

# Домашние задания по курсу «Математическая логика»

ИТМО, группы М3236..М3239

Осень 2016 г.

## Общие замечания

Для всех программ кодировка входных и выходных файлов должна быть либо CP1251, либо UTF8.

## Задача 1. Проверка вывода

Написать программу, проверяющую вывод  $\gamma_1, \dots, \gamma_n \vdash \alpha$  в исчислении высказываний на корректность. Входной файл соответствует следующей грамматике:

$$\begin{aligned} \langle \text{файл} \rangle & ::= \langle \text{заголовок} \rangle \backslash \mathbf{n} \{ \langle \text{выражение} \rangle \backslash \mathbf{n} \}^* \\ \langle \text{заголовок} \rangle & ::= [ \langle \text{выражение} \rangle \{ ', ' \langle \text{выражение} \rangle \}^* ] \backslash \text{'-'} \langle \text{выражение} \rangle \\ \langle \text{выражение} \rangle & ::= \langle \text{дизъюнкция} \rangle \mid \langle \text{дизъюнкция} \rangle \text{'->'} \langle \text{выражение} \rangle \\ \langle \text{дизъюнкция} \rangle & ::= \langle \text{конъюнкция} \rangle \mid \langle \text{дизъюнкция} \rangle \text{'\|'} \langle \text{конъюнкция} \rangle \\ \langle \text{конъюнкция} \rangle & ::= \langle \text{отрицание} \rangle \mid \langle \text{конъюнкция} \rangle \text{'\&'} \langle \text{отрицание} \rangle \\ \langle \text{отрицание} \rangle & ::= ( \text{'A'} \dots \text{'Z'} ) \{ \text{'A'} \dots \text{'Z'} \mid \text{'0'} \dots \text{'9'} \}^* \mid \text{'!'} \langle \text{отрицание} \rangle \mid ( \text{'('} \langle \text{выражение} \rangle \text{'})'} \end{aligned}$$

В первой строке входного файла (заголовок) перечислены предположения  $\gamma_i$  (этот список может быть пустым) и доказываемое утверждение  $\alpha$ . В последующих строках указаны формулы, составляющие вывод формулы  $\alpha$ . Пробелы, символы табуляции и возврата каретки (ASCII-код 13<sub>10</sub>) должны игнорироваться. Символ  $\backslash$  имеет ASCII-код 124<sub>10</sub>.

Результатом работы программы должен быть файл с проаннотированным текстом доказательства, в котором первая строка — это заголовок из входного файла, каждая же последующая строка — соответствующая строка из вывода, расширенная в соответствии с грамматикой:

$$\begin{aligned} \langle \text{строка} \rangle & ::= ( \text{'('} \langle \text{номер} \rangle \text{'})'} \text{' ' } \langle \text{выражение} \rangle \text{' ('} \langle \text{аннотация} \rangle \text{'})'} \\ \langle \text{аннотация} \rangle & ::= \text{'Сх. акс. ' } \langle \text{номер} \rangle \\ & \quad \mid \text{'Предп. ' } \langle \text{номер} \rangle \\ & \quad \mid \text{'М.Р. ' } \langle \text{номер} \rangle \text{' , ' } \langle \text{номер} \rangle \\ & \quad \mid \text{'Не доказано'} \\ \langle \text{номер} \rangle & ::= \{ \text{'0'} \dots \text{'9'} \}^+ \end{aligned}$$

Выражение не должно содержать пробелов, номер от выражения и выражение от аннотации должны отделяться одним пробелом. Выражения в доказательстве должны нумероваться подряд натуральными числами с 1. Если выражение  $\delta_n$  получено из  $\delta_i$  и  $\delta_j$ , где  $\delta_j \equiv \delta_i \rightarrow \delta_n$  путём применения правила Modus Ponens, то аннотация должна выглядеть как 'М.Р.  $i, j$ ', обратный порядок номеров не допускается.

Уделите внимание производительности: ваша программа должна проверять доказательство в 5000 выражений (общим объемом 1Мб) на Intel Core i5-2520M (2.5 GHz) за несколько секунд.

