Московский Государственный
Университет им. М.В.Ломоносова

Механико-математический факультет

Вероятностные алгоритмы вычисления однократных формул

КУРСОВАЯ РАБОТА

студента 3-го курса
каф. Математической Логики и Теории Алгоритмов
Евфимьевского Александра Валентиновича

Научный руководитель:

д. ф.-м. н. Николай Константинович Верещагин

Москва, 1995 г.

1 Основные определения

Монотонная формула (в дальнейшем просто *формула*) F — дерево, в листьях которого находятся переменные, а в развилках — знаки $\vee$ (дизъюнкция) и $\wedge$ (конъюнкция). *Однократная формула* — такая формула, в которую каждая переменная входит не более одного раза. *Бинарная формула* содержит развилки только с двумя входами. Выход у любой развилки всегда один — в сторону корня.

Переменные в листьях могут принимать значения из множества $\mathbf{B} = \{0,1\}$. Если всем листьям формулы приписать некоторые значения, то можно вычислить значение формулы, двигаясь от листьев к корню и вычисляя конъюнкции и дизъюнкции. Ясно, что результат не зависит от порядка подсчета.

Направленным алгоритмом (F,μ) будем называть формулу F, где каждой развилке приписано распределение вероятностей на множестве перестановок ее входов. Т.е. если развилка A имеет n входов, то ей должен соответствовать набор чисел $\mu^A = (\mu^A_\sigma \mid \sigma : \{1 \ldots n\} \leftrightarrow \{1 \ldots n\})$, где $\forall\sigma\ \mu^A_\sigma \in [0,1] \subset \mathbf{R}$ и $\sum_\sigma \mu^A_\sigma = 1$. Число μ^A_σ назовем *вероятностью* перестановки σ в развилке A.

Вычисление с помощью направленного алгоритма.

Пусть всем переменным в листьях формулы приписаны некоторые значения из множества $\mathbf{B}$. Вычисления выполняются машиной Тьюринга с генератором случайных чисел заданного распределения вероятностей. Определение ведем по индукции: пусть необходимо найти значение формулы F; если она состоит из одного листа, то нужно запросить значение переменной в листе и выдать его. Если F имеет вид развилки (дизъюнкции или конъюнкции) A с n входами, где на входах находятся подформулы $F_1, F_2, \ldots, F_n$, которые мы уже умеем вычислять по индукции, нужно, воспользовавшись генератором случайных чисел, выбрать перестановку σ чисел $1, 2, \ldots, n$ с распределением вероятностей на перестановках μ^A. Затем вычислять значения $F_{\sigma(1)}, F_{\sigma(2)}, \ldots$ до тех пор, пока не определится значение F (для дизъюнкции: пока не попадется $F_{\sigma(i)} = 1$ или пока не будут вычислены все $F_{\sigma(i)}$; для конъюнкции — пока не попадется 0).

2 Постановка задачи

Предположим, что получение значения переменной в листе — операция дорогостоящая, и нам бы хотелось опрашивать как можно меньше переменных. Обозначив значения листьев-переменных как $x_1, x_2, \ldots, x_N$, а стоимость вопроса к ним — $c(1), c(2), \ldots, c(N)$, определим *сложность направленного алгоритма* (F,μ) *при входе* $\bar{x}$ как математическое ожидание (по всем μ^A как независимым распределениям) суммы стоимостей опрошенных при вычислении переменных, т.е.

$$C^F(\mu,\bar{x}) := \sum_{(\sigma^A)} \left(\sum_{i-\text{опрош.}} c(i) \right) \prod_A \mu^A_{\sigma^A}\ .$$

Далее, естественно определить *сложность направленного алгоритма* (F,μ) как его сложность при наихудшем входе, а *сложность формулы* F — как минимальную сложность направленного алгоритма для этой формулы:

$$C^F(\mu) := \max_{\bar{x}} C^F(\mu,\bar{x}); \quad C^F := \min_{(F,\mu)} C^F(\mu).$$

Несколько обобщив ситуацию, можно считать, что стоимость вопроса к листу-переменной различна в случае, если переменная равна 0 или 1 ($c_0(i) \neq$

$\neq c_1(i)$). Из дальнейшего станет ясной естественность такого обобщения. Обозначив через $F(\bar{x})$ значение формулы F при значениях переменных $\bar{x}$, введем определения *0-* и *1-сложности* формул и алгоритмов следующим образом:

$$C_0^F(\mu) := \max_{F(\bar{x})=0} C^F(\mu,\bar{x}); \qquad C_0^F := \min_{(F,\mu)} C_0^F(\mu);$$
$$C_1^F(\mu) := \max_{F(\bar{x})=1} C^F(\mu,\bar{x}); \qquad C_1^F := \min_{(F,\mu)} C_1^F(\mu).$$

Возникает вопрос: как по заданной формуле F за полиномиальное время найти направленный алгоритм (F,μ), имеющий минимальную сложность (т.е. такой, что $C^F(\mu) = C^F$)? Для бинарных однократных формул этот вопрос был решен в работе [M.Saks, A.Wigderson, Probabilistic boolean decision trees and the complexity of evaluating game trees]; здесь же будет изложено решение его для произвольных однократных формул путем сведения к случаю бинарных.

