}()$ сколь угодно большой глубины вложенности.

Альтернативное обозначение: $a.\text{str}(1, j).\text{end}(1, k) = a.\text{str}(1, j).\text{str}(k, j)$

$$(т.4.9) \text{str}(a, 1, j) = \text{str}(\text{str}(a, 1, j), 1, k) \cdot \text{str}(\text{str}(a, 1, j), k', j)$$

Альтернативное обозначение: $a.\text{str}(1, j) = a.\text{str}(1, j).\text{str}(1, k) \cdot a.\text{str}(1, j).\text{str}(k', j)$

II. Алфавит строк

$$(т.5.1) \text{Ach}(\text{Chr}(i)) = \min(i, l'_{\text{ChrLim}})$$

III. Концы строк, соединение строк

$$(т.6.1) 0 < i \leq j \Rightarrow (\text{str}(a, i, 1) = \ominus \Rightarrow \text{str}(a, j, 1) = \ominus)$$

$$(т.6.2) \text{len}(a) \neq \ominus \Rightarrow \text{str}(a, \text{len}(a)', 1) = \ominus \wedge (\text{len}(a) = 0 \vee \text{str}(a, \text{len}(a), 1) \neq \ominus)$$

$$(т.6.3) \text{len}(a) \neq \ominus \Rightarrow (\text{len}(a) < i \Rightarrow \text{str}(a, i, 1) = \ominus)$$

$$(т.6.4) j \leq i \Rightarrow \text{str}(\text{beg}(a, i), j, 1) = \text{str}(a, j, 1)$$

Альтернативное обозначение: $0 < j \leq i \Rightarrow a.\text{beg}(i).\text{str}(j, 1) = a.\text{str}(j, 1)$

(т.6.5) $i \neq 0 \wedge j \neq 0 \Rightarrow \text{str}(\text{end}(a, i), j, 1) = \text{str}(a, i + j - 1, 1)$

Альтернативное обозначение: $i \neq 0 \wedge j \neq 0 \Rightarrow a.\text{end}(i).\text{str}(j, 1) = a.\text{str}(i + j - 1, 1)$

(т.6.6) $\text{len}(a) \neq \ominus \wedge 0 < i \leq \text{len}(a) \Rightarrow \text{str}(a, i, 1) \neq \ominus$

(т.6.7) $\text{len}(a) \neq \ominus \wedge i \leq \text{len}(a) \Rightarrow \text{len}(\text{beg}(a, i)) = i \wedge (\text{len}(\text{end}(a, i)) = \text{len}(a) - i + 1 \vee i = 0)$

(с.6.7) $\text{len}(a) \neq \ominus \wedge i \neq 0 \wedge i + j - 1 \leq \text{len}(a) \Rightarrow \text{len}(\text{str}(a, i, j)) = j$

(т.6.8) $\text{len}(a) = \ominus \Rightarrow \text{len}(\text{beg}(a, i)) = i \wedge (\text{len}(\text{end}(a, i)) = \ominus \vee i = 0)$

(т.6.9) $\text{len}(a) \neq \ominus \wedge \text{len}(a) < i \Rightarrow \text{len}(\text{beg}(a, i)) = \text{len}(a) \wedge \text{len}(\text{end}(a, i)) = 0$

(с.6.9) $\text{len}(a) \neq \ominus \Rightarrow \text{len}(\text{beg}(a, i)) = \min(i, \text{len}(a)) \wedge (\text{len}(\text{end}(a, i)) = \max(\text{len}(a), i - 1) - i + 1 \vee i =$

0)

(т.7.1) $\text{len}(a_1) \neq \ominus \wedge \dots \wedge \text{len}(a_n) \neq \ominus \Rightarrow \text{len}(a_1 \cdot \dots \cdot a_n) = \text{len}(a_1) + \dots + \text{len}(a_n)$

(т.7.2) $i \leq l_{\text{ChrLim}} \Rightarrow \text{len}(\text{Chr}(i)) = 1$

IV. Равенство строк

Ассоциативность конкатенации:

(т.8.1) $(a \cdot b) \cdot c = a \cdot (b \cdot c)$

(т.8.2) $\text{len}(a) \neq \ominus \Rightarrow a = \text{str}(a, 1, \text{len}(a))$, в силу чего в отношении конечной строки a , в виде $\text{str}(a, 1, \text{len}(a))$, можно применять теоремы (т.4.4) и от (т.4.6) до (т.4.9) включительно

(с.8.2) $\text{len}(a) \neq \ominus \Rightarrow (\text{len}(a) \leq i \Rightarrow a = \text{str}(a, 1, i))$

(т.8.3) $a = a \cdot \ominus = \ominus \cdot a$

(т.8.4) $a = b \Leftrightarrow \forall i(0 < i \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))$

(т.8.5) $\text{len}(a) \neq \ominus \Rightarrow [a = b \Leftrightarrow \forall i((\text{len}(a) = \text{len}(b) \wedge 0 < i \wedge i \leq \text{len}(a)) \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))]$

(т.8.6) $\text{len}(a) \neq \ominus \Rightarrow a = \text{beg}(a \cdot u, \text{len}(a))$

(т.8.7) $\text{len}(a) \neq \ominus \Rightarrow u = \text{end}(a \cdot u, \text{len}(a)')$

(т.8.8) $a = \ominus \Leftrightarrow \text{len}(a) = 0$

(с.8.8) $\text{len}(a) = \ominus \Leftrightarrow \text{len}(\text{len}(a)) = 0$

V. Сравнение, поиск, вставка

(л.9.1) Лемма о минимальной позиции отличия в строках:

$\text{str}(a, n, 1) \neq \text{str}(b, n, 1) \Rightarrow$

$\exists_1 m(\text{str}(a, m, 1) \neq \text{str}(b, m, 1) \wedge (\forall i < m : \text{str}(a, i, 1) = \text{str}(b, i, 1)))$

Лемма об эквивалентном добавлении дизъюнктивных членов:

(л.9.2) $A \Rightarrow ((A \Rightarrow \neg C) \Rightarrow (B \Leftrightarrow (B \vee C)))$

(с.9.1) $\text{len}(a) \neq \ominus \Rightarrow [0 < \text{CompIn}(a, b) \leq \text{len}(a) \Leftrightarrow \forall u \text{CompIn}(a \cdot u, b) = \text{CompIn}(a, b)]$, и

то же верно, если в этом следствии заменить $\text{CompIn}(a, b)$ и $\text{CompIn}(a \cdot u, b)$ на $\text{CompIn}(b, a)$ и $\text{CompIn}(b, a \cdot u)$ соответственно.

Критерий локального сравнения данных:

(с.9.2) $\text{len}(a) \neq \ominus \Rightarrow [0 < \text{CompIn}(a, b) \leq \text{len}(a) \Leftrightarrow \forall u \text{Comp}(a \cdot u, b) = \text{Comp}(a, b)]$, и то же верно, если в этом следствии заменить $\text{CompIn}(a, b)$, $\text{Comp}(a \cdot u, b)$ и $\text{Comp}(a, b)$ на $\text{CompIn}(b, a)$, $\text{Comp}(b, a \cdot u)$ и $\text{Comp}(b, a)$ соответственно.

(л.9.3) Лемма о минимальной позиции совпадения в строке:

$what \neq \ominus \wedge \text{len}(what) \neq \ominus \wedge \text{str}(in, n, \text{len}(what)) = what \Rightarrow$
 $\exists_1 m(\text{str}(in, m, \text{len}(what)) = what \wedge (\forall i < m : \text{str}(in, i, \text{len}(what)) \neq what))$
 (л.9.4) $a \neq a \cdot \text{Chr}(0) \Leftrightarrow \text{len}(a) \neq \ominus$

VII. Критерий локального изменения данных

(т.9.1) $\text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus \Rightarrow \text{str}(a \cdot b \cdot u, \text{len}(a)', \text{len}(b)) = b$

(т.9.2) $\text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus \Rightarrow \text{end}(a \cdot b \cdot u, \text{len}(a)') = b \cdot u$

(т.9.3) $\text{IsIns}(\text{len}(a), i, \text{len}(b)) = 1 \Rightarrow \text{len}(\text{Ins}(a, i, b)) = \text{len}(a)$

(т.9.4) Критерий локальной вставки:

$(\text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus) \Rightarrow$

$(\text{IsIns}(\text{len}(a), i, \text{len}(b)) = 1 \Leftrightarrow \forall u \text{Ins}(a \cdot u, i, b) = \text{Ins}(a, i, b) \cdot u)$

VIII. Критерий локального извлечения данных

(т.9.5) $\text{len}(a) \neq \ominus \Rightarrow (\text{IsStr}(\text{len}(a), i, j) = 1 \Leftrightarrow \forall u \text{str}(a \cdot u, i, j) = \text{str}(a, i, j))$

Теперь будем записывать аксиомы и определения теории строк, разбирать их семантику и доказывать теоремы.

I. Разбиение строк

Аксиома о делении строки на начальную и конечную части в произвольной позиции i :

$$(1) a = \text{beg}(a, i) \cdot \text{end}(a, i')$$

Пояснение. Разумеется, если длина строки 6 символов, то деление с 8 позиции приведет к конкатенации исходной строки и пустой строки, обозначаемой значком \ominus :

$$\text{«строка»} = \text{beg}(\text{«строка»}, 8) \cdot \text{end}(\text{«строка»}, 9) = \text{«строка»} \cdot \ominus$$

Если же то же самое делать с 3 позиции, то получится обычное:

$$\text{«строка»} = \text{beg}(\text{«строка»}, 3) \cdot \text{end}(\text{«строка»}, 4) = \text{«стр»} \cdot \text{«ока»}$$

Впрочем, всё это будет понятно из дальнейших аксиом.

Кстати, «конкатенация» – это операция соединения строк. В данной аксиоме – соединение «начала» и «конца» строки, разбитой по границе между местом строки i и i' . Конкатенация в данном цикле статей будет обозначаться значком « \cdot » – отняв это обозначение у умножения чисел. Для умножения мы будем использовать значок « \times ».

В данной группе аксиом («I. Разбиение строк») мы рассматриваем только соединение частей одной строки между собой. Эти аксиомы не могут решить вопрос о соединении (конкатенации) разных строк между собой – как будет показано в группе аксиом «IV. Соединение строк».

Нумерация символов в строке начинается с 1, чтобы использовать место номер «0» для «технических нужд» - что будет видно в этой и следующих статьях цикла. И длина строки при такой нумерации совпадает с номером последнего символа в ней.

Аксиомы пустых строк:

(2.1) $\text{beg}(a, 0) = \ominus$ - если пытаемся взять начало до 1-го символа (не включая 1й символ), то получаем пустую строку;

(2.2) $\text{end}(a, 0) = \ominus$ - если пытаемся брать окончание строки с позиции перед первой позицией, то ничего не обнаруживаем, потому что перед 1-м местом нет символа, который можно вернуть со всей цепочкой символов после него;

(2.3) $\text{end}(\text{beg}(a, i), i') = \ominus$ - если мы взяли в качестве новой строки начало старой строки до i -го символа включительно, то после этого символа в новой строке нет символов.

Альтернативное обозначение: $a.\text{beg}(i).\text{end}(i') = \ominus$

«Альтернативное обозначение» – из объектно-ориентированного программирования (ООП), когда у объектов есть свойства, а чтобы узнать значение свойства – надо задать после имени объекта через точку имя свойства со списком аргументов. Вот так:

$$\text{object.property}(Arg_1, \dots, Arg_n)$$

В данной аксиоме объектом является строка, а свойством – подстрока в начале данной строки. А затем объектом становится полученная строка (подстрока тоже является строкой) и его свойство – подстрока в конце этой строки. А аргумент задаёт границу, до которого нас интересует подстрока в начале, а затем граница, от которой берём подстроку в конце. И результатом тоже является строка, что снова позволяет обратиться к свойству этой новой строки. И так – с любой глубиной композиции свойств.

Такое обозначение гораздо удобней для композиций глубокой вложенности, чем традиционное. Потому что обычное обозначение приводит к необходимости читать исполнение вложенных функций справа налево, а аргументы – слева направо, при этом разрыв нарастает и всё это – с трудно уловимым вложенным соответствием скобок.

Понимая, что новые обозначения не все быстро осваивают, будем придерживаться в основном традиционного обозначения. Кроме случаев слишком большой вложенности, по крайней мере.

(2.4) $\text{beg}(\ominus, x) = \ominus$, начало пустой строки – пустая строка, независимо от того, является второй аргумент числом или нет.

(2.5) $\text{end}(\ominus, x) = \ominus$, то же самое можно сказать про конец пустой строки.

(2.6) $i \neq \ominus$, никакое число (включая ноль) не является пустой строкой. Тут выясняется, что в теории строк есть объект (хотя бы один), отличающийся от любого числа.

Аксиомы уменьшения вложенности:

(3.1) $\text{beg}(a, i) = \text{beg}(\text{beg}(a, i), i)$

Альтернативное обозначение: $a.\text{beg}(i) = a.\text{beg}(i).\text{beg}(i)$

(3.2) $\text{beg}(a, i) = \text{beg}(\text{beg}(a, i), i')$

Альтернативное обозначение: $a.\text{beg}(i) = a.\text{beg}(i).\text{beg}(i')$

(3.3) $\text{beg}(a, i) = \text{beg}(\text{beg}(a, i'), i)$

Альтернативное обозначение: $a.\text{beg}(i) = a.\text{beg}(i').\text{beg}(i)$

Пояснение для аксиом (3.1)-(3.3). Если отбрасывать все символы в строке 2 раза – после i -го и $i + 1$ -го символов, то независимо от последовательности этих 2 операций останется только i начальных символов после отбрасывания.

По индукции легко доказать, что

(т.3.1) $\text{beg}(a, \min(i, j)) = \text{beg}(\text{beg}(a, i), j)$

Альтернативное обозначение: $a.\text{beg}(\min(i, j)) = a.\text{beg}(i).\text{beg}(j)$

Нюанс: «легко доказать» - если использовать обычные правила работы с равенством. Вопрос о равенстве в данном разделе будет обсуждаться особо в подразделе IV.

Основные моменты доказательства (т.3.1) на примере. Из предположения истинности для $i < j$ предложения $\text{beg}(a, \min(i, j)) = \text{beg}(\text{beg}(a, i), j)$ получим:

$\text{beg}(\text{beg}(a, i), j') = \text{beg}(\text{beg}(\text{beg}(a, i), j), j') = \text{beg}(\text{beg}(a, i), j) = \text{beg}(a, \min(i, j)) = \text{beg}(a, \min(i, j'))$

(3.4) $a = \text{end}(a, 0')$

Пояснение: если брать всю цепочку символов, начиная с 1-го, то получим всю исходную строку.

(3.5) $i \neq 0 \Rightarrow \text{end}(a, i') = \text{end}(\text{end}(a, i), 0'')$

Альтернативное обозначение: $i \neq 0 \Rightarrow a.\text{end}(i') = a.\text{end}(i).\text{end}(0'')$

Пояснение. Удаление первых i символов в строке – это то же самое, что удалить $i - 1$ первых символов, а затем удалить ещё 1-й символ в оставшейся строке. Отсюда легко доказывается по индукции, как и для (т.3.1) и с той же оговоркой про равенство:

(т.3.2) $(i \neq 0 \wedge j \neq 0) \Rightarrow \text{end}(a, i + j - 1) = \text{end}(\text{end}(a, i), j)$

Альтернативное обозначение: $(i \neq 0 \wedge j \neq 0) \Rightarrow a.\text{end}(i + j - 1) = a.\text{end}(i).\text{end}(j)$

(т.3.3) $i < k \Rightarrow \text{end}(\text{beg}(a, i), k) = \ominus$

Альтернативное обозначение: $i < k \Rightarrow a.\text{beg}(i).\text{end}(k) = \ominus$

Выводится из (2.3) $\text{end}(\text{beg}(a, i), i') = \ominus$, (2.5) $\text{end}(\ominus, i) = \ominus$ и (3.5)

Доказывается индукцией по k от $k = i'$, что совпадает с (2.3). Допустим

$\text{end}(\text{beg}(a, i), k) = \ominus$ и докажем из этого, что $\text{end}(\text{beg}(a, i), k') = \ominus$.

$\text{end}(\text{beg}(a, i), k') = \text{end}(\text{end}(\text{beg}(a, i), k), 0'')$, по (3.5)

$\text{end}(\text{end}(\text{beg}(a, i), k), 0'') = \text{end}(\ominus, 0'')$, по предположению индукции

$\text{end}(\ominus, 0'') = \ominus$, по (2.5). Это завершает доказательство по индукции.

Определение функции $\text{str}()$ и аксиома перестановки в композиции $\text{beg}()$ с $\text{end}()$:

(4.1) $\text{str}(a, i, j) = \text{beg}(\text{end}(a, i), j)$

Альтернативное обозначение: $\text{str}(a, i, j) = a.\text{end}(i).\text{beg}(j)$

Пояснение. Это обычная для программирования функция взятие подстроки в строке. В теории оказалось удобнее разбить логику её работы на 2 этапа: удаление начальной части строки до $i - 1$ символа включительно (функция $\text{end}()$) и, затем, удаление в оставшейся части строки всех символов от $j + 1$ -го включительно и всех символов после него (функция $\text{beg}()$).

Функция $\text{str}()$ чрезвычайно важна для логики строк, так как раскрывает структуру строк до уровня их «атомов» через $\text{str}(a, i, 1)$. На любом месте в строке находится либо символ, либо пустая строка. Но сначала нам необходима аксиома, на базе которой выводится способ построения подстроки из её минимальных «атомов». Речь пока не про произвольную строку, а только про результат применения функции $\text{str}()$ к произвольной строке.

После аксиом понижения вложенности и следующей аксиомы перестановки можно любую композицию из вложенных функций $\text{str}()$ с конкретными числами свести либо к одной функции $\text{str}()$, либо к пустой строке \ominus .

Если дело сведётся к одной функции $\text{str}()$ вместо их композиции, то на месте 2-го и 3-го аргументов этой единственной функции $\text{str}()$ будут некие арифметические результаты из всех чисел в исходной композиции. И следующая аксиома перестановки – вместе с изложенными выше аксиомами – делает это возможным:

(4.2) $\text{end}(\text{beg}(a, i), i) = \text{beg}(\text{end}(a, i), 1)$

Альтернативное обозначение: $a.\text{beg}(i).\text{end}(i) = a.\text{end}(i).\text{beg}(1)$

Пояснение. У нас есть 3 способа получить одинаковый результат:

1. Взять в качестве результата то (символ или пустую строку), что находится на i -м месте в исходной строке (способ, производный для данной теории от 2-х следующих);

2. Удалить все символы после (не включая) i -го символа и затем удалить в оставшейся строке все символы до (не включая) i -го символа;

3. Удалить в исходной строке все символы до (не включая) i -го символа и затем удалить в оставшейся строке все символы после (не включая) 1-го.

(т.4.1) $(i \neq 0 \wedge j \neq 0) \Rightarrow \text{beg}(\text{end}(a, i), j) = \text{end}(\text{beg}(a, i + j - 1), i)$

Альтернативное обозначение: $(i \neq 0 \wedge j \neq 0) \Rightarrow a.\text{end}(i).\text{beg}(j) = a.\text{beg}(i + j - 1).\text{end}(i)$

Доказывать будем в виде

$\forall x \text{ beg}(\text{end}(x, i'), j') = \text{end}(\text{beg}(x, i' + j), i')$ – квантор общности тут, чтоб подчеркнуть, что доказательство по индукции ведётся для всех строк сразу, а не для некоторой фиксированной строки a . И переменные тут в сравнении с (т.4.1) имеют вид i' вместо i и j' вместо j . Это сделано для устранения из формулы посылки ($i \neq 0 \wedge j \neq 0$). Потому что любое натуральное число m , не равное нулю, может быть представлено как n' для некоторого n . И, обратно, любое число вида n' – не равно нулю.

Доказывать будем, отталкиваясь от (4.2)

$$\text{beg}(\text{end}(a, i), 1) = \text{end}(\text{beg}(a, i), i)$$

Допустим, у нас доказано «предположение индукции»

$$\forall x \text{ beg}(\text{end}(x, i'), j') = \text{end}(\text{beg}(x, i' + j), i'), \text{ что для } j = 0 \text{ верно из-за (4.2)}$$

Нам надо доказать

$$\forall x \text{ beg}(\text{end}(x, i'), j'') = \text{end}(\text{beg}(x, i' + j'), i')$$

То, что у нас квантор общности – подчёркивает, что доказательство идёт сразу для всех строк, потому что в процессе вывода мы будем использовать предположение индукции не только для одной конкретной строки a . Но, фактически, мы будем исходить из «вторичного» предположения индукции:

$$\text{beg}(\text{end}(a, i'), j') = \text{end}(\text{beg}(a, i' + j), i')$$

А доказывать будем искомое следствие индукции без квантора общности:

$$\text{beg}(\text{end}(a, i'), j'') = \text{end}(\text{beg}(a, i' + j'), i')$$

Обоснованием такого способа действия является аксиома логики (см. раздел 4):

$$(a) \forall x A(x, j) \Rightarrow A(a, j)$$

Что позволяет нам перейти от предположения индукции к вторичному предположению индукции.

Точнее, мы воспользуемся этим предположением два раза, и получим результат вида:

$$\forall x A(x, j) \Rightarrow A(a, j) \wedge A(b, j)$$

Затем мы получим искомое следствие без квантора общности, то есть – докажем импликацию вида:

$$A(a, j) \wedge A(b, j) \Rightarrow A(a, j')$$

На основании силлогизма по последним двум импликациям будем иметь:

$$\forall x A(x, j) \Rightarrow A(a, j')$$

по правилу вывода логики (см. раздел 4):

$$(a) \text{ Если верно } U \Rightarrow B(a), \text{ то верно } U \Rightarrow \forall x B(x)$$

Получим результат вида:

$$\forall x A(x, j) \Rightarrow \forall x A(a, j')$$

И, таким образом, завершим доказательство индукции. Начнём же его:

Для этого надо рассмотреть 2 ветки для 2 сторон данного равенства. Каждая ветка будет построена на основании (1)

$$a = \text{beg}(a, i) \cdot \text{end}(a, i')$$

$$1\text{-я ветка: } \text{beg}(\text{end}(a, i'), j'') = \text{beg}(\text{beg}(\text{end}(a, i'), j''), j') \cdot \text{end}(\text{beg}(\text{end}(a, i'), j''), j'')$$

$$2\text{-я ветка: } \text{end}(\text{beg}(a, i' + j'), i') = \text{beg}(\text{end}(\text{beg}(a, i' + j'), i'), j') \cdot \text{end}(\text{end}(\text{beg}(a, i' + j'), i'), j'')$$

И теперь надо доказать их равенство на основании предположения индукции.

1-я ветка 1-й член конкатенации:

$$\text{beg}(\text{beg}(\text{end}(a, i'), j''), j') = \text{beg}(\text{end}(a, i'), j'), \text{ по (3.3)}$$

1-я ветка 2-й член конкатенации:

$$\text{end}(\text{beg}(\text{end}(a, i'), j''), j'') = \text{beg}(\text{end}(\text{end}(a, i'), j''), 1), \text{ по (4.2) для «внешней» композиции}$$

$$\text{beg}(\text{end}(\text{end}(a, i'), j''), 1) = \text{beg}(\text{end}(a, i' + j'), 1), \text{ из (т.3.2) для «внутренней» композиции}$$

2-я ветка 1-й член конкатенации:

$$\text{beg}(\text{end}(\text{beg}(a, i' + j'), i'), j') = \text{end}(\text{beg}(\text{beg}(a, i' + j'), i' + j), i'), \text{ по предположению индукции,}$$

где вместо a подставлено $\text{beg}(a, i' + j')$

$$\text{end}(\text{beg}(\text{beg}(a, i' + j'), i' + j), i') = \text{end}(\text{beg}(a, i' + j), i'), \text{ по (3.3) для внутренней композиции}$$

$$\text{end}(\text{beg}(a, i' + j), i') = \text{beg}(\text{end}(a, i'), j'), \text{ по предположению индукции.}$$

Совпадение с 1-й веткой 1-м членом конкатенации

2-я ветка 2-й член конкатенации:

$$\text{end}(\text{end}(\text{beg}(a, i' + j'), i'), j'') = \text{end}(\text{beg}(a, i' + j'), i' + j'), \text{ из (т.3.2) для «внешней» композиции}$$

$$\text{end}(\text{beg}(a, i' + j'), i' + j') = \text{beg}(\text{end}(a, i' + j'), 1), \text{ по (4.2)}$$

Совпадение с 1-й веткой 2-м членом конкатенации

Поскольку в обеих конкатенациях совпали 1-е члены с 1-ми и 2-е со 2-ми, то обе конкатенации одинаковы и доказано $\text{beg}(\text{end}(a, i'), j'') = \text{end}(\text{beg}(a, i' + j'), i')$ на основании индуктивного предположения. Это завершает доказательство по индукции.

$$(т.4.2) (i \neq 0 \wedge j \neq 0) \Rightarrow \text{str}(a, i, j) = \text{end}(\text{beg}(a, i + j - 1), i)$$

$$\text{Альтернативное обозначение: } (i \neq 0 \wedge j \neq 0) \Rightarrow a.\text{str}(i, j) = a.\text{beg}(i + j - 1).\text{end}(i)$$

Пояснение. Это та же теорема (т.4.1), переписанная с учётом определения (4.1).

У теоремы (т.4.2) есть практическое значение – если у нас имеется очень длинная (даже бесконечная) строка a , то на практике для получения подстроки $\text{str}(a, i, j)$ нужно поступить иначе, чем $\text{beg}(\text{end}(a, i), j)$. Потому что делать буквально по определению (4.1) – это значит, в промежуточном вычислении $\text{end}(a, i)$ создавать значение из «хвоста» значения a от символа i включительно. А «хвост» значения a может быть как угодно велик, и там может быть сколько угодно символов, не нужных для окончательного результата. Поэтому лучше исполнить промежуточное вычисление $\text{beg}(a, i + j - 1)$, создав значение из «начала» значения a до $i + j - 1$ символа включительно, и лишь затем сформировать результата из «хвоста» этого промежуточного значения. При втором методе «хвост» промежуточного значения целиком попадает в окончательный результат без обработки ненужных символов.

$$(т.4.3) \text{str}(a, i, j') = \text{str}(a, i, j) \cdot \text{str}(a, i + j, 1)$$

Пояснение. Это простое следствие из анализа доказательства (т.4.1). Конкатенация тут состоит из 1-го и 2-го членов конкатенации ветки 1 (Или ветки 2 – без разницы).

Теорема (т.4.3) раскрывает нам то, как подстрока «построена» из своих «атомов». Всё начинается с «атома» $\text{str}(a, i, 1)$, а дальше «атомы» добавляются к самому концу строки по одному, до j -го «атома» из строки a , если речь у нас о подстроке $\text{str}(a, i, j)$

$$(т.4.4) j < k \Rightarrow \text{str}(\text{str}(a, i, j), k, 1) = \ominus$$

Альтернативное обозначение: $j < k \Rightarrow a.\text{str}(i, j).\text{str}(k, 1) = \ominus$

Вытекает из (т.3.3) $i < k \Rightarrow \text{end}(\text{beg}(a, i), k) = \ominus$. Действительно:

$\text{str}(\text{str}(a, i, j), k, 1) = \text{beg}(\text{end}(\text{beg}(\text{end}(a, i), j), k), 1)$, по определению для $\text{str}()$;

$\text{beg}(\text{end}(\text{beg}(\text{end}(a, i), j), k), 1) = \text{end}(\text{beg}(\text{beg}(\text{end}(a, i), j), k), k)$, по (4.2) для «внешней» композиции;

$\text{end}(\text{beg}(\text{beg}(\text{end}(a, i), j), k), k) = \text{end}(\text{beg}(\text{end}(a, i), j), k)$, из (т.3.1) для «средней» композиции, так как $k > j$;

$\text{end}(\text{beg}(\text{end}(a, i), j), k) = \ominus$, из (т.3.3) для «внешней» композиции, так как $k > j$. Это завершает доказательство.

(т.4.5) $1 < k \Rightarrow \text{str}(\text{str}(a, i, 1), k, 1) = \ominus$

Альтернативное обозначение: $1 < k \Rightarrow a.\text{str}(i, 1).\text{str}(k, 1) = \ominus$

Очевидное следствие из (т.4.4), которое показывает, что подстрока, полученная с одного (i -го) места строки, не имеет продолжения со 2-й своей позиции включительно.

Следующие далее теоремы для подстроки $\text{str}(a, 1, j)$ будут позже перенесены и на любые конечные строки, потому, что мы докажем, что для конечных строк верно $a = \text{str}(a, 1, \text{len}(a))$

(т.4.6) $\text{str}(a, 1, j) = \text{beg}(a, j)$

Действительно:

$\text{str}(a, 1, j) = \text{beg}(\text{end}(a, 1), j)$ по определению $\text{str}(a, 1, j)$

$\text{beg}(\text{end}(a, 1), j) = \text{beg}(a, j)$ по (3.4).

(т.4.7) $j \leq k \Rightarrow \text{str}(\text{str}(a, 1, j), 1, k) = \text{str}(a, 1, j)$

Альтернативное обозначение: $j \leq k \Rightarrow a.\text{str}(1, j).\text{str}(1, k) = a.\text{str}(1, j)$

Очевидное следствие теоремы (т.3.1) $\text{beg}(a, \min(i, j)) = \text{beg}(\text{beg}(a, i), j)$ и предыдущей теоремы.

(т.4.8) $\text{end}(\text{str}(a, 1, j), k) = \text{str}(\text{str}(a, 1, j), k, j)$, притом вместо одного данного $\text{end}()$ может быть композиция из подобных $\text{end}()$ сколь угодно большой глубины вложенности.

