Полиномиальные случаи решения задачи об F-выполнимости булевых формул

Е. А. Поцелуевкая

Введение

На сегодняшний день в основе многих систем обеспечения информационной безопасности лежат различные NP-полные задачи, не решаемые в общем случае за полиномиальное время. Одной из них является проблема формульной выполнимости. Для данной задачи большое практическое значение имеет выявление подклассов задач, которые могут быть решены за полиномиальное время. В частности, несколько подобных классов было найдено Шеффером, описание данных классов приведено в его работе *The complexity of satisfiability problems*. В настоящей работе предпринята попытка отыскать новые классы функций, для которых задача выполнимости решалась бы также быстро.

1. Постановка задачи

В общем случае задача о F-выполнимости булевых формул ставится следующим образом.

Пусть $\mathbf{F} = \{F_1, \ldots, F_s\}$ — любое конечное множество формул (функциональных символов). Определим F-формулу как конъюнкцию $F_{i_1}(\cdot)F_{i_2}(\cdot)\ldots F_{i_t}(\cdot)$ с переменными $x_1, \ldots, x_n$, расставленными некоторым образом.

Проблема F-выполнимости — это проблема выполнимости F-формул. В общем случае данная задача является NP-полной.

Пример 1. Пусть $F(x,y,z)$ — булева формула с таблицей истинности $(1,0,0),(0,1,0),(0,0,1)$. Тогда формула $F(x,y,z)F(x,y,u)F(u,u,y)$ выполнима и $(x,y,z,u)=(0,1,0,0)$ — ее выполняющий набор.

Требуется выявить для задачи об F-выполнимости случаи (классы задач), когда она разрешима за полиномиальное время.

2. Методика проведения исследований

Основным методом исследования данного вопроса послужило введение дополнительных ограничений на заданные функции, позволяющих решать поставленную задачу за полиномиальное время. Рассматривается случай, когда все функции F_{i_k} зависят не более чем от трех переменных и заданы таблицей истинности либо формулой в конъюнктивной нормальной форме. Для этого случая разработан алгоритм, сочетающий в себе перебор определенного подмножества S переменных x_i и решение для каждого фиксированного набора значений переменных из S полиномиальной подзадачи о 2-выполнимости. В соответствии с данным алгоритмом разработана программа, получающая на вход функции $F_{i_1}(\cdot),F_{i_2}(\cdot),\ldots,F_{i_t}(\cdot)$ и выдающая ответ, выполнима ли данная формула, выполняющий набор (в случае выполнимости) и время работы программы.

3. Основные понятия и утверждения

Для того, чтобы отыскать полиномиальные случаи решения задачи об F-выполнимости булевых формул, введем упомянутые выше ограничения на функции $F_1,F_2,\ldots,F_s$:

- $F_i(\cdot)$ $(i=1\ldots s)$ зависят не более, чем от трех переменных;
- $F_i(\cdot)$ $(i=1\ldots s)$ заданы таблицами истинности или КНФ.

Назовем задачу F-выполнимости при указанных ограничениях задачей $3-F$-выполнимости и докажем, что она NP-полна.

Теорема 1. *Задача $3-F$-выполнимости булевых формул NP-полна.*

Доказательство. Пусть все функции $F_i(\cdot)$ $(i = 1 \ldots s)$ заданы таблицами истинности, тогда для каждой функции $F_i(\cdot)$ возможны следующие варианты:

1) $F_i(\cdot) \equiv 1$, тогда добавление F_i в какую-то F-формулу никак не повлияет на её выполнимость и нахождение выполняющего набора, и такую функцию можно исключить из рассмотрения. В случае же если F-формула состоит из одной F_i, формула выполнима и этот частный случай задачи $3 - F$-выполнимости решается за полиномиальное время.
2) $F_i(\cdot) \not\equiv 1$, тогда для данной функции можно по таблице истинности построить её СКНФ.

Таким образом, случай, когда все функции заданы таблицами истинности, сводится к случаю, когда все функции заданы КНФ за полиномиальное время.

Пусть теперь все $F_i(\cdot)$ $(i = 1 \ldots s)$ заданы КНФ. Тогда данная задача очевидным образом сводится к задаче 3-выполнимости КНФ, которая является NP-полной. Теорема доказана.

