5.3. Определение равносильности логических формул

Составить программу, проверяющую равносильность (эквивалентность) двух заданных формул алгебры логики. В формулах используются логические константы TRUE и FALSE, логические переменные и логические операции: отрицания (¬), конъюнкции (∧), дизъюнкции (∨), импликации (→) и эквиваленции (≡), а также круглые скобки. В качестве имен переменных могут быть взяты произвольные идентификаторы. Например, логической формулой является запись a1 → ¬ (b1 ∧¬ a2 ∨ c) ≡ (c ∧ b2).

Порядок выполнения операций в формуле алгебры логики определяется согласно общепринятому приоритету логических операций и записанным скобкам. Операция эквиваленции ≡ имеет меньший приоритет, чем операция импликации →, а импликация – меньший, чем операции отрицания, конъюнкции и дизъюнкции.

В логической формуле могут быть опущены незначащие скобки. Пара скобок считается *незначащей* (избыточной), если после её удаления получается формула, равносильная исходной.

Две логические формулы называются *равносильными*, если для любого набора значений входящих в них переменных значения этих формул совпадают (т.е. они реализуют одну и ту же логическую функцию). Например, равносильны формулы x → y и ¬x ∨ y.

Логические формулы и реализуемые ими функции могут содержать фиктивные (несущественные) переменные. Переменная x_k является *фиктивной* для функции $f(x_1, \dots, x_n)$, $n \geq 1$, $k \leq n$, если для любого набора логических значений $a_1, \dots, a_{k-1}, a_{k+1}, \dots, a_n$

$$f(a_1,\dots,a_{k-1},0,a_{k+1},\dots,a_n) \equiv f(a_1,\dots,a_{k-1},1,a_{k+1},\dots,a_n)$$

Например, фиктивной является переменная x для функции, реализуемой формулой (¬ x ∨ x) ≡ y ∧ z.

Программа проверки равносильности формул выполняет:

- ввод исходных логических формул с клавиатуры или из файла (направление ввода определяется командами пользователя или выясняется в диалоге с ним);
- проверку синтаксической правильности введённых формул и выдачу диагностических сообщений в случае обнаружения синтаксических ошибок;
- проверку равносильности двух синтаксически правильных логических формул;

- нахождение для каждой из двух формул всех фиктивных переменных и их вывод.

В ходе выполнения задания необходимо описать синтаксис формул алгебры логики в виде БНФ-правил.

Рефал-программа должна выявлять и диагностировать следующие виды синтаксических ошибок:

- нарушение баланса открывающихся и закрывающихся скобок;
- недопустимая операция;
- пропуск операции или операнда;
- неверная запись логической константы или переменной.

5.8. Методические указания к вариантам

Во всех вариантах рекомендуется выделить следующие предварительные этапы обработки исходных символьных выражений – *лексический* и *синтаксический* анализ.

Задача лексического анализа – выделение лексем исходного выражения и, возможно, перевод их во внутреннее представление, в котором:

- числовые константы (последовательности цифр) преобразованы в символы-цифры или макроцифры;
- имена переменных (последовательности букв и цифр, начинающиеся с буквы) заключены в структурные скобки или преобразованы в символ-метки;
- имена функций (`sin`, `cos` и др. в варианте дифференцирования выражения) или служебные имена (`begin`, `end`, `case`, `integer` и др. в вариантах интерпретации и трансляции паскаль-программы) преобразованы в соответствующие символы-метки (`/sin/`, `/cos/`, `/begin/` и т.п.);
- знаки операций, состоящие из нескольких символов, заменены на соответствующие им символы-метки (например, в варианте вычисления выражения языка С: знаки операции ++ можно заменить на символ-метку `/pp/`, а знаки *= на символ-метку `/mulassign/`);
- унарные знаки + и – заменены на другие символы – для того, чтобы отличать их от знаков бинарных операций + и –.

Основная задача синтаксического анализа – выявление структуры обрабатываемого выражения и перевод его во *внутреннее представление*, удобное для дальнейших преобразований. Для этого:

- необходимо заменить в исходном выражении символы круглых скобок на структурные скобки;
- в варианте вычисления выражения языка С в ходе расстановки структурных скобок целесообразно считать знаки тернарной операции ? и : особым видом парных скобок и заменить каждую пару таких

символов на пару структурных скобок и символов-меток, например, `/question/` и `/two-spot/`, или же заключить в структурные скобки каждую пару символов `?:` вместе со стоящим между ними выражением;

- в вариантах интерпретации и трансляции паскаль-программы после замены обычных круглых скобок на структурные следует зафиксировать вложенность операторов паскаль-программы расстановкой структурных скобок вокруг фрагментов операторов, начинающихся и заканчивающихся соответственно символами-метками `/begin/` и `/end/`, `/then/` и `/else/`, `/case/` и `/endcase/`, `/repeat/` и `/until/` (введёнными на этапе лексического анализа).

Таким образом, в ходе лексического и синтаксического анализа входное символьное выражение будет преобразовано во внутреннее представление, упрощающее его последующую обработку (дифференцирование, вычисление, интерпретацию, трансляцию и т.п. – в зависимости от варианта задания).

Для хранения значений переменных в ходе вычислений целесообразно использовать копилку.