Альтернативное обозначение: $a.\text{str}(1, j).\text{end}(1, k) = a.\text{str}(1, j).\text{str}(k, j)$

В отношении «сколь угодно большой глубины вложенности» – это очевидно из (т.3.2):

$(i \neq 0 \wedge j \neq 0) \Rightarrow \text{end}(a, i + j - 1) = \text{end}(\text{end}(a, i), j)$

Понятно, что для случая равенства 0 второго аргумента в хотя бы одном из $\text{end}()$ равенство в (т.4.8) будет выполнено. Так как обе его стороны будут равны \ominus в силу аксиом пустых строк (2) и определения $\text{str}()$. Если же числовые аргументы не равны 0, то любая вложенность $\text{end}()$ сводится к одному $\text{end}()$. Поэтому, если (т.4.8) истинна для одного $\text{end}()$, то если

$b = \text{str}(\text{str}(a, 1, j), k, j)$

тогда и дальше можно использовать прежнее число j для дальнейшей замены функции $\text{end}()$ на функцию $\text{str}()$:

$\text{end}(b, n) = \text{str}(b, n, j)$

Поэтому нам достаточно рассмотреть единственную функцию $\text{end}()$, а не их композицию, при этом $k \neq 0$ в (т.4.8).

$\text{end}(\text{str}(a, 1, j), k) = \text{end}(\text{beg}(a, j), k)$, по (т.4.6) – 1-я ветка для левой части равенства (т.4.8)

Теперь разберём 2-ю ветку для правой части равенства (т.4.8)

$$\text{str}(\text{str}(a, 1, j), k, j) = \text{str}(\text{beg}(a, j), k, j), \text{ из (т.4.6)}$$

$$\text{str}(\text{beg}(a, j), k, j) = \text{end}(\text{beg}(\text{beg}(a, j), k + j - 1), k), \text{ из (т.4.3)}$$

$$\text{end}(\text{beg}(\text{beg}(a, j), k + j - 1), k) = \text{end}(\text{beg}(a, j), k), \text{ из (т.3.1) и } k \neq 0$$

Первая и вторая ветка совпали, поэтому (т.4.8) доказана.

$$\text{(т.4.9) } \text{str}(a, 1, j) = \text{str}(\text{str}(a, 1, j), 1, k) \cdot \text{str}(\text{str}(a, 1, j), k', j)$$

$$\text{Альтернативное обозначение: } a.\text{str}(1, j) = a.\text{str}(1, j).\text{str}(1, k) \cdot a.\text{str}(1, j).\text{str}(k', j)$$

Действительно:

$$\text{str}(a, 1, j) = \text{beg}(\text{str}(a, 1, j), k) \cdot \text{end}(\text{str}(a, 1, j), k'), \text{ по (1)}$$

1-я ветка для 1-го члена конкатенации:

$$\text{beg}(\text{str}(a, 1, j), k) = \text{str}(\text{str}(a, 1, j), 1, k), \text{ из (т.4.6)}$$

2-я ветка для 2-го члена конкатенации:

$$\text{end}(\text{str}(a, 1, j), k') = \text{str}(\text{str}(a, 1, j), k', j), \text{ из (т.4.8)}$$

Обе ветки совпали с соответствующими членами конкатенации в (т.4.9). Теорема доказана.

II. Алфавит строк

Аксиомы об «атомах», из которых образованы строки:

$$(5.1) \forall a \forall i \exists j \text{ str}(a, i, 1) = \text{Chr}(j)$$

Пояснение. У нас есть «алфавит» (в следующей аксиоме (5.2)) и на произвольном месте i в произвольной строке a находится некоторый вариант значения (из конечного набора значений) этого алфавита. И рассматриваем мы при этом только одну позицию в строке, поэтому третий аргумент в функции $\text{str}(a, i, 1)$ равен 1.

(5.2) Аксиома об «алфавите»:

$$(\forall i > l_{\text{ChrLim}} : \text{Chr}(i) = \ominus)$$

$$\wedge (\forall i \leq l_{\text{ChrLim}} : \text{Chr}(i) \neq \ominus \wedge \text{Chr}(i) = \text{str}(\text{Chr}(i), 1, 1))$$

$$\wedge (\forall i \leq l_{\text{ChrLim}} \forall j \leq l_{\text{ChrLim}} : \text{Chr}(i) = \text{Chr}(j) \Rightarrow i = j)$$

Пояснение:

Данная аксиома описывает «алфавит», из символов и пустой строки, из которого строятся строки. Символы нумеруются от 0 до l_{ChrLim} включительно, а пустая строка соответствует любому номеру, большему, чем l_{ChrLim} . Будем называть «элементом алфавита» как любой символ, так и пустую строку. Символом будем называть любой элемент алфавита, отличный от пустой строки.

Для названий переменных будем придерживаться таких условностей, соответствующих их значениям:

smb, smb_1, smb_2, \dots для переменных, имеющих в качестве своего значения символ;

abc, abc_1, abc_2, \dots для переменных, имеющих в качестве своего значения элемент алфавита.

Теперь разберём аксиому (5.2).

Если число i больше, чем l_{ChrLim} то функция $\text{Chr}(i)$ возвращает пустую строку:

$$\forall i > l_{\text{ChrLim}} : \text{Chr}(i) = \ominus$$

Как мы увидим ниже, у пустой строки даже длина равна 0, а у символа длина равна 1 в теории строк. Но из всех аксиом выше, включая аксиому алфавита, такой вывод получить невозможно, как будет показано далее.

Если же число i не превышает l_{ChrLim} , то результат функция $\text{Chr}(i)$ – символ. При этом символ по свойствам совпадает со своей собственной подстрокой из единственной первой позиции:

$$\forall i \leq l_{\text{ChrLim}} : \text{Chr}(i) \neq \ominus \wedge \text{Chr}(i) = \text{str}(\text{Chr}(i), 1, 1)$$

Кстати, пустая строка тоже равна подстроке из своей первой позиции – но это можно вывести из аксиом (2.2), (2.4) и определения (4.1) для $\text{str}()$.

При этом все символы отличаются друг от друга, если их номера отличаются:

$$\forall i \leq l_{\text{ChrLim}} \forall j \leq l_{\text{ChrLim}} : \text{Chr}(i) = \text{Chr}(j) \Rightarrow i = j$$

Данная аксиома вводит предметную константу – некоторое число l_{ChrLim} . Первоначально я записывал аксиому с квантором существования по l_{ChrLim} , но, по сути, l_{ChrLim} может быть любым натуральным числом (даже нулём), чтобы в алфавите был хотя бы один символ, отличный от пустой строки.

Можно, конечно, представить себе кодировку, в которой у первого символа «номер» отличается от нуля, но это легко привести к «стандартному» отсчёту от нуля и теоретического смысла

рассматривать 2 предметных константы (ещё и номер первого символа алфавита), вместо одной (номер последнего символа) – нет. С прикладной точки зрения кодировки, в которых номер первого символа больше нуля – тоже не используются. Поэтому и с практической точки зрения нет смысла считать, что номер первого символа может отличаться от нуля.

Разумеется, аксиома алфавита независима от всех предыдущих аксиом, так как можно построить систему с бесконечным количеством символов, которые тоже образуют «цепочки», отвечающие всем аксиомам до (не включая) аксиомы алфавита. Например – массивы натуральных чисел, которые используются в программировании.

Поэтому сформулируем метатеорему – то есть, теорему о теории:

(м.5.1) Аксиома алфавита (5.2) независима от предыдущих аксиом, так как массивы произвольных натуральных чисел удовлетворяют предыдущим аксиомам, но не удовлетворяют аксиоме (5.2).

(5.3) Определение функции $Ach(u)$, обратной к функции $Chr(i)$:

$$(Ach(u) = l'_{ChrLim} \wedge u = \ominus)$$

$$\vee (Ach(u) = i \wedge u \neq \ominus \wedge Chr(i) = u)$$

$$\vee (Ach(u) = \ominus \wedge \forall i Chr(i) \neq u)$$

Функция $Ach(u)$ не «идеально» обратная к функции $Chr(i)$ – для пустой строки она возвращает минимально возможный номер i , при котором $Chr(i) = \ominus$. Для символа же (то есть – не пустой строки) номер вполне однозначный в силу

$$(\forall i \leq l_{ChrLim} \forall j \leq l_{ChrLim} : Chr(i) = Chr(j) \Rightarrow i = j), \text{ из (5.2)}$$

Поэтому легко доказать, что

$$\text{(т.5.1)} \quad Ach(Chr(i)) = \min(i, l'_{ChrLim})$$

Кроме того, если аргумент u в $Ach(u)$ не является элементом алфавита, то $Ach(u) = \ominus$.

Формула (5.3) является определением – то есть, в (5.3) аксиоматизируется лишь обозначение, но никакой дополнительной логики в теории данная формула не создаёт.

Это следует из метода «введения новых функциональных букв и предметных констант». Подробнее смотри раздел 4, пункт VI.5 данной статьи, там имеется и ссылка на источник.

В частности, данное определение опирается на теорему:

$$\exists_1 q ((q = l'_{ChrLim} \wedge u = \ominus) \vee (q = i \wedge u \neq \ominus \wedge Chr(i) = u) \vee (q = \ominus \wedge \forall i Chr(i) \neq u))$$

Её доказательство и является обоснованием того, что (5.3) является всего лишь определением. И данное доказательство нужно «всего лишь» для того, чтобы не вводить аксиому для равенства, применительно к функции $Ach(u)$. Потому что раз функция не имеет дополнительной относительно остальных аксиом логики, то аксиомы равенства будут для неё автоматически выполнены, раз они выполнены для теории, в которой она определена.

Доказательство данной теоремы о существовании и единственности интересно только с точки зрения минимизации количества исходных утверждений теории – её аксиом. В противном случае достаточно просто дописать ещё одну аксиому равенства – пусть она и избыточна, но никакого противоречия в этом нет. Кому же доказательство всё же интересно, то вот такая у него примерная схема:

Дадим обозначение $C(q, u)$ для формулы:

$$(q = l'_{ChrLim} \wedge u = \ominus) \vee (q = i \wedge u \neq \ominus \wedge Chr(i) = u) \vee (q = \ominus \wedge \forall j Chr(j) \neq u)$$

Сначала докажем существование q для произвольного u .

Допустим, истинно $u = \ominus$, тогда истинно

$$C(l'_{ChrLim}, u)$$

Это очевидно по первому члену дизъюнкции в $C(l'_{ChrLim}, u)$.

На основании теоремы дедукции делаем вывод об истинности импликации

$$u = \ominus \Rightarrow C(l'_{ChrLim}, u)$$

По аксиоме логики (см. Раздел 4. Приложение):

$$(b) A(a) \Rightarrow \exists x A(x)$$

Получаем

$$C(l'_{ChrLim}, u) \Rightarrow \exists q C(q, u)$$

По правилу силлогизма из последней формулы и предыдущей $u = \ominus \Rightarrow C(l'_{ChrLim}, u)$ имеем:

$$u = \ominus \Rightarrow \exists q C(q, u)$$

Аналогичным образом доказываем ещё 2 формулы:

$$(u \neq \ominus \wedge Chr(i) = u) \Rightarrow \exists q C(q, u)$$

$$(\forall j Chr(j) \neq u) \Rightarrow \exists q C(q, u)$$

Из предпоследней формулы на основании правила вывода логики (см. Раздел 4. Приложение):

$$(\beta) \text{ Если верно } B(a) \Rightarrow U, \text{ то верно } \exists x B(x) \Rightarrow U$$

получим:

$$(\exists j (u \neq \ominus \wedge Chr(j) = u)) \Rightarrow \exists q C(q, u)$$

Перепишем в эквивалентном виде (эквивалентность – см. Раздел 4. Приложение, в VI.8 поменять стороны в эквивалентности на их отрицания и применить VI.7):

$$(u \neq \ominus \wedge \exists j Chr(j) = u) \Rightarrow \exists q C(q, u)$$

Теперь у нас доказаны 3 формулы:

$$u = \ominus \Rightarrow \exists q C(q, u)$$

$$(u \neq \ominus \wedge \exists j Chr(j) = u) \Rightarrow \exists q C(q, u)$$

$$(\forall j Chr(j) \neq u) \Rightarrow \exists q C(q, u)$$

Из этих трёх формул и аксиомы (см. Раздел 4. Приложение)

$$III.3) (A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \vee B \Rightarrow C))$$

Применив её 2 раза, получим:

$$((u = \ominus) \vee (u \neq \ominus \wedge \exists j Chr(j) = u) \vee (\forall j Chr(j) \neq u)) \Rightarrow \exists q C(q, u)$$

Посылка этой импликации является истинной и может быть отброшена по правилу MP. После чего существование доказано.

Насчёт истинной посылки обоснование такое –

Раскроем по закону дистрибутивности часть дизъюнкции

$$(u = \ominus) \vee (u \neq \ominus \wedge \exists j Chr(j) = u)$$

Получится эквивалентное выражение (то есть, такое, которое можно подставлять вместо исходного в логическую формулу без изменения истинности результата):

$$(u = \ominus \vee u \neq \ominus) \wedge (u = \ominus \vee \exists j \text{Chr}(j) = u)$$

Сократив заведомо истинный член конъюнкции, получим эквивалентное:

$$(u = \ominus \vee \exists j \text{Chr}(j) = u)$$

Так как верно

$u = \ominus \Rightarrow \exists j \text{Chr}(j) = u$, по аксиоме (5.1) $\forall a \forall i \exists j \text{str}(a, i, 1) = \text{Chr}(j)$ при $i = 0$, то предыдущая дизъюнкция может быть переписана в эквивалентном виде:

$$\exists j \text{Chr}(j) = u$$

Теперь вся посылка

$$((u = \ominus) \vee (u \neq \ominus \wedge \exists j \text{Chr}(j) = u)) \vee (\forall j \text{Chr}(j) \neq u)$$

Приобрела вид:

$$(\exists j \text{Chr}(j) = u) \vee (\forall j \text{Chr}(j) \neq u)$$

Истинность последнего утверждения очевидна и легко выводится. Поэтому мы её сокращаем по правилу МР и существование $\exists q C(q, u)$ доказано.

Теперь надо доказать единственность, то есть, надо доказать, что

$$C(q_1, u) \wedge C(q_2, u) \Rightarrow q_1 = q_2$$

Доказывать будем аналогично тому, как для существования – перебирая возможные посылки.

Начнём с $u = \ominus$

Так как все члены дизъюнкции в $C(q_1, \ominus)$ ложные, кроме первой, то верно:

$$q_1 = l'_{ChrLim} \wedge u = \ominus$$

Отсюда:

$$q_1 = l'_{ChrLim}$$

Аналогично:

$$q_2 = l'_{ChrLim}$$

Из этих 2-х равенств получаем:

$$q_1 = q_2$$

Поэтому по теореме дедукции имеем:

$$u = \ominus \Rightarrow (C(q_1, u) \wedge C(q_2, u) \Rightarrow q_1 = q_2)$$

Перебрав оставшиеся 2 посылки и соединив их дизъюнкцией в единую истинную посылку по аналогии с доказательством существования, сократим эту истинную посылку и получим:

$$C(q_1, u) \wedge C(q_2, u) \Rightarrow q_1 = q_2.$$

Единственность тоже доказана. Поэтому (5.3) является определением функции $\text{Ach}(u)$.

(5.4) Определение функции $\text{Comp}(\text{Chr}(i), \text{Chr}(j))$, сравнивающей произвольные элементы алфавита $\text{Chr}(i)$ и $\text{Chr}(j)$):

$$(\text{Comp}(\text{Chr}(i), \text{Chr}(j)) = 0 \wedge i > l_{ChrLim} \wedge j > l_{ChrLim})$$

$$\vee (\text{Comp}(\text{Chr}(i), \text{Chr}(j)) = 0 \wedge i \leq l_{ChrLim} \wedge j \leq l_{ChrLim} \wedge i = j)$$

$$\vee (\text{Comp}(\text{Chr}(i), \text{Chr}(j)) = 1 \wedge i \leq l_{ChrLim} \wedge j \leq l_{ChrLim} \wedge i < j)$$

$$\vee (\text{Comp}(\text{Chr}(i), \text{Chr}(j)) = 1 \wedge i > l_{ChrLim} \wedge j \leq l_{ChrLim})$$

$$\vee (\text{Comp}(\text{Chr}(i), \text{Chr}(j)) = 2 \wedge i \leq l_{ChrLim} \wedge j \leq l_{ChrLim} \wedge i > j)$$

$$\vee (\text{Comp}(\text{Chr}(i), \text{Chr}(j)) = 2 \wedge i \leq l_{ChrLim} \wedge j > l_{ChrLim})$$

Функция $\text{Comp}(abc_1, abc_2)$ здесь сравнивает между собой символы/пустые строки, которые представлены в 1-м и 2-м аргументах. И для одинаковых – возвращает 0. А для разных возвращает 1, когда значение 1-го аргумента меньше второго из-за его меньшего номера в алфавите, либо того, что в первом аргументе – пустая строка, в отличие от второго аргумента. В остальных случаях (2ой аргумент «меньше» 1-го) функция возвращает 2.

Заметим, что функция определена пока только для элементов алфавита, а не для произвольных строк, как $\text{Ach}(u)$. Дело в том, что для произвольных строк функция $\text{Comp}(a, b)$ будет доопределена после формулировки аксиомы равенства для строк.

Пока же, как видно из (5.4), формула строится даже не на элементах алфавита, а на числах. И ещё желательно доказать корректность такого построения. Ведь даже сама формулировка $\text{Comp}(\text{Chr}(i), \text{Chr}(j))$ – это композиция функций, «на входе» которой два числа i, j и результат – тоже число. Но вопрос, разве функция $\text{Chr}(i)$ никогда не уничтожает какие-то необходимые для работы функции $\text{Comp}(\text{Chr}(i), \text{Chr}(j))$ подробности о числе i ?

Перечислим детали определения.

Если оба аргумента являются пустыми строками – то функция равна 0. А если оба аргумента – символы (не пустые строки) и при этом имеют один номер в алфавите – то функция тоже возвращает 0:

$$\begin{aligned} (\text{Comp}(\text{Chr}(i), \text{Chr}(j)) = 0 \wedge i > l_{\text{ChrLim}} \wedge j > l_{\text{ChrLim}}) \\ \vee (\text{Comp}(\text{Chr}(i), \text{Chr}(j)) = 0 \wedge i \leq l_{\text{ChrLim}} \wedge j \leq l_{\text{ChrLim}} \wedge i = j) \end{aligned}$$

Если аргументы являются символами и номер первого символа в алфавите меньше номера второго, то функция возвращает 1. Если же значением первого аргумента является пустая строка, а значением второго является символ (не пустая строка), то функция тоже возвращает 1:

$$\begin{aligned} (\text{Comp}(\text{Chr}(i), \text{Chr}(j)) = 1 \wedge i \leq l_{\text{ChrLim}} \wedge j \leq l_{\text{ChrLim}} \wedge i < j) \\ \vee (\text{Comp}(\text{Chr}(i), \text{Chr}(j)) = 1 \wedge i > l_{\text{ChrLim}} \wedge j \leq l_{\text{ChrLim}}) \end{aligned}$$

Если аргументы являются символами и номер первого символа в алфавите больше номера второго, то функция возвращает 2. Если же значением первого аргумента является символом, а значением второго является пустая строка, то функция тоже возвращает 2:

$$\begin{aligned} (\text{Comp}(\text{Chr}(i), \text{Chr}(j)) = 2 \wedge i \leq l_{\text{ChrLim}} \wedge j \leq l_{\text{ChrLim}} \wedge i > j) \\ \vee (\text{Comp}(\text{Chr}(i), \text{Chr}(j)) = 2 \wedge i \leq l_{\text{ChrLim}} \wedge j > l_{\text{ChrLim}}) \end{aligned}$$

Для тех, кому интересна минимизация количества аксиом равенства, приведу схему доказательства того, что (5.4) является определением для $\text{Comp}(\text{Chr}(i), \text{Chr}(j))$.

Докажем это в два этапа.

Первый этап – доказательство

$$\exists_1 n C(n, i, j)$$

где $C(n, i, j)$ служит обозначением для формулы

$$\begin{aligned} (n = 0 \wedge i > l_{\text{ChrLim}} \wedge j > l_{\text{ChrLim}}) \\ \vee (n = 0 \wedge i \leq l_{\text{ChrLim}} \wedge j \leq l_{\text{ChrLim}} \wedge i = j) \\ \vee (n = 1 \wedge i \leq l_{\text{ChrLim}} \wedge j \leq l_{\text{ChrLim}} \wedge i < j) \\ \vee (n = 1 \wedge i > l_{\text{ChrLim}} \wedge j \leq l_{\text{ChrLim}}) \end{aligned}$$

$$\forall (n = 2 \wedge i \leq l_{ChrLim} \wedge j \leq l_{ChrLim} \wedge i > j)$$

$$\forall (n = 2 \wedge i \leq l_{ChrLim} \wedge j > l_{ChrLim})$$

Доказательство существования и единственности для n тут проводится аналогично со случаем (5.3). Только вариантов посылок тут будет шесть, а не три. Но очевидно, что они охватят все варианты возможных значений для i, j и дизъюнкция посылок в итоге станет истинной и будет отброшена по правилу вывода МР как при доказательстве существования, так и при доказательстве единственности. Поэтому

$\exists_1 n C(n, i, j)$ доказано.

Второй этап – перейти от чисел к элементам алфавита.

Можно вывести в рамках арифметики и достаточно очевидно, что:

$$C(u, i, j) \Leftrightarrow C(u, \min(i, l'_{ChrLim}), \min(j, l'_{ChrLim}))$$

Доказывается данная эквивалентность на основе эквивалентности каждого члена дизъюнкции в $C(u, i, j)$ и $C(u, \min(i, l'_{ChrLim}), \min(j, l'_{ChrLim}))$ соответственно.

С другой стороны,

$$\min(i, l'_{ChrLim}) = \text{Ach}(\text{Chr}(i)), \text{ из-за (т.5.1)}$$

Таким образом:

$$C(u, i, j) \Leftrightarrow C(u, \text{Ach}(\text{Chr}(i)), \text{Ach}(\text{Chr}(j)))$$

Поэтому:

$$C(u, i, j) \Leftrightarrow C(u, \text{Ach}(abc_1), \text{Ach}(abc_2))$$

Мы построили формулу, эквивалентную $C(u, i, j)$ для которой в силу эквивалентности выполнено:

$$\exists_1 n C(u, \text{Ach}(abc_1), \text{Ach}(abc_2))$$

И эта формула представляет собой основу для определения (5.4), так как зависит уже от элементов алфавита, а не от их номеров. Поэтому то, что (5.4) является определением для $\text{Comp}(\text{Chr}(i), \text{Chr}(j))$ – доказано.

III. Концы строк, соединение строк

Аксиома о конце строки, для случая, когда у строки есть конец:

$$(6.1) (i \neq 0 \wedge \text{str}(a, i, 1) = \ominus) \Rightarrow \text{str}(a, i', 1) = \ominus$$

Пояснение.

Аксиома конца строк не может быть выведена из предыдущих аксиом. Действительно, рассмотрим случай, когда все «цепочки» элементов алфавита бесконечны, а после \ominus в «цепочке» может идти любой другой элемент алфавита. А функция $\text{beg}()$ (и $\text{str}()$ соответственно) возвращает тоже бесконечные «цепочки» элементов алфавита – добавляя после последнего элемента алфавита, извлеченному из исходной «цепочки», ещё и бесконечную «цепочку» из \ominus .

Изложенная в предыдущем абзаце интерпретация может быть построена в реальности, поэтому не противоречива и эти «цепочки» соответствуют всем изложенным до аксиомы (6.1) аксиомам – не включая аксиому (6.1).

Поэтому доказана метатеорема:

(м.6.1) Аксиома (6.1) независима от предыдущих аксиом, так как случай, когда пустая строка может чередоваться с другими элементами алфавита внутри «строки», соответствует всем аксиомам перед аксиомой (6.1), а аксиоме (6.1) – не соответствует.

Именно аксиома (6.1) придаёт символу \ominus особый смысл, превращая его наличие в строке – на первом месте или после любого символа (не равного пустой строке \ominus , разумеется) строки – в критерий окончания строки.

Очевидно (доказывается по индукции), что:

$$(т.6.1) 0 < i \leq j \Rightarrow (\text{str}(a, i, 1) = \ominus \Rightarrow \text{str}(a, j, 1) = \ominus)$$

(6.2) Определение для $\text{len}(a)$:

$$(\text{len}(a) = 0 \wedge \text{str}(a, 1, 1) = \ominus)$$

$$\vee (\text{len}(a) = i \wedge \text{str}(a, i, 1) \neq \ominus \wedge \text{str}(a, i', 1) = \ominus)$$

$$\vee (\text{len}(a) = \ominus \wedge \forall i \text{ str}(a, i, 1) \neq \ominus)$$

Пояснение.

Если на первом же месте строки стоит пустая строка (то есть, если $\text{str}(a, 1, 1) = \ominus$), то длина строки считается нулевой:

$$\text{len}(a) = 0 \wedge \text{str}(a, 1, 1) = \ominus$$

Если же в строке есть символ (отличный от пустой строки элемент алфавита), то номер места последнего такого символа в строке считается длиной строки:

$$\text{len}(a) = i \wedge \text{str}(a, i, 1) \neq \ominus \wedge \text{str}(a, i', 1) = \ominus$$

А то, что после такого символа уже нет символов – обеспечено из-за (т.6.1).

Если же строка a вообще нигде не кончается – то есть, на каждом её месте имеется символ (отличный от пустой строки элемент алфавита), то значением функции $\text{len}(a)$ является пустая строка \ominus :

$$\text{len}(a) = \ominus \wedge \forall i \text{ str}(a, i, 1) \neq \ominus$$

Случай $\text{len}(a) = i$ отличается от случая $\text{len}(a) = \ominus$ из-за того, что $i \neq \ominus$ по (2.6).

Доказательство того, что (6.2) является определением, проводится аналогично доказательству для (5.3).

$$(т.6.2) \text{ len}(a) \neq \ominus \Rightarrow \text{str}(a, \text{len}(a)', 1) = \ominus \wedge (\text{len}(a) = 0 \vee \text{str}(a, \text{len}(a), 1) \neq \ominus)$$

Следует из определения (6.2). Вариант посылки

$$(\text{len}(a) = i \wedge \text{str}(a, i, 1) \neq \ominus \wedge \text{str}(a, i', 1) = \ominus)$$

Очевидно соответствует (т.6.2).

Вариант посылки:

$$(\text{len}(a) = 0 \wedge \text{str}(a, 1, 1) = \ominus)$$

Тоже соответствует в силу того, что в нём и записано

$$\text{str}(a, \text{len}(a)', 1) = \ominus, \text{ так как } 1 = \text{len}(a)' \text{ в данном случае.}$$

К полученным импликациям добавляем «лишнюю» посылку, что не меняет истинности, и получим 2 истинных импликации:

$$\text{len}(a) \neq \ominus$$

$$\Rightarrow ((\text{len}(u) = i \wedge \text{str}(a, i, 1) \neq \ominus \wedge \text{str}(a, i', 1) = \ominus) \Rightarrow \text{str}(a, \text{len}(a)', 1) = \ominus) \wedge (\text{len}(a) = 0 \vee \text{str}(a, \text{len}(a), 1) \neq \ominus)$$

$$\text{len}(a) \neq \ominus$$

$$\Rightarrow ((\text{len}(u) = 0 \wedge \text{str}(a, 1, 1) = \ominus) \Rightarrow \text{str}(a, \text{len}(a)', 1) = \ominus \wedge (\text{len}(a) = 0 \vee \text{str}(a, \text{len}(a), 1) \neq \ominus))$$

Заодно добавим истинное:

$$\text{len}(a) \neq \ominus$$

$$\Rightarrow ((\text{len}(u) = \ominus \wedge \forall i \text{ str}(a, i, 1) \neq \ominus) \Rightarrow \text{str}(a, \text{len}(a)', 1) = \ominus \wedge (\text{len}(a) = 0 \vee \text{str}(a, \text{len}(a), 1) \neq \ominus))$$

Последнее верно потому, что вложенные импликации эквивалентны импликации с конъюнкцией посылок, а в данном случае у нас противоречие в конъюнкции посылок:

$$\text{len}(a) \neq \ominus \wedge \text{len}(u) = \ominus \text{ А из противоречия следует всё.}$$

Теперь во всех 3 импликациях меняем местами вторую посылку с первой (это даёт эквивалентное утверждение) и получим:

$$Part_1 \Rightarrow (\text{len}(a) \neq \ominus \Rightarrow \text{str}(a, \text{len}(a)', 1) = \ominus \wedge (\text{len}(a) = 0 \vee \text{str}(a, \text{len}(a), 1) \neq \ominus))$$

$$Part_2 \Rightarrow (\text{len}(a) \neq \ominus \Rightarrow \text{str}(a, \text{len}(a)', 1) = \ominus \wedge (\text{len}(a) = 0 \vee \text{str}(a, \text{len}(a), 1) \neq \ominus))$$

$$Part_3 \Rightarrow (\text{len}(a) \neq \ominus \Rightarrow \text{str}(a, \text{len}(a)', 1) = \ominus \wedge (\text{len}(a) = 0 \vee \text{str}(a, \text{len}(a), 1) \neq \ominus))$$

Понятно, что понимается под $Part_1, Part_2, Part_3$. Из этих 3 импликаций получим по аксиоме логики III.3 из Раздела 4. Приложение, применив её 2 раза:

$$Part_1 \vee Part_2 \vee Part_3 \Rightarrow (\text{len}(a) \neq \ominus \Rightarrow \text{str}(a, \text{len}(a)', 1) = \ominus \wedge (\text{len}(a) = 0 \vee \text{str}(a, \text{len}(a), 1) \neq \ominus))$$

Но посылка в данной импликации – это определение (6.2), то есть – истинное утверждение. Отбрасываем её по правилу MP. Теорема доказана.

$$(т.6.3) \text{ len}(a) \neq \ominus \Rightarrow (\text{len}(a) < i \Rightarrow \text{str}(a, i, 1) = \ominus)$$

Доказательство – из (т.6.1) $0 < i \leq j \Rightarrow (\text{str}(a, i, 1) = \ominus \Rightarrow \text{str}(a, j, 1) = \ominus)$ и (т.6.2) $\text{len}(a) \neq \ominus \Rightarrow \text{str}(a, \text{len}(a)', 1) = \ominus$.

Следующие 2 теоремы позволяют «поглощать» функции $\text{end}()$ и $\text{beg}()$ – а, значит, и $\text{str}()$ – во вложенной композиции тем $\text{str}()$, который применяется к ним в композиции и извлекает элемент алфавита только из одной позиции строки.