3 Теорема о сложности однократных формул

Теорема 1 *Пусть F — однократная формула с корнем-развилкой A, имеющей n входов с подформулами $F_1, F_2, \ldots, F_n$; пусть также для $\forall i = 1, 2, \ldots, n$ задан направленный алгоритм (F_i,μ_i) такой, что $C_0^{F_i} = C_0^{F_i}(\mu_i)$ и $C_1^{F_i} = C_1^{F_i}(\mu_i)$. Тогда, чтобы найти направленный алгоритм (F,μ) такой, что $C_0^F = C_0^F(\mu)$ и $C_1^F = C_1^F(\mu)$, достаточно взять $\mu = (\mu_1, \mu_2, ..., \mu_n, \mu^A)$, где $(\hat{A},\mu^A)$ — направленный алгоритм, в котором формула $\hat{A}$ состоит из одной развилки A и n листьев-переменных со стоимостями $c_0(i) = C_0^{F_i}$, $c_1(i) = C_1^{F_i}$, а μ^A — любое распределение вероятностей на A, минимизирующее сложность $C^{\hat{A}}(\mu^A)$.*

Доказательство. Без ограничения общности будем считать, что в развилке A стоит дизъюнкция (для конъюнкции 0 и 1 поменяются ролями). В этом случае вычисление по направленному алгоритму (F,μ) на входе $\bar{x}$ начинается с получения случайной перестановки σ чисел $1,2,\ldots,n$ с использованием распределения вероятностей μ^A; затем вычисляются алгоритмы $(F_{\sigma(1)},\mu_{\sigma(1)}),(F_{\sigma(2)},\mu_{\sigma(2)}),\ldots$ до тех пор, пока не получится $F_{\sigma(i)}(\bar{x}) = 1$ (тогда $F(\bar{x}) = 1$) или пока не выяснится, что все $F_{\sigma(i)}(\bar{x}) = 0$, $i = 1,2,\ldots,n$ (тогда $F(\bar{x}) = 0$). Можно заметить, что если $F(\bar{x}) = 0$, то сложность алгоритма (F,μ) при входе $\bar{x}$ не зависит от распределения μ^A (какое бы σ не выпало, считать все равно все (F_i,μ_i)). Отсюда следующие выражения для сложности ($\bar{x}_i$ — значения переменных, входящих в F_i):

$$F(\bar{x}) = 0 \quad \to \quad C^F(\mu,\bar{x}) = \sum_{i=1}^{n} C^{F_i}(\mu_i,\bar{x}_i);$$

$$F(\bar{x}) = 1 \quad \to \quad C^F(\mu,\bar{x}) = \sum_{\substack{\sigma:\{1,\ldots,n\}\leftrightarrow \\ \leftrightarrow\{1,\ldots,n\}}} \mu^A_\sigma \left(\sum_{i=1}^{\substack{\text{до первого } i, \text{ что} \\ F_{\sigma(i)}(\bar{x}_{\sigma(i)})=1 \text{ вкл.}}} C^{F_{\sigma(i)}}(\mu_{\sigma(i)},\bar{x}_{\sigma(i)}) \right).$$

Найдем теперь 0- и 1-сложность алгоритма (F,μ). Их нужно выразить через 0- и 1-сложности (F_i,μ_i):

$$\begin{aligned} C^F_0(\mu) &= \max_{F(\bar{x})=0} \sum_{i=1}^{n} C^{F_i}(\mu_i,\bar{x}_i) = \max_{\forall i\, F_i(\bar{x}_i)=0} \sum_{i=1}^{n} C^{F_i}(\mu_i,\bar{x}_i) = \\ &= \sum_{i=1}^{n} \max_{F_i(\bar{x}_i)=0} C^{F_i}(\mu_i,\bar{x}_i) = \sum_{i=1}^{n} C^{F_i}_0(\mu_i). \end{aligned} \tag{1}$$

