

Математическая логика. Логика предикатов

Занятие кружка по информатике ФИМЛИ 5

г. Долгопрудный, МО
Грицуляк РТ,
30 сентября 2016

Базовые операции

- Как обсуждали, наличие элемента в множестве — можно представить битом.
- Разберем битовую арифметику подробнее.

$a \wedge b$ — операция И (and): $1 \wedge 1 = 1; 1 \wedge 0 = 0 \wedge 1 = 0 \wedge 0 = 0$

в C++: $a \& b \equiv a \wedge b$ в математических обозначениях

$a \vee b$ — операция ИЛИ (or): $1 \vee 1 = 1; 1 \vee 0 = 0 \vee 1 = 1; 0 \vee 0 = 0$

в C++: $a | b \equiv a \vee b$

- Исключающее ИЛИ(XOR): $a \wedge b$ в C++
 $1 \wedge 1 == 0 \wedge 0 == 0$; $1 \wedge 0 == 0 \wedge 1 == 1$
- \sim дополнение (отрицание);
- \gg сдвиг вправо
- \ll сдвиг влево

Задачи

- Написать умножение на 2 с использованием целых беззнаковых (без использования + - * / с использованием if, for, и операций с логикой) (1 *)
- Сложить два целых числа (с ограничением предыдущей задачи) (2 *)
- Умножить (3 *)
- Разделить (4 *)

```

struct book{
    string book_name;
    string book_id;
    vector <string> references;
    book(string name,string id, vector<string> ref):
        book_name(name),book_id(id),references(ref)
    {}
    string to_string() {
        auto ret = book_name + ";" + book_id + ";";
        for(auto val:references) ret+=val+";";
        return ret;
    }
};

```

```

int main(int argc, char ** argv)
{
    vector <book> library;
    vector<string> references;

    book book1("book1","ISDN-book1",references);
    library.push_back(book1);
    references.push_back("ISDN-book1");
    references.push_back("ISDN-catalog");

    book catalog1("catalog1","ISDN-
catalog",references);

    library.push_back(catalog1);
}

```

Каталоги

- Нельзя
- Допустим можно. Тогда этот каталог должен входить сам в себя, чего быть не может по определению каталога.
- То есть нельзя построить множество множеств построенных по принципу каталогов (выше) так чтобы в нем были все множества, какие то будут выпадать.
- Можно исключить, или выделить в отдельный тип.

Исчисление высказываний

- Изучение этих проблем привело к вычислению высказываний.
- Формальная система исчисления высказываний описывается:
 $\{L, F, A, R\}$
- L — алфавит
- F — формулы
- A — аксиомы
- R — правила
- Для A и R будем пока использовать интуитивное понимание

L — алфавит, состоит из

- {символы} — переменные
- $\&, \vee, \rightarrow, \sim, \leftrightarrow$

где $A \leftrightarrow B : \neg(A \wedge B)$

$A \rightarrow B$: Из A выводимо B

Скобки ()

Формулы

- Если A — формула, то $A \& B$ формула
- Если A и B формулы — то и $A \& B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$, $\sim A$ тоже формулы
- Примеры
 - $(x \wedge y)$
 - $x \rightarrow (y \& z)$

Логика предикатов

- Рассмотрим некоторые понятия из исчисления высказываний и логики предикатов.

- Для тех кому прошлые задачи простые:

Библиографические каталоги — это книги, которые описывают другие книги. Некоторые каталоги могут описывать другие каталоги. Некоторые каталоги могут описывать даже сами себя. Можно ли составить каталог, всех каталогов, которые не описывают сами себя?

-

Задача:

А Тавтологично В — если формула для всех значений истинна.

- Простая (1*) Реализовать функцию $\text{arrow}(a,b)$ реализующую операцию $a \rightarrow b$
- Средняя (3*) Для формул из 10 переменных проверить, являются ли они тавтологиями
- Сложная (5*) Тавтология - Написать генератор более коротких тавтологий для формул.

Подсказка

- Использовать таблицы истинности
- Пример таблицы:

A	B	C	F(A,B,C)
0	0	1	1
0	1	0	0
			ИТД

Исчисление предикатов

В алфавит добавим:

- $P_n(x_1, x_2, \dots, x_n)$ — предикатные буквы ($n=0, 1, \dots$)
- $\forall \exists$

В формулы:

- $P_n(x_1, \dots, x_n)$ — элементарные (атомарные формулы)
- Если $A(x)$ — формула со свободной

– То $\forall x A(x), \exists x A(x)$ — формулы

Задача

- Написать проверку, является ли строка утверждением в исчислении высказываний.
- Принять что алфавит ограничен x_1, x_2 ,

Использовать символы

– @ и ? вместо - $\forall \exists$

Вопросы и надеюсь ответы

- roman.gritsulyak@gmail.com