Конспект по дискретной математике

October 1, 2019

1 Преобразование Мёбиуса

$$f(x_1\dots x_n) \underset{\alpha_1\dots\alpha_n\in A_f}{\oplus} x_1^{\alpha_1}\dots x_n^{\alpha_n}$$

$$x_i^1=x_i, x_i^0=1$$

$$f(\dots)=0 \Leftrightarrow \exists x_i: x_i=0, \alpha_i=1$$

$$f(\dots)=1 \Leftrightarrow \forall i: x_i=1 \ or \ \alpha_i=0 \Leftrightarrow \overline{x}\succeq \overline{\alpha}$$

$$a_{\alpha_1\dots\alpha_n}=\begin{cases} 1,\alpha_1\dots\alpha_n\in A_f\\ 0,\text{иначе} \end{cases}$$

Определение. Доминирование - частничый порядок на \mathbb{B}^n , такой что $\forall i \ x_i \geq y_i$. Обозначается $x_1 \dots x_n \succeq y_1 \dots y_n$.

$$f(x_1 \dots x_n) = \bigoplus_{\alpha_1 \dots \alpha_n} a_{\alpha_1 \dots \alpha_n} x_1^{\alpha_1} \dots x_n^{\alpha_n} = \bigoplus_{\alpha_1 \dots \alpha_n : X \succeq \alpha} a_{\alpha_1 \dots \alpha_n}$$
$$a_{\alpha_1 \dots \alpha_n} \mapsto f(\overline{x})$$
$$f(x_1 \dots x_n) \mapsto a_{\alpha_1 \dots \alpha_n}$$

Теорема 1. Преобразование Мёбиуса: $a_{\alpha_1\dots\alpha_n}=\bigoplus\limits_{\alpha_1\dots\alpha_n:\alpha\succeq X}f(x_1\dots x_n)-\mathbb{B}^{2^n}\to\mathbb{B}^{2^n}$

$$xy$$
 $x\lor y$ 00 0 $\alpha_{00}=0$ Пример $(x\lor y)$: 01 1 $\alpha_{01}=1$ 10 1 $\alpha_{10}=1$ 11 0 $\alpha_{11}=1$

Доказательство. Индукция по числу 1 в $\alpha_1 \dots \alpha_n$.

База:
$$\alpha_{0...0} = f(0...0)$$

Переход:
$$\bigoplus_{x:\alpha\succeq x} f(x_1\dots x_n) = \bigoplus_{x:\alpha\succeq x} \bigoplus_{\beta:x\succeq\beta} a_{\beta_1\dots\beta_n} = a_{\alpha_1\dots\alpha_n} \oplus \bigoplus_{\substack{x,\beta\\ \alpha\succeq x\succeq\beta\\ \text{неверно}:\alpha=x=\beta}} a_{\beta_1\dots\beta_n} = a_{\alpha_1\dots\alpha_n}$$

Определение. Инволюция — преобразование, тождественное себе обратному.

Преобразование Мёбиуса является инволюцией.

Можно заметить, что при записи через некий базис функций от множества переменных, формулы получаются длинными. Поэтому существуют другие формы записи — схемы из функциональных элементов и линейные программы.

2 Схемы из функциональных элементов

Записывается в форме ациклического ориентированного графа — direct acyclic graph — dag.

Теорема 2. Если G - ациклический ориентированный граф, тогда его вершинам можно присвоить номера так, что $uv \in E \Rightarrow \#u < \#v$. Это называется топологическая сортировка

Лемма 1. Ациклический граф содержит вершину, из которой не выходят ребра.

Определение. Схема из функциональных элементов над базисом F — ациклический ориентированный граф, у которого есть n выделенных вершин, в которые ничего не входит, помеченные "вход" и есть одна вершина, из которой ничего не выходит, помеченная "выход". Остальные вершины называются внутренние, они — функции $f \in F$, в которые входят k ребер, каждое из которых помечено числом от 1 до k.

Чтобы построить таблицу истинности схемы из функциональных элементов, проводится слеюущий порядок действий:

- 1. Поместим во входные узлы значения аргументов схемы.
- 2. В остальные узлы в порядке топологической сортировки помещаются значения, равные значению функции, соответствующей этому узлы, если принять значения из входных узлов как аргументы данной функции.
- 3. Значение в выходном узле будет значением схемы.

Самый правый узел - выход.

Глубина:
$$d(u) = \max_{V:V \to u} (d(u) + 1)$$

Можно заметить, что с помощью схемы можно использовать промежуточные значения несколько раз, этим эта форма записи лучше формулы.

M3137y2019

Определение. Сложность функции f в базисе B — минимальный размер схемы, которая вычисляет эту функцию, обозначается $size_B(f) = \min\{size(C) | C$ вычисл. $f, C \in B\}$. $size_{\vee,\wedge,\neg}(\oplus_2) < 5$ $size_{\wedge,\oplus,1}(\oplus_2)=1$ **Теорема 3**. Пусть B_1 и B_2 - базисы. Тогда $\exists c>0$, которая зависит от этих двух базисов и больше не зависит ни от чего, что $\forall f: size_{B_1}(f) \leq c \cdot size_{B_2}(f)$ Таким образом, выбор базиса для сложности меняет только константу, т.е. ассимптотически сложность функции одинакова во всех базисах. **Лемма 2**. Если B - базис, то $\forall f$ можно задать схемой из функциональных элементов над B. Доказательство. Построим схему через дерево обхода - поставим стрелки вверх и объединим все входы. Доказательство. Построим для f оптимальную схему в B_2 . $B_2 = \{g_1, g_2 \dots g_t\}$. Рассмотрим схемы для функций из B_2 над базисом B_1 . Для каждой этой функции построим такую схему C_i . $C := \max size(C_i)$. Вместо каждой функции подставим соответствующую схему C.

Доказательство. Глубиной функции f в базисе B называется минимальная глубина схемы, которая вычисляет f, обозначается $depth_B(f) = \min\{depth(C)|C$ вычисл. $f, C \in A$

 U_2 - множество всех функций от 2 аргументов. $T_2 = U_2 \setminus \{ \oplus_{\cdot} = \}$

B}.