Данные теоремы уместно было бы разместить в подразделе «l. Разбиение строк», но они нужны сейчас, после определения функции $\text{len}()$ длины строки. Так как с их помощью будут доказаны следующие за ними теоремы о длине подстрок.

(т.6.4) $j \leq i \Rightarrow \text{str}(\text{beg}(a, i), j, 1) = \text{str}(a, j, 1)$

Альтернативное обозначение: $0 < j \leq i \Rightarrow a.\text{beg}(i).\text{str}(j, 1) = a.\text{str}(j, 1)$

Доказательство при гипотезе дедукции $0 < j \leq i$:

$\text{str}(\text{beg}(a, i), j, 1) = \text{end}(\text{beg}(\text{beg}(a, i), j), j) = \text{end}(\text{beg}(a, j), j) = \text{str}(a, j, 1)$

Осталось рассмотреть случай $j = 0$:

$\text{str}(\text{beg}(a, i), j, 1) = \text{str}(\text{beg}(a, i), 0, 1) = \text{beg}(\text{end}(\text{beg}(a, i), 0), 1) = \text{beg}(\ominus, 1) = \ominus$

$\text{str}(a, j, 1) = \text{str}(a, 0, 1) = \text{beg}(\text{end}(a, 0), 1) = \text{beg}(\ominus, 1) = \ominus$

Поэтому и для случая $j = 0$ доказано:

$\text{str}(\text{beg}(a, i), j, 1) = \text{str}(a, j, 1)$

Теорема (т.6.4) доказана.

(т.6.5) $i \neq 0 \wedge j \neq 0 \Rightarrow \text{str}(\text{end}(a, i), j, 1) = \text{str}(a, i + j - 1, 1)$

Альтернативное обозначение: $i \neq 0 \wedge j \neq 0 \Rightarrow a.\text{end}(i).\text{str}(j, 1) = a.\text{str}(i + j - 1, 1)$

Доказательство при гипотезе дедукции $i \neq 0 \wedge j \neq 0$:

$\text{str}(\text{end}(a, i), j, 1) = \text{beg}(\text{end}(\text{end}(a, i), j), 1) = \text{beg}(\text{end}(i + j - 1), 1) = \text{str}(a, i + j - 1, 1)$

Теорема (т.6.5) доказана.

(т.6.6) $\text{len}(a) \neq \ominus \wedge 0 < i \leq \text{len}(a) \Rightarrow \text{str}(a, i, 1) \neq \ominus$

Доказательство при дедуктивном предположении $\text{len}(a) \neq \ominus \wedge 0 < i \leq \text{len}(a)$:

Допустим, имеется такой i , что:

$\text{str}(a, i, 1) = \ominus$

Тогда в силу (т.6.1) имеем:

$0 < i \leq \text{len}(a) \Rightarrow (\text{str}(a, i, 1) = \ominus \Rightarrow \text{str}(a, \text{len}(a), 1) = \ominus)$

В силу чего доказано:

$\text{str}(a, \text{len}(a), 1) = \ominus$

Но это противоречит (т.6.2)

$\text{len}(a) \neq \ominus \Rightarrow \text{str}(a, \text{len}(a)', 1) = \ominus \wedge (\text{len}(a) = 0 \vee \text{str}(a, \text{len}(a), 1) \neq \ominus)$

Так как при имеющимся дедуктивном предположении верно:

$\text{str}(a, \text{len}(a), 1) \neq \ominus$

Полученное противоречие завершает доказательство.

(т.6.7) $\text{len}(a) \neq \ominus \wedge i \leq \text{len}(a) \Rightarrow \text{len}(\text{beg}(a, i)) = i \wedge (\text{len}(\text{end}(a, i)) = \text{len}(a) - i + 1 \vee i = 0)$

Доказательство.

Для случая $i = 0$ очевидно, так как $\text{len}(\text{beg}(a, 0)) = 0 \wedge i = 0$.

Поэтому докажем заключение предложения (т.6.7) при дедуктивном предположении:

$\text{len}(a) \neq \ominus \wedge 0 < i \leq \text{len}(a)$

Тогда

$\text{str}(\text{beg}(a, i), i, 1) = \text{str}(a, i, 1)$, из-за (т.6.4)

$\text{str}(a, i, 1) \neq \ominus$, из-за (т.6.6)

Поэтому

$$\text{str}(\text{beg}(a, i), i, 1) \neq \ominus$$

С другой стороны:

$\text{str}(\text{beg}(a, i), i', 1) = \ominus$, из-за (т.4.4) $j < k \Rightarrow \text{str}(\text{str}(a, i, j), k, 1) = \ominus$ при $i = 1$ и равенстве (т.4.6) $\text{str}(a, 1, j) = \text{beg}(a, j)$.

Два доказанных утверждения дают в силу определение (6.2) для $\text{len}(u)$:

$\text{len}(\text{beg}(a, i)) = i$, теорема (т.6.7) доказана для первого члена конъюнкции заключения.

$\text{str}(\text{end}(a, i), \text{len}(a) - i + 1, 1) = \text{str}(a, i + (\text{len}(a) - i + 1) - 1, 1)$, из-за (т.6.5).

Поэтому:

$$\text{str}(\text{end}(a, i), \text{len}(a) - i + 1, 1) = \text{str}(a, \text{len}(a), 1)$$

$\text{str}(a, \text{len}(a), 1) \neq \ominus$, из-за (т.6.2) $\text{len}(a) \neq \ominus \Rightarrow \dots \wedge (\text{len}(a) = 0 \vee \text{str}(a, \text{len}(a), 1) \neq \ominus)$.

Получили:

$$\text{str}(\text{end}(a, i), \text{len}(a) - i + 1, 1) \neq \ominus$$

Аналогично:

$$\text{str}(\text{end}(a, i), (\text{len}(a) - i + 1)', 1) = \text{str}(a, \text{len}(a)', 1) = \ominus$$

Два последних утверждения доказывают:

$$\text{len}(\text{end}(a, i)) = \text{len}(a) - i + 1$$

Теорема (т.6.7) доказана теперь и для второго члена конъюнкции заключения при $i \neq 0$.

Поэтому теорема доказана для случаев $i = 0$ и $i \neq 0$. Доказана полностью.

(с.6.7) $\text{len}(a) \neq \ominus \wedge i \neq 0 \wedge i + j - 1 \leq \text{len}(a) \Rightarrow \text{len}(\text{str}(a, i, j)) = j$

Доказательство при предположении дедукции $\text{len}(a) \neq \ominus \wedge i \neq 0 \wedge j \neq 0 \wedge i + j - 1 \leq \text{len}(a)$.

Случай 1. $j = 0$

На основании определения $\text{str}()$ имеем:

$$\text{str}(a, i, j) = \text{beg}(\text{end}(a, i), 0)$$

На основании аксиомы (2.1) $\text{beg}(a, 0) = \ominus$ получаем:

$$\text{str}(a, i, j) = \ominus$$

На основании определения $\text{len}()$ и предыдущего равенства получаем:

$$\text{len}(\text{str}(a, i, 0)) = 0$$

То есть, для Случая 1 ($j = 0$) следствие (с.6.7) доказано:

$$\text{len}(\text{str}(a, i, j)) = j$$

Случай 2. $j \neq 0$

На основании предположении дедукции, Случая 2 и теоремы (т.4.2) которая (напоминание):

$$(i \neq 0 \wedge j \neq 0) \Rightarrow \text{str}(a, i, j) = \text{end}(\text{beg}(a, i + j - 1), i)$$

Получаем:

$$\text{str}(a, i, j) = \text{end}(\text{beg}(a, i + j - 1), i)$$

На основании теоремы (т.6.7) которая (напоминание):

$$\text{len}(a) \neq \ominus \wedge i \leq \text{len}(a) \Rightarrow \text{len}(\text{beg}(a, i)) = i \wedge (\text{len}(\text{end}(a, i)) = \text{len}(a) - i + 1 \vee i = 0)$$

Получаем:

$$\text{len}(\text{beg}(a, i + j - 1)) = i + j - 1$$

Применяя теорему (т.б.7) ещё раз – теперь к $\text{beg}(a, i + j - 1)$, получаем:

$$\text{len}(\text{end}(\text{beg}(a, i + j - 1), i)) = \text{len}(\text{beg}(a, i + j - 1)) - i + 1 = (i + j - 1) - i + 1 = j$$

То есть, для Случая 2 ($j \neq 0$) следствие (с.б.7) тоже доказано.

Следствие доказано.

$$\text{(т.б.8)} \quad \text{len}(a) = \ominus \Rightarrow \text{len}(\text{beg}(a, i)) = i \wedge (\text{len}(\text{end}(a, i)) = \ominus \vee i = 0)$$

Доказательство.

Для случая $i = 0$ теорема очевидно выполнена. Рассмотрим случай $i \neq 0$.

При $\text{len}(a) = \ominus$ все $\text{str}(a, i, 1) \neq \ominus$ при $i \neq 0$.

Поэтому для $\text{beg}(a, i)$ в доказательстве ничего не меняется, в сравнении с (т.б.7), поэтому:

$$\text{len}(\text{beg}(a, i)) = i$$

А для $\text{end}(a, i)$ все символы оказываются из a , то есть – для любого $j \neq 0$ верно:

$$\text{str}(\text{end}(a, i), i, 1) \neq \ominus.$$

Поэтому, по определению (б.2) для $\text{len}()$ получаем:

$$\text{len}(\text{end}(a, i)) = \ominus$$

Теорема доказана.

$$\text{(т.б.9)} \quad \text{len}(a) \neq \ominus \wedge \text{len}(a) < i \Rightarrow \text{len}(\text{beg}(a, i)) = \text{len}(a) \wedge \text{len}(\text{end}(a, i)) = 0$$

Доказательство при дедуктивном предположении $\text{len}(a) \neq \ominus \wedge \text{len}(a) < i$.

Случай 1. $\text{len}(a) = 0$

На основании (4.1) которая (напоминание):

$$\text{str}(a, i, j) = \text{beg}(\text{end}(a, i), j)$$

Запишем:

$$\text{str}(\text{beg}(a, i), 1, 1) = \text{beg}(\text{end}(\text{beg}(a, i), 1), 1)$$

На основании (3.4) которая (напоминание):

$$a = \text{end}(a, 0')$$

убираем $\text{end}()$:

$$\text{str}(\text{beg}(a, i), 1, 1) = \text{beg}(\text{beg}(a, 1), 1)$$

На основании (т.3.1) $\text{beg}(a, \min(i, j)) = \text{beg}(\text{beg}(a, i), j)$ получим:

$$\text{str}(\text{beg}(a, i), 1, 1) = \text{beg}(a, 1)$$

На основании (т.4.6) $\text{str}(a, 1, j) = \text{beg}(a, j)$ получим:

$$\text{str}(\text{beg}(a, i), 1, 1) = \text{str}(a, 1, 1)$$

Но так как $\text{len}(a) = 0$, то по определению $\text{len}()$ имеем:

$$\text{str}(a, 1, 1) = \ominus$$

Поэтому:

$$\text{str}(\text{beg}(a, i), 1, 1) = \ominus$$

И по определению $\text{len}()$ получаем:

$$\text{len}(\text{beg}(a, i)) = 0$$

$$\text{len}(\text{beg}(a, i)) = \text{len}(a)$$

Теорема (т.б.9) для $\text{beg}(a, i)$ и Случая 1 доказана.

Теперь рассмотрим $\text{str}(\text{end}(a, i), 1, 1)$, тоже раскрывая $\text{str}()$ по определению (4.1):

$$\text{str}(\text{end}(a, i), 1, 1) = \text{beg}(\text{end}(\text{end}(a, i), 1), 1)$$

На основании (3.4) которая (напоминание):

$$a = \text{end}(a, 0')$$

убираем тот $\text{end}()$, где второй аргумент равен 1:

$$\text{str}(\text{end}(a, i), 1, 1) = \text{beg}(\text{end}(a, i), 1)$$

На основании (4.1) (см. тут чуть выше) получим:

$$\text{str}(\text{end}(a, i), 1, 1) = \text{str}(a, i, 1)$$

Но для рассматриваемого Случая 1 мы уже получили при рассмотрении $\text{str}(\text{beg}(a, i), 1, 1)$:

$$\text{str}(a, 1, 1) = \ominus$$

На основании (т.6.1) которая (напоминание):

$$0 < i \leq j \Rightarrow (\text{str}(a, i, 1) = \ominus \Rightarrow \text{str}(a, j, 1) = \ominus)$$

Получаем:

$$\text{str}(a, i, 1) = \ominus$$

Поэтому

$$\text{str}(\text{end}(a, i), 1, 1) = \ominus$$

Из определения для $\text{len}()$ из последнего равенства получаем:

$$\text{len}(\text{end}(a, i)) = 0$$

Теорема (т.6.9) для $\text{end}(a, i)$ и Случая 1 доказана.

Теорема (т.6.9) для Случая 1 доказана.

Случай 2. $\text{len}(a) \neq 0$

Так как из-за (т.4.2) ($i \neq 0 \wedge j \neq 0$) $\Rightarrow \text{str}(a, i, j) = \text{end}(\text{beg}(a, i + j - 1), i)$, то:

$$\text{str}(\text{beg}(a, i), \text{len}(a), 1) = \text{end}(\text{beg}(\text{beg}(a, i), \text{len}(a)), 1)$$

Так как (т.3.1) $\text{beg}(a, \min(i, j)) = \text{beg}(\text{beg}(a, i), j)$, то:

$$\text{str}(\text{beg}(a, i), \text{len}(a), 1) = \text{end}(\text{beg}(a, \text{len}(a)), 1)$$

И используя снова (т.4.2), но теперь «в другую сторону», получаем:

$$\text{str}(\text{beg}(a, i), \text{len}(a), 1) = \text{str}(a, \text{len}(a), 1)$$

Аналогично доказывается

$$\text{str}(\text{beg}(a, i), \text{len}(a)', 1) = \text{str}(a, \text{len}(a)', 1)$$

Так как $\text{len}(a) \neq 0$, то по определению имеем:

$$\text{str}(a, \text{len}(a), 1) \neq \ominus$$

$$\text{str}(a, \text{len}(a)', 1) = \ominus$$

Но раз для $\text{beg}(a, i)$ выполнены те же 2 соотношения, что для a , то верно:

$$\text{len}(\text{beg}(a, i)) = \text{len}(a)$$

Теорема (т.6.9) для $\text{beg}(a, i)$ и Случая 2 доказана.

$$\text{str}(\text{end}(a, i), 1, 1) = \text{beg}(\text{end}(a, i), 1)$$

На основании (4.1) (см. тут чуть выше) получим:

$$\text{str}(\text{end}(a, i), 1, 1) = \text{str}(a, i, 1)$$

Но для рассматриваемого Случая 2 мы уже получили при рассмотрении $\text{str}(\text{beg}(a, i), \text{len}(a)', 1)$:

$$\text{str}(a, \text{len}(a)', 1) = \ominus$$

На основании (т.б.1) которая (напоминание):

$$0 < i \leq j \Rightarrow (\text{str}(a, i, 1) = \ominus \Rightarrow \text{str}(a, j, 1) = \ominus)$$

Получаем, так как $\text{len}(a)' \leq i$ из-за дедуктивного предположения $\text{len}(a) < i$:

$$\text{str}(a, i, 1) = \ominus$$

Поэтому

$$\text{str}(\text{end}(a, i), 1, 1) = \ominus$$

Из определения для $\text{len}()$ из последнего равенства получаем:

$$\text{len}(\text{end}(a, i)) = 0$$

Теорема (т.б.9) для $\text{end}(a, i)$ и Случая 2 доказана.

Теорема (т.б.9) для Случая 2 доказана.

Теорема (т.б.9) доказана.

$$\text{(с.б.9)} \quad \text{len}(a) \neq \ominus \Rightarrow \text{len}(\text{beg}(a, i)) = \min(i, \text{len}(a)) \wedge (\text{len}(\text{end}(a, i)) = \max(\text{len}(a), i - 1) - i + 1 \vee i = 0)$$

Это следствие очевидно из предыдущих теорем - (т.б.7), которая (напоминание):

$$\text{len}(a) \neq \ominus \wedge i \leq \text{len}(a) \Rightarrow \text{len}(\text{beg}(a, i)) = i \wedge (\text{len}(\text{end}(a, i)) = \text{len}(a) - i + 1 \vee i = 0)$$

И теоремы (т.б.9), которая (напоминание):

$$\text{len}(a) \neq \ominus \wedge \text{len}(a) < i \Rightarrow \text{len}(\text{beg}(a, i)) = \text{len}(a) \wedge (\text{len}(\text{end}(a, i)) = 0)$$

Как видим из этих теорем:

Значение $\text{len}(\text{beg}(a, i))$ всегда совпадает с тем, что меньше среди двух вариантов: i и $\text{len}(a)$. То есть – равно $\min(i, \text{len}(a))$.

И

Значение $\text{len}(\text{end}(a, i))$ либо 0 при $i = 0$, либо от $\text{len}(a)$ отнимается $(i - 1)$, когда есть от чего отнимать (в (т.б.7), $i - 1 < \text{len}(a)$), либо от уменьшаемого $\text{len}(a)$ ничего не остаётся (в (т.б.9), $\text{len}(a) \leq i - 1$), когда уменьшаемое меньше вычитаемого. То есть, $\text{len}(\text{end}(a, i))$ оказывается равным арифметической разности между $\text{len}(a)$ и $(i - 1)$. А арифметическая разность записывается, например, формулой $\max(\text{len}(a), i - 1) - i + 1$.

Следствие доказано.

(7.1) Аксиома о результате конкатенации при конечной 1-й строке:

$$\text{len}(a) \neq \ominus \Rightarrow (i \leq \text{len}(a) \Rightarrow \text{str}(a \cdot b, i, 1) = \text{str}(a, i, 1)) \wedge (i > \text{len}(a) \Rightarrow \text{str}(a \cdot b, i, 1) = \text{str}(b, i - \text{len}(a), 1))$$

Пояснение.

Аксиома о результате конкатенации для произвольных строк декларирует сохранении прежнего порядка (как в исходных строках) элементов алфавита на своих новых местах в новой строке относительно «соседей». То есть – начало второй «цепочки» символов «цепляется» к концу первой «цепочки» символов, но исходные «цепочки» не разрываются при создании новой строки.

Аксиома о порядке символов в конкатенации необходима: иначе слияние может быть, например, слиянием 2-х упорядоченных массивов в упорядоченный массив. И все предыдущие аксиомы будут соответствовать такой интерпретации.

При интерпретации «упорядоченные массивы символов» бесконечный 2-й массив полностью «вытесняет» первый массив при их конкатенации, если символы 2-го упорядоченного массива символов идут раньше в алфавите, чем 1-ого. И все аксиомы для упорядоченных массивов символов при разбиении строк остаются в силе – потому что такое извлечение части из массива упорядоченных символов – тоже даёт упорядоченный массив.

Кстати, видимо, можно включить в рассмотрение и случай бесконечной первой строки в конкатенации:

(7.2) Аксиома о результате конкатенации при бесконечной 1-й строке:

$$\text{len}(a) = \ominus \Rightarrow a = a \cdot b$$

(м.7.1) Аксиомы (7.1) и (7.2) независимы от предыдущих аксиом, так как случай упорядоченных массивов символов соответствует всем аксиомам перед аксиомами (7.1) и (7.2), но не соответствует аксиомам (7.1) и (7.2).

Данная теорема доказана чуть выше. Разумеется, доказательство метатеорем не может быть формализовано в рамках теории, о которой проведено данное доказательство, так как касается вопросов, выходящих за рамки самой теории, о которой сформулированы метатеоремы.

(т.7.1) $\text{len}(a_1) \neq \ominus \wedge \dots \wedge \text{len}(a_n) \neq \ominus \Rightarrow \text{len}(a_1 \cdot \dots \cdot a_n) = \text{len}(a_1) + \dots + \text{len}(a_n)$

Очевидное следствие для $n = 2$ из аксиомы конкатенации и определения $\text{len}()$. Доказывается по индукции, так как $\text{len}(a_1 \cdot \dots \cdot a_n) = \text{len}((a_1 \cdot \dots \cdot a_n) \cdot a_n)$, поэтому вопрос для n сводится к предыдущему n и случаю $n = 2$.

(т.7.2) $i \leq l_{ChrLim} \Rightarrow \text{len}(\text{Chr}(i)) = 1$

Очень удобная теорема вместе с предыдущей – так как не пустая строка равна конкатенации собственных символов. Впрочем, утверждение про «равна конкатенации своих символов» невозможно доказать без теоремы о равенстве строк в следующем разделе.

Но сейчас мы уже можем доказать, что длина конечной строки равна количеству этих символов. Доказательство теоремы (т.7.2) не требует аксиом (7), поэтому её можно было бы доказать перед аксиомой (7.1), но она удобна в связке с предыдущей теоремой (т.7.1), поэтому расположил эти теоремы рядом.

Доказательство при предположении дедукции $i \leq l_{ChrLim}$.

Так как в аксиоме (5.2) есть конъюнктивный член:

$$\forall i \leq l_{ChrLim} : \text{Chr}(i) \neq \ominus \wedge \text{Chr}(i) = \text{str}(\text{Chr}(i), 1, 1)$$

То верно:

$$\text{str}(\text{Chr}(i), 1, 1) = \text{Chr}(i) \neq \ominus.$$

С другой стороны, из-за (т.4.5) которая (напоминание):

$$1 < k \Rightarrow \text{str}(\text{str}(a, i, 1), k, 1) = \ominus$$

Получим:

$$\text{str}(\text{Chr}(i), 2, 1) = \text{str}(\text{str}(\text{Chr}(i), 1, 1), 2, 1) = \ominus$$

Таким образом у нас выполнено:

$$\text{str}(\text{Chr}(i), 1, 1) \neq \ominus \wedge \text{str}(\text{Chr}(i), 2, 1) = \ominus$$

В соответствии с определением (6.2) для $\text{len}(a)$ это соответствует 2-му члену в дизъюнкции этого определения при $i = 1$:

$$\text{len}(a) = i \wedge \text{str}(a, i, 1) \neq \ominus \wedge \text{str}(a, i', 1) = \ominus$$

Поэтому при сделанном дедуктивном предположении $i \leq l_{ChrLim}$ верно:

$$\text{len}(\text{Chr}(i)) = 1$$

Теорема доказана.

IV. Равенство строк

Аксиома о равенстве строк:

$$(8.1) a = b \Leftrightarrow \forall i \text{ str}(a, i, 1) = \text{str}(b, i, 1)$$

Пояснения.

Аксиома равенства строк – независима от предыдущих. Так как функции взятия подстроки могут сохранять «ссылку» на исходный объект и этим отличаться от «нового» (возникшего в результате конкатенации, например) объекта – это опять опыт из программирования. И без данной аксиомы теорию, составленную из предыдущих аксиом (до 8.1, не включая) можно расширить таким образом, что будет верно:

$\text{str}(\langle \text{строка} \rangle, 1, 3) \neq \langle \text{стр} \rangle$, при этом никаких противоречий не возникнет.

Действительно, результат $\langle \text{строка} \rangle.\text{str}(\langle \text{строка} \rangle, 1, 3)$ может допускать возможность выяснить – с какой позиции была получена данная подстрока и из какой строки. И получать эти значения можно при помощи свойств $i_{in}(a)$ и $\text{parent}(a)$ соответственно. И тогда будет верно:

$$\langle \text{строка} \rangle.\text{str}(1, 3).i_{in} = 1$$

$$\langle \text{строка} \rangle.\text{str}(1, 3).\text{parent} = \langle \text{строка} \rangle.$$

В то время как у подстроки, содержащей все символы, свойство parent может возвращать (если так запрограммировать) результат $\langle \text{стр} \rangle$:

$$\langle \text{стр} \rangle.\text{parent} = \langle \text{стр} \rangle$$

Получается, что НЕ верно:

$$\langle \text{строка} \rangle.\text{str}(1, 3) = \langle \text{стр} \rangle$$

Так как, в соответствии с аксиомами равенства (см. Раздел 4 Приложение):

$$(J_1) a = a$$

$$(J_2) a = b \Rightarrow (A(a) \Rightarrow A(b))$$

И в случае равенства должно было быть:

$$\langle \text{строка} \rangle.\text{str}(1, 3).\text{parent} = \langle \text{стр} \rangle.\text{parent}, \text{ что неверно в разбираемом примере.}$$

А это значит, что без аксиомы (8.1), а только на основе изложенных ранее аксиом, невозможно доказать, что:

$\text{str}(\langle \text{строка} \rangle, 1, 3) = \langle \text{стр} \rangle$, потому что у нас есть непротиворечивое расширение теории, в которой доказывается отрицание данного равенства.

Тут есть нюанс: до аксиомы алфавита (не включая аксиому алфавита) все строки (аргументы, которые находятся на месте для произвольных строк в функциях) являются одной и той же строкой, либо её производной (результатом некоторых функций от общей произвольной строки-предка). Для таких аксиом все равенства подразумевают одинаковые дополнительные свойства ($i_{in}(a)$ и $\text{parent}(a)$).

Но аксиома алфавита и аксиома о результате конкатенации являются единственными аксиомами перед аксиомой о равенстве строк, где присутствуют произвольные строки, которые не являются производными (через некоторые функции) от общей строки-предка. Однако, в обоих этих аксиомах с обеих сторон равенства всегда находятся строки из единственного элемента алфавита каждая.

Поэтому для непротиворечивого построения теории со свойствами $i_{in}(a)$ и $parent(a)$ надо считать, что они сохраняются лишь для строк, длина которых больше 1. Как только из строки получена пустая строка или подстрока длиной 1 – эти свойства меняются на «базовые». Например, имеется аксиома:

$$a.str(i, 1).parent = a.str(i, 1)$$

То есть, строка из одного символа и пустая строки всегда – «независимы» и имеют в качестве «предков» самих себя.

А это значит, что всё же можно построить непротиворечивую теорию, расширяющую аксиомы перед аксиомой равенства строк, если считать, что равенство имеется только тогда, когда у равных строк не только последовательность символов $a.str(i, 1)$ одинакова по i , но одинакова последовательность символов по i и в $a.parent.str(i, 1)$.

Таким образом, доказана следующая метатеорема:

(м.8.1) Аксиома (8.1) независима от предыдущих аксиом, так как есть пример из ООП (объектно-ориентированного программирования), когда функции $beg()$ и $end()$ создают «детей» у «независимых» строк. И между «независимыми» строками и «детьми» есть разница помимо последовательности символов. Такие объекты ООП соответствуют всем аксиомам перед аксиомой (8.1), но не соответствуют аксиоме (8.1).

Как уже упоминалось, доказательство у метатеорем всегда неформально и в силу этого есть риск пропустить в рассуждениях важный нюанс, из-за пропуска которого доказательство окажется ошибочным. Например, в последней метатеореме можно было упустить нюанс с «независимостью» элементов алфавита. Поэтому к доказательствам метатеорем надо относиться с особой осторожностью.

Впрочем, метатеоремы не влияют на внутреннюю логику теории, к которой они относятся, даже если метатеорема ошибочна. Но в случае ошибочности метатеоремы будет не совсем верным наше неформальное понимание «внутренней логики» теории – что всё равно не создаёт риска построения ошибочных логических выводов средствами самой теории.

Для того, чтобы мы могли использовать знак равенства в нашей теории «обычным» образом (а мы именно это и делали) необходимо, чтобы для теории были истинны 2 следующих аксиомы теории первого порядка с равенством (см. Раздел 4. Приложение):

$$(J_1) a = a$$

$$(J_2) a = b \Rightarrow (A(a) \Rightarrow A(b))$$

Где $A(a)$ представляет собой любую логически корректную формулу, построенную из функциональных букв, предикатных букв теории, кванторов (в область действия которых не попадают a, b) и логических связок.

Но на самом деле – тут тонкий момент, чтобы то, что ты понимаешь под знаком равенства, действительно обладало свойствами равенства. И пояснения к аксиоме (8.1) – тому пример. Просто написание знака « $=$ » в некоторых аксиомах не делает предикаты на его основе соответствующим утверждениям (J_1) и (J_2) .

Однако у нас есть критерий, который позволяет отнести теорию первого порядка в категорию

теории первого порядка с равенством – см. Раздел 4. Приложение, пункт VI.4. Вот в соответствии с этим пунктом мы и допишем необходимые нам для получения теории первого порядка с равенством утверждения. И допишем их в качестве аксиом:

а. Для каждой функциональной буквы $f_n()$ с аргументами x_1, \dots, x_m (где m – количество аргументов функции $f_n(x_1, \dots, x_m)$) должно быть истинно:

$$(x_1 = y_1 \wedge \dots \wedge x_m = y_m) \Rightarrow (f_n(x_1, \dots, x_m) = f_n(y_1, \dots, y_m))$$

Поэтому:

$$(8.2) (x_1 = y_1 \wedge x_2 = y_2) \Rightarrow x_1 \cdot x_2 = y_1 \cdot y_2$$

$$(8.3) (x = y \wedge i = j) \Rightarrow \text{beg}(x, i) = \text{beg}(y, j)$$

$$(8.3) (x = y \wedge i = j) \Rightarrow \text{end}(x, i) = \text{end}(y, j)$$

Тут, вроде, надо было бы написать ещё аксиому для функции следования (i'). Но для неё всё нужное уже доказано в теории Пеано – как и для других арифметических операций (сравнения чисел определяются в арифметике, поэтому тоже соответствуют). А теория Пеано тоже является частью нашей теории и будет явно включена в теорию строк в следующем разделе.

$$(8.4) (i = j) \Rightarrow \text{Chr}(i) = \text{Chr}(j)$$

Заметим, что для тех функций, которые были определены – мы аксиом равенства не пишем, так как соответствующие утверждения выводятся из других аксиом, которые не являются определениями.

б. Для знака равенства должна быть истинна рефлексивность $x = x$. Поэтому:

$$(8.5) x = x$$

В теории Пеано данное равенство не аксиоматизируется, так как оно следует для натуральных чисел из аксиомы $x + 0 = x$. Но в теории строк я это утверждение аксиоматизирую.

с. Для знака равенства должна быть истинна транзитивность $x = y \Rightarrow (z = x \Rightarrow z = y)$. Поэтому:

$$(8.6) x = y \Rightarrow (z = x \Rightarrow z = y)$$

Возможно, что некоторые (или даже все) из аксиом равенства (8.2)-(8.6) можно вывести из предыдущих аксиом. Но это уже второстепенная задача, а у любителя математики (я про себя) нет возможности заниматься ещё и этим исследованием. К тому же задача по аккуратному построению теории строк является задачей для математического сообщества, и неправильно взваливать её на одного человека – даже если бы речь шла о профессионале.