Мы использовали здесь однократность формулы F (когда проносили максимум за сумму); видно, что 0-сложность (F,μ) не зависит от μ^A. Аналогичное рассуждение применимо и в случае 1-сложности ($\bar{y} := (y_1,\ldots,y_n)$):

$$\begin{aligned} C^F_1(\mu) &= \max_{F(\bar{x})=1} \sum_{\sigma} \mu^A_\sigma \left(\sum_{i=1}^{\substack{\text{до первого } i, \text{ что} \\ F_{\sigma(i)}(\bar{x}_{\sigma(i)})=1 \text{ вкл.}}} C^{F_{\sigma(i)}}(\mu_{\sigma(i)},\bar{x}_{\sigma(i)}) \right) = \\ &= \max_{\bar{y}\neq\bar{0}} \max_{\forall i\, F_i(\bar{x}_i)=y_i} \sum_{\sigma} \mu^A_\sigma \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ y_{\sigma(i)}=1 \text{ вкл.}}} C^{F_{\sigma(i)}}(\mu_{\sigma(i)},\bar{x}_{\sigma(i)}) \right) = \end{aligned}$$

$$
= \max_{\bar{y}\neq\bar{0}} \sum_{\sigma} \mu^A_\sigma \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ y_{\sigma(i)}=1 \text{ вкл.}}} \max_{F_{\sigma(i)}(\bar{x}_{\sigma(i)})=y_{\sigma(i)}} C^{F_{\sigma(i)}}(\mu_{\sigma(i)}, \bar{x}_{\sigma(i)}) \right) =
$$

$$
= \max_{\bar{y}\neq\bar{0}} \sum_{\sigma} \mu^A_\sigma \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ y_{\sigma(i)}=1 \text{ вкл.}}} C^{F_{\sigma(i)}}_{y_{\sigma(i)}}(\mu_{\sigma(i)}) \right). \tag{2}
$$

При каких $\mu = (\mu^A, \mu_1, \ldots, \mu_n)$ 0- и 1-сложность алгоритма (F, μ) принимают наименьшее значение? Из формулы (1) следует, что $C^F_0(\mu) = C^F_0$ тогда и только тогда, когда все $C^{F_i}_0(\mu_i) = C^{F_i}_0$, причем распределение μ^A не играет роли; что же касается 1-сложности, то, так как все $C^{F_i}_{0,1}(\mu_i)$ входят в нее как слагаемые с неотрицательными коэффициентами, мы заведомо минимизируем ее, если возьмем μ_i, при которых одновременно минимизируются все эти сложности, а затем найдем оптимальное μ^A. Условие теоремы предоставляет нам возможность получить такие μ_i.

Итак, пусть μ_i такие, что $\forall i\ C^{F_i}_0(\mu_i) = C^{F_i}_0, C^{F_i}_1(\mu_i) = C^{F_i}_1$. 0-сложность алгоритма (F, μ) минимизируется автоматически; чтобы одновременно минимизировать и 1-сложность, нужно найти

$$
\mu^A = \arg\min_{\mu^A} \max_{\bar{y}\neq\bar{0}} \sum_{\sigma} \mu^A_\sigma \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ y_{\sigma(i)}=1 \text{ вкл.}}} C^{F_{\sigma(i)}}_{y_{\sigma(i)}} \right). \tag{3}
$$

Но точно такая же формула (3) возникает, если мы будем искать оптимальный направленный алгоритм вида $(\hat{A}, \mu^A)$ для формулы $\hat{A}$, состоящей из одной развилки A с листьями-переменными $y_1, y_2, \ldots, y_n$ и их стоимостями $c_0(i) = C^{F_i}_0$, $c_1(i) = C^{F_i}_1$. Таким образом, направленный алгоритм (F, μ), составленный из алгоритмов $(\hat{A}, \mu^A)$ и $(F_i, \mu_i), i = 1, \ldots, n$, окажется оптимальным одновременно по 0- и 1-сложности.

Теорема доказана. ■

Задача о нахождении оптимального алгоритма (одновременно по 0- и 1-сложности) для произвольной однократной формулы F теперь сведена к задаче о нахождении такого алгоритма для только одной развилки: умея оптимизировать развилки, можно, двигаясь от листьев к корню, найти все требуемые распределения вероятностей на перестановках. Ясно, что алгоритм (F, μ), оптимальный одновременно по 0- и 1-сложности, оптимален и по просто сложности: пусть, например, $C^F_0 \le C^F_1$; всякий другой алгоритм (F, μ') с той же формулой будет такой, что $C^F_1 \le C^F_1(\mu')$; так как, с одной стороны, $C^F(\mu) = C^F_1(\mu) = C^F_1$, а с другой, $C^F_1(\mu') \le C^F(\mu')$, то имеем $C^F(\mu) \le C^F(\mu')$, что и требовалось.

В следующих разделах будет изучаться задача об оптимизации одной развилки. Без ограничения общности будем считать ее дизъюнкцией. Далее все формулы предполагаются однократными.