Пусть $X = \{x_1, \ldots, x_n\}$ — множество переменных. Пусть каждая из булевых функций из множества $\mathbf{F} = \{F_1, \ldots, F_s\}$ зависит не более чем от трех переменных из X и задана своей конъюнктивной нормальной формой. Тогда функция $F = F_{i_1}(\cdot) \ldots F_{i_t}(\cdot)$ также приводится к КНФ, и все дизъюнкты, входящие в данную запись КНФ, могут быть разбиты на следующие три множества:

$$D_1 = \{x_i^\alpha | x_i \in X, \alpha \in \{0,1\}, x_i^\alpha \& F = F\};$$

$$D_2 = \{(x_i^\alpha \vee x_j^\beta) | x_i, x_j \in X, i \neq j, \alpha, \beta \in \{0,1\}, (x_i^\alpha \vee x_j^\beta) \& F = F\};$$

$$D_3 = \{(x_i^\alpha \vee x_j^\beta \vee x_k^\gamma) | x_i, x_j, x_k \in X, i \neq j, j \neq k, i \neq k, \alpha, \beta, \gamma \in \{0,1\}, \\ (x_i^\alpha \vee x_j^\beta \vee x_k^\gamma) \& F = F\}.$$

То есть множества дизъюнктов КНФ, зависящих от одной, двух и трех переменных соответственно. Таким образом, F можно представить в виде

$$F = F_{i_1}(\cdot) \ldots F_{i_t}(\cdot) = \mathop{\&}_{i=1}^{|D_1|} d_i^1 \mathop{\&}_{j=1}^{|D_2|} d_j^2 \mathop{\&}_{k=1}^{|D_3|} d_k^3, \text{ где } d_i^l \in D_l,\ l \in \{1, 2, 3\}.$$

Пример 2. Пусть $F_1(x_1, x_2, x_3) = x_1(x_2 \vee \bar{x}_3)(\bar{x}_1 \vee x_2)(x_1 \vee x_2 \vee x_3)$, $F_2(x_1, x_2, x_4) = \bar{x}_2(\bar{x}_1 \vee x_2)(x_2 \vee x_4)$, $F_3(x_1, x_3, x_5) = x_1 x_3 x_5(x_1 \vee \bar{x}_3 \vee x_5)$. Тогда $D_1 = \{x_1, \bar{x}_2, x_3, x_5\}$, $D_2 = \{(x_2 \vee \bar{x}_3), (\bar{x}_1 \vee x_2), (x_2 \vee x_4)\}$, $D_3 = \{(x_1 \vee x_2 \vee x_3), (x_1 \vee \bar{x}_3 \vee x_5)\}$.

Обозначение. Пусть $d \in D_l$, $l \in \{1, 2, 3\}$. Для переменной $x \in X$ будем обозначать $x \in d$, если x входит в соответствующую дизъюнкцию.

Обозначение. Введем следующие обозначения множеств: для переменной $x \in X$ обозначим $D(x) = \{d \in D_3 | x \in d\}$, для множества M переменных из X обозначим $D(M) = \{d \in D_3 | \exists x \in M \ x \in d\}$.

Определение. Будем говорить, что множество переменных M *покрывает* множество дизъюнктов $D \subseteq D_3$, если $D \backslash D(M) = \varnothing$.

Для множества D_3 определим множество S как минимальное множество переменных, покрывающее все дизъюнкции из D_3, то есть наименьшее по мощности множество переменных, таких, что в каждой дизъюнкции из D_3 задействована хотя бы одна из них. То есть $S = \min\limits_{|M|}\{M\}$, где $M = \{x \in X | \ \exists d \in D_3, \text{ такое что } x \in d\}$.

Пример 3. Для случая, рассмотренного в предыдущем примере $D_3 = \{(x_1 \vee x_2 \vee x_3), (x_1 \vee \bar{x}_3 \vee x_5)\}$, тогда $S = \{x_1\}$ либо $S = \{x_3\}$. Данный пример, в частности, показывает, что множество S определяется не однозначно.

Пример 4. Пусть $D_3 = \{(x_1 \vee x_2 \vee x_3), (x_1 \vee \bar{x}_2 \vee x_3), (x_1 \vee x_4 \vee \bar{x}_5), (x_4 \vee x_5 \vee \bar{x}_6)\}$, тогда S может выглядеть следующим образом: $\{x_1, x_4\}, \{x_1, x_5\}, \{x_1, x_6\}, \{x_2, x_4\}, \{x_2, x_5\}, \{x_3, x_4\}, \{x_3, x_5\}$.

Пусть $S = \bigcup S_i$, $S_i \bigcap S_j = \varnothing$, $i \neq j$ — разбиение множества S на непересекающиеся подмножества.

Обозначение. Для подмножества $S_i \in S$ и дизъюнкта $d \in D_3$ будем обозначать $d \in S_i$, если в d входит хотя бы одна переменная из S_i.

Пусть S_i — непересекающиеся по дизъюнкциям подмножества S, то есть в каждое из множеств S_i входят только те переменные S, для которых все переменные, входящие в дизъюнкции с элементами из S_i, не встречаются больше нигде в D_3. То есть $\forall d \in S_i \ \forall j \neq i \ d \notin S_j$.