Кстати, в учебнике Э. Мендельсона «Введение в математическую логику» доказательство того, что теория Пеано – теория первого порядка с равенством – занимает несколько страниц в Главе 3. Формальная арифметика. Раздел 1. Система аксиом. От леммы 3.1. до следствия 3.3 включительно. Боюсь, что в теории строк аналогичные доказательства заняли бы значительно больше места – а придумать их намного труднее, чем прочитать.

В учебнике А. С. Герасимова «Курс математической логики и теории вычислимости» нужные для равенства утверждения не доказываются, а аксиоматизируются для теории Пеано. Что никаких противоречий не создаёт и тоже вполне возможно. Занятно, что в отличном учебнике Дж. Булос, Р Джеффри – «Вычислимость и логика» вопрос о равенстве для теории Пеано оказывает-

ся упущенным из вида, и среди аксиом теории Пеано нет необходимой аксиомы о транзитивности равенства.

Можно сказать, что вопрос о равенстве для теории строк в данной статье не упущен, но до идеала не «отполирован», чего едва ли можно требовать от статьи.

Ассоциативность конкатенации:

$$(т.8.1) \quad (a \cdot b) \cdot c = a \cdot (b \cdot c)$$

Очевидная теорема, следует из аксиом (7) и (8.1). Если бы вместо аксиом (7) у нас была аксиома для слияния упорядоченных массивов символов в упорядоченный массив символов, то помимо ассоциативности была доказана ещё и коммутативность. Но конкатенация строк не обладает таким свойством, разумеется.

(т.8.2) $\text{len}(a) \neq \ominus \Rightarrow a = \text{str}(a, 1, \text{len}(a))$, в силу чего в отношении конечной строки a , в виде $\text{str}(a, 1, \text{len}(a))$, можно применять теоремы (т.4.4) и от (т.4.6) до (т.4.9) включительно.

Доказательство.

Исходим из предположения $\text{len}(a) \neq \ominus$.

Будем использовать (т.4.6) $\text{str}(a, 1, j) = \text{beg}(a, j)$.

При $\text{len}(a) = 0$ теорема истинна, так как тогда $a = \ominus$ и $\text{str}(a, 1, \text{len}(a)) = \text{str}(a, 1, 0) = \ominus$, поэтому:

$$a = \text{str}(a, 1, \text{len}(a))$$

Поэтому нам осталось рассмотреть случай $\text{len}(a) = 0 \wedge \text{len}(a) \neq \ominus$.

Из (т.4.2) $(i \neq 0 \wedge j \neq 0) \Rightarrow \text{str}(a, i, j) = \text{end}(\text{beg}(a, i + j - 1), i)$ при $j = 1$ получим:

$$\text{str}(\text{beg}(a, \text{len}(a)), i, 1) = \text{end}(\text{beg}(\text{beg}(a, \text{len}(a)), i), i)$$

При $i \leq \text{len}(a)$, из (т.3.1) $\text{beg}(a, \min(i, j)) = \text{beg}(\text{beg}(a, i), j)$ получим:

$$\text{end}(\text{beg}(\text{beg}(a, \text{len}(a)), i), i) = \text{end}(\text{beg}(a, i), i)$$

По (4.2) $\text{end}(\text{beg}(a, i), i) = \text{beg}(\text{end}(a, i), 1)$ и определению (4.1) для $\text{str}()$ получим:

$$\text{end}(\text{beg}(a, i), i) = \text{str}(a, i, 1)$$

Таким образом (раз доказано для случая $i \neq 0 \wedge i \leq \text{len}(a)$):

$$0 < i \leq \text{len}(a) \Rightarrow \text{str}(\text{beg}(a, \text{len}(a)), i, 1) = \text{str}(a, i, 1)$$

При $\text{len}(a) < i$ из (т.4.4) $j < k \Rightarrow \text{str}(\text{str}(a, i, j), k, 1) = \ominus$ получим:

$$\text{str}(\text{str}(a, 1, \text{len}(a)), i, 1) = \ominus$$

Так как, (т.4.6) $\text{str}(a, 1, \text{len}(a)) = \text{beg}(a, \text{len}(a))$ то доказано:

$$\text{len}(a) < i \Rightarrow \text{str}(\text{beg}(a, \text{len}(a)), i, 1) = \ominus$$

С другой стороны, (т.6.3) $\text{len}(a) \neq \ominus \Rightarrow (\text{len}(a) < i \Rightarrow \text{str}(a, 1, i) = \ominus)$, поэтому при данных условиях

$$\text{len}(a) < i \Rightarrow \text{str}(a, 1, i) = \ominus$$

Из последних 2 равенств получаем:

$$\text{len}(a) < i \Rightarrow \text{str}(\text{beg}(a, \text{len}(a)), i, 1) = \text{str}(a, 1, i)$$

Таким образом, у нас при сделанном предположении $\text{len}(a) \neq \ominus$ доказаны 2 импликации:

$$0 < i \leq \text{len}(a) \Rightarrow \text{str}(\text{beg}(a, \text{len}(a)), i, 1) = \text{str}(a, i, 1)$$

$$\text{len}(a) < i \Rightarrow \text{str}(\text{beg}(a, \text{len}(a)), i, 1) = \text{str}(a, 1, i)$$

Добавляем очевидную третью импликацию, где в заключении стоит равенство, в которой обе стороны равны пустой строке \ominus :

$$i = 0 \Rightarrow \text{str}(\text{beg}(a, \text{len}(a)), i, 1) = \text{str}(a, i, 1)$$

Объединяя их в одну импликацию по аксиоме логики III.3 (см. Раздел 4. Приложение) получим:
 $(0 < i \leq \text{len}(a) \vee \text{len}(a) < i \vee i = 0) \Rightarrow \text{str}(\text{beg}(a, \text{len}(a)), i, 1) = \text{str}(a, i, 1)$

В посылке стоит истинное утверждение, поэтому отбрасываем его по правилу вывода *MP*.

Получаем утверждение:

$$\text{str}(\text{beg}(a, \text{len}(a)), i, 1) = \text{str}(a, i, 1)$$

По аксиоме равенства строк (8.1) получаем:

$$a = \text{beg}(a, \text{len}(a))$$

То же самое иными обозначениями:

$$a = \text{str}(a, 1, \text{len}(a))$$

оно истинно при истинности предположения нашего доказательства:

$$\text{len}(a) \neq \ominus$$

Поэтому по теореме дедукции получаем:

$$\text{len}(a) \neq \ominus \Rightarrow \text{str}(a, 1, \text{len}(a))$$

Что и требовалось доказать.

$$(c.8.2) \text{len}(a) \neq \ominus \Rightarrow (\text{len}(a) \leq i \Rightarrow a = \text{str}(a, 1, i))$$

Очевидное следствие теорем (т.8.2) и (т.4.7) $j \leq k \Rightarrow \text{str}(\text{str}(a, 1, j), 1, k) = \text{str}(a, 1, j)$

$$(т.8.3) a = a \cdot \ominus = \ominus \cdot a$$

Очевидные равенства – из свойств пустой строки \ominus , аксиом для конкатенации и для равенства строк.

$$(т.8.4) a = b \Leftrightarrow \forall i(0 < i \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))$$

Очевидное следствие из аксиомы для равенства строк (8.1), так как для случая $i = 0$ равенство функций:

$$\text{str}(a, 0, 1) = \text{str}(b, 0, 1)$$

выполнено по (4.1) – определению $\text{str}()$, по (2.2) $\text{end}(a, 0) = \ominus$ и (2.4) $\text{beg}(\ominus, q) = \ominus$.

$$(т.8.5) \text{len}(a) \neq \ominus \Rightarrow [a = b \Leftrightarrow \forall i((\text{len}(a) = \text{len}(b) \wedge 0 < i \wedge i \leq \text{len}(a)) \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))]$$

Тоже очевидное следствие – уже из (т.8.4) для равенства конечных строк. Если их длины не равны, то для более короткой (пусть b) будет $\text{str}(b, \text{len}(b)', 1) = \ominus$ при $\text{str}(a, \text{len}(b)', 1) \neq \ominus$, то есть – строки не равны тогда. Если же длины равны, а внутри своих длин строки тоже равны, то за пределами своей длины (одинаковой для них) $\text{str}()$ для каждой такой позиции в этих строках возвращает пустую строку и они равны там тоже:

$\text{len}(a) < i \Rightarrow \text{str}(a, i, 1) = \ominus$, что верно и для строки b из (т.6.3) которая (напоминание):

$$\text{len}(a) \neq \ominus \Rightarrow (\text{len}(a) < i \Rightarrow \text{str}(a, 1, i) = \ominus).$$

$$(т.8.6) \text{len}(a) \neq \ominus \Rightarrow a = \text{beg}(a \cdot u, \text{len}(a))$$

Доказательство. Предположение дедукции - $\text{len}(a) \neq \ominus$.

Сравним $\text{beg}(a \cdot u, \text{len}(a))$ и $\text{beg}(a, \text{len}(a))$. Про последнее мы знаем, что

$$\text{beg}(a, \text{len}(a)) = a, \text{ из-за (т.8.2) } \text{len}(a) \neq \ominus \Rightarrow a = \text{str}(a, 1, \text{len}(a)) \text{ и (т.4.6) } \text{str}(a, 1, j) = \text{beg}(a, j).$$

Случай 1. $i \leq \text{len}(a)$

Из-за теоремы (т.6.4) $j \leq i \Rightarrow \text{str}(\text{beg}(a, i), j, 1) = \text{str}(a, j, 1)$ получаем:

$$\text{str}(\text{beg}(a \cdot u, \text{len}(a)), i, 1) = \text{str}(a \cdot u, i, 1)$$

По аксиоме (7.1) которая (напоминание):

$$\text{len}(a) \neq \ominus$$

$$\Rightarrow (i \leq \text{len}(a) \Rightarrow \text{str}(a \cdot b, i, 1) = \text{str}(a, i, 1)) \wedge (i > \text{len}(a) \Rightarrow \text{str}(a \cdot b, i, 1) = \text{str}(b, i - \text{len}(a), 1))$$

Получаем с учётом $i \leq \text{len}(a)$:

$$\text{str}(a \cdot u, i, 1) = \text{str}(a, i, 1)$$

$$\text{str}(\text{beg}(a \cdot u, \text{len}(a)), i, 1) = \text{str}(a, i, 1)$$

Устраняя условие Случая 1, получим:

$$i \leq \text{len}(a) \Rightarrow \text{str}(a, i, 1) = \text{str}(\text{beg}(a \cdot u, \text{len}(a)), i, 1)$$

Случай 2. $\text{len}(a) < i$

$$\text{str}(\text{beg}(a \cdot u, \text{len}(a)), i, 1) = \text{str}(\text{str}(a \cdot u, 1, \text{len}(a)), i, 1)$$

Из теоремы (т.4.4) $j < k \Rightarrow \text{str}(\text{str}(a, i, j), k, 1) = \ominus$ получаем:

$$\text{str}(\text{str}(a \cdot u, 1, \text{len}(a)), i, 1) = \ominus$$

Аналогично:

$$\text{str}(a, i, 1) = \text{str}((\text{str}(a, 1, \text{len}(a))), i, 1) = \ominus$$

Из двух последних предложений:

$$\text{str}(a, i, 1) = \text{str}(\text{beg}(a \cdot u, \text{len}(a)), i, 1)$$

Устраняя условие Случая 2, получим:

$$\text{len}(a) < i \Rightarrow \text{str}(a, i, 1) = \text{str}(\text{beg}(a \cdot u, \text{len}(a)), i, 1)$$

Объединяя Случай 1 и Случай 2 по III.3 из раздела 4. Приложение, получим:

$$(i \leq \text{len}(a) \vee \text{len}(a) < i) \Rightarrow \text{str}(a, i, 1) = \text{str}(\text{beg}(a \cdot u, \text{len}(a)), i, 1)$$

Посылка истинная, поэтому отбрасываем её по МР и получаем:

$$\text{str}(a, i, 1) = \text{str}(\text{beg}(a \cdot u, \text{len}(a)), i, 1)$$

В соответствии с аксиомой (8.1) равенства строк получаем:

$$a = \text{beg}(a \cdot u, \text{len}(a))$$

Устраняя предположение дедукции, завершаем доказательство:

$$\text{len}(a) \neq \ominus \Rightarrow a = \text{beg}(a \cdot u, \text{len}(a)), \text{ что и требовалось доказать.}$$

$$\text{(т.8.7) } \text{len}(a) \neq \ominus \Rightarrow u = \text{end}(a \cdot u, \text{len}(a)')$$

Доказательство. Предположение дедукции - $\text{len}(a) \neq \ominus$

Если $\text{len}(a) = 0$, то $a \cdot u = \ominus \cdot u = u$. Откуда $\text{end}(a \cdot u, \text{len}(a)') = \text{end}(u, 1) = u$.

Если $\text{len}(u) = 0$, то $a \cdot u = a \cdot \ominus = a$. Откуда $\text{end}(a \cdot u, \text{len}(a)') = \text{end}(a, \text{len}(a)') = \ominus = u$.

Поэтому для $\text{len}(a) = 0 \vee \text{len}(u) = 0$ теорема истинна и исходим из «внутреннего предположения дедукции №1» ($\text{len}(a) \neq 0 \wedge \text{len}(u) \neq 0$), будем пропускать эту посылку при использовании аксиом (она выполнена).

$$\text{str}(\text{end}(a \cdot u, \text{len}(a)'), i, 1) = \text{beg}(\text{end}(\text{end}(a \cdot u, \text{len}(a)'), i), 1), \text{ по (4.1) – определению str()}$$

$$\text{beg}(\text{end}(\text{end}(a \cdot u, \text{len}(a)'), i), 1) = \text{beg}(\text{end}(a \cdot u, \text{len}(a) + i), 1),$$

Из (т.3.2) $(i \neq 0 \wedge j \neq 0) \Rightarrow \text{end}(a, i + j - 1) = \text{end}(\text{end}(a, i), j)$, но только при $i \neq 0$ или эквивалентном $0 < i$ («внутреннее предположение дедукции №2»):

$\text{beg}(\text{end}(a \cdot u, \text{len}(a) + i), 1) = \text{str}(a \cdot u, \text{len}(a) + i, 1)$, по (4.1) – определению $\text{str}()$

$\text{str}(a \cdot u, \text{len}(a) + i, 1) = \text{str}(u, \text{len}(a) + i - \text{len}(a), 1)$, по (7.1) о конкатенации с конечной строкой и того, что $i > 0$.

Поэтому по всей цепочке предыдущих равенств приравниваем начало с концом:

$\text{str}(\text{end}(a \cdot u, \text{len}(a)'), i, 1) = \text{str}(u, i, 1)$

Устраняем «внутреннее предположение дедукции №2»:

$0 < i \Rightarrow \text{str}(\text{end}(a \cdot u, \text{len}(a)'), i, 1) = \text{str}(u, i, 1)$

Обобщаем данную формулу по i :

$\forall i(0 < i \Rightarrow \text{str}(\text{end}(a \cdot u, \text{len}(a)'), i, 1) = \text{str}(u, i, 1))$

И из (т.8.4) $a = b \Leftrightarrow \forall i(0 < i \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))$ получаем:

$u = \text{end}(a \cdot u, \text{len}(a)'),$

Устраняем «внутреннее предположение дедукции №1»:

$(\text{len}(a) \neq 0 \wedge \text{len}(u) \neq 0) \Rightarrow u = \text{end}(a \cdot u, \text{len}(a)'),$

Поскольку ранее было доказано:

$(\text{len}(a) = 0 \vee \text{len}(u) = 0) \Rightarrow u = \text{end}(a \cdot u, \text{len}(a)'),$

То соединение этих предложений даёт:

$u = \text{end}(a \cdot u, \text{len}(a)'),$

Теперь устраняем предположение дедукции и получаем:

$\text{len}(a) \neq \ominus \Rightarrow u = \text{end}(a \cdot u, \text{len}(a)'),$ что и требовалось доказать.

(т.8.8) $a = \ominus \Leftrightarrow \text{len}(a) = 0$

В части $a = \ominus \Rightarrow \text{len}(a) = 0$ эта эквивалентность выводится из определения для $\text{len}(\ominus)$, на основании дедуктивного предположения $a = \ominus$ и результата расчёта $\text{len}(a) = 0$. Но, так как в данном подразделе мы разбираем равенство для теории строк, то упомяну, на чём основывается возможность логического вывода $a = \ominus \Rightarrow \text{len}(a) = 0$ для равенства на уровне свойств равенства.

В рамках свойств равенства (если обходиться без теоремы дедукции) выводится:

$a = \ominus \Rightarrow \text{len}(a) = \text{len}(\ominus)$

А затем надо воспользоваться выводимым свойством транзитивности для равенства:

$\text{len}(\ominus) = 0 \Rightarrow (\text{len}(a) = \text{len}(\ominus) \Rightarrow \text{len}(a) = 0)$

Посылка этой импликации истинная и она отбрасывается, после чего остаётся:

$\text{len}(a) = \text{len}(\ominus) \Rightarrow \text{len}(a) = 0$

И из этого утверждения и ранее полученного $a = \ominus \Rightarrow \text{len}(a) = \text{len}(\ominus)$ получаем по правилу силлогизма:

$a = \ominus \Rightarrow \text{len}(a) = 0$

Все подробности вывода свойств для равенства можно прочитать в любом учебнике логики, но они и так самоочевидны для читателя с математической подготовкой уровня программиста, например.

Осталось доказать эквивалентность в части:

$$\text{len}(a) = 0 \Rightarrow a = \ominus$$

Гипотеза теоремы дедукции: $\text{len}(a) = 0$

Из определения (6.2) для $\text{len}(a)$ при $\text{len}(a) = 0$ устрояем 2 ложных члена дизъюнкции, превратив их в истинные посылки. Вот по такой схеме:

$A \vee B \vee C$ – исходное определение. Преобразуем его в эквивалентное:

$\neg B \Rightarrow (A \vee C)$, но B ложное, так как для него надо $\text{len}(a) = i$, а по условию $\text{len}(a) = 0$. Поэтому $\neg B$ истинное и его можно убрать по МР. Аналогично с C , где надо $\text{len}(a) = \ominus$, а имеем $\text{len}(a) = 0$ и по (2.6) $i \neq \ominus$. После удаления по МР члена C остался член дизъюнкции A . В нашем случае это:

$$\text{len}(a) = 0 \wedge \text{str}(a, 1, 1) = \ominus$$

В истинной конъюнкции истинным является каждый член конъюнкции, поэтому получаем:

$$\text{str}(a, 1, 1) = \ominus$$

А это значит, что

$$\text{str}(a, i, 1) = \ominus, \text{ из-за (т.6.1) } i \leq j \Rightarrow (\text{str}(a, i, 1) = \ominus \Rightarrow \text{str}(a, j, 1) = \ominus)$$

что совпадает с: $\text{str}(\ominus, i, 1) = \ominus$

И по аксиоме равенства строк получаем: $a = \ominus$

Поэтому по теореме дедукции выведено: $\text{len}(a) = 0 \Rightarrow a = \ominus$

Вторая часть эквивалентности доказана, а вместе с ней доказана и эквивалентность (т.8.8).

$$\text{(с.8.8) } \text{len}(a) = \ominus \Leftrightarrow \text{len}(\text{len}(a)) = 0$$

Очевидное следствие из теоремы (т.8.8), которое получается при подстановке в (т.8.8) выражения $\text{len}(a)$ вместо a .

V. Сравнение, поиск, вставка

Лемма о минимальной позиции отличия в строках:

(л.9.1) $\text{str}(a, n, 1) \neq \text{str}(b, n, 1) \Rightarrow \exists_1 m (\text{str}(a, m, 1) \neq \text{str}(b, m, 1) \wedge (\forall i < m : \text{str}(a, i, 1) = \text{str}(b, i, 1)))$

Пояснение: Очевидная лемма о том, что если строки не равны, то есть такой номер m позиции в этих строках, что именно на нем встречается самое первое отличие в строках, а до него (если есть позиции раньше) у строк одинаковые элементы алфавита на соответствующих местах.

Существование m доказывается по индукции для n' . Рассматриваются 2 варианта: при первом варианте имеется i :

$i \leq n$, такой что $\text{str}(a, i, 1) \neq \text{str}(b, i, 1)$,

а при втором варианте для каждого i :

$i \leq n$ означает $\text{str}(a, i, 1) = \text{str}(b, i, 1)$.

При первом варианте для n' теорема верна в силу предположения индукции, а во втором – сама позиция n' является искомым решением. Это завершает доказательство по индукции для существования.

Единственность m тоже очевидна: если бы были два отличных m_1 и m_2 , то для меньшего – пусть m_1 – было бы верно:

$\text{str}(a, m_1, 1) \neq \text{str}(b, m_1, 1)$

А для второго было бы верно:

$i < m_2 \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1)$, в частности, для $i = m_1$ получается (так как $m_1 < m_2$):

$\text{str}(a, m_1, 1) = \text{str}(b, m_1, 1)$

Что противоречит написанному чуть выше неравенству. Поэтому единственность тоже доказана.

Лемма об эквивалентном добавлении дизъюнктивных членов:

(л.9.2) $A \Rightarrow ((A \Rightarrow \neg C) \Rightarrow (B \Leftrightarrow (B \vee C)))$

Это тавтология, что легко выявить прямой проверкой, но приведу схему вывода ради демонстрации некоторых технических приёмов.

Для доказательства леммы (л.9.2) достаточно доказать, что:

$\neg C \Rightarrow (B \Leftrightarrow (B \vee C))$, так как лемма (л.9.2) следует по правилу силлогизма из данного предложения и тавтологии:

$A \Rightarrow ((A \Rightarrow \neg C) \Rightarrow \neg C)$.

Поэтому сводим эквивалентность к эквивалентной эквивалентности отрицаний:

$\neg C \Rightarrow (\neg B \Leftrightarrow (\neg B \wedge \neg C))$

А последнее предложение очевидно – при дедуктивном предположении $\neg C$ из очевидных тавтологий:

$\neg B \Rightarrow \neg B$

$\neg C \Rightarrow (\neg B \Rightarrow \neg B)$

$\neg C \Rightarrow (\neg B \Rightarrow \neg C)$

$\neg C \Rightarrow (\neg B \Rightarrow \neg B \wedge \neg C)$

$\neg B \wedge \neg C \Rightarrow \neg B$

$$\neg C \Rightarrow (\neg B \wedge \neg C \Rightarrow \neg B)$$

$$\neg C \Rightarrow (\neg B \Leftrightarrow (\neg B \wedge \neg C))$$

Теорема доказана.

(9.1) Определение для $\text{CompIn}(a, b)$:

$$(\text{str}(a, \text{CompIn}(a, b), 1) \neq \text{str}(b, \text{CompIn}(a, b), 1) \wedge \forall i (i < \text{CompIn}(a, b) \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))) \vee (\text{CompIn}(a, b) = 0 \wedge a = b)$$

Пояснение. Эта функция возвращает первое место, на котором в строке a и строке b находятся разные элементы алфавита. А если такого места нет, то $\text{CompIn}(a, b)$ возвращает 0.

Для доказательства того, что (9.1) является определением, будем рассматривать следующую формулу, которую обозначим как $C(m, a, b)$:

$$(\text{str}(a, m, 1) \neq \text{str}(b, m, 1) \wedge (\forall i < m : \text{str}(a, i, 1) = \text{str}(b, i, 1))) \vee (m = 0 \wedge a = b)$$

И надо доказать:

$$\exists_1 m C(m, a, b)$$

Ветка 1.

Докажем данную формулу при условии

$$\text{str}(a, n, 1) \neq \text{str}(b, n, 1)$$

Тогда у нас уже доказано по (л.9.1):

$$\exists_1 m (\text{str}(a, m, 1) \neq \text{str}(b, m, 1) \wedge (\forall i < m : \text{str}(a, i, 1) = \text{str}(b, i, 1)))$$

Верной является следующее предложение:

$$1.1. \text{str}(a, n, 1) \neq \text{str}(b, n, 1) \Rightarrow \neg(m = 0 \wedge a = b)$$

Это следует из

$$a = b \Rightarrow (\forall i \text{str}(a, i, 1) = \text{str}(b, i, 1))$$

$$(\forall i \text{str}(a, i, 1) = \text{str}(b, i, 1)) \Rightarrow \text{str}(a, n, 1) = \text{str}(b, n, 1)$$

И, применяя контрапозицию к краям этой короткой цепочки импликаций, получим:

$$\text{str}(a, n, 1) \neq \text{str}(b, n, 1) \Rightarrow \neg(a = b)$$

К заключению импликации мы можем добавлять любой дизъюнктивный член по аксиоме III.1 раздела 4. Приложение. И по правилу силлогизма получим:

$$\text{str}(a, n, 1) \neq \text{str}(b, n, 1) \Rightarrow (\neg(a = b) \vee \neg(m = 0))$$

Если вынести знак отрицания за скобки в заключении данной импликации, то получим (докажем) нужное нам предложение:

$$1.1. \text{str}(a, n, 1) \neq \text{str}(b, n, 1) \Rightarrow \neg(m = 0 \wedge a = b).$$

Кроме того, в соответствии с леммой (л.9.2) имеем:

$$\text{str}(a, n, 1) \neq \text{str}(b, n, 1) \Rightarrow$$

$$((\text{str}(a, n, 1) \neq \text{str}(b, n, 1) \Rightarrow \neg(m = 0 \wedge a = b)))$$

$$\Rightarrow [(\text{str}(a, m, 1) \neq \text{str}(b, m, 1) \wedge (\forall i < m : \text{str}(a, i, 1) = \text{str}(b, i, 1))) \Leftrightarrow C(m, a, b)]$$

)

Посылки можно переставлять местами, поэтому предложение 1.1 на первое место и убираем её по правилу МР, так как оно истинно. Получим:

$$\text{str}(a, n, 1) \neq \text{str}(b, n, 1)$$

$$\Rightarrow [(\text{str}(a, m, 1) \neq \text{str}(b, m, 1) \wedge (\forall i < m : \text{str}(a, i, 1) = \text{str}(b, i, 1))) \Leftrightarrow C(m, a, b)]$$

В силу леммы (л.9.1) и последней эквивалентности имеем (заменяя в лемме (л.9.1) формулу под квантором \exists_1 на эквивалентную формулу):

$$\text{str}(a, n, 1) \neq \text{str}(b, n, 1) \Rightarrow \exists_1 m C(m, a, b). \text{ Ветка 1 доказана.}$$

Ветка 2.

Докажем $\exists_1 m C(m, a, b)$ при условии

$$a = b$$

Очевидно, что

$$a = b \Rightarrow \exists_1 m (m = 0 \wedge a = b)$$

Теперь аналогично предложению 1.1 в ветке 1 получаем предложение:

$$2.1. a = b \Rightarrow \neg(\text{str}(a, m, 1) \neq \text{str}(b, m, 1) \wedge (\forall i < m : \text{str}(a, i, 1) = \text{str}(b, i, 1)))$$

Действительно:

$$a = b \Rightarrow (\forall i \text{ str}(a, i, 1) = \text{str}(b, i, 1))$$

А далее по цепочке силлогизмов (многоточие в посылке – это заключение предыдущей импликации):

$$\dots \Rightarrow \text{str}(a, m, 1) = \text{str}(b, m, 1)$$

$$\dots \Rightarrow \text{str}(a, m, 1) = \text{str}(b, m, 1) \vee \neg(\forall i < m : \text{str}(a, i, 1) = \text{str}(b, i, 1))$$

$$\dots \Rightarrow \neg(\text{str}(a, m, 1) \neq \text{str}(b, m, 1) \wedge (\forall i < m : \text{str}(a, i, 1) = \text{str}(b, i, 1)))$$

Истинность импликации 2.1 очевидна, так как неравенство элементов алфавита на местах m невозможно для равных строк. И данный вывод не требует правил вывода помимо МР, так как опирается на силлогизм и аксиому (а) $\forall x A(x) \Rightarrow A(a)$ из Раздела 4. Приложение.

Это предложение 2.1 истинно, и мы точно так же сокращаем его после применения леммы (л.9.2), как и в ветке 1. Получим:

$$a = b \Rightarrow [(m = 0 \wedge a = b) \Leftrightarrow C(m, a, b)]$$

Поэтому из последней формулы и написанного выше предложения

$a = b \Rightarrow \exists_1 m (m = 0 \wedge a = b)$, получим (заменяя в данном предложении формулу под квантором \exists_1 на эквивалентную формулу):

$$a = b \Rightarrow \exists_1 m C(m, a, b). \text{ Ветка 2 доказана.}$$

Объединяя ветки 1 и 2, получим:

$$(\text{str}(a, n, 1) \neq \text{str}(b, n, 1) \vee a = b) \Rightarrow \exists_1 m C(m, a, b)$$

По правилу вывода из раздела 4. Приложение:

$$(\beta) \text{ Если верно } B(a) \Rightarrow U, \text{ то верно } \exists x B(x) \Rightarrow U$$

Получим:

$$(\exists n \text{ str}(a, n, 1) \neq \text{str}(b, n, 1) \vee a = b) \Rightarrow \exists_1 m C(m, a, b)$$

Теперь, чтобы доказать, что $\exists_1 m C(m, a, b)$, достаточно доказать посылку последней импликации и отбросить её по правилу МР. Итак, осталось доказать:

$$\exists n (\text{str}(a, n, 1) \neq \text{str}(b, n, 1) \vee a = b)$$

По пункту VI.9 из раздела 4. Приложение:

$$\exists n (\text{str}(a, n, 1) \neq \text{str}(b, n, 1) \vee a = b) \Leftrightarrow (a = b \vee \exists n \text{ str}(a, n, 1) \neq \text{str}(b, n, 1))$$

По (8.1) $a = b \Leftrightarrow \forall i \text{ str}(a, i, 1) = \text{str}(b, i, 1)$ и по VI.7 из раздела 4. Приложение:

$a = b \Leftrightarrow \neg(\exists n \text{ str}(a, n, 1) \neq \text{str}(b, n, 1))$, поэтому:

$\exists n(\text{str}(a, n, 1) \neq \text{str}(b, n, 1) \vee a = b) \Leftrightarrow (a = b \vee \neg(a = b))$

Правая часть эквивалентности доказана из-за тавтологии

$A \vee \neg A$

Таким образом,

$\exists n(\text{str}(a, n, 1) \neq \text{str}(b, n, 1) \vee a = b)$ доказано, а с ним доказано и $\exists_1 m C(m, a, b)$, в силу чего (9.1) является определением.