4 Развилка с двумя входами

Рассмотрим формулу $\hat{A}$, состоящую из одной развилки-дизъюнкции A с двумя входами-переменными x_1 и x_2, имеющими стоимости соответственно C_0^1, C_1^1 и C_0^2, C_1^2. Если μ^A — распределение вероятностей на перестановках входов этой развилки, то $\mu^A = (\alpha, 1-\alpha)$, где $0 \le \alpha \le 1$; α — вероятность перестановки $(1,2)$, $1-\alpha$ — вероятность $(2,1)$. Найдем теперь сложности алгоритма $(\hat{A}, \mu^A)$. Возможные значения переменных x_1, x_2 таковы: $00, 01, 10, 11$. Из них 00 дает на выходе 0, а остальные — 1. 0-сложность определяется очень просто:

$$C_0(\alpha) = C(\alpha, 00) = C_0^1 + C_0^2 = C_0.$$

Чтобы выписать сложности на входах 01, 10 и 11, рассуждаем так: при входе 01 вход x_2 будет опрошен в любом случае, так как другого входа с 1 нет; вход же x_1 будет опрошен только в случае выпадения перестановки $(1,2)$, а ее вероятность $= \alpha$; значит, математическое ожидание $C(\alpha, 01) = \alpha C_0^1 + C_1^2$. Аналогичные рассуждения для других входов приводят к формулам:

$$\begin{cases} C(\alpha,01) &=& \alpha C_0^1 + C_1^2; \\ C(\alpha,10) &=& (1-\alpha)C_0^2 + C_1^1; \\ C(\alpha,11) &=& \alpha C_1^1 + (1-\alpha)C_1^2. \end{cases}$$

По определению, $C_1 = \min\limits_{0\le\alpha\le1} \max\{C(\alpha,01), C(\alpha,10), C(\alpha,11)\}$; ясно, что $C(\alpha,11)$ из максимума можно выбросить: при $\alpha \in [0,1]$ $C(\alpha,11) \le \max\{C_1^1, C_1^2\} \le \max\{C(\alpha,10), C(\alpha,01)\}$. Функции $C(\alpha,01)$ и $C(\alpha,10)$ линейны по α и могут быть расположены на отрезке $\alpha \in [0,1]$ следующими тремя способами:

1. Не пересекаются; везде на отрезке $C(\alpha,01) < C(\alpha,10)$. Этот случай эквивалентен условию $C_1^2 + C_0^1 < C_1^1$; будем говорить, что вход x_1 *перетягивает* вход x_2, и обозначать $x_1 \gg x_2$. $C_1(\alpha) = C(\alpha,10)$, оптимальное $\alpha = 1$, т.е. перестановка $(1,2)$ с вероятностью 1. 1-сложность формулы $\hat{A}$: $C_1 = C_1^1$.

2. Не пересекаются; везде на отрезке $C(\alpha,10) < C(\alpha,01)$. Этот случай эквивалентен условию $C_1^1 + C_0^2 < C_1^2$; будем говорить, что вход x_2 *перетягивает* вход x_1, и обозначать $x_1 \ll x_2$. $C_1(\alpha) = C(\alpha,01)$, оптимальное $\alpha = 0$, т.е. перестановка $(2,1)$ с вероятностью 1. 1-сложность формулы $\hat{A}$: $C_1 = C_1^2$.

3. Точка пересечения $\tilde\alpha$ лежит на отрезке $[0,1]$. Оптимальное $\alpha=\tilde\alpha$, 1-сложность формулы $\hat A$: $C_1=C(\tilde\alpha,01)=C(\tilde\alpha,10)\ge\max\{C_1^1,C_1^2\}$. Нетрудно посчитать, что в этом случае
$$C_1=\frac{C_0^1C_0^2+C_0^1C_1^1+C_0^2C_1^2}{C_0^1+C_0^2}. \tag{4}$$

Итак, случай бинарной развилки-дизъюнкции полностью разобран. Для конъюнкции надо всего лишь везде заменить 0 на 1 и 1 на 0. Теперь попробуем свести любую развилку к бинарным, не увеличивая сложность.

5 Нижняя оценка на 1-сложность дизъюнкции

Пусть $\hat A$ — формула из одной развилки-дизъюнкции A с n входами x_1, $x_2,\dots,x_n$ стоимости C_0^i,C_1^i, $i=1,\dots,n$. 0- и 1-сложность самой формулы выражаются следующим образом:
$$C_0=\sum_{i=1}^n C_0^i;\quad C_1=\min_{(\mu_\sigma)}\max_{\bar x\ne\bar 0}\sum_\sigma\mu_\sigma\left(\sum_{i=1}^{\substack{\text{до перв. } i:\\ x_{\sigma(i)}=1\text{ вкл.}}} C^{\sigma(i)}_{x_{\sigma(i)}}\right).$$

Лемма 1 *Верна следующая нижняя оценка для C_1:*
$$C_1\ge\frac{\sum\limits_{i=1}^n C_0^iC_1^i+\sum\limits_{i=1}^n\sum\limits_{j<i}C_0^iC_0^j}{\sum\limits_{i=1}^n C_0^i}. \tag{5}$$