## Задача 2. Исчисление предикатов

Написать программу, проверяющую доказательство в формальной арифметике на корректность, а также преобразующую вывод  $\Gamma, \alpha \vdash \beta$  в вывод  $\Gamma \vdash \alpha \rightarrow \beta$

Грамматика входного файла:

$$\begin{aligned} \langle \text{Файл} \rangle & ::= \langle \text{заголовок} \rangle \backslash \mathbf{n} \langle \text{доказательство} \rangle \\ \langle \text{заголовок} \rangle & ::= [ \{ \langle \text{выражение} \rangle \text{' , ' } \}^* \langle \text{выражение} \rangle ] \backslash \text{'-'} \langle \text{выражение} \rangle \\ \langle \text{доказательство} \rangle & ::= \{ \langle \text{выражение} \rangle \backslash \mathbf{n} \}^* \end{aligned}$$

$$\begin{aligned}
\langle \text{выражение} \rangle &::= \langle \text{дизъюнкция} \rangle \mid \langle \text{дизъюнкция} \rangle \text{'-'} \langle \text{выражение} \rangle \\
\langle \text{дизъюнкция} \rangle &::= \langle \text{конъюнкция} \rangle \mid \langle \text{дизъюнкция} \rangle \text{'|'} \langle \text{конъюнкция} \rangle \\
\langle \text{конъюнкция} \rangle &::= \langle \text{унарное} \rangle \mid \langle \text{конъюнкция} \rangle \text{'\&'} \langle \text{унарное} \rangle \\
\langle \text{унарное} \rangle &::= \langle \text{предикат} \rangle \mid \text{'!' } \langle \text{унарное} \rangle \mid \text{'(' } \langle \text{выражение} \rangle \text{' )'} \\
&\quad \mid \text{'@'} \mid \text{'?' } \langle \text{переменная} \rangle \langle \text{унарное} \rangle \\
\langle \text{переменная} \rangle &::= \text{'a' } \dots \text{'z'} \{ \text{'0'} \dots \text{'9'} \}^* \\
\langle \text{предикат} \rangle &::= \text{'A'} \dots \text{'Z'} \{ \text{'0'} \dots \text{'9'} \}^* [ \text{'(' } \langle \text{терм} \rangle \{ \text{' , ' } \langle \text{терм} \rangle \}^* \text{' )'} ] \\
&\quad \mid \langle \text{терм} \rangle \text{'=' } \langle \text{терм} \rangle \\
\langle \text{терм} \rangle &::= \langle \text{слагаемое} \rangle \mid \langle \text{терм} \rangle \text{'+' } \langle \text{слагаемое} \rangle \\
\langle \text{слагаемое} \rangle &::= \langle \text{умножаемое} \rangle \mid \langle \text{слагаемое} \rangle \text{'*'} \langle \text{умножаемое} \rangle \\
\langle \text{умножаемое} \rangle &::= \text{'a'} \dots \text{'z'} \{ \text{'0'} \dots \text{'9'} \}^* \text{'(' } \langle \text{терм} \rangle \{ \text{' , ' } \langle \text{терм} \rangle \}^* \text{' )'} \\
&\quad \mid \langle \text{переменная} \rangle \mid \text{'(' } \langle \text{терм} \rangle \text{' )'} \\
&\quad \mid \text{'0'} \mid \langle \text{умножаемое} \rangle \text{' ' }
\end{aligned}$$

Символ '@' обозначает квантор всеобщности, символ '?' — квантор существования. Грамматика не различает переменные и константы; мы можем предполагать, что константы в формуле — это свободные переменные.

Входной файл может содержать некорректный вывод — а именно, некоторые переходы в выводе могут быть некорректными; входной файл по-прежнему синтаксически корректен. В этом случае программа должна выдать текст вида:

Вывод некорректен начиная с формулы номер <N>[: <ошибка>]

где вместо <N> подставлен номер первой некорректной формулы. Необязательное поле <ошибка> должно появляться, если формула не является аксиомой или допущением, не может быть выведена из предыдущих, но являлась бы (или могла бы быть выведена), если бы не нарушение ограничений на переменные. Возможные варианты ошибок:

терм <X> не свободен для подстановки в формулу <Y> вместо переменной <a>.

переменная <a> входит свободно в формулу <X>.

используется <правило|схема аксиом> с квантором по переменной <a>, входящей свободно в допущение <X>.

Выходной файл — либо измененное доказательство утверждения  $\Gamma \vdash \alpha \rightarrow \beta$ , при этом вывод должен соответствовать грамматике для ввода, либо сообщение об ошибке. Если в заголовке доказательства отсутствуют гипотезы, то требуется выдать исходное доказательство.

### Задача 3. Неравенства

Написать программу, по двум целым числам  $a$  и  $b$  ( $0 \leq a, b \leq 100$ ), указанным во входном файле, строящую доказательство:

- Для групп 38-39:  $\bar{a} \leq \bar{b}$  или  $-\bar{a} \leq \bar{b}$ , в зависимости от соотношения  $a$  и  $b$  (знак «меньше или равно» раскрывается естественно — как  $\exists p(a + p = b)$ ).
- Для групп 36-37:  $(\bar{a} + \bar{1}) \cdot (\bar{a} + \bar{1}) = \bar{a} \cdot \bar{a} + 2 \cdot \bar{a} + \bar{1}$

Доказательство должно быть записано в соответствии с грамматикой выходного файла из задачи 2.