(с.9.1) $\text{len}(a) \neq \ominus \Rightarrow [0 < \text{CompIn}(a, b) \leq \text{len}(a) \Leftrightarrow \forall u \text{ CompIn}(a \cdot u, b) = \text{CompIn}(a, b)]$, и то же верно, если в этом следствии заменить $\text{CompIn}(a, b)$ и $\text{CompIn}(a \cdot u, b)$ на $\text{CompIn}(b, a)$ и $\text{CompIn}(b, a \cdot u)$ соответственно.

Пояснение: если строка a конечна ($\text{len}(a) \neq \ominus$) и отличие от строки b есть ($0 < \text{CompIn}(a, b)$) и находится в пределах строки a ($\text{CompIn}(a, b) \leq \text{len}(a)$), то это тот и только тот случай, когда для любой строки $a \cdot u$, начало которой равно конечной строке a , имеется первое отличие от строки b в той же позиции, в которой в строке a находится первое отличается от строки b .

Тут мы впервые доказываем возможность игнорировать в строке всё, кроме её ограниченного начала, но при этом получить интересующий нас результат для всей строки при определённых условиях.

Доказательство.

Предполагаем, что $\text{len}(a) \neq \ominus$ и в условиях этой гипотезы дедукции нам для доказательства (с.9.1) достаточно доказать эквивалентность:

$0 < \text{CompIn}(a, b) \leq \text{len}(a) \Leftrightarrow \forall u \text{ CompIn}(a \cdot u, b) = \text{CompIn}(a, b)$

Для доказательства данной эквивалентности надо доказать две импликации:

Ветка 1: $0 < \text{CompIn}(a, b) \leq \text{len}(a) \Rightarrow \forall u \text{ CompIn}(a \cdot u, b) = \text{CompIn}(a, b)$

Ветка 2: $\forall u \text{ CompIn}(a \cdot u, b) = \text{CompIn}(a, b) \Rightarrow 0 < \text{CompIn}(a, b) \leq \text{len}(a)$

Ветка 1.

Определение (9.1) для $\text{CompIn}(a, b)$, когда $\text{CompIn}(a, b) \neq 0$, и $\text{CompIn}(a, b) \leq \text{len}(a)$ (наши дедуктивные гипотезы):

$(\text{str}(a, \text{CompIn}(a, b), 1) \neq \text{str}(b, \text{CompIn}(a, b), 1) \wedge (\forall i i < \text{CompIn}(a, b) \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1)))$

Так как для всех i (под квантором общности) в данном предложении верно $i \leq \text{len}(a)$, то будет верно в силу аксиомы (7.1) о результате конкатенации при конечной 1-й строке:

$\text{str}(a \cdot u, i, 1) = \text{str}(a, i, 1)$

И то же самое можно сказать про

$\text{str}(a \cdot u, \text{CompIn}(a, b), 1) = \text{str}(a, \text{CompIn}(a, b), 1)$

Поэтому, по аксиомам для равенства можно заменить в первом предложении Ветки 1 все $\text{str}(a, i, 1)$ на $\text{str}(a \cdot u, i, 1)$, а $\text{str}(a, \text{CompIn}(a, b), 1)$ можно заменить на $\text{str}(a \cdot u, \text{CompIn}(a, b), 1)$.

И тогда выяснится, что число $\text{CompIn}(a, b)$ полностью удовлетворяет определению для $\text{CompIn}(a \cdot u, b)$.

Таким образом, доказано:

$$0 < \text{CompIn}(a, b) \leq \text{len}(a) \Rightarrow \text{CompIn}(a \cdot u, b) = \text{CompIn}(a, b)$$

И по схеме вывода из Раздела 4. Приложение

(α) Если верно $U \Rightarrow B(a)$, то верно $U \Rightarrow \forall x B(x)$

Получим:

$$0 < \text{CompIn}(a, b) \leq \text{len}(a) \Rightarrow \forall u \text{CompIn}(a \cdot u, b) = \text{CompIn}(a, b), \text{ ветка 1 доказана.}$$

Ветка 2.

Возьмём контрапозицию, эквивалентную доказываемой импликации:

$$[\text{CompIn}(a, b) = 0 \vee \text{len}(a) < \text{CompIn}(a, b)] \Rightarrow \neg \forall u \text{CompIn}(a \cdot u, b) = \text{CompIn}(a, b)$$

Правую часть перепишем в соответствии с п. VI.7 Раздела 4. Приложение. И разобьём доказательство полученной импликации на доказательство 2-х импликаций:

Ветка 2.1: $\text{CompIn}(a, b) = 0 \Rightarrow \exists u \text{CompIn}(a \cdot u, b) \neq \text{CompIn}(a, b)$

Ветка 2.2: $\text{len}(a) < \text{CompIn}(a, b) \Rightarrow \exists u \text{CompIn}(a \cdot u, b) \neq \text{CompIn}(a, b)$

После того, как Ветки 2.1 и 2.2 будут доказаны, из них выводится исходная Ветку 2 в силу тавтологии:

$$(A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow ((A \vee B) \Rightarrow C)), \text{ п.III.3 Раздела 4. Приложение.}$$

Ветка 2.1. $\text{CompIn}(a, b) = 0$ – предположение дедукции.

Так как $\text{CompIn}(a, b) = 0$, то из определения (9.1) получаем:

$$a = b$$

Положим:

$$u = \text{Chr}(0)$$

Очевидно (легко выводится из $a = b$ и $\text{len}(a) \neq \ominus$):

$$\text{CompIn}(a \cdot \text{Chr}(0), b) \neq \text{CompIn}(a, b)$$

Так как левая часть неравенства ($\text{CompIn}(a \cdot \text{Chr}(0), b)$) имеет значение $\text{len}(a)'$, а правая ($\text{CompIn}(a, b)$) – имеет значение 0 (ноль).

Из аксиомы (b) Раздела 4. Приложение:

$$\text{CompIn}(a \cdot u, b) \neq \text{CompIn}(a, b) \Rightarrow \exists u \text{CompIn}(a \cdot u, b) \neq \text{CompIn}(a, b)$$

А, значит, верно, при подстановке в свободную (не под квантором) переменную u :

$$\text{CompIn}(a \cdot \text{Chr}(0), b) \neq \text{CompIn}(a, b) \Rightarrow \exists u \text{CompIn}(a \cdot u, b) \neq \text{CompIn}(a, b)$$

Так как посылка импликации истинна, то получаем:

$\exists u \text{CompIn}(a \cdot u, b) \neq \text{CompIn}(a, b)$, вспоминая предположение дедукции ветки 2.1 – получаем её доказательство.

Ветка 2.2. $\text{len}(a) < \text{CompIn}(a, b)$ – предположение дедукции.

Так как $\text{len}(a) < \text{CompIn}(a, b)$, то из определения (9.1) получаем:

$$\forall i \leq \text{len}(a) : \text{str}(a, i, 1) = \text{str}(b, i, 1)$$

Поэтому, из (т.8.5) получаем

$$\text{beg}(b, \text{len}(a)) = a$$

Положим:

$$u = \text{end}(b, \text{len}(a)')$$

Тогда по (1) получаем:

$$b = a \cdot \text{end}(b, \text{len}(a)')$$

Поэтому:

$$\text{CompIn}(a \cdot \text{end}(b, \text{len}(a)'), b) \neq \text{CompIn}(a, b)$$

Последнее неравенство истинное, потому что слева – 0 (ноль), а справа $\text{CompIn}(a, b) > 0$.

Дальше вывод аналогичен ветке 2.1, поэтому ветка 2.2. доказана.

Следствие (с.9.1) доказано.

(9.2) Определение для $\text{Comp}(a, b)$:

$$\text{Comp}(a, b) = \text{Comp}(\text{str}(a, \text{CompIn}(a, b), 1), \text{str}(b, \text{CompIn}(a, b), 1))$$

Очевидно, что это определение.

1. Если строки не равны, то результат $\text{Comp}(a, b)$ совпадает с результатом сравнения элементов алфавита на том месте (в обоих строках), где это отличие возникло впервые от 1 места и до данного, Вопрос сводится к сравнению элементов алфавита в силу аксиомы (5.1) $\forall a \forall i \exists j \text{str}(a, i, 1) = \text{Chr}(j)$. А сравнение для элементов алфавита было определено в аксиоме (5.4).

2. Если строки равны, то результат сравнения будет равен нулю, потому что

$$\text{CompIn}(a, b) = 0 \text{ для равных строк, в силу чего}$$

$$\text{Comp}(\text{str}(a, 0, 1), \text{str}(b, 0, 1)) = \text{Comp}(\ominus, \ominus) = 0$$

В последней строке вывода были использованы определение (4.1) $\text{str}(a, i, j) = \text{beg}(\text{end}(a, i), j)$ и аксиома (2.2) $\text{end}(a, 0) = \ominus$, и ещё аксиома (2.4) $\text{beg}(\ominus, i) = \ominus$.

Критерий локального сравнения данных:

(с.9.2) $\text{len}(a) \neq \ominus \Rightarrow [0 < \text{CompIn}(a, b) \leq \text{len}(a) \Leftrightarrow \forall u \text{Comp}(a \cdot u, b) = \text{Comp}(a, b)]$, и то же верно, если в этом следствии заменить $\text{CompIn}(a, b)$, $\text{Comp}(a \cdot u, b)$ и $\text{Comp}(a, b)$ на $\text{CompIn}(b, a)$, $\text{Comp}(b, a \cdot u)$ и $\text{Comp}(b, a)$ соответственно.

Заметим, что в левой и правой частях эквивалентностей используются разные функции. В левой - $\text{CompIn}(a, b)$, как в следствии (с.9.1), а в правой - $\text{Comp}(a, b)$, которая определена в последней аксиоме (9.2).

Доказательство.

По аналогии с доказательством (с.9.1) рассмотрим две импликации при гипотезе дедукции $\text{len}(a) \neq \ominus$.

Ветка 1.

Из доказательства Ветки 1 в (с.9.1) знаем, что когда $\text{CompIn}(a, b) \neq 0$, и $\text{CompIn}(a, b) \leq \text{len}(a)$, то

$$\text{CompIn}(a \cdot u, b) = \text{CompIn}(a, b)$$

И из доказательства (с.9.1) для всех $i < \text{len}(a)$ верно

$$\text{str}(a \cdot u, i, 1) = \text{str}(a, i, 1)$$

Поэтому из определения (9.2) – заменяя сразу $\text{CompIn}(a \cdot u, b)$ и $\text{str}(a \cdot u, \dots, 1)$ на равные им $\text{CompIn}(a, b)$ и $\text{str}(a, \dots, 1)$:

$$\text{Comp}(a \cdot u, b) = \text{Comp}(\text{str}(a, \text{CompIn}(a, b), 1), \text{str}(b, \text{CompIn}(a, b), 1)) = \text{Comp}(a, b)$$

Поэтому Ветка 1 доказана.

Ветка 2.1: $\text{CompIn}(a, b) = 0 \Rightarrow \exists u \text{Comp}(a \cdot u, b) \neq \text{Comp}(a, b)$

Как и в доказательстве Ветки 2.1 следствия (с.9.1) рассматриваем

$u = \text{Chr}(0)$ и выводим заключение импликации из

$$\begin{aligned} \text{Comp}(a \cdot \text{Chr}(0), b) &= \text{Comp}(\text{str}(a \cdot \text{Chr}(0), \text{CompIn}(a \cdot \text{Chr}(0), b), 1), \text{str}(b, \text{CompIn}(a \cdot \text{Chr}(0), b), 1)) \\ &= \text{Comp}(\text{str}(a \cdot \text{Chr}(0), \text{len}(a)', 1), \text{str}(b, \text{len}(a)', 1)) = \text{Comp}(\text{Chr}(0), \ominus) = 2 \end{aligned}$$

Результат 2 отличается от $\text{Comp}(a, b) = 0$.

Завершение доказательства – по аналогии с Веткой 2.1 из доказательства для (с.9.1).

Ветка 2.2: $\text{len}(a) < \text{CompIn}(a, b) \Rightarrow \exists u \text{Comp}(a \cdot u, b) \neq \text{Comp}(a, b)$

В качестве предположения дедукции-2 возьмем не просто $\text{len}(a) < \text{CompIn}(a, b)$, но:

$\text{CompIn}(a, b) \neq 0 \wedge \text{len}(a) < \text{CompIn}(a, b)$, в соответствии с теоремой VI.10 из раздела 4. Приложение.

Как и в Ветке 2.2 доказательства (с.9.1) рассматриваем

$u = \text{end}(b, \text{len}(a)')$ и выводим

$\text{Comp}(a \cdot u, b) = 0$, так как мы добились равенства $a \cdot u = b$.

И этот результат 0 (ноль) отличается от $\text{Comp}(a, b) \neq 0$ из предположения дедукции-2.

Завершение доказательства – по аналогии с Веткой 2.1

Следствие (с.9.2) доказано.

(л.9.3) Лемма о минимальной позиции совпадения в строке:

$$\text{what} \neq \ominus \wedge \text{len}(\text{what}) \neq \ominus \wedge \text{str}(in, n, \text{len}(\text{what})) = \text{what} \Rightarrow$$

$$\exists_1 m(\text{str}(in, m, \text{len}(\text{what})) = \text{what} \wedge (\forall i < m : \text{str}(in, i, \text{len}(\text{what})) \neq \text{what}))$$

Пояснение. Это лемма о том, что если в строке in имеется подстрока (не пустая и не бесконечная), равная строке $what$, то можно найти такую подстроку, равную $what$, которая начиналась бы в строке in раньше (на позиции с самым маленьким номером), чем любые другие такие строки.

Логика данной формулы идентична логике формулы (л.9.1) – меняется только способ сравнения: вместо поиска самого раннего отличия в данной позиции у данной строки ищется самое раннее совпадение начиная с данной позиции у данной строки. Доказательство аналогичное доказательству (л.9.1).

(9.3) Определение для $\text{find}(in, \text{what}, 0)$:

$$(\text{what} \neq \ominus \wedge \text{len}(\text{what}) \neq \ominus \wedge \text{str}(in, \text{find}(in, \text{what}, 0), \text{len}(\text{what})) = \text{what}$$

$$\wedge \forall j(j < \text{find}(in, \text{what}, 0) \Rightarrow \text{str}(in, j, \text{len}(\text{what})) \neq \text{what})$$

$$) \vee (\text{find}(in, \text{what}, 0) = 0 \wedge (\text{what} = \ominus \vee \text{len}(\text{what}) = \ominus \vee \forall j \text{str}(in, j, \text{len}(\text{what})) \neq \text{what}))$$

Пояснение.

Первый член дизъюнкции соответствует случаю, когда найдена позиция в строке in , с которой начинается такая подстрока в строке in , которая равна строке $what$. Притом раньше найденной позиции нет никакой другой позиции с таким же совпадением со строкой $what$, а строка $what$ не является ни пустой, ни бесконечной.

Второй член дизъюнкции – это случай равенства нулю: это случай, когда строка $what$ является либо пустой, либо бесконечной, либо ни на каких местах в строке in не начинается подстрока, равная строке $what$.

В принципе, можно было бы включить в состав находимых и варианты с бесконечной подстрокой, но не вижу в этом необходимости с практической точки зрения. Если же необходимость возникнет, то можно будет тогда определить функцию расширенного поиска бесконечной подстроки.

Сейчас же логика функции $\text{find}(in, what, 0)$ почти совпадает с логикой функции $\text{CompIn}(a, b)$. Даже проще, так как случай $what = \ominus$ сразу попадет в «прочие» для $\text{find}(in, what, 0)$, в отличие от $b = \ominus$ для $\text{CompIn}(a, b)$.

Так как логика формулы (9.3) аналогична логике формулы (9.1), как логика леммы (л.9.3) аналогична логике леммы (л.9.1), то:

Доказательство на основе леммы (л.9.3) того, что аксиома (9.3) является определением – проводится почти идентично тому (с учетом разницы в способе признания результата в данной строке в данной позиции – искомым), как доказывалось на основе леммы (л.9.1) то, что аксиома (9.1) является определением.

Поэтому считаем доказанным то, что аксиома (9.3) является определением для $\text{find}(in, what, 0)$.

(9.4) Определение для $\text{find}(in, what, i)$:

$$\text{find}(in, what, i) = \text{find}(\text{end}(in, i'), what, 0)$$

Очевидно, что аксиома (9.4) является определением для функции $\text{find}(in, what, i)$.

От функции $\text{find}(in, what, 0)$ она отличается тем, что ищет позицию совпадения со строкой $what$ в строке in не с первой позиции строки in , а только с (i') позиции включительно, пропустив первые i символов в строке in .

(9.5) Определение для $if_0(a, b, c)$:

$$(if_0(a, b, c) = b \wedge a = 0) \vee (if_0(a, b, c) = c \wedge a \neq 0)$$

Пояснение. Если первый аргумент в данной функции равен нулю, то функция принимает значение второго аргумента. В противном случае функция принимает значение третьего аргумента.

Очевидно, что данная аксиома является определением, то есть, предложение:

$$\exists_1 u (u = b \wedge a = 0) \vee (u = c \wedge a \neq 0) \text{ легко выводится.}$$

(л.9.4) $a \neq a \cdot \text{Chr}(0) \Leftrightarrow \text{len}(a) \neq \ominus$

Действительно, если верно $a \neq a \cdot \text{Chr}(0)$, то для бесконечной строки это не так, в силу применив к аксиоме о конкатенации с бесконечной строкой (7.2) контрапозиции и подстановки:

$$a \neq a \cdot \text{Chr}(0) \Rightarrow \text{len}(a) \neq \ominus$$

В обратную сторону эта импликация доказывается из (т.7.1) которая (напоминание):

$$\text{len}(a_1) \neq \ominus \wedge \dots \wedge \text{len}(a_n) \neq \ominus \Rightarrow \text{len}(a_1 \dots a_n) = \text{len}(a_1) + \dots + \text{len}(a_n), \text{ так как } \text{len}(a) \neq \ominus:$$

$$\text{len}(a \cdot \text{Chr}(0)) = \text{len}(a) + \text{len}(\text{Chr}(0))$$

Но из (т.7.2) которая (напоминание):

$$i \leq l_{\text{ChrLim}} \Rightarrow \text{len}(\text{Chr}(i)) = 1$$

Получаем:

$$\text{len}(\text{Chr}(0)) = 1, \text{ и находим:}$$

$$\text{len}(a \cdot \text{Chr}(0)) = \text{len}(a)'$$

Из арифметики известно – так как $\text{len}(a)$ число – что:

$$\text{len}(a)' \neq \text{len}(a)$$

откуда

$$\text{len}(a \cdot \text{Chr}(0)) \neq \text{len}(a)$$

А из теоремы (т.8.5)

$$\text{len}(a) \neq \ominus \Rightarrow [a = b \Leftrightarrow \forall i((\text{len}(a) = \text{len}(b) \wedge 0 < i \wedge i \leq \text{len}(a)) \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))]$$

Получаем поэтому (из $\text{len}(a \cdot \text{Chr}(0)) \neq \text{len}(a)$ и из (т.8.5), меняя обе стороны в эквивалентности на свои отрицания):

$$a \neq a \cdot \text{Chr}(0)$$

Значит, эквивалентность

$$a \neq a \cdot \text{Chr}(0) \Leftrightarrow \text{len}(a) \neq \ominus$$

доказана по теореме дедукции и в «обратную» сторону.

Принципиальная необходимость следующих трёх определений будет понятна после трёх подразделов, которые следуют за текущим подразделом.

(9.6) Определение для функции $\text{Ins}(a, i, b)$:

$$\text{Ins}(a, i, b) = \text{beg}(a, \max(i, 1) - 1) \cdot b \cdot \text{end}(a, \text{if}_0(\min(\text{len}(\text{len}(b)), i), 0, \text{len}(b) + i))$$

Данная функция представляет собой композицию аксиоматизированных и определённых ранее или в арифметике функций, поэтому нет нужды в доказательстве того, что (9.6) – определение.

Пояснение. Функция $\text{Ins}(a, i, b)$ «вставляет» строку b внутрь строки a , заменяя своими символами все те символы строки a , номера мест которых в строке a начинаются с i -го (включительно) и заканчиваются $i + \text{len}(b) - 1$ -м. По крайней мере, так происходит, когда значение переменной a имеет достаточно большой размер. То есть, когда в строке a на местах номер с i -го по $i + \text{len}(b) - 1$ находятся символы.

Если же размер $\text{len}(a) < i$, то строка b просто допишется (при помощи конкатенации) к концу строки a .

Если же размер $\text{len}(a) \geq i$, но при этом вся строка b не может «влезть» в исходные пределы строки a (потому что $\text{len}(a) < i + \text{len}(b) - 1$), то от строки a останется только $\text{beg}(a, i - 1)$, а дальше будет дописана строка b .

Но всё сказанное о значении функции $\text{Ins}(a, i, b)$ верно лишь в том случае, если $i \neq 0$. Потому что если $i = 0$, то значением функции $\text{Ins}(a, i, b)$ будет b .

Для чего нам нужна эта функция? Для того, чтобы наша теория была способна выразить такую работу алгоритма, при которой алгоритм изменяет лишь часть данных (то, что у нас в теории будет «в начале» переменной). И чтобы получить результат этой работы, алгоритму не требуется ни чтения «лишней» части данных, ни изменения этих «лишних» данных. И, как мы увидим дальше, такая выразительность не может быть достигнута ни рекурсивными функциями, ни в рамках арифметики.

Более формально говоря, нам необходимо сформулировать такие «корректные» условия для «локальной вставки», при которых будет верно:

$$\forall u \text{Ins}(a \cdot u, i, b) = \text{Ins}(a, i, b) \cdot u$$

При этом «хвост» u не должен «сдвинуться» в строке $\text{Ins}(a, i, b) \cdot u$, если сравнивать его положение с положением в строке $a \cdot u$:

$$\text{len}(a) = \text{len}(\text{Ins}(a, i, b))$$

Теперь мы приступим к реализации всей этой программы.

Определим следующую функцию проверки выполнения комплекса условий для локальной вставки:

(9.7) Определение для функции $\text{IsIns}(q_a, i, q_b)$:

$$(\text{IsIns}(q_a, i, q_b) = 1 \wedge i \neq 0 \wedge q_a \neq \ominus \wedge q_b \neq \ominus \wedge (q_b = 0 \vee q_a \geq i + q_b - 1))$$

$$\vee (\text{IsIns}(q_a, i, q_b) = 0 \wedge (i = 0 \vee q_a = \ominus \vee q_b = \ominus \vee (q_b \neq 0 \wedge q_a < i + q_b - 1)))$$

Очевидно, что это определение, потому что первый член дизъюнкции описывает набор всех «корректных» условий (пока «корректность» не доказана – пишу кавычки), а второй член дизъюнкции описывает случай отрицания первого набора. И для «корректного» набора условий есть единственное значение функции – равное 1. А для любого «некорректного» набора тоже единственный результат функции – равный 0. И эти результаты не равны ($1 \neq 0$).

Но не только «вставку» новой строки b в старую строку a можно сделать так, что при любом продолжении старой строки a строкой u не было никаких изменений в строке u ни в содержании, ни в положении строки u внутри $a \cdot u$ после изменения:

$$\forall u \text{Ins}(a \cdot u, i, b) = \text{Ins}(a, i, b) \cdot u$$

Точно так же и получать значения из строки можно независимо от того, началом в какой строке $a \cdot u$ является строка a :

$$\forall u \text{str}(a \cdot u, i, j) = \text{str}(a, i, j)$$

И для вычисления того, когда данное выражение истинно, мы определим соответствующую функцию.

(9.8) Определение для функции $\text{IsStr}(q_a, i, j)$:

$$(\text{IsStr}(q_a, i, j) = 1 \wedge q_a \neq \ominus \wedge (i = 0 \vee j = 0 \vee q_a \geq i + j - 1))$$

$$\vee (\text{IsStr}(q_a, i, j) = 0 \wedge (q_a = \ominus \vee (i \neq 0 \wedge j \neq 0 \wedge q_a < i + j - 1)))$$

Очевидно, что это определение, по тем же причинам, что были написаны для определения (9.7).

Пояснение. Разница между извлечением подстроки и вставкой в строку состоит в том, что при вставке с нулевого символа – всё заменяется новым значением. А при извлечении с нулевого места в качестве результата получается пустая строка. И такое значение результата не зависит от того, какая строка u продолжает строку a в строке $a \cdot u$. Поэтому данная функция $\text{IsStr}(q_a, i, j)$ возвращает 1 (в отличие от нуля функции $\text{IsIns}(q_a, i, q_b)$ при аналогичных условиях), если подстроку пытаются получить с нулевого места строки a (или строки $a \cdot u$). А в остальном функции $\text{IsIns}(q_a, i, q_b)$ и $\text{IsStr}(q_a, i, j)$ работают схожим образом.

Только функция $\text{IsIns}(q_a, i, q_b)$ делает дополнительную проверку по аргументу q_b – не соответствует ли его значение строке с бесконечной длиной, а в функции $\text{IsStr}(q_a, i, j)$ изначально используется j для извлечения конечной строки длины j (или меньше, если в строке с длиной q_a не хватит символов).

(9.9) Определение для $\text{ss}(i)$ и $\text{ss}_k(i)$:

$\text{len}(\text{ss}(i)) = i \wedge \forall j \leq i : \text{str}(\text{ss}(i), j, 1) = \text{Chr}(0)$. Функция $\text{ss}_k(i)$ определяется так же, но только вместо $\text{Chr}(0)$ используется $\text{Chr}(k)$, где $k \leq l_{\text{ChrLim}}$.

То, что (9.9) является определением – очевидно. Формально существование $\text{ss}(i)$ можно вывести по индукции из $\text{ss}(0) = \ominus$ и $\text{ss}(i') = \text{ss}(i) \cdot \text{Chr}(0)$. А единственность этого существования следует из (т.8.5):

$$\text{len}(a) \neq \ominus \Rightarrow [a = b \Leftrightarrow \forall i ((\text{len}(a) = \text{len}(b) \wedge 0 < i \wedge i \leq \text{len}(a)) \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))]$$

VI. Алгоритмы, не сводимые к (частично) рекурсивным функциям и арифметике

За неимением теории строк, теория первого порядка, в которой «представимы» алгоритмы в современных книгах по теории алгоритмов – это арифметика, а представимость алгоритмов в арифметике – это представимость частично рекурсивных функций (ЧРФ далее в этом подразделе). Но множество ЧРФ не включает в себя все необходимые для исследования в теории алгоритмов алгоритмы.

Дело в том, что примитивно рекурсивные функции (ПРФ далее в этом подразделе) используют только такие базовые функции, и такие операторы подстановки и примитивной рекурсии, которые имеют дело только с целостными значениями переменных. Ничего в этом плане не меняет и оператор минимизации – расширяющий множество примитивно рекурсивных функций до множества ЧРФ.

Напомню базовые функции и способы создания из них всех остальных функций данного множества, взятые из Википедии (на начало дня 25.02.2020):

Начало цитаты

К числу базовых ПРФ относятся функции следующих трёх видов:

1. Нулевая функция 0 — функция без аргументов, всегда возвращающая 0 .
2. Функция следования S одного переменного, сопоставляющая любому натуральному числу непосредственно следующее за ним натуральное число $x + 1$.
3. Функции I_n^m , где $0 < m \leq n$, от n переменных, сопоставляющие любому упорядоченному набору x_1, \dots, x_n натуральных чисел число x_m из этого набора.

Операторы подстановки и примитивной рекурсии определяются следующим образом:

Оператор суперпозиции (иногда — оператор подстановки). Пусть f — функция от m переменных, а g_1, \dots, g_m — упорядоченный набор функций от n переменных каждая. Тогда результатом суперпозиции функций g_k в функцию f называется функция h от n переменных, сопоставляющая любому упорядоченному набору x_1, \dots, x_n натуральных чисел число

$$h(x_1, \dots, x_n) = f(g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n)).$$

Оператор примитивной рекурсии. Пусть f — функция от n переменных, а g — функция от $n + 2$ переменных. Тогда результатом применения оператора примитивной рекурсии к паре функций f и g называется функция h от $n + 1$ переменной вида

$$h(x_1, \dots, x_n, 0) = f(x_1, \dots, x_n),$$

$$h(x_1, \dots, x_n, y + 1) = g(x_1, \dots, x_n, y, h(x_1, \dots, x_n, y)).$$

В данном определении переменную y можно понимать как счётчик итераций, f — как исходную функцию в начале итерационного процесса, выдающего некую последовательность функций n переменных, начинающуюся с f , и g — как оператор, принимающий на вход n переменных x_1, \dots, x_n , номер шага итерации (y), функцию $h(x_1, \dots, x_n, y)$ на данном шаге итерации, и возвращающий функцию на следующем шаге итерации.

...

ЧРФ определяется аналогично ПРФ, только к двум операторам суперпозиции и примитивной рекурсии добавляется ещё третий оператор — минимизации аргумента.

Оператор минимизации аргумента. Пусть f – функция от n натуральных переменных. Тогда результатом применения оператора минимума аргумента к функции f называется функция h от $n - 1$ переменной, задаваемая следующим определением:

$$h(x_1, \dots, x_{n-1}, 0) = \min y, \text{ при условии } f(x_1, \dots, x_{n-1}, y) = 0$$

То есть функция h возвращает минимальное значение последнего аргумента функции f , при котором её значение равно 0.

...

Общерекурсивная функция – частично рекурсивная функция, определённая для всех значений аргументов.

Конец цитаты

Как видим, никакая базовая функция не может работать с какой-то частью своего входного аргумента, не зная его «хвоста». И любая производная от базовых функций функция на базе указанных выше правил – тоже не может оказаться функцией, использующей свой аргумент частично, не «читая» его целиком. Действительно:

Нулевая функция не имеет аргументов, Функции I_n^m использует только готовые аргументы, никак не получая из аргумента часть его значения, функция следования – увеличивает аргумент на 1. Чтобы это сделать – необходимо прочитать аргумент целиком.