Доказательство. Достаточно доказать, что для любого распределения (μ_σ) выполняется неравенство
$$\max_{\bar x\ne\bar 0}\sum_\sigma\mu_\sigma\left(\sum_{i=1}^{\substack{\text{до перв. } i:\\ x_{\sigma(i)}=1\text{ вкл.}}} C^{\sigma(i)}_{x_{\sigma(i)}}\right)\ge\frac{\sum\limits_{i=1}^n C_0^iC_1^i+\sum\limits_{i=1}^n\sum\limits_{j<i}C_0^iC_0^j}{\sum\limits_{i=1}^n C_0^i}.$$

Возьмем произвольное (μ_σ) и проведем некоторые упрощения:
$$\max_{\bar x\ne\bar 0}\sum_\sigma\mu_\sigma\left(\sum_{i=1}^{\substack{\text{до перв. } i:\\ x_{\sigma(i)}=1\text{ вкл.}}} C^{\sigma(i)}_{x_{\sigma(i)}}\right)\ge\max_{\substack{\bar x=(0\dots1_i\dots0)\\ i=1,\dots,n}}\sum_\sigma\mu_\sigma\left(\sum_{k=1}^{\substack{\text{до перв. } k:\\ x_{\sigma(k)}=1\text{ вкл.}}} C^{\sigma(k)}_{x_{\sigma(k)}}\right)=$$

$$= \max_{i=1,\dots,n} \sum_{\sigma} \mu_\sigma \left(C_1^i + \sum_{k=1}^{k<\sigma^{-1}(i)} C_0^{\sigma(k)} \right) = \max_{i=1,\dots,n} \left(C_1^i + \sum_{\sigma} \mu_\sigma \sum_{k=1}^{k<\sigma^{-1}(i)} C_0^{\sigma(k)} \right) \geq^1$$

$$\geq \frac{1}{\sum_{i=1}^{n} C_0^i} \sum_{i=1}^{n} C_0^i \left(C_1^i + \sum_{\sigma} \mu_\sigma \sum_{k=1}^{k<\sigma^{-1}(i)} C_0^{\sigma(k)} \right) =$$

$$= \frac{1}{\sum_{i=1}^{n} C_0^i} \left(\sum_{i=1}^{n} C_0^i C_1^i + \sum_{\sigma} \mu_\sigma \left(\sum_{i=1}^{n} \sum_{k=1}^{k<\sigma^{-1}(i)} C_0^i C_0^{\sigma(k)} \right) \right).$$

Покажем теперь, что для любой перестановки σ чисел $1, 2, \dots, n$ верно

$$\sum_{i=1}^{n} \sum_{k=1}^{k<\sigma^{-1}(i)} C_0^i C_0^{\sigma(k)} = \sum_{i=1}^{n} \sum_{j<i} C_0^i C_0^j. \tag{6}$$

Достаточно показать для $\forall i_0, j_0 \in \{1, .., n\}$, $i_0 \neq j_0$, что в левую часть входит одно и только одно из двух слагаемых $C_0^{i_0} C_0^{j_0}$, $C_0^{j_0} C_0^{i_0}$. Если не входит ни одно из них, то $\sigma^{-1}(j_0) \geq \sigma^{-1}(i_0)$ и $\sigma^{-1}(i_0) \geq \sigma^{-1}(j_0)$, т.е. они равны и $i_0 = j_0$. Если же входят оба слагаемых, то $\sigma^{-1}(j_0) < \sigma^{-1}(i_0)$ и $\sigma^{-1}(i_0) < \sigma^{-1}(j_0)$, что невозможно. Тем самым равенство (6) доказано, и

$$\frac{1}{\sum_{i=1}^{n} C_0^i} \left(\sum_{i=1}^{n} C_0^i C_1^i + \sum_{\sigma} \mu_\sigma \left(\sum_{i=1}^{n} \sum_{k=1}^{k<\sigma^{-1}(i)} C_0^i C_0^{\sigma(k)} \right) \right) =$$

$$= \frac{1}{\sum_{i=1}^{n} C_0^i} \left(\sum_{i=1}^{n} C_0^i C_1^i + \sum_{\sigma} \mu_\sigma \left(\sum_{i=1}^{n} \sum_{j<i} C_0^i C_0^j \right) \right) = \frac{\sum_{i=1}^{n} C_0^i C_1^i + \sum_{i=1}^{n} \sum_{j<i} C_0^i C_0^j}{\sum_{i=1}^{n} C_0^i}.$$

Лемма доказана. ■

Оценка (5) иногда достигается: например, в случае бинарной развилки без перетяжек (когда $x_1 \not\lll x_2$ и $x_2 \not\lll x_1$, см. разд. 4). Во всяком случае, если некоторый алгоритм для развилки имеет 1-сложность, равную нижней оценке, то он заведомо оптимален.