Итак, базовые функции работают только, используя (читая, в частности) аргумент целиком.

Теперь разберем методы построения новых функций из базовых функций.

Оператор суперпозиции никак не меняет того, как составляющие его функции работают с аргументами, и сводится к ним в плане работы с аргументами. Оператор примитивной рекурсии – уменьшает аргумент при своем построении на 1. Но для получения значения, которое меньше входного аргумента на 1 надо прочитать входной аргумент целиком.

Таким образом, ПРФ могут работать только с целыми значениями входных аргументов. Потому что они строятся на основе базовых функций, которые работают с целостными значениями входных аргументов. А методы построения ПРФ сохраняют тот же метод работы со значениями входных аргументов – читая и используя эти значения целиком.

Меняет ли что-то в этой ситуации оператор минимизации? В смысле входных аргументов он всего лишь сокращает их число.

Таким образом, ЧРФ работают со значениями входных аргументов только целостно – используя (читая, в частности) все входные данные в каждом используемом ими аргументе. «Используемые» аргументы не следует путать с переданными функции аргументами: функция I_n^m , например, использует только один аргумент, но тоже целиком.

Это значит, что в рамках множества ЧРФ невозможно отличить алгоритм, использующий только первый символ в своём входном аргументе от алгоритма, который работает со всем своим входным аргументом. Поэтому решать вопрос об эффективности (скорости) работы алгоритмов, находясь в рамках ЧРФ невозможно в общем случае:

Ведь разница между размерами одного символа (или любого ограниченного их числа) и возможным размером входного аргумента, не может быть ограничена не то, что каким-либо полиномом

от этого ограниченного числа, но вообще никакой функцией.

Вот в теории строк 1-я же аксиома сообщает о возможности разбиения строки в позиции с произвольным номером i с выделением начальной части $\text{beg}(a, i)$:

$$(1) a = \text{beg}(a, i) \cdot \text{end}(a, i')$$

Мало того, что строка a в теории строк может быть представлена как

$$a = b \cdot \text{Unknown}$$

Но и работать со строкой $a = b \cdot \text{Unknown}$ в теории строк можно – при определенных условиях – так, словно вместо всей строки a есть только её начало b . Эти условия видны в следствии (с.9.2) для работы со строкой при помощи функции $\text{Comp}()$, эти условия представлены функциями $\text{IsIns}(q_a, i, q_b)$ и $\text{IsStr}(q_a, i, j)$, как будет доказано в следующем подразделе.

Но помимо теории строк – ещё и в реальном мире у нас есть возможность «локальной» работы со строкой. Например: я могу прочитать только начало книги и решить, что она мне не годится. И для принятия такого решения мне нет нужды читать книгу до конца, потому что начало книги несёт такую информацию, которая никак не меняется от дальнейшего текста. И время на чтение этого начала я трачу одинаково вне зависимости от того – целая книга у меня в руках, или только вырванный из книги блок её первых страниц.

Но для того, чтобы в теории я мог воспользоваться возможностями «локальной работы» со строками из реального мира, мне нужно иметь в теории соответствующие инструменты. А таких инструментов нет в формализме частично рекурсивных функций.

Проблема ЧРФ при работе с входными данными чрезмерно (сколь угодно «чрезмерно») большого размера, которые не используются алгоритмом в своей работе, кроме некоторой начальной части этих данных, порождает формулировки вроде:

«Подтверждающий сертификат должен иметь размер, ограниченный полиномом от размера проверяемого слова» и «Если среди сертификатов допустимого размера нет сертификата, который является подтверждающим для данного слова, то данное слово не принадлежит языку».

Возникает вопрос: а кто/что проверяет – является ли допустимым размер у проверяемого сертификата? Разве проверка такого рода не является вычислением? И разве время на эту проверку не столь же уместно учитывать в качестве времени работы алгоритма проверки, как и время после того, когда недопустимые по своим размерам сертификаты отброшены?

Разумеется, время на проверку размеров сертификатов учитывать надо. И правильная формулировка состоит в том, что время работы алгоритма проверки не должно зависеть от размера чрезмерного по своим размерам сертификата. Но контроль размера и выбраковку сертификатов с чрезмерно большими размерами алгоритм проверки осуществляет самостоятельно.

И выбраковка чрезмерно больших сертификатов должна осуществляться за время, полиномиальное от критического («граничного») размера сертификата – при чтении первого символа, находящегося вне допустимых – для размера входных данных – пределов. А вовсе не в результате обработки всего «входа».

Но в рамках ЧРФ решить подобную задачу невозможно. Однако одной заменой ЧРФ на что-то более выразительное для теории алгоритмов обойтись не удастся, как мы увидим чуть ниже в

данном подразделе – арифметика тоже не обладает необходимой для теории алгоритмов выразительностью.

Поэтому необходимо создать такую теорию (теорию строк, например) и такую модель исполнения алгоритмов (2-я статья, которая будет после данной), которая устраняла бы неразрешимую в рамках ЧРФ проблему работы с «входными данными», которые используются алгоритмами лишь в незначительной начальной части этих данных.

К данному подразделу мы уже в достаточной степени построили теорию строк для того, чтобы доказать, что арифметика (не только теория Пеано, но и её любые возможные арифметические расширения) недостаточно выразительна для решения вопросов теории алгоритмов в общем случае.

(м.9.1) В теории строк с односимвольным алфавитом ($l_{ChrLim} = 0$) невозможно в общем случае представить работу алгоритмов со строками многосимвольного алфавита таким образом, чтобы размер логического вывода для результата работы такого алгоритма был в полиномиальных (или любых иных) пределах относительно времени работы этого алгоритма.

Доказательство.

Пусть у нас есть алгоритм проверки русского правописания, который должен отвергать (возвращая в качестве результата своей работы ноль, например) предложенный для проверки текст сразу, как только обнаружит ошибку в этом тексте. Пусть на вход данному алгоритму подан роман, первое же слово в котором начинается на «Ъ».

Разумеется, наш алгоритм отвергнет данный роман как неграмотно написанный по первой же его букве. Потому что в русском языке нет слов, которые начинаются на «Ъ». Замечу, что тут речь идёт о работе алгоритма со строкой, составленной на базе многосимвольного алфавита. И для простоты будем считать, что на любую другую букву русское слово начинаться может, что проще реального русского языка, где есть ещё буква «Ь».

Вопрос: можно ли в односимвольной теории строк построить такое представление для данного алгоритма проверки (сопоставив как-то многосимвольным строкам строки из символов односимвольного алфавита), что логический вывод данного результата (доказательство истинности результата, равного нулю) имел бы размер, не выходящий за некие полиномиальные ограничения относительно времени работы данного алгоритма?

Если бы это было так, то в логическом выводе не мог бы присутствовать весь роман, вообще говоря. Потому что размер романа может быть сколь угодно велик по сравнению с «Ъ» и временем работы алгоритма проверки.

Если же предположить, что в логическом выводе теории строк с односимвольным алфавитом присутствует лишь часть строки, то эта часть состоит из одних и тех же символов – будем считать «0», например. А размер всей строки, которая соответствует роману – неизвестен.

Допустим, логический вывод дает нужный нам результат 0 (отвергая роман как ошибочный) на основании подстроки «000...0» длиной в фиксированное количество n символов «0». Количество символов, представляющих весь роман – неизвестно. Это значит, что про роман, который представлен строкой, в которой есть фиксированное число n символов «0» заранее известно, что он

начинается на «Ъ». Я тут упрощаю (это напоминание), потому что при реальном русском языке ошибка будет и при «Ь», но разбираемая модель понятна.

А это означает, что все романы, которые представлены строками односимвольного алфавита длиной от n символов начинаются на символ «Ъ». А это означает, что все романы, которые не начинаются на символ «Ъ» должны соответствовать конечному числу вариантов, которые могут выразить строки (из символа односимвольного алфавита) длиной меньше, чем фиксированное число n символов «0».

Но таких романов – бесконечное количество, а вариантов разных строк, длина которых меньше n – конечное количество. Поэтому случай из предыдущего абзаца невозможен.

Полученное противоречие доказывает, что предположение о том, что:

В односимвольной теории строк логический вывод о том, что многосимвольный текст начинается на некий «запрещённый» символ может обойтись лишь частью строки, которая содержится в строке односимвольного алфавита, соответствующей данному тексту многосимвольного алфавита –

Является ошибочным.

Поскольку размер текста в сравнении с размером одного символа может быть как угодно велик, то мы доказали, что размер логического вывода и время на его проверку в односимвольной теории строк может быть сколь угодно велики в сравнении со временем работы алгоритма проверки. Метатеорема доказана.

(м.9.2) В арифметике невозможно в общем случае представить работу алгоритмов со строками многосимвольного алфавита таким образом, чтобы размер логического вывода для результата работы такого алгоритма был в полиномиальных (или любых иных) пределах относительно времени работы этого алгоритма.

Доказательство.

Выразительность арифметики не превосходит выразительности теории строк с односимвольным алфавитом, потому что арифметика является частью теории строк. Поэтому то, что невозможно логически вывести в теории строк с односимвольным алфавитом – невозможно вывести и в арифметике. Теорема доказана.

Как видим, благодаря теории строк удаётся понять недостаточную выразительность арифметики для того, чтобы пытаться решать в её рамках вопросы теории алгоритмов. И это понимание возникает за счёт «взгляда со стороны» на саму теорию строк, а уже поняв разницу между разными её вариантами – удаётся понять и в чём недостаточная выразительность арифметики для теории алгоритмов. И, разумеется, приведённые доказательства метатеорем не являются формальными и не могут быть формализованы в рамках теории строк – потому что это информация не теории строк, а про теорию строк.

Из метатеоремы (м.9.2) следует, что гёделевы номера – которые определяются в арифметике в качестве способа представить тексты логических формул, логических выводов, функций арифметики и т.п. – являются недостаточно выразительными для задач теории алгоритмов, как и сама арифметика.

А почему нельзя считать нужным представлением формулу, выражающую то, что нужная строка начинается на «Ъ»?

Просто потому, что мы в реальности не можем сказать – есть ли «Ъ» в многосимвольной строке, соответствующей данной «односимвольной» строке – не прочитав данную строку односимвольного алфавита до конца. А раз это невозможно в реальности, то невозможно и в теории – если эта теория непротиворечива и, если соответствующие ей функции мы строим в нашей реальности в возможных в этой реальности интерпретациях.

Но в любом случае мы опираемся на реальность в доказательстве метатеорем. В частности – не можем себе представить, как можно было бы решить вопрос с «Ъ» в начале, не прочитав всю строку из символа односимвольного алфавита.

В следующем разделе мы разберем связь между теорией строк и теорией Пеано (или любым расширением теории Пеано). И там будут представлен и вариант аксиоматизации для «односимвольной» связи строк и числе.

Для арифметики, получается, совершенно закономерно иметь интерпретацию, в которой необходимо читать число до конца. Потому что то, как «зашифровано» число – это никак не задано в самой арифметике. И поэтому ЧРФ никак не «портят» арифметику тем, что работают с входным аргументом только целиком – потому что арифметика «поступает» точно так же.

А в чём разница с многосимвольным алфавитом? В том, что в любой строке данной длины n из символов многосимвольного алфавита есть ещё варианты, кроме одного данного. И «клинча» - «тогда и остальные такие же» - не возникает. А вот при односимвольном алфавите логический вывод, сделанный на основании части длиной n от всей строки (произвольного размера), будет «таким же» для любых строк, имеющих длину больше или равную n .

Кстати, даже «частичное чтение» не позволило бы в арифметике сделать вывод, что число больше или равно числу прочитанных «символов». Действительно, ведь на базе арифметики можно аксиоматизировать следующую связь между числами и строками при односимвольном алфавите:

Если n – чётное число, но не степень 2, то число записано в виде строки из символов «0» длиной $n + 1$: «00...0». Если n – нечётное число, то число n записано как строка из символов «0»: «00...0». И количество символов «0» в этой строке (длина строки) будет $2^n + 1$. А вот число 2^n будет записана тоже как строка из «0», но длина строки в этой записи будет $n + 1$.

И тогда нет возможности в общем случае по чтению нескольких начальных символов утверждать, что число больше некоторого числа – в общем случае. Потому что оно может быть равно нечётному числу, хотя символов ты уже прочитал больше, чем данное нечётное число, но прочитал их пока меньше, чем до конца.

Заметим, что в реальности строки обладают свойством доступности для частичного чтения. Теперь недостаточная выразительность арифметики стала обоснованной в результате доказательства метатеоремы (м.9.2).

Если мы посмотрим в современные учебники теории алгоритмов, то увидим, что вычислимость на машине Тьюринга рассматривается в духе стандартной интерпретации, то есть, когда «входные аргументы» представлены последовательностью «счётных палочек», разделённых пустой клеткой

(условно скажем «символом нуля»), например. Для нуля (речь про обозначение числа, а не про пустую клетку-разделитель) используется «1», для 1 – «11», для 2 – «111» и т.д.

Я проверил по книгам:

«Теория алгоритмов» В.И. Игошин, «Курс математической логики и теории вычислимости» А.С. Герасимов, «Вычислимость и логика» Дж. Булос, Р. Джеффри.

То есть, не идёт речь о позиционном представлении чисел, хотя все реальные компьютеры работают именно с позиционным представлением чисел.

Понятно, что такое содержание учебников связано с отсутствием теории строк во время их написания. А в рамках арифметики (теории Пеано, например) просто нет средств для того, чтобы представить строки и записать числа в позиционном виде.

И представимость алгоритмов машины Тьюринга в арифметике доказывается не напрямую, а через ЧРФ. При этом машина Тьюринга работает со «стандартной интерпретацией» когда числа представлены блоками единиц. Про время работы вопрос не ставился. «Представимость» алгоритмов доказывалась в первой половине 20 века, в период великих логических открытий (неполнота, неразрешимость, теории первого и второго порядка, программа Гильберта и т.д. и т.п.). И всех интересовал вопрос возможности или невозможности тех или иных «вычислений» в принципе, независимо от времени этого вычисления – важно лишь, чтобы это время было конечным.

Тезис Чёрча понимался как возможность свести произвольное вычисление (которое завершается за конечное время) к некоторой ЧРФ, чтобы в итоге вычисления получить правильный результат – независимо от того времени, которое потребуется для этого – если время конечно. И независимо от размера логического вывода для доказательства того, что будет получен правильный результат – если размер логического вывода конечный.

И тезис Чёрча оказывается ошибочным, если понимать его как возможность свести произвольное вычисление к некоторой ЧРФ, без потери важных для теории алгоритмов свойств вычисления.

Что же теперь будет построено вместо ЧРФ? Будет построена – в статье номер 2 – «машина компьютерных алгоритмов» (МКА далее), и в ней будет реализована возможность частичного чтения входного аргумента.

Можно ли в качестве замены ЧРФ взять машину Тьюринга? Возможно – но это не удобно. И если в рамках прежнего построения «представимости» сопоставление между работой машины Тьюринга и теорией Пеано шло через ЧРФ и касалось только ЧРФ, то заменяя ЧРФ на МКА мы только упростим ситуацию и сделаем необходимое исправление – взяв верный метод с учётом задач теории алгоритмов, вместо неверного метода – в рамках указанных задач – который использовал ЧРФ.

VII. Критерий локального изменения данных

Теперь мы сформулируем и докажем теоремы, которые нам потребуются для доказательства «полиномиальной представимости» машины компьютерных алгоритмов в теории строк. Теория строк излагается в данной (1-й) статье, машина компьютерных алгоритмов строится в следующей (2-й) статье. Доказательство «полиномиальной представимости» будет дано в 3-й статье, а некоторые нюансы (оговоренные в 3-й статье) будут уточнены в 4-й статье.

$$(т.9.1) \text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus \Rightarrow \text{str}(a \cdot b \cdot u, \text{len}(a)', \text{len}(b)) = b$$

Доказательство при дедуктивном предположении $\text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus$.

Ветка 1. Дедуктивное предположение-2: $\text{len}(b) \neq 0$

Тогда из-за (т.4.2) ($i \neq 0 \wedge j \neq 0$) $\Rightarrow \text{str}(a, i, j) = \text{end}(\text{beg}(a, i + j - 1), i)$ получим:

$$1.1. \text{str}(a \cdot b \cdot u, \text{len}(a)', \text{len}(b)) = \text{end}(\text{beg}(a \cdot b \cdot u, \text{len}(a) + \text{len}(b)), \text{len}(a)')$$

Так как $\text{len}(a \cdot b) = \text{len}(a) + \text{len}(b)$ из-за (т.7.1), и в силу (т.8.6) $\text{len}(a) \neq \ominus \Rightarrow a = \text{beg}(a \cdot u, \text{len}(a))$

получаем:

$$\text{beg}((a \cdot b) \cdot u, \text{len}(a) + \text{len}(b)) = (a \cdot b)$$

Поэтому из п.1.1 и предыдущего равенства получим:

$$1.2. \text{str}(a \cdot b \cdot u, \text{len}(a)', \text{len}(b)) = \text{end}((a \cdot b), \text{len}(a)')$$

Но $\text{end}((a \cdot b), \text{len}(a)') = b$, из-за (т.8.7) $\text{len}(a) \neq \ominus \Rightarrow u = \text{end}(a \cdot u, \text{len}(a)'),$ поэтому

$$1.3. \text{str}(a \cdot b \cdot u, \text{len}(a)', \text{len}(b)) = b$$

Ветка 1 доказана.

Ветка 2. Дедуктивное предположение-2: $\text{len}(b) = 0$

Тогда в силу определения $\text{str}()$ и из-за (2.1) $\text{beg}(a, 0) = \ominus$ получим:

$$2.1. \text{str}(a \cdot b \cdot u, \text{len}(a)', \text{len}(b)) = \text{str}(a \cdot b \cdot u, \text{len}(a)', 0) = \ominus$$

С другой стороны, из-за (т.8.8) $a = \ominus \Leftrightarrow \text{len}(a) = 0$ получим:

$$2.2. b = \ominus$$

Из пунктов 2.1 и 2.2 получаем:

$$\text{str}(a \cdot b \cdot u, \text{len}(a)', \text{len}(b)) = b$$

Ветка 2 для случая $\text{len}(b) = 0$ тоже доказана. Вместе с веткой 1 для случая $\text{len}(b) \neq 0$, все варианты рассмотрены.

Теорема (т.9.1) доказана.

$$(т.9.2) \text{len}(a) \neq \ominus \Rightarrow \text{end}(a \cdot b \cdot u, \text{len}(a)') = b \cdot u$$

Доказательство при дедуктивном предположении $\text{len}(a) \neq \ominus$.

Из (т.8.7) $\text{len}(a) \neq \ominus \Rightarrow u = \text{end}(a \cdot u, \text{len}(a)'),$ получим, используя ассоциативность конкатенации:

$$\text{end}(a \cdot (b \cdot u), \text{len}(a)') = b \cdot u$$

Теорема (т.9.2) доказана.

$$(т.9.3) \text{IsIns}(\text{len}(a), i, \text{len}(b)) = 1 \Rightarrow \text{len}(\text{Ins}(a, i, b)) = \text{len}(a)$$

Доказательство при дедуктивном предположении $\text{IsIns}(\text{len}(a), i, \text{len}(b)) = 1$.

Из определения (9.7) и дедуктивного предположения получаем:

$$i \neq 0 \wedge \text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus \wedge (\text{len}(b) = 0 \vee \text{len}(a) \geq i + \text{len}(b) - 1)$$

Теперь просто вычислим, чему равна функция $\text{Ins}(a, i, b)$ при истинности предыдущего утверждения:

$\text{Ins}(a, i, b) = \text{beg}(a, \max(i, 1) - 1) \cdot b \cdot \text{end}(a, if_0(\min(\text{len}(\text{len}(b)), i), 0, \text{len}(b) + i))$, по определению (9.6)

В силу дедуктивного предположения $(\max(i, 1) - 1) = i - 1$ и в силу ещё и того, что $\text{len}(b) \neq \ominus$ (а, значит, $\text{len}(\text{len}(b)) \neq 0$ из-за (с.8.8)) верно $\min(\text{len}(\text{len}(b)), i) > 0$. Из последнего неравенства и определения if_0 получаем $if_0(\min(\text{len}(\text{len}(b)), i), 0, \text{len}(b) + i) = \text{len}(b) + i$. Поэтому:

$$\text{Ins}(a, i, b) = \text{beg}(a, i - 1) \cdot b \cdot \text{end}(a, \text{len}(b) + i)$$

Если $\text{len}(b) = 0$, то $b = \ominus$ из-за (т.8.8) и тогда его можно исключить из конкатенации в силу её ассоциативности и (т.8.3) $a = a \cdot \ominus = \ominus \cdot a$. Поэтому тогда:

$$\text{Ins}(a, i, b) = \text{beg}(a, i - 1) \cdot \ominus \cdot \text{end}(a, 0 + i) = \text{beg}(a, i - 1) \cdot \text{end}(a, i)$$

Но по (1) имеем:

$$a = \text{beg}(a, i - 1) \cdot \text{end}(a, i)$$

То есть, при $\text{len}(b) = 0$ верно:

$$\text{Ins}(a, i, b) = a$$

$$\text{len}(\text{Ins}(a, i, b)) = \text{len}(a)$$

Теперь рассмотрим $\text{len}(b) \neq 0$, тогда из дедуктивного предположения верно:

$$\text{len}(a) \geq i + \text{len}(b) - 1$$

Чтобы посчитать $\text{len}(\text{Ins}(a, i, b))$, будем считать длины каждого члена конкатенации в равенстве:

$$\text{Ins}(a, i, b) = \text{beg}(a, i - 1) \cdot b \cdot \text{end}(a, \text{len}(b) + i)$$

А что их сумма будет равна длине их конкатенации обеспечено теоремой (т.7.1) которая (напоминание):

$$\text{len}(a_1) \neq \ominus \wedge \dots \wedge \text{len}(a_n) \neq \ominus \Rightarrow \text{len}(a_1 \cdot \dots \cdot a_n) = \text{len}(a_1) + \dots + \text{len}(a_n)$$

Поэтому:

$$\text{len}(\text{Ins}(a, i, b)) = \text{len}(\text{beg}(a, i - 1)) + \text{len}(b) + \text{len}(\text{end}(a, \text{len}(b) + i))$$

Из-за теоремы (т.6.7) которая (напоминание):

$$\text{len}(a) \neq \ominus \wedge i \leq \text{len}(a) \Rightarrow \text{len}(\text{beg}(a, i)) = i \wedge (\text{len}(\text{end}(a, i)) = \text{len}(a) - i + 1 \vee i = 0)$$

Получаем (так как $(i - 1) \leq \text{len}(a)$):

$$\text{len}(\text{beg}(a, i - 1)) = i - 1$$

Слагаемое $\text{len}(b)$ не требует вычисления, осталось найти $\text{len}(\text{end}(a, \text{len}(b) + i))$

Случай 1.

Для вычисления данного слагаемого нам может пригодиться либо упомянутая теорем (6.7) в части:

$\text{len}(\text{end}(a, i)) = \text{len}(a) - i + 1$, но только если $i \leq \text{len}(a)$ в теореме и $\text{len}(b) + i \leq \text{len}(a)$ в нашем конкретном случае. Тогда:

$$\text{len}(\text{end}(a, \text{len}(b) + i)) = \text{len}(a) - (\text{len}(b) + i) + 1 = \text{len}(a) - \text{len}(b) - i + 1$$

$$\text{len}(\text{Ins}(a, i, b)) = (i - 1) + \text{len}(b) + (\text{len}(a) - \text{len}(b) - i + 1) = \text{len}(a)$$

Для Случая 1 теорема доказана.

Случай 2.

Либо для вычисления $\text{len}(\text{end}(a, \text{len}(b)+i))$ нам пригодится теорема (6.9) которая (напоминание):

$$\text{len}(a) \neq \ominus \wedge \text{len}(a) < i \Rightarrow \text{len}(\text{beg}(a, i)) = \text{len}(a) \wedge (\text{len}(\text{end}(a, i)) = 0)$$

Нас интересует следующая часть теоремы (6.9):

$\text{len}(\text{end}(a, i)) = 0$, но только если $\text{len}(a) < i$ в теореме и $\text{len}(a) < \text{len}(b) + i$ в нашем конкретном случае. Тогда вместе с условием $\text{len}(a) \geq i + \text{len}(b) - 1$ из дедуктивного предположения имеем:

$$\text{len}(a) \geq i + \text{len}(b) - 1 \wedge \text{len}(a) < \text{len}(b) + i$$

Перепишем для большей наглядности так:

$$\text{len}(b) + i - 1 \leq \text{len}(a) < \text{len}(b) + i$$

В арифметике из этого выводится:

$$\text{len}(a) = \text{len}(b) + i - 1$$

$$\text{len}(b) = \text{len}(a) - i + 1$$

Теперь можем посчитать всю сумму длин:

$$\text{len}(\text{Ins}(a, i, b)) = (i - 1) + \text{len}(b) + 0 = i - 1 + \text{len}(a) - i + 1 = \text{len}(a)$$

Для Случая 2 теорема тоже доказана.

Теорема доказана.

(т.9.4) Критерий локальной вставки:

$$(\text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus) \Rightarrow$$

$$(\text{IsIns}(\text{len}(a), i, \text{len}(b))) = 1 \Leftrightarrow \forall u \text{Ins}(a \cdot u, i, b) = \text{Ins}(a, i, b) \cdot u$$

Для доказательства этой эквивалентности нужно доказать, что в условиях 1-го дедуктивного предположения:

$\text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus$ истинны 2 импликации:

$\text{IsIns}(\text{len}(a), i, \text{len}(b)) = 1 \Rightarrow \forall u \text{Ins}(a \cdot u, i, b) = \text{Ins}(a, i, b) \cdot u$ – первая импликация;

$\forall u \text{Ins}(a \cdot u, i, b) = \text{Ins}(a, i, b) \cdot u \Rightarrow \text{IsIns}(\text{len}(a), i, \text{len}(b)) = 1$ – вторая импликация.

Для доказательства 1-й импликации делаем 2-е дедуктивное предположение:

$$\text{IsIns}(\text{len}(a), i, \text{len}(b)) = 1$$

Из определения (9.7) и 2-го дедуктивного предположения получаем:

$$i \neq 0 \wedge \text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus \wedge (\text{len}(b) = 0 \vee \text{len}(a) \geq i + \text{len}(b) - 1)$$

Теперь просто вычислим, чему равна функция $\text{Ins}(a \cdot u, i, b)$ из заключения 1-й импликации при истинности предыдущего утверждения:

$\text{Ins}(a \cdot u, i, b) = \text{beg}(a \cdot u, \max(i, 1) - 1) \cdot b \cdot \text{end}(a \cdot u, \text{if}_0(\min(\text{len}(\text{len}(b)), i), 0, \text{len}(b) + i))$, по определению (9.6)

В силу 2-го дедуктивного предположения $\max(i, 1) - 1 = i - 1$ и в силу ещё и того, что $\text{len}(b) \neq \ominus$ (из 1-го дедуктивного предположения) верно $\min(\text{len}(\text{len}(b)), i) > 0$ (так как $\text{len}(b) \neq \ominus \Leftrightarrow \text{len}(\text{len}(b)) \neq 0$ из-за (с.8.8)) Из последнего неравенства и определения if_0 получаем $\text{if}_0(\min(\text{len}(\text{len}(b)), i), 0, \text{len}(b) + i) = \text{len}(b) + i$. Поэтому:

$\text{Ins}(a \cdot u, i, b) = \text{beg}(a \cdot u, i - 1) \cdot b \cdot \text{end}(a \cdot u, \text{len}(b) + i)$, но:

$$\text{beg}(a \cdot u, i - 1) = \text{str}(a \cdot u, 1, i - 1), \text{ из-за (т.4.6),}$$

Поэтому:

$$\text{Ins}(a \cdot u, i, b) = \text{str}(a \cdot u, 1, i - 1) \cdot b \cdot \text{end}(a \cdot u, \text{len}(b) + i)$$

Ветка 1 для 1-й импликации. Случай $\text{len}(b) = 0$.

Если $\text{len}(b) = 0$, то $b = \ominus$ из-за (т.8.8) и тогда его можно исключить из конкатенации в силу её ассоциативности и (т.8.3) $a = a \cdot \ominus = \ominus \cdot a$. Поэтому тогда:

$$\text{Ins}(a \cdot u, i, b) = \text{str}(a \cdot u, 1, i - 1) \cdot \text{end}(a \cdot u, 0 + i)$$

Так $\text{str}(a \cdot u, 1, i - 1) = \text{beg}(a \cdot u, i - 1)$ из-за (т.4.6) $\text{str}(a, 1, j) = \text{beg}(a, j)$, то

$$\text{Ins}(a \cdot u, i, b) = \text{beg}(a \cdot u, i - 1) \cdot \text{end}(a \cdot u, i)$$

В силу (1) поэтому при $\text{len}(b) = 0$ получаем:

$$\text{Ins}(a \cdot u, i, b) = a \cdot u$$

Так как

$$\text{Ins}(a, i, b) = \text{Ins}(a \cdot \ominus, i, b) = a \cdot \ominus = a$$

То

$$\text{Ins}(a \cdot u, i, b) = \text{Ins}(a, i, b) \cdot u$$

Для ветки 1 импликация 1 доказана.

Ветка 2 для 1-й импликации. Случай $\text{len}(b) > 0$. Из этого и из 2-го дедуктивного предположения получается $\text{len}(a) \geq i + \text{len}(b) - 1$.

У нас доказано при 1-м и 2-м предположениях дедукции

$$\text{Ins}(a \cdot u, i, b) = \text{str}(a \cdot u, 1, i - 1) \cdot b \cdot \text{end}(a \cdot u, \text{len}(b) + i)$$

Но $i - 1 \leq \text{len}(a)$, так как выше мы видели, что $\text{len}(b) + i - 1 \leq \text{len}(a)$. Поэтому:

$\text{str}(a \cdot u, 1, i - 1) = \text{str}(a, 1, i - 1)$, что выводится из (т.8.6) $\text{len}(a) \neq \ominus \Rightarrow a = \text{beg}(a \cdot u, \text{len}(a))$, после того, как разбить a в соответствии с (1) $a = \text{beg}(a, i - 1) \cdot \text{end}(a, i)$ и воспользоваться теоремой (т.6.7) чтобы доказать $\text{len}(\text{beg}(a, i - 1)) = i - 1$, подобно тому, как мы делали чуть выше.

Таким образом:

$$\text{Ins}(a \cdot u, i, b) = \text{str}(a, 1, i - 1) \cdot b \cdot \text{end}(a \cdot u, \text{len}(b) + i)$$

У нас остался только один член конкатенации, зависящий от u . Поэтому теперь надо доказать, что:

$$\text{end}(a \cdot u, \text{len}(b) + i) = \text{end}(a, \text{len}(b) + i) \cdot u$$

Запишем на основании аксиомы (1) переменную a в виде:

$$a = \text{beg}(a, \text{len}(b) + i - 1) \cdot \text{end}(a, \text{len}(b) + i)$$

Тогда

$$\text{end}(a \cdot u, \text{len}(b) + i) = \text{end}(\text{beg}(a, \text{len}(b) + i - 1) \cdot \text{end}(a, \text{len}(b) + i) \cdot u, \text{len}(b) + i)$$

В силу 2-го дедуктивного и предположения Ветки 2 верно $\text{len}(a) \geq i + \text{len}(b) - 1$, поэтому $\text{len}(b) + i - 1 \leq \text{len}(a)$

Поэтому в силу (т.6.7) которая (напоминание):

$$\text{len}(a) \neq \ominus \wedge i \leq \text{len}(a) \Rightarrow \text{len}(\text{beg}(a, i)) = i \wedge (\text{len}(\text{end}(a, i)) = \text{len}(a) - i + 1 \vee i = 0)$$

Имеем:

$$\text{len}(\text{beg}(a, \text{len}(b) + i - 1)) = \text{len}(b) + i - 1$$

А поэтому из теоремы (т.9.2) которая (напоминание):

$$\text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus \Rightarrow \text{end}(a \cdot b \cdot u, \text{len}(a)') = b \cdot u$$

Получим

$$\text{end}(\text{beg}(a, \text{len}(b) + i - 1) \cdot \text{end}(a, \text{len}(b) + i) \cdot u, \text{len}(b) + i) = \text{end}(a, \text{len}(b) + i) \cdot u$$

Таким образом, мы получили:

$$\text{end}(a \cdot u, \text{len}(b) + i) = \text{end}(a, \text{len}(b) + i) \cdot u$$

И теперь мы можем переписать предложение

$$\text{Ins}(a \cdot u, i, b) = \text{str}(a, 1, i - 1) \cdot b \cdot \text{end}(a \cdot u, \text{len}(b) + i)$$

Таким образом:

$$\text{Ins}(a \cdot u, i, b) = \text{str}(a, 1, i - 1) \cdot b \cdot \text{end}(a, \text{len}(b) + i) \cdot u$$

Если рассмотреть случай $u = \ominus$, то получим:

$$\text{Ins}(a, i, b) = \text{Ins}(a \cdot \ominus, i, b) = \text{str}(a, 1, i - 1) \cdot b \cdot \text{end}(a, \text{len}(b) + i)$$

Из последних 2 формул получаем:

$$\text{Ins}(a \cdot u, i, b) = \text{Ins}(a, i, b) \cdot u$$

Ветка 2 для импликации 1 доказана.

Чтобы перейти к квантору общности, надо для результата этих 2х веток (сделав из каждой импликацию по теореме дедукции, соединив обе эти импликации в одну и отбросив дизъюнкцию частных посылок по МР) воспользоваться правилом вывода:

$$(\alpha) \text{ Если верно } U \Rightarrow B(a), \text{ то верно } U \Rightarrow \forall x B(x)$$

После чего импликация 1 доказана.

Теперь надо доказать в рамках $\text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus$ 1-го дедуктивного предположения 2-ю импликацию:

$$\forall u \text{Ins}(a \cdot u, i, b) = \text{Ins}(a, i, b) \cdot u \Rightarrow \text{IsIns}(\text{len}(a), i, \text{len}(b)) = 1$$

Перепишем её в эквивалентном виде при помощи контрапозиции, превращая квантор всеобщности в квантор существования при помощи теоремы VI.7 из раздела 4. Приложение:

$$\text{IsIns}(\text{len}(a), i, \text{len}(b)) \neq 1 \Rightarrow \exists u \text{Ins}(a \cdot u, i, b) \neq \text{Ins}(a, i, b) \cdot u$$

Очевидно, что

$$\text{IsIns}(\text{len}(a), i, \text{len}(b)) \neq 1 \Leftrightarrow \text{IsIns}(\text{len}(a), i, \text{len}(b)) = 0$$

Поэтому перепишем 2ю импликацию так:

$$\text{IsIns}(\text{len}(a), i, \text{len}(b)) = 0 \Rightarrow \exists u \text{Ins}(a \cdot u, i, b) \neq \text{Ins}(a, i, b) \cdot u$$

Для доказательства 2-й импликации делаем 2-е дедуктивное предположение:

$$\text{IsIns}(\text{len}(a), i, \text{len}(b)) = 0$$

Из определения (9.7) и 2-го дедуктивного предположения получаем «рабочий вариант» 2-го дедуктивного предположения:

$$i = 0 \vee \text{len}(a) = \ominus \vee \text{len}(b) = \ominus \vee (\text{len}(b) \neq 0 \wedge \text{len}(a) < i + \text{len}(b) - 1)$$

Теперь нам нужно доказать, что из каждого члена дизъюнкции приведённого рабочего варианта 2-го дизъюнктивного предположения следует нужное нам заключение

$$\exists u \text{Ins}(a \cdot u, i, b) \neq \text{Ins}(a, i, b) \cdot u$$

А после этого мы сможем объединить все импликации с разными посылками и одинаковым заключением в импликацию, где посылка будет дизъюнкцией при помощи

$$\text{III.3 } (A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \vee B \Rightarrow C)), \text{ из раздела 4. Приложение.}$$

Поэтому доказываем последовательно для каждой такой посылки нужное нам заключение

Ветка 1. $i = 0$

Положим

$u = \text{Chr}(0)$, тогда из определения (9.6)

$$\text{Ins}(a \cdot u, i, b) = \text{beg}(a \cdot u, \max(i, 1) - 1) \cdot b \cdot \text{end}(a \cdot u, if_0(\min(\text{len}(\text{len}(b)), i), 0, \text{len}(b) + i))$$

Подстановкой соответствующего $i = 0$ получим для двух членов конкатенации:

$$\text{beg}(a \cdot u, \max(i, 1) - 1) = \text{beg}(a \cdot u, \max(0, 1) - 1) = \text{beg}(a \cdot u, 0) = \ominus$$

$$\text{end}(a \cdot u, if_0(\min(\text{len}(\text{len}(b)), i), 0, \text{len}(b) + i)) = \text{end}(a \cdot u, if_0(0, 0, \text{len}(b) + i)) = \ominus$$

Поэтому:

$$\text{Ins}(a \cdot u, i, b) = \ominus \cdot b \cdot \ominus = b$$

Что касается $\text{Ins}(a, i, b) \cdot u$, то по аналогичному выводу получим (подставляя $u = \text{Chr}(0)$):

$$\text{Ins}(a, i, b) \cdot u = b \cdot u = b \cdot \text{Chr}(0)$$

Очевидно, что длины строк, получающиеся из 2-х последних формул не равны:

$$\text{len}(\text{Ins}(a \cdot u, i, b)) \neq \text{len}(\text{Ins}(a, i, b) \cdot u)$$

А не равны они из-за теоремы (т.7.1) которая (напоминание):

$$\text{len}(a_1) \neq \ominus \wedge \dots \wedge \text{len}(a_n) \neq \ominus \Rightarrow \text{len}(a_1 \cdot \dots \cdot a_n) = \text{len}(a_1) + \dots + \text{len}(a_n)$$

Поэтому

$$\text{len}(b) < \text{len}(b \cdot \text{Chr}(0)) = \text{len}(b) + \text{len}(\text{Chr}(0)) = \text{len}(b) + 1$$

Кстати, про $\text{Chr}(0)$ мы можем заведомо утверждать, что $\text{Chr}(0) \neq \ominus$, а поэтому и длина его равна 1 из-за теоремы (т.7.2) которая (напоминание):

$$i \leq l_{\text{ChrLim}} \Rightarrow \text{len}(\text{Chr}(i)) = 1:$$

Ведь для случая $i = 0$ заведомо выполнено $i \leq l_{\text{ChrLim}}$, каким бы натуральным числом l_{ChrLim} ни был.

А в силу того, что длины строк не равны – не равны и сами строки из-за теоремы (т.8.5) которая (напоминание):

$$\text{len}(a) \neq \ominus \Rightarrow [a = b \Leftrightarrow \forall i((\text{len}(a) = \text{len}(b) \wedge 0 < i \wedge i \leq \text{len}(a)) \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))]$$

Поэтому для $u = \text{Chr}(0)$ верно:

$$\text{Ins}(a \cdot u, i, b) \neq \text{Ins}(a, i, b) \cdot u$$

То есть, верно – после подстановки – следующее неравенство:

$$\text{Ins}(a \cdot b \cdot \text{Chr}(0), i, b) \neq \text{Ins}(a, i, b) \cdot b \cdot \text{Chr}(0)$$

В силу аксиомы логики предикатов из раздела 4. Приложение:

$$(b) A(a) \Rightarrow \exists x A(x)$$

Верно:

$$\text{Ins}(a \cdot u, i, b) \neq \text{Ins}(a, i, b) \cdot u \Rightarrow \exists u \text{Ins}(a \cdot u, i, b) \neq \text{Ins}(a, i, b) \cdot u$$

И мы можем подставить на место свободных переменных конкретное значение $u = \text{Chr}(0)$:

$$\text{Ins}(a \cdot \text{Chr}(0), i, b) \neq \text{Ins}(a, i, b) \cdot \text{Chr}(0) \Rightarrow \exists u \text{Ins}(a \cdot u, i, b) \neq \text{Ins}(a, i, b) \cdot u$$

Посылка данной импликации доказана чуть выше, поэтому её можно отбросить по правилу вывода МР. И мы получаем:

$$\exists u \text{Ins}(a \cdot u, i, b) \neq \text{Ins}(a, i, b) \cdot u$$

Что и требовалось доказать для Ветки 1.

Ветка 2. $\text{len}(a) = \ominus \vee \text{len}(b) = \ominus$

В этой ветке 2-е дедуктивное предположение является точным логическим отрицанием 1-го с его $\text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus$. А из противоречия доказано всё. Поэтому для Ветки 2 заключение импликации 2 тоже доказано.

Ветка 3. $(\text{len}(b) \neq 0 \wedge \text{len}(a) < i + \text{len}(b) - 1)$

А точнее, мы рассмотрим Ветку 3 в таком «рабочем виде»:

$i \neq 0 \wedge \text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus \wedge (\text{len}(b) \neq 0 \wedge \text{len}(a) < i + \text{len}(b) - 1)$

Поясню такую «метаморфозу».

У нас 2-е дедуктивное предположение:

$i = 0 \vee \text{len}(a) = \ominus \vee \text{len}(b) = \ominus \vee (\text{len}(b) \neq 0 \wedge \text{len}(a) < i + \text{len}(b) - 1)$

Если записать его в виде:

$A \vee B$

Где член дизъюнкции A – уже рассмотрен в «Ветка 1» и «Ветка 2», а сейчас мы рассматриваем в качестве посылки член дизъюнкции B . И «вдруг» подменяем его на:

$(\neg A \wedge B)$

Это возможно в том случае, если 2-е дедуктивное предположение в виде $A \vee B$ эквивалентно следующему:

$A \vee (\neg A \wedge B)$

А оно действительно эквивалентно в силу теоремы VI.10 $(A \vee B) \Leftrightarrow (A \vee (\neg A \wedge B))$ из раздела 4. Приложение.

Поэтому наша «подмена» дедуктивного предположения-2 Ветки 3:

$\text{len}(b) \neq 0 \wedge \text{len}(a) < i + \text{len}(b) - 1$

на «Рабочую посылку 3»:

$i \neq 0 \wedge \text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus \wedge (\text{len}(b) \neq 0 \wedge \text{len}(a) < i + \text{len}(b) - 1)$

является корректной.

Из рабочей посылки 3 нам нужно вывести заключение 2-й импликации:

$\exists u \text{Ins}(a \cdot u, i, b) \neq \text{Ins}(a, i, b) \cdot u$

Для этого положим:

$u = \text{ss}(i + \text{len}(b) - 1 - \text{len}(a))$

В качестве аргумента в $\text{ss}()$ тут – натуральное число, не равное нулю, в силу

$\text{len}(a) < i + \text{len}(b) - 1$, из рабочей посылки 3.

Теперь просто вычислим размер $\text{Ins}(a \cdot u, i, b)$ для которого верно:

$\text{Ins}(a \cdot u, i, b) = \text{Ins}(a \cdot \text{ss}(i + \text{len}(b) - 1 - \text{len}(a)), i, b)$

Так как:

$\text{len}(a \cdot \text{ss}(i + \text{len}(b) - 1 - \text{len}(a))) = \text{len}(a) + \text{len}(\text{ss}(i + \text{len}(b) - 1 - \text{len}(a)))$

То учтём, что по определению $\text{ss}(i)$ верно: $\text{len}(\text{ss}(i)) = i$, поэтому:

$\text{len}(a \cdot \text{ss}(i + \text{len}(b) - 1 - \text{len}(a))) = \text{len}(a) + i + \text{len}(b) - 1 - \text{len}(a) = i + \text{len}(b) - 1$

А это значит, что наша реализация

$a \cdot u = a \cdot \text{ss}(i + \text{len}(b) - 1 - \text{len}(a))$

удовлетворяет условию:

$$\text{len}(a \cdot u) \leq i + \text{len}(b) - 1$$

Кроме того, верно:

$$\text{len}(a \cdot u) \neq \ominus$$

И выполнено:

$$\text{IsIns}(a \cdot u, i, b) = 1$$

А поэтому воспользуемся теоремой (т.9.3), которая (напоминание):

$$\text{IsIns}(\text{len}(a), i, \text{len}(b)) = 1 \Rightarrow \text{len}(\text{Ins}(a, i, b)) = \text{len}(a)$$

Из неё мы непосредственно получаем:

$$\text{len}(\text{Ins}(a \cdot u, i, b)) = \text{len}(a \cdot u) = i + \text{len}(b) - 1, \text{ при } u = \text{ss}(i + \text{len}(b) - 1 - \text{len}(a)).$$

А теперь посчитаем $\text{len}(\text{Ins}(a, i, b) \cdot u)$:

$$\text{len}(\text{Ins}(a, i, b) \cdot u) = \text{len}(\text{Ins}(a, i, b)) + \text{len}(u)$$

Из определения (9.6) для $\text{Ins}(a, i, b)$:

$$\text{Ins}(a, i, b) = \text{beg}(a, \max(i, 1) - 1) \cdot b \cdot \text{end}(a, \text{if}_0(\min(\text{len}(\text{len}(b)), i), 0, \text{len}(b) + i))$$

Для $\text{beg}(a, \max(i, 1) - 1)$ верно:

$$\text{len}(\text{beg}(a, \max(i, 1) - 1)) = \text{len}(\text{beg}(a, i - 1)) = \min(\text{len}(a), i - 1)$$

Действительно, $\max(i, 1) = i$, так как в Рабочей посылке 3 выполнено $i \neq 0$, а последняя строка в цепочке равенств – из следствия (с.6.9), которая (напоминание):

$$\text{len}(a) \neq \ominus \Rightarrow \text{len}(\text{beg}(a, i)) = \min(i, \text{len}(a)) \wedge (\text{len}(\text{end}(a, i)) = \max(\text{len}(a), i + 1) - i + 1 \vee i = 0)$$

Для $\text{end}(a, \text{if}_0(\min(\text{len}(\text{len}(b)), i), 0, \text{len}(b) + i))$ верно:

$$\text{len}(\text{end}(a, \text{if}_0(\min(\text{len}(\text{len}(b)), i), 0, \text{len}(b) + i))) = \text{len}(\text{end}(a, \text{len}(b) + i))$$

Действительно, $\min(\text{len}(\text{len}(b)), i) \neq 0$, так как в Рабочей посылке 3 выполнено $i \neq 0$, $\text{len}(b) \neq 0$, $\text{len}(b) \neq \ominus$. Из $\text{len}(b) \neq \ominus$ следует $\text{len}(\text{len}(b)) \neq 0$ в силу (с.8.8) $\text{len}(a) = \ominus \Leftrightarrow \text{len}(\text{len}(a)) = 0$. А поэтому $\text{if}_0(\min(\text{len}(\text{len}(b)), i), 0, \text{len}(b) + i)$ равно $\text{len}(b) + i$.

Таким образом:

$$\text{len}(\text{Ins}(a, i, b)) = \min(\text{len}(a), i - 1) + \text{len}(b) + \text{len}(\text{end}(a, \text{len}(b) + i))$$

Так как в Рабочей посылке 3 верно $\text{len}(a) < i + \text{len}(b) - 1$, то тем более $\text{len}(a) < i + \text{len}(b)$.

Поэтому в силу теоремы (т.6.9), которая (напоминание):

$$\text{len}(a) \neq \ominus \wedge \text{len}(a) < i \Rightarrow \text{len}(\text{beg}(a, i)) = \text{len}(a) \wedge (\text{len}(\text{end}(a, i)) = 0)$$

Получаем:

$$\text{len}(\text{end}(a, \text{len}(b) + i)) = 0$$

Значит:

$$\text{len}(\text{Ins}(a, i, b)) = \min(\text{len}(a), i - 1) + \text{len}(b) + 0 = \min(\text{len}(a), i - 1) + \text{len}(b)$$

Вспоминая, что мы рассматриваем $u = \text{ss}(i + \text{len}(b) - 1 - \text{len}(a))$ где $\text{len}(u) = i + \text{len}(b) - 1 - \text{len}(a)$ получим:

$$\text{len}(\text{Ins}(a, i, b) \cdot u) = \min(\text{len}(a), i - 1) + \text{len}(b) + \text{len}(u) = \min(\text{len}(a), i - 1) + \text{len}(b) + i + \text{len}(b) - 1 - \text{len}(a)$$

$$\text{len}(\text{Ins}(a, i, b) \cdot u) = \min(\text{len}(a), i - 1) + i - 1 - \text{len}(a) + 2 \times \text{len}(b)$$

И сравним это с полученным выше:

$$\text{len}(\text{Ins}(a \cdot u, i, b)) = i + \text{len}(b) - 1$$

Тогда:

$$\text{len}(\text{Ins}(a, i, b) \cdot u) - \text{len}(\text{Ins}(a \cdot u, i, b)) = [\min(\text{len}(a), i - 1) + i - 1 - \text{len}(a) + 2 \times \text{len}(b)] - [i + \text{len}(b) - 1]$$

$$\text{len}(\text{Ins}(a \cdot u, i, b)) - \text{len}(\text{Ins}(a, i, b) \cdot u) = \min(\text{len}(a), i - 1) + \text{len}(b) - \text{len}(a)$$

Так как в Рабочей посылке 3 верно $\text{len}(a) < i + \text{len}(b) - 1$, то результат последнего выражения больше 0, так как если $\min(\text{len}(a), i - 1) = i - 1$, то $\text{len}(a)$ вычитается от $i + \text{len}(b) - 1$, которая больше $\text{len}(a)$. А если $\min(\text{len}(a), i - 1) = \text{len}(a)$, то результат выражения равен $\text{len}(b)$, а $\text{len}(b) \neq 0$ по Рабочей посылке 3.

Таким образом:

$$\text{len}(\text{Ins}(a \cdot u, i, b)) \neq \text{len}(\text{Ins}(a, i, b) \cdot u)$$

Значит, по теореме (т.8.5), которая (напоминание)::

$$\text{len}(a) \neq \ominus \Rightarrow [a = b \Leftrightarrow \forall i((\text{len}(a) = \text{len}(b) \wedge 0 < i \wedge i \leq \text{len}(a)) \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))]$$

Получаем:

$$\text{Ins}(a \cdot u, i, b) \neq \text{Ins}(a, i, b) \cdot u, \text{ при } u = \text{ss}(i + \text{len}(b) - 1 - \text{len}(a)).$$

От этого конкретного значения u переходим к нужному нам заключению:

$$\exists u \text{Ins}(a \cdot u, i, b) \neq \text{Ins}(a, i, b) \cdot u$$

Для перехода от предпоследнего предложения к последнему предложению используем тот же способ, который использовался при рассмотрении Ветки 1.

Все варианты посылок рассмотрены, поэтому импликация 2 тоже доказана.

Теорема доказана.

VIII. Критерий локального извлечения данных

Критерий локального извлечения данных:

$$(т.9.5) \text{ len}(a) \neq \ominus \Rightarrow (\text{IsStr}(\text{len}(a), i, j) = 1 \Leftrightarrow \forall u \text{ str}(a \cdot u, i, j) = \text{str}(a, i, j))$$

Для доказательства этой эквивалентности нужно доказать, что в условиях 1-го дедуктивного предположения:

$\text{len}(a) \neq \ominus$ истинны 2 импликации:

$\text{IsStr}(\text{len}(a), i, j) = 1 \Rightarrow \forall u \text{ str}(a \cdot u, i, j) = \text{str}(a, i, j)$ – первая импликация;

$\forall u \text{ str}(a \cdot u, i, j) = \text{str}(a, i, j) \Rightarrow \text{IsStr}(\text{len}(a), i, j) = 1$ – вторая импликация.

Для доказательства 1-й импликации делаем 2-е дедуктивное предположение:

$$\text{IsStr}(\text{len}(a), i, j) = 1$$

Из определения (9.8) и 2-го дедуктивного предположения получаем:

$$\text{len}(a) \neq \ominus \wedge (i = 0 \vee j = 0 \vee \text{len}(a) \geq i + j - 1)$$

Ветка 1. $i = 0$

Тогда из определения $\text{str}()$ и из-за (2.2) $\text{end}(a, 0) = \ominus$ и (2.4) $\text{beg}(\ominus, q) = \ominus$ получим:

$$\text{str}(a \cdot u, i, j) = \text{str}(a \cdot u, 0, j) = \text{beg}(\text{end}(a \cdot u, 0), j) = \ominus$$

Аналогично:

$$\text{str}(a, i, j) = \ominus$$

Поэтому

$$\text{str}(a \cdot u, i, j) = \text{str}(a, i, j)$$

Первая импликация для Ветки 1 доказана из последнего равенства в виде:

$$\text{IsStr}(\text{len}(a), i, j) = 1 \Rightarrow \forall u \text{ str}(a \cdot u, i, j) = \text{str}(a, i, j)$$

А переход к квантору общности делается по аксиоме из Раздела 4. Приложение:

$$(\alpha) \text{ Если верно } U \Rightarrow B(a), \text{ то верно } U \Rightarrow \forall x B(x)$$

Ветка 2. $j = 0$

Тогда из определения $\text{str}()$ и из-за (2.1) $\text{beg}(a, 0) = \ominus$ получим:

$$\text{str}(a \cdot u, i, j) = \text{str}(a \cdot u, i, 0) = \text{beg}(\text{end}(a \cdot u, i), 0) = \ominus$$

Аналогично:

$$\text{str}(a, i, j) = \ominus$$

Поэтому

$$\text{str}(a \cdot u, i, j) = \text{str}(a, i, j)$$

Завершение доказательства Ветки 2 с квантором общности – как для Ветки 1.

Ветка 3. $(i \neq 0 \wedge j \neq 0 \wedge \text{len}(a) \geq i + j - 1)$

Почему рассмотренные Ветки 1 и 2 ($i = 0 \vee j = 0$) превратились в своё отрицание, и добавились как член конъюнкции к оставшемуся для рассмотрения варианту – см. теорему VI.10 $(A \vee B) \Leftrightarrow (A \vee (\neg A \wedge B))$ в разделе 4. Приложение.

Теперь, чтобы вычислить, чему равна функция $\text{str}(a \cdot u, i, j)$, представим переменную a следующим образом в соответствии с аксиомой (1) и с учётом $i \neq 0$ получим:

$$a = \text{beg}(a, i - 1) \cdot \text{end}(a, i)$$

$$\text{end}(a, i) = \text{beg}(\text{end}(a, i), j) \cdot \text{end}(\text{end}(a, i), j')$$

На основании (т.4.6) $\text{str}(a, 1, j) = \text{beg}(a, j)$ получаем:

$$\text{beg}(a, i - 1) = \text{str}(a, 1, i - 1)$$

На основании определения (4.1) для $\text{str}()$ получаем:

$$\text{beg}(\text{end}(a, i), j) = \text{str}(a, i, j)$$

На основании (т.3.2) $(i \neq 0 \wedge j \neq 0) \Rightarrow \text{end}(a, i + j - 1) = \text{end}(\text{end}(a, i), j)$ получаем:

$$\text{end}(\text{end}(a, i), j') = \text{end}(a, i + j)$$

Теперь можно переписать a следующим образом:

$$a = \text{str}(a, 1, i - 1) \cdot \text{str}(a, i, j) \cdot \text{end}(a, i + j)$$

На основании следствия (с.6.7) которое (напоминание):

$$\text{len}(a) \neq \ominus \wedge i \neq 0 \wedge i + j - 1 \leq \text{len}(a) \Rightarrow \text{len}(\text{str}(a, i, j)) = j$$

Получаем с учётом того, что $\text{len}(a) \geq i + j - 1$:

$$\text{len}(\text{str}(a, 1, i - 1)) = i - 1$$

$$\text{len}(\text{str}(a, i, j)) = j$$

На основании трёх последних предложений (не считая (с.6.7)), ассоциативности конкатенации и теоремы (т.9.1) которая (напоминание):

$$\text{len}(a) \neq \ominus \wedge \text{len}(b) \neq \ominus \Rightarrow \text{str}(a \cdot b \cdot u, \text{len}(a)', \text{len}(b)) = b$$

Получаем:

$$\text{str}(a \cdot u, i, j) = \text{str}(\text{str}(a, 1, i - 1) \cdot \text{str}(a, i, j) \cdot \text{end}(a, i + j) \cdot u, i, j) = \text{str}(a, i, j)$$

Завершение доказательства Ветки 3 – как для Ветки 1.

Первая импликация доказана.

Теперь надо доказать вторую импликацию:

$$\forall u \text{str}(a \cdot u, i, j) = \text{str}(a, i, j) \Rightarrow \text{IsStr}(\text{len}(a), i, j) = 1$$

Перепишем её в эквивалентном виде при помощи контрапозиции, превращая квантор общности в квантор существования при помощи теоремы VI.7 из раздела 4. Приложение:

$$\text{IsStr}(\text{len}(a), i, j) \neq 1 \Rightarrow \exists u \text{str}(a \cdot u, i, j) \neq \text{str}(a, i, j)$$

Очевидно, что

$$\text{IsStr}(\text{len}(a), i, j) \neq 1 \Leftrightarrow \text{IsStr}(\text{len}(a), i, j) = 0$$

Поэтому перепишем 2ю импликацию так:

$$\text{IsStr}(\text{len}(a), i, j) = 0 \Rightarrow \exists u \text{str}(a \cdot u, i, j) \neq \text{str}(a, i, j)$$

Для доказательства 2-й импликации делаем 2-е дедуктивное предположение:

$$\text{IsStr}(\text{len}(a), i, j) = 0$$

Из определения (9.8) и 2-го дедуктивного предположения получаем «рабочий вариант» 2-го дизъюнктивного предположения:

$$\text{len}(a) = \ominus \vee (i \neq 0 \wedge j \neq 0 \wedge \text{len}(a) < i + j - 1)$$

Член дизъюнкции $\text{len}(a) = \ominus$ в данном предложении является отрицанием 1-го дедуктивного предположения, поэтому удаляем его из дизъюнкции. И тогда первое и второе дедуктивные предположения дают общее дедуктивное предположение:

$$\text{len}(a) \neq \ominus \wedge i \neq 0 \wedge j \neq 0 \wedge \text{len}(a) < i + j - 1$$

Из общего дедуктивного предположения нам нужно вывести заключение 2-й импликации:

$\exists u \text{str}(a \cdot u, i, j) \neq \text{str}(a, i, j)$

Для этого положим:

$$u = \text{ss}(i + j - 1 - \text{len}(a))$$

Тогда, учитывая, что по определению $\text{ss}(i)$ верно: $\text{len}(\text{ss}(i)) = i$, получим:

$$\text{len}(a \cdot u) = \text{len}(a) + \text{len}(u) = \text{len}(a) + \text{len}(\text{ss}(i + j - 1 - \text{len}(a)))$$

$$\text{len}(a \cdot u) = \text{len}(a) + i + j - 1 - \text{len}(a) = i + j - 1$$

На основании следствия (с.6.7) которое (напоминание):

$$\text{len}(a) \neq \ominus \wedge i \neq 0 \wedge i + j - 1 \leq \text{len}(a) \Rightarrow \text{len}(\text{str}(a, i, j)) = j$$

Получаем с учётом того, что $\text{len}(a \cdot u) \geq i + j - 1$:

$$\text{len}(\text{str}(a \cdot u, i, j)) = j$$

С другой стороны, из-за (т.4.2) $(i \neq 0 \wedge j \neq 0) \Rightarrow \text{str}(a, i, j) = \text{end}(\text{beg}(a, i + j - 1), i)$ получим:

$$\text{str}(a, i, j) = \text{end}(\text{beg}(a, i + j - 1), i)$$

На основании (т.8.2) $\text{len}(a) \neq \ominus \Rightarrow a = \text{str}(a, 1, \text{len}(a))$ и (т.4.6) $\text{str}(a, 1, j) = \text{beg}(a, j)$ получаем:

$$\text{str}(a, i, j) = \text{end}(\text{beg}(\text{beg}(a, \text{len}(a)), i + j - 1), i)$$

На основании (т.3.1) $\text{beg}(a, \min(i, j)) = \text{beg}(\text{beg}(a, i), j)$ и $\text{len}(a) < i + j - 1$ из общего дедуктивного предположения получаем:

$$\text{str}(a, i, j) = \text{end}(\text{beg}(a, \text{len}(a)), i)$$

Используя снова (т.4.6) $\text{str}(a, 1, j) = \text{beg}(a, j)$ и (т.8.2) $\text{len}(a) \neq \ominus \Rightarrow a = \text{str}(a, 1, \text{len}(a))$ получаем:

$$\text{str}(a, i, j) = \text{end}(a, i)$$

Тогда на основании $i \neq 0$ из общего дедуктивного предположения и следствия (с.6.9), которая (напоминание):

$$\text{len}(a) \neq \ominus \Rightarrow \text{len}(\text{beg}(a, i)) = \min(i, \text{len}(a)) \wedge (\text{len}(\text{end}(a, i)) = \max(\text{len}(a), i - 1) - i + 1 \vee i = 0)$$

Получаем:

$$\text{len}(\text{str}(a, i, j)) = \text{len}(\text{end}(a, i)) = \max(\text{len}(a), i - 1) - i + 1$$

и надо его сравнить с полученным ранее

$$\text{len}(\text{str}(a \cdot u, i, j)) = j$$

При том, что $\text{len}(a) < i + j - 1$ из общего дедуктивного предположения.