[1]так как

$$\nu_i > 0 \;\Rightarrow\; \max\{a_1, \dots, a_n\} \geq \frac{\nu_1 a_1 + \dots + \nu_n a_n}{\nu_1 + \dots + \nu_n}$$

Чтобы построить оптимальный алгоритм для развилки-дизъюнкции с n входами, мы будем заменять развилку на некоторую однократную формулу из бинарных дизъюнкций, имеющую ту же 1-сложность, что и исходная развилка (0-сложность всегда будет равна сумме 0-сложностей входов). Любая замена развилки-дизъюнкции на однократную формулу из дизъюнкций эквивалентна ограничению на множество распределений вероятностей на перестановках входов этой развилки. В самом деле, выбрав определенные перестановки на всех развилках формулы, можно очевидным образом найти порядок опроса всех переменных, а значит, соответствующую перестановку исходной развилки; взяв некоторый алгоритм на формуле, можно найти соответствующее распределение вероятностей на исходной развилке. Если мы научимся заменять произвольную развилку на бинарную однократную формулу из одних дизъюнкций той же 1-сложности, то мы ответим на вопрос работы, т.к. тогда, пользуясь теоремой 1, мы сможем быстро строить оптимальный алгоритм.

6 Бинарная формула без перетяжек

Лемма 2 *Пусть F — бинарная однократная формула с одними дизъюнкциями и без перетяжек, т.е. для любой развилки A с входами-подформулами F_1 и F_2 выполняются неравенства:*

$$C_1^{F_2} + C_0^{F_1} \geq C_1^{F_1}, \quad C_1^{F_1} + C_0^{F_2} \geq C_1^{F_2}.$$

Тогда 1-сложность формулы F равна нижней оценке (5).

Доказательство. По индукции: пусть A — бинарная развилка-дизъюнкция с входами F_1 и F_2; F_1 — подформула с переменными $x_1, \ldots, x_m$, F_2 — подформула с переменными $x_{m+1}, \ldots, x_n$, обе подформулы удовлетворяют лемме:

$$C_1^{F_1} = \frac{\sum\limits_{i=1}^{m} C_0^i C_1^i + \sum\limits_{i=1}^{m} \sum\limits_{j<i} C_0^i C_0^j}{\sum\limits_{i=1}^{m} C_0^i}, \quad C_1^{F_2} = \frac{\sum\limits_{i=m+1}^{n} C_0^i C_1^i + \sum\limits_{i=m+1}^{n} \sum\limits_{m<j<i} C_0^i C_0^j}{\sum\limits_{i=m+1}^{n} C_0^i}.$$

Пользуясь теоремой 1 и формулой (4) (верной вследствие отсутствия перетягивания), получим:

$$C_1^F = \frac{C_0^{F_1} C_0^{F_2} + C_0^{F_1} C_1^{F_1} + C_0^{F_2} C_1^{F_2}}{C_0^{F_1} + C_0^{F_2}} =$$

$$= \frac{\left(\sum_{i=1}^{m} C_0^i\right)\left(\sum_{i=m+1}^{n} C_0^i\right) + \sum_{i=1}^{m} C_0^i C_1^i + \sum_{i=1}^{m}\sum_{j<i} C_0^i C_0^j + \sum_{i=m+1}^{n} C_0^i C_1^i + \sum_{i=m+1}^{n}\sum_{m<j<i} C_0^i C_0^j}{\sum_{i=1}^{m} C_0^i + \sum_{i=m+1}^{n} C_0^i} =$$

$$= \frac{\sum_{i=1}^{n} C_0^i C_1^i + \sum_{i=m+1}^{n}\sum_{j=1}^{m} C_0^i C_0^j + \sum_{i=1}^{m}\sum_{j<i} C_0^i C_0^j + \sum_{i=m+1}^{n}\sum_{m<j<i} C_0^i C_0^j}{\sum_{i=1}^{n} C_0^i} =$$

$$= \frac{\sum_{i=1}^{n} C_0^i C_1^i + \sum_{i=1}^{n}\sum_{j<i} C_0^i C_0^j}{\sum_{i=1}^{n} C_0^i},$$ что и требовалось доказать. ■

Из этой леммы следует, что если развилку удается заменить бинарной формулой из дизъюнкций без перетяжек, то цель достигнута: оптимальный алгоритм для этой формулы будет оптимальным и для развилки.

7 Некоторые свойства однократных формул из одних дизъюнкций

Лемма 3 *Пусть A — развилка-дизъюнкция с n входами $x_1, \ldots, x_n$ и сложностью C_0, C_1, а A' — дизъюнкция с $m < n$ входами и сложностью C_0', C_1', причем стоимости у входов A' те же, что и у первых m входов A. Тогда $C_1 \geq C_1'$.*

Доказательство. Надо получить следующее неравенство ($\bar{y} \in \mathbf{B}^m$):

$$\min_{(\mu_\sigma)} \max_{\bar{x} \neq \bar{0}} \sum_{\sigma} \mu_\sigma \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ x_{\sigma(i)}=1 \text{ вкл.}}} C_{x_{\sigma(i)}}^{\sigma(i)} \right) \geq \min_{(\mu'_{\sigma'})} \max_{\bar{y} \neq \bar{0}} \sum_{\sigma'} \mu'_{\sigma'} \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ y_{\sigma'(i)}=1 \text{ вкл.}}} C_{y_{\sigma'(i)}}^{\sigma'(i)} \right).$$

Для этого достаточно доказать, что $\forall (\mu_\sigma)\ \exists (\mu'_{\sigma'})\ \forall \bar{y} \neq 0\ \exists \bar{x} \neq 0$

$$\sum_{\sigma} \mu_\sigma \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ x_{\sigma(i)}=1 \text{ вкл.}}} C_{x_{\sigma(i)}}^{\sigma(i)} \right) \geq \sum_{\sigma'} \mu'_{\sigma'} \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ y_{\sigma'(i)}=1 \text{ вкл.}}} C_{y_{\sigma'(i)}}^{\sigma'(i)} \right). \tag{7}$$

Возьмем произвольное распределение вероятностей (μ_σ) на перестановках σ входов $x_1, \ldots, x_n$. Распределение вероятностей $(\mu'_{\sigma'})$ на перестановках σ' входов $y_1, \ldots, y_m$ можно получить путем сложения некоторых μ_σ друг с другом, а именно: $\mu'_{\sigma'} = \sum\limits_{\sigma \in \Xi(\sigma')} \mu_\sigma$, где $\Xi(\sigma')$ — множество таких перестановок σ, что если написать последовательность $\sigma(1), \sigma(2), \ldots, \sigma(n)$ и вычеркнуть числа, большие m, то получится последовательность $\sigma'(1)$, $\sigma'(2), \ldots, \sigma'(m)$. Ясно, что $\Xi(\sigma')$ — разбиение σ на непересекающиеся подмножества и что построенное $(\mu'_{\sigma'})$ является распределением вероятностей.

Пусть теперь дан набор значений $\bar{y}$ переменных развилки A'. Соответствующий набор значений $\bar{x}$ для A выглядит так: $\bar{x} = (y_1, \ldots, y_m, 0, \ldots, 0)$. Докажем теперь неравенство (7). Обозначим

$$\tilde{C}^i_{0,1} = \begin{cases} C^i_{0,1}, & i \le m \\ 0, & i > m \end{cases} \qquad (\text{Ясно, что } \tilde{C}^i_{0,1} \le C^i_{0,1})$$

и проведем преобразования:

$$\sum_\sigma \mu_\sigma \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ x_{\sigma(i)}=1 \text{ вкл.}}} C^{\sigma(i)}_{x_{\sigma(i)}} \right) \ge \sum_\sigma \mu_\sigma \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ x_{\sigma(i)}=1 \text{ вкл.}}} \tilde{C}^{\sigma(i)}_{x_{\sigma(i)}} \right) =$$

$$= \sum_{\sigma'} \left(\sum_{\sigma \in \Xi(\sigma')} \mu_\sigma \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ x_{\sigma(i)}=1 \text{ вкл.}}} \tilde{C}^{\sigma(i)}_{x_{\sigma(i)}} \right) \right) =^2 \sum_{\sigma'} \left(\sum_{\sigma \in \Xi(\sigma')} \mu_\sigma \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ y_{\sigma'(i)}=1 \text{ вкл.}}} C^{\sigma'(i)}_{y_{\sigma'(i)}} \right) \right) =$$

$$= \sum_{\sigma'} \mu'_{\sigma'} \left(\sum_{i=1}^{\substack{\text{до перв. } i: \\ y_{\sigma'(i)}=1 \text{ вкл.}}} C^{\sigma'(i)}_{y_{\sigma'(i)}} \right), \text{ что и требовалось. } \blacksquare$$

Следствие 1 *Пусть A — развилка-дизъюнкция с n входами $x_1, \ldots, x_n$, F — однократная формула из одних дизъюнкций с теми же листьями-переменными $x_1, \ldots, x_n$, в корне у которой бинарная развилка B с входами-подформулами F_1 и F_2, причем F_1 перетягивает F_2. Пусть также F_1 есть одна развилка с входами-переменными или формула, 1-сложность которой равна 1-сложности соответствующей развилки. Тогда: $C^A_1 = C^F_1 = C^{F_1}_1$.*

---

[2]Выражение $\sum\limits_{i=1}^{\substack{\text{до перв. } i: \\ x_{\sigma(i)}=1 \text{ вкл.}}} \tilde{C}^{\sigma(i)}_{x_{\sigma(i)}}$ зависит только от того, в каком порядке при перечислении $\sigma(i)$ попадаются числа, $\le m$, т.е. от того, какому $\Xi(\sigma')$ принадлежит σ.