Перепишем последнее неравенство так:

$$\text{len}(a) - (i - 1) < j$$

Левая часть $\text{len}(a) - (i - 1)$ может быть и отрицательной, но тогда

$$\max(\text{len}(a), i - 1) - i + 1 = 0 < j$$

А если $\text{len}(a) - (i - 1)$ не отрицательна, то

$$\max(\text{len}(a), i - 1) - i + 1 = \text{len}(a) - (i - 1) < j$$

То есть, в любом случае:

$$\max(\text{len}(a), i - 1) - i + 1 < j$$

$$\text{len}(\text{str}(a, i, j)) < \text{len}(\text{str}(a \cdot u, i, j))$$

$$\text{len}(\text{str}(a, i, j)) \neq \text{len}(\text{str}(a \cdot u, i, j))$$

Тогда на основании теоремы (т.8.5), которая (напоминание):

$\text{len}(a) \neq \emptyset \Rightarrow [a = b \Leftrightarrow \forall i((\text{len}(a) = \text{len}(b) \wedge 0 < i \wedge i \leq \text{len}(a)) \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))]$

Получаем:

$\text{str}(a, i, j) \neq \text{str}(a \cdot u, i, j)$

Поэтому для $u = \text{ss}(i + j - 1 - \text{len}(a))$ верно:

$\text{str}(a, i, j) \neq \text{str}(a \cdot u, i, j)$

То есть, верно – после подстановки – следующее неравенство:

$\text{str}(a, i, j) \neq \text{str}(a \cdot u = \text{ss}(i + j - 1 - \text{len}(a)), i, j)$

В силу аксиомы логики предикатов из раздела 4. Приложение:

(b) $A(a) \Rightarrow \exists x A(x)$

Верно:

$\text{str}(a, i, j) \neq \text{str}(a \cdot u, i, j) \Rightarrow \exists u \text{str}(a, i, j) \neq \text{str}(a \cdot u, i, j)$

И мы можем подставить на место свободных переменных конкретное значение $u = \text{ss}(i + j - 1 - \text{len}(a))$:

$\text{str}(a, i, j) \neq \text{str}(a \cdot \text{ss}(i + j - 1 - \text{len}(a)), i, j) \Rightarrow \exists u \text{str}(a, i, j) \neq \text{str}(a \cdot u, i, j)$

Посылка данной импликации доказана чуть выше, поэтому её можно отбросить по правилу вывода МР. И мы получаем:

$\exists u \text{str}(a, i, j) \neq \text{str}(a \cdot u, i, j)$

Что и требовалось доказать из общего дедуктивного предположения, поэтому импликация 2 тоже доказана.

Теорема доказана.

3 Числа в теории строк

То, что в теории строк мы опираемся в «конструировании» объектов не только на ноль, создаёт свойства для строк сверх тех свойств, которые есть у чисел в теории Пеано. У строк теперь есть собственная структура «цепочка символов». Вопрос – как эту же структуру придать числам, которые используются в теории строк? Это будет та структура, которой нет у чисел арифметики (арифметики Пеано, в частности), но есть у чисел в позиционном представлении.

Что касается чисел, которые использованы в аксиомах теории строк, то они подчиняются обычным свойствам арифметики, и тут мы имеем (в неявном виде пока) ещё аксиомы теории Пеано или её арифметического расширения. Запишем явным образом аксиомы теории Q , достаточной для представления рекурсивных функций, а затем и её расширение до теории Пеано.

Дж. Булос, Р. Джеффри «Вычислимость и логика» Глава 14. Представимость в Q . Аксиомы теории Q .

$$(Q_1) \forall i \forall j (i' = j' \Rightarrow i = j)$$

$$(Q_2) \forall i 0 \neq i'$$

$$(Q_3) \forall i (i \neq 0 \Rightarrow \exists j i = j')$$

$$(Q_4) \forall i i + 0 = i$$

$$(Q_5) \forall i \forall j (i + j' = (i + j)')$$

$$(Q_6) \forall i i \times 0 = 0$$

$$(Q_7) \forall i \forall j (i \times (j') = (i \times j) + i)$$

Замечу, что для знака умножения использован символ « \times », потому что символ « \cdot » уже занят для обозначения конкатенации.

Глава 15 Неразрешимость, неопределимость и полнота. В той же книге. Добавление аксиом индукции к теории Q и переход к теории Пеано – «теории Z ».

$((A(0) \wedge \forall i (A(i) \Rightarrow A(i')))) \Rightarrow \forall i (A(i))$ – все формулы такого вида на языке теории строк, но с навешиванием на такие формулы квантора общности $\forall i$.

Разумеется, для теории строк формула $A(i)$ понимается как формула, которая может зависеть и от разных строковых аргументов, которые не являются числами.

При этом аксиому Q_3 можно исключить, так как она выводится из аксиомы индукции и аксиомы Q_2 .

До данного момента мы разделяли между собой просто строки и строки, которые являются ещё и числами. Вот аксиомы Пеано – это аксиомы, которые выполнены для не просто строк, но для строк со свойствами чисел. Теперь, сохраняя те же условности обозначения для строк, имеющих свойства чисел (для переменных со значением числа мы используем оговоренные ранее буквы) зададим связь между алфавитом для строк и числами.

Чтобы придать числам свойства строк, достаточно задать строку, которая будет 0 (нулём) и задать операцию добавления единицы (функцию следования). После этого остальные арифметические операции легко выводятся на основании свойств строк и аксиом арифметики Пеано.

Вот нужная для «превращения» некоторых строк ещё и в числа аксиома, которую можно при

желании разбить на отдельные аксиомы – отдельная аксиома из каждого члена следующей конъюнкции:

$$\begin{aligned} &(0 \leq l_{NumBeg} < l_{NumLim} \leq l_{ChrLim}) \\ &\wedge 0 = \text{Chr}(l_{NumBeg}) \\ &\wedge (l_{NumBeg} \leq j < l_{NumLim} \Rightarrow (\text{Chr}(j))' = \text{Chr}(j')) \\ &\wedge \text{Chr}(l_{NumLim})' = (\text{Chr}(l_{NumBeg})') \cdot \text{Chr}(l_{NumBeg}) \\ &\wedge ((k \neq 0 \wedge l_{NumBeg} \leq j < l_{NumLim}) \Rightarrow (k \cdot \text{Chr}(j))' = k \cdot \text{Chr}(j')) \\ &\wedge (k \neq 0 \Rightarrow (k \cdot \text{Chr}(l_{NumLim}))' = (k') \cdot \text{Chr}(l_{NumBeg})) \end{aligned}$$

Пояснения:

1. Мы задаём «алфавит в алфавите» - для цифр.
2. Первая цифра в этом алфавите – символ «0». Вот его мы и объявляем числом 0 (ноль) в этой части из приведённой аксиомы:

$$0 = \text{Chr}(l_{NumBeg})$$

Дальше идёт логика добавления 1 в разных случаях.

3. Сначала рассматриваем случай, когда у нас число состоит из любой цифры, кроме последней цифры «алфавита цифр»:

$$l_{NumBeg} \leq j < l_{NumLim} \Rightarrow (\text{Chr}(j))' = \text{Chr}(j')$$

При увеличении такого числа на 1 происходит просто замена прежней цифры на следующую. Например, для десятичной системы:

$$5' = 6 \text{ или } 5 + 1 = 6$$

4. Затем рассматриваем случай, когда у нас число состоит из последней цифры «алфавита цифр». В десятичной системе это – 9:

$$\text{Chr}(l_{NumLim})' = (\text{Chr}(l_{NumBeg})') \cdot \text{Chr}(l_{NumBeg})$$

На примере 9 это правило выглядит так:

$$9' = 10 \text{ или } 9 + 1 = 10$$

Заметим, что тут мы используем строковую операцию конкатенации, соединяя в правой части равенства следующую цифру после 0 (так как $1 = 0'$) и сам 0, который возник вместо 9.

5. Затем рассматриваем случай, когда последняя цифра – не 9 (если разбирать на примере десятичной системы) и она не единственная в числе (усложнение пункта 3). Тогда последняя цифра просто заменяется на следующую в алфавите цифру, когда число увеличено на 1:

$$23345' = 23346$$

В данном примере $k = \langle 2334 \rangle$ и он соединён (конкатенация) с цифрой «5». Условие

$$k \neq 0$$

понятно, потому что числа у нас не имеют вида «05», например. Поэтому разбираемая часть аксиомы имеет вид:

$$(k \neq 0 \wedge l_{NumBeg} \leq j < l_{NumLim}) \Rightarrow (k \cdot \text{Chr}(j))' = k \cdot \text{Chr}(j')$$

6. И в заключение рассматриваем случай, когда последняя цифра в числе является последней цифрой ещё и в алфавите цифр:

$$(k \neq 0 \Rightarrow (k \cdot \text{Chr}(l_{NumLim}))' = (k') \cdot \text{Chr}(l_{NumBeg}))$$

Тогда последняя цифра заменяется на 0 (ноль), а вся предыдущая часть числа заменяется на число, которое больше этой предыдущей части на 1. На примере десятичного числа:

$$123799' = 123800$$

Видно, что число у нас из двух соединённых строк – «12379» и «9». При увеличении такого числа на 1 получается соединение из 2 новых строк – «12380» и «0».

Этой аксиоматизации достаточно для задания всех чисел – от 0 и до бесконечности. Из этой аксиомы и остальных аксиом для строк и арифметики можно уже вывести правила сложения, умножения, вычитания и деления «в столбик» и любые другие правила для операций с числами в позиционном представлении, записанных при помощи строк.

Замечу, что в аксиоме использованы для обозначения чисел те буквы, которые были оговорены для этого в начале предыдущего раздела: i, j, k, l, m, n . Неявно это означает, что в каждой аксиоме имеется посылка о том, что используемый на месте числа объект – число (а не просто строка). Например, полная запись аксиомы (8.1) из предыдущего раздела будет:

$$a = b \Leftrightarrow (\forall i \text{ Num}(i) \Rightarrow \text{str}(a, i, 1) = \text{str}(b, i, 1))$$

Где одноместная предикатная буква $\text{Num}(i)$ как раз «проверяет», является ли данная строка числом. Но раз мы условились об обозначении числа буквой i , то считаем это условие (про то, что тут число) исполненным и посылку с $\text{Num}(i)$ пропускаем:

$$a = b \Leftrightarrow \forall i \text{ str}(a, i, 1) = \text{str}(b, i, 1)$$

Это обычная практика – как, например, в теории множеств принято изображать множества строчными, а классы – прописными буквами, а посылки с проверками на то, что объект – множество – типа $M(x)$ – тогда не пишут.

Тем не менее, одноместная предикатная буква $\text{Num}(i)$ о проверке строки на то, что она ещё и число – имеется в теории. И в связи с аксиомой о связи чисел и строк можно вывести:

$$\text{Num}(a) \Leftrightarrow$$

$$\text{len}(a) \neq \emptyset \wedge (\text{str}(a, 1, 1) = \text{Chr}(l_{\text{NumBeg}}) \Rightarrow a = \text{str}(a, 1, 1))$$

$$\wedge \forall i \leq \text{len}(a) : l_{\text{NumBeg}} \leq \text{Ach}(\text{str}(a, i, 1)) \leq l_{\text{NumLim}}$$

Пояснение: при вышеизложенной аксиоматизации о связи чисел и строк получается, что число представляет собой конечную строку из одних цифр, притом цифра «0» (ноль) может стоять на первом месте только в числе 0 (ноль).

Но представленная аксиоматизация для связи чисел и строк – совершенно не единственно возможная. Мы вполне можем написать аксиомы для представления чисел в духе «стандартной интерпретации». Вот так:

Аксиоматизация нуля:

$$0 = \text{Chr}(0)$$

Аксиоматизация увеличения на 1:

$$i' = i \cdot \text{Chr}(0)$$

Пояснение: Символы с номером 0 (ноль) мы используем как «счётные палочки». И их количество минус 1 и есть само число. А «кучи» этих «счётных палочек» мы организуем при помощи конкатенации.

При такой аксиоматизации можно вывести:

$$i + j = \text{end}(i \cdot j, 0'')$$

$$i - j = \text{end}(i, \text{len}(j)), \text{ для } i > j$$

Ну, а случай

$$i \times j$$

Окажется конкатенацией из $\text{len}(j) - 1$ одинаковых строк $\text{end}(i, 0'')$ и конкатенация $\text{Chr}(0)$ сверх того:

$$\text{end}(i, 0'') \cdot \text{end}(i, 0'') \dots \text{end}(i, 0'') \cdot \text{end}(i, 0'') \cdot \text{Chr}(0)$$

Ну, или конкатенацией из $\text{len}(i) - 1$ строк $\text{end}(j, 0'')$ и конкатенация $\text{Chr}(0)$ сверх того:

$$\text{end}(j, 0'') \cdot \text{end}(j, 0'') \dots \text{end}(j, 0'') \cdot \text{end}(j, 0'') \cdot \text{Chr}(0)$$

Напоминаю, что для знака умножения использован символ « \times », потому что символ « \cdot » уже занят для обозначения конкатенации.

При таких аксиомах связи чисел и строк будет верно:

$$\text{Num}(a) \Leftrightarrow \text{len}(a) \neq \ominus \wedge \text{len}(a) \neq 0 \wedge (\forall i 1 \leq i \leq \text{len}(a) \Rightarrow \text{str}(a, i, 1) = \text{Chr}(0))$$

Как видим, одни и те же аксиомы арифметики допускают представление чисел и в таком виде, когда размер представления числа равен величине числа. Это совершенно не годится для правильной оценки размеров данных и времени работы с ними при рассмотрении алгоритмов, работающих со строками и числами в позиционном представлении. Впрочем, это уже было доказано в предыдущем разделе.

4 Приложение. Логика теорий первого порядка с равенством

В данном разделе приведены аксиомы логики исчисления предикатов, аксиомы равенства и сведения о некоторых базовых теоремах логики. Использована аксиоматика из книги Д. Гильберт, П. Бернайс «Основания математики. Логические исчисления и формализация арифметики». Далее кратко «Основания математики».

Выбранная аксиоматика представляется наиболее интуитивно понятной и удобной для построения логических выводов среди известных мне. К тому же опора на общеизвестный классический труд в значительной мере страхует от каких-то незамеченных читателями неточностей.

«Основания математики», Глава III. Исчисление высказываний. Параграф 3. Дедуктивная логика высказываний. Подпараграф 2. Одна система исходных формул для дедуктивной логики высказываний: полнота этой системы.

I. Формулы для импликации:

- 1) $A \Rightarrow (B \Rightarrow A)$
- 2) $(A \Rightarrow (A \Rightarrow B)) \Rightarrow (A \Rightarrow B)$
- 3) $(A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C))$

II. Формулы для конъюнкции:

- 1) $A \wedge B \Rightarrow A$
- 2) $A \wedge B \Rightarrow B$
- 3) $(A \Rightarrow B) \Rightarrow ((A \Rightarrow C) \Rightarrow (A \Rightarrow B \wedge C))$

III. Формулы для дизъюнкции:

- 1) $A \Rightarrow A \vee B$
- 2) $B \Rightarrow A \vee B$
- 3) $(A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \vee B \Rightarrow C))$

IV. Формулы для эквивалентности:

- 1) $(A \Leftrightarrow B) \Rightarrow (A \Rightarrow B)$
- 2) $(A \Leftrightarrow B) \Rightarrow (B \Rightarrow A)$
- 3) $(A \Rightarrow B) \Rightarrow ((B \Rightarrow A) \Rightarrow (A \Leftrightarrow B))$

V. Формулы для отрицания:

- 1) $(A \Rightarrow B) \Rightarrow (\neg B \Rightarrow \neg A)$
- 2) $A \Rightarrow \neg\neg A$
- 3) $\neg\neg A \Rightarrow A$

«Основания математики», Глава IV. Формализация процесса вывода II: Исчисление предикатов. Параграф 2. Связанные переменные и правила для кванторов. Подпараграф 4. Окончательная формулировка правил исчисления предикатов; изображение форм категорических суждений; случай пустой индивидуальной области.

В качестве исходных формул разрешается брать тождественные формулы исчисления высказываний. К ним мы добавляем обе основные формулы:

- (a) $\forall x A(x) \Rightarrow A(a)$
- (b) $A(a) \Rightarrow \exists x A(x)$

Тут я от себя добавлю пояснение, что аксиома (b) – это бесконечное множество импликаций. И мы можем выбирать ту, которая удобна нам. Например, при доказательстве существования для (5.3) в подразделе II раздела 2 мы взяли как бы $A(l'_{ChrLim})$, за счёт чего воспользовались правилом силлогизма и эта посылка исчезла из итоговой формулы. Но за счёт аксиомы (b) мы получаем возможность выбрать путь (значение переменной a), который позволяет достичь нужной цели.

В качестве схем вывода, позволяющих получить новые формулы из ранее полученных, у нас имеется, наша первоначальная схема заключения <MP>

Если верно U и верно $U \Rightarrow R$, то верно R .

И кроме того, две новые схемы:

(α) Если верно $U \Rightarrow B(a)$, то верно $U \Rightarrow \forall x B(x)$

и

(β) Если верно $B(a) \Rightarrow U$, то верно $\exists x B(x) \Rightarrow U$

Обе они могут применяться лишь тогда, когда в первой формуле схемы переменная a встречается лишь на местах, указанных в качестве аргумента, а x не входит в $B(a)$.

«Основания математики», Глава V. Исчисление предикатов с равенством. Полнота одноместного исчисления предикатов:

(J_1) $a = a$

(J_2) $a = b \Rightarrow (A(a) \Rightarrow A(b))$

Где $A(a)$ представляет собой любую логически корректную формулу, построенную из функциональных букв, предикатных букв теории, кванторов (в область действия которых не попадают a, b) и логических связок. При этом заменять a на b можно как везде, где написано a , так и в некоторых (произвольно выбранных) местах написания переменной a .

Теперь перечислим некоторые ключевые теоремы про логику первого порядка с равенством.

VI. Ключевые теоремы

1) Теорема дедукции

Простой вариант:

Если при добавлении к теории T утверждения A можно вывести утверждение B без использования правил вывода (α) и (β), то в теории T без добавления утверждения A можно вывести утверждение $A \Rightarrow B$.

Но мы использовали и более тонкий вариант этой теоремы:

Использовать правила вывода (α) и (β) можно, но только так, чтобы при этом не связывалась на одна свободная переменная утверждения A .

Или, записывая более формально:

Если есть вывод $T, A \vdash B$ без использования (α) и (β) в отношении свободных переменных утверждения A , то есть вывод $T \vdash A \Rightarrow B$.

2) Вложенные посылки эквивалентны конъюнкции посылок

Утверждение:

$A_1 \Rightarrow (A_2 \Rightarrow \dots \Rightarrow (A_N \Rightarrow B) \dots)$

Эквивалентно утверждению:

$$(A_1 \wedge A_2 \wedge \dots \wedge A_N) \Rightarrow B$$

Поэтому последовательность посылок во вложенных импликациях можно менять произвольным образом.

3) Правило С. Смотри, например, Э. Мендельсон, «Введение в математическую логику», Глава 2. Теория первого порядка. Раздел 7. Правило С.

Если у нас в теории T имеется истинное утверждение вида $\exists x A(x)$, то есть, имеется вывод:

$$T \vdash \exists x A(x)$$

И имеется ещё такой вывод:

$$T, A(u) \vdash B, \text{ без использования } (\alpha) \text{ и } (\beta) \text{ по } u,$$

То найдётся вывод:

$$T \vdash B$$

4) Определение равенства. Смотри, например, Э. Мендельсон, «Введение в математическую логику», Глава 2. Теории первого порядка. Раздел 8. Теория первого порядка с равенством. Предложение 2.26. <критерий того, что в теории первого порядка истинны аксиомы равенства>

Если в аксиомах теории есть только функциональные буквы и знак равенства, то аксиомы равенства (J_1) и (J_2) для неё будут выполнены тогда и только тогда, когда

а. Для каждой функциональной буквы $f_n()$ с аргументами x_1, \dots, x_m (где m – количество аргументов функции $f_n(x_1, \dots, x_m)$) истинно:

$$(x_1 = y_1 \wedge \dots \wedge x_m = y_m) \Rightarrow (f_n(x_1, \dots, x_m) = f_n(y_1, \dots, y_m))$$

б. Для знака равенства истинна рефлексивность:

$$x = x$$

с. Для знака равенства истинна транзитивность:

$$x = y \Rightarrow (z = x \Rightarrow z = y)$$

5) Введение новых функциональных букв и предметных констант. Смотри, например, Э. Мендельсон, «Введение в математическую логику», Глава 2. Теории первого порядка. Раздел 9. Введение новых функциональных букв и предметных констант. Следствие 3.3.

Если для формулы $A(u, x_1, \dots, x_m)$ выполнено:

$$\exists_1 u A(u, x_1, \dots, x_m)$$

То можно аксиоматизировать функцию $f(x_1, \dots, x_m)$ следующей формулой:

$$A(f(x_1, \dots, x_m), x_1, \dots, x_m)$$

И добавление данной аксиомы к теории не добавляет никакой дополнительной логики, кроме нового обозначения. В частности, такая функция будет соответствовать аксиомам равенства, если данная теория до введения этого обозначения была теорией первого порядка с равенством.

Напоминание – обозначение «существует и единственно»

$\exists_1 u A(u, x_1, \dots, x_m)$ означает выполнение 2 условий:

а. $\exists u A(u, x_1, \dots, x_m)$

б. $A(u_1, x_1, \dots, x_m) \wedge A(u_2, x_1, \dots, x_m) \Rightarrow u_1 = u_2$

6) Эквивалентные формулы логики высказывания взаимозаменяемы внутри любых формул логики. Например, если доказана формула

$$A \Rightarrow (\neg B \vee C)$$

И при этом верно:

$$B \Leftrightarrow D$$

То будет доказана и формула:

$$A \Rightarrow (\neg D \vee C)$$

$$7) \neg \forall n \neg A(n) \Leftrightarrow \exists n A(n)$$

Доказательство.

Ветка 1

$$\forall n \neg A(n) \Rightarrow \neg A(i), \text{ по (a) } \forall x A(x) \Rightarrow A(a)$$

$$A(i) \Rightarrow \neg \forall n \neg A(n), \text{ из предыдущего, контрапозиция}$$

$$\exists n A(n) \Rightarrow \neg \forall n \neg A(n), \text{ из предыдущего по } (\beta) \text{ Если верно } B(a) \Rightarrow U, \text{ то верно } \exists x B(x) \Rightarrow U$$

Ветка 2

$$\neg A(i) \Rightarrow \exists n \neg A(n), \text{ по (b) } A(a) \Rightarrow \exists x A(x)$$

$$\neg \exists n \neg A(n) \Rightarrow A(i), \text{ из предыдущего, контрапозиция}$$

$$\neg \exists n \neg A(n) \Rightarrow \forall n A(n), \text{ из предыдущего по } (\alpha) \text{ Если верно } U \Rightarrow B(a), \text{ то верно } U \Rightarrow \forall x B(x)$$

$$\neg \forall n A(n) \Rightarrow \exists n \neg A(n), \text{ из предыдущего, контрапозиция}$$

$\neg \forall n \neg A(n) \Rightarrow \exists n A(n)$, из предыдущего, заменяя $A(n)$ на $\neg A(n)$ и $\neg(\neg A(n))$ на $A(n)$ в соответствии с п.6)

Соединяя ветку 1 и 2 получим:

$$\neg \forall n \neg A(n) \Leftrightarrow \exists n A(n)$$

$$8) \forall i (B \vee A(i)) \Leftrightarrow B \vee \forall i (A(i))$$

Доказательство

Ветка 1.

$$A(n) \Rightarrow \exists i A(i)$$

$$B \wedge A(n) \Rightarrow B \wedge \exists i A(i)$$

$$\exists i (B \wedge A(n)) \Rightarrow B \wedge \exists i A(i)$$

$$\neg B \vee \neg \exists i A(i) \Rightarrow \neg \exists i (B \wedge A(n))$$

$$\neg B \vee \forall i \neg A(i) \Rightarrow \forall i \neg (B \wedge A(n))$$

$$\neg B \vee \forall i \neg A(i) \Rightarrow \forall i (\neg B \vee \neg A(n))$$

$$B \vee \forall i A(i) \Rightarrow \forall i (B \vee A(n))$$

Ветка 2.

$$\forall i (B \vee A(i)) \Rightarrow (B \vee A(n))$$

$$\forall i (B \vee A(i)) \Rightarrow (\neg B \Rightarrow A(n))$$

$$[\neg B \wedge \forall i (B \vee A(i))] \Rightarrow A(n)$$

$$[\neg B \wedge \forall i (B \vee A(i))] \Rightarrow \forall i A(i)$$

$$\forall i (B \vee A(i)) \Rightarrow (\neg B \Rightarrow \forall i A(i))$$

$$\forall i(B \vee A(i)) \Rightarrow (B \vee \forall i A(i))$$

Доказано.

$$\mathbf{9)} \exists i(B \vee A(i)) \Leftrightarrow B \vee \exists i A(i)$$

Доказательство.

Ветка 1.

$$\forall i A(i) \Rightarrow A(n)$$

$$B \wedge \forall i A(i) \Rightarrow B \wedge A(n)$$

$$B \wedge \forall i A(i) \Rightarrow \forall i(B \wedge A(i))$$

$$\neg \forall i(B \wedge A(i)) \Rightarrow \neg B \vee \neg \forall i A(i)$$

$$\exists i \neg(B \wedge A(i)) \Rightarrow \neg B \vee \exists i \neg A(i)$$

$$\exists i(\neg B \vee \neg A(i)) \Rightarrow \neg B \vee \exists i \neg A(i)$$

$$\exists i(B \vee A(i)) \Rightarrow B \vee \exists i A(i)$$

Ветка 2.

$$A(n) \Rightarrow B \vee A(n)$$

$$B \Rightarrow B \vee A(n)$$

$$B \vee A(n) \Rightarrow \exists i(B \vee A(i))$$

$$A(n) \Rightarrow \exists i(B \vee A(i))$$

$$B \Rightarrow \exists i(B \vee A(i))$$

$$\exists i A(u) \Rightarrow \exists i(B \vee A(i))$$

$$B \vee \exists i A(u) \Rightarrow \exists i(B \vee A(i))$$

Доказано.

$$\mathbf{10)} (A \vee B) \Leftrightarrow (A \vee (\neg A \wedge B))$$

Пояснение.

Данная эквивалентность часто используется, когда доказываются импликации вида:

$$(A_1 \vee A_2 \vee \dots \vee A_n) \Rightarrow D$$

Для доказательства тогда надо доказывать импликации вида:

$$A_1 \Rightarrow D$$

$$A_2 \Rightarrow D$$

...

$$A_n \Rightarrow D$$

А после этого использовать аксиому III.3) $(A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \vee B \Rightarrow C))$ несколько раз.

Но зачастую в приведённой цепочке импликаций вместо импликации $A_n \Rightarrow D$ доказывают импликацию:

$$(\neg A_1 \wedge \neg A_2 \wedge \dots \wedge A_n) \Rightarrow D$$

И то, что стоит слева от A_n в последней импликации – это как раз $\neg A$, если считать, что $(A_1 \vee A_2 \vee \dots \vee A_{n-1})$ – это A , а A_n – это B из доказываемой нами формулы VI.10. Поэтому и происходит подмена этого B в импликации $B \Rightarrow D$ на $(\neg A \wedge B)$, что вместо исходной посылки:

$$(A_1 \vee A_2 \vee \dots \vee A_n)$$

Можно заменить её на эквивалентную:

$$(A_1 \vee A_2 \vee \dots \vee (\neg A_1 \wedge \neg A_2 \wedge \dots \wedge A_n))$$

И такая замена оказывается возможной как раз благодаря данной теореме VI.10.

Данную импликацию VI.10 можно проверить по таблицам истинности, можно дать логический вывод, но поясню, почему эта эквивалентность истинная – третьим способом – через «булеву алгебру».

$A \vee B$ эквивалентно $A \vee (\neg A \wedge B)$ по следующей причине:

$$B \Leftrightarrow (A \vee \neg A) \wedge B \Leftrightarrow (A \wedge B) \vee (\neg A \wedge B)$$

Поэтому:

$$(A \vee B) \Leftrightarrow A \vee (A \wedge B) \vee (\neg A \wedge B)$$

Но так как для A тоже верно:

$$A \Leftrightarrow (A \wedge B) \vee (A \wedge \neg B)$$

То

$$A \Leftrightarrow A \wedge (A \wedge B)$$

Ведь $A \vee (A \wedge B)$ ничем не отличается от $(A \wedge B) \vee (A \wedge \neg B) \vee (A \wedge B)$, где последний член просто повторяет первый, а дизъюнкция первых двух – эквивалентна A .

A в силу последней эквивалентности верно:

$$A \vee (A \wedge B) \vee (\neg A \wedge B) \Leftrightarrow A \vee (\neg A \wedge B)$$

Откуда и получаем искомое:

$$(A \vee B) \Leftrightarrow A \vee (\neg A \wedge B)$$

Теорема доказана.

Список литературы

- [1] *Дж. Булос, Р. Джеффри.* Вычислимость и логика. Мир, М. 1994.
- [2] *Д. Гильберт, П. Бернайс.* Основания математики. Логические исчисления и формализация арифметики Наука, М. 1979.
- [3] *Э. Мендельсон.* Введение в математическую логику Наука, М. 1984