Доказательство. По доказанной лемме 1-сложность формулы F_1 меньше или равна 1-сложности развилки A, т.к. F_1 столь же 1-сложна, сколь и развилка с некоторыми входами из A. Вследствие перетяжки 1-сложность F есть 1-сложность F_1; мы имеем теперь:

$$C_1^F = C_1^{F_1} \leq C_1^A.$$

С другой стороны, 1-сложность A не превышает 1-сложности F, т.к. всякий набор перестановок для F соответствует некоторой перестановке входов A, и всякий направленный алгоритм для F есть некоторое распределение вероятностей для A; F — ограниченное A. Отсюда:

$$C_1^A \leq C_1^F = C_1^{F_1} \leq C_1^A \quad \Rightarrow \quad C_1^A = C_1^F = C_1^{F_1}.$$

Следствие доказано. ■

Докажем теперь несколько свойств отношения перетягивания ($\ll$). Будем обозначать формулу, состоящую из бинарной дизъюнкции формул F_1 и F_2, как $F_1 \vee F_2$.

Лемма 4 *Пусть $\mathcal{F}$ — множество формул из дизъюнкций, причем ни одна переменная не встречается в формулах из $\mathcal{F}$ дважды. Тогда отношение $\ll$ задает частичный порядок на $\mathcal{F}$, и выполняется свойство:*

$$F_1 \gg F_2 \gg F_3 \quad \Rightarrow \quad F_1 \gg (F_2 \vee F_3).$$

Доказательство. Пусть сложности формулы $F_i \in \mathcal{F}$ равны C_0^i, C_1^i. Условие $F_1 \gg F_2$ эквивалентно неравенству $C_1^1 > C_1^2 + C_0^1$. Ясно, что не бывает вместе $F_1 \gg F_2$ и $F_2 \gg F_1$ (сложности всегда положительны). Транзитивность:

$$C_1^1 > C_1^2 + C_0^1, C_1^2 > C_1^3 + C_0^2 \quad \Rightarrow \quad C_1^1 > C_1^3 + C_0^1 + C_0^2 > C_1^3 + C_0^1.$$

Теперь доказываем свойство:

$$\begin{aligned} F_2 \gg F_3 \quad &\Rightarrow \quad C_1^{F_2 \vee F_3} = C_1^2; \\ F_1 \gg F_2 \quad &\Rightarrow \quad C_1^1 > C_1^2 + C_0^1 = C_1^{F_2 \vee F_3} + C_0^1. \end{aligned}$$

Лемма доказана. ■

8 Построение алгоритма замены развилки бинарной формулой

В этом заключительном разделе будет построен алгоритм для быстрого нахождения оптимального распределения вероятностей для развилки-дизъюнкции с произвольным числом входов, а значит, и для любой однократной формулы вообще. Более того, будет предложен способ замены развилки на бинарную формулу той же 1-сложности (и, естественно, той же 0-сложности).

Пусть $x_1, x_2, \ldots, x_n$ — n переменных с некоторыми стоимостями C_0^i, C_1^i. Вначале алгоритм заполняет множество $\mathcal{F}$ этими переменными (кроме того, он хранит 0- и 1-сложности элементов $\mathcal{F}$). Затем он циклически выполняет следующую процедуру: просматривая всевозможные пары различных формул из $\mathcal{F}$, он ищет несравнимые по перетягиванию, т.е. такие F_1 и F_2, что $F_1 \not\gg F_2$ и $F_2 \not\gg F_1$; найдя их, он удаляет их из $\mathcal{F}$, а на их место записывает формулу $F_1 \vee F_2$. Алгоритм прекращает выполнение процедуры, когда либо все формулы в $\mathcal{F}$ сравнимы, либо если осталась одна формула.

На каждом шаге алгоритма в $\mathcal{F}$ содержатся только бинарные однократные формулы из дизъюнкций без перетяжек. Если после завершения алгоритма осталась только одна формула, то она и есть формула той же сложности, что и развилка с n входами $x_1, \ldots, x_n$ (леммы 1 и 2). Если осталось несколько формул, то они все сравнимы по перетягиванию, а значит, линейно упорядочены. Тогда возьмем наименьшие две формулы и заменим на их дизъюнкцию (с перетяжкой). По лемме 4 линейная упорядоченность сохранится. Так будем делать, пока не останется две формулы: F_1 и F_2. Будем иметь $F_1 \gg F_2$, F_1 — бинарная однократная формула без перетяжек, F_2 — с перетяжками. По леммам 1 и 2 F_1 столь же 1-сложна, как и соответствующая развилка. Взяв $F = F_1 \vee F_2$, мы оказываемся в ситуации из следствия 1, откуда получаем, что формула F является оптимальной.