Пример 5. Для случая, рассмотренного в Примере 4, при $S = \{x_1, x_4\}$ разбиение состоит из одного множества $S_1 = S$.

Пример 6. Пусть теперь $D_3 = \{(x_1 \vee x_2 \vee x_3), (x_1 \vee \bar{x}_2 \vee x_3), (x_1 \vee x_4 \vee \bar{x}_5), (x_6 \vee x_7 \vee \bar{x}_8)\}$, тогда $S = \{x_1, x_6\}$ и $S_1 = \{x_1\}$, $S_2 = \{x_6\}$.

Обозначим D_3^i — множество дизъюнкций из D_3, в которые входят переменные из S_i. То есть $D_3^i = D(S_i) = \{d \in D_3 \mid \exists x \in S_i$, такое что $x \in d\}$.

Пример 7. Для случая, рассмотренного в Примере 6, $D_3^1 = \{(x_1 \vee x_2 \vee x_3), (x_1 \vee \bar{x}_2 \vee x_3), (x_1 \vee x_4 \vee \bar{x}_5)\}$, $D_3^2 = \{(x_6 \vee x_7 \vee \bar{x}_8)\}$.

Таким образом, используя введенные обозначения, можно сформулировать следующее утверждение.

Утверждение 1. *Конъюнкция дизъюнктов из D_3 выполнима тогда и только тогда, когда одновременно выполнимы конъюнкции дизъюнктов из всех D_3^i.*

Доказательство. Данное утверждение следует из того, что если $D_3 = \bigcup\limits_{i=1}^{p} D_3^i$, то $\&_{j=1}^{|D_3|} d_j = \&_{i=1}^{p} \left(\&_{l=1}^{|D_3^i|} d_{i_l} \right)$.

Таким образом, для того, чтобы проверить выполнимость конъюнкции элементов из D_3 достаточно проверить выполнимость каждой из конъюнкций D_3^i, причем проверки можно осуществлять независимо друг от друга, так как все переменные, входящие в них, различны (по построению D_3^i).

4. Алгоритм решения задачи об F-выполнимости

В общем виде разработанный алгоритм решения задачи об F-выполнимости булевых формул выглядит следующим образом.

Входные данные:

1) Количество функций в F-формуле t.
2) КНФ функций $F_{i_1}(\cdot), F_{i_2}(\cdot), \ldots F_{i_t}(\cdot)$, зависящих не более чем от трех переменных.

Порядок действий:

1) По КНФ функций $F_{i_1}(\cdot), F_{i_2}(\cdot), \dots F_{i_t}(\cdot)$ строятся множества D_1, D_2, D_3.
2) В случае если $D_3 = \varnothing$, задача сводится к проблеме 2-выполнимости КНФ и решается методом резолюций за полиномиальное время.
3) В случае если $D_3 \neq \varnothing$, осуществляется поиск множества M, покрывающего D_3, с помощью следующего алгоритма:
   - для каждой переменной x, входящей в дизъюнкции D_3, считается суммарное количество вхождений этой переменной $k(x) = |\{d \in D_3 | x \in d\}|$;
   - переменные из D_3 упорядочиваются по убыванию функции $k(x)$. Без ограничения общности переименуем эти переменные в соответствии с данным порядком: $x_1, x_2, \dots, x_k$, $k(x_1) \geqslant k(x_2) \geqslant \dots \geqslant k(x_k)$;
   - $M = \{x_1\}$;
   - пусть $D(x_1) = \{d \in D_3 | x_1 \in d\}$ — множество дизъюнкций D_3, в которые входит переменная x_1, тогда если $D_3 \backslash D(x_1) \neq \varnothing$, то для каждой переменной x_j, задействованной в $D_3 \backslash D(x_1)$ вычисляем величину $k_M(x_j) = |\{d \in D_3 \backslash D(x_1) | x_j \in d\}|$ и добавляем к M переменную x_j, для которой k_M достигает максимального значения ($k_M(x_j) > k_M(x_p), p \neq j$). Если максимальных значений несколько ($k_M(x_j) \geqslant k_M(x_p), p \neq j$), то выбираем среди них такое, для которого $k(x_j) > k(x_p), p \neq j$. Если и здесь значения совпадают, то в M добавляется любой из элементов, для которых максимальны обе величины: k_M и k. Переходим к следующему шагу;
   - если $D \backslash D(M) \neq \varnothing$, то рассмотрим переменные из $D \backslash D(M)$. Для них считаем $k_M(x_j) = |\{d \in D_3 \backslash D(M) | x_j \in d\}|$ и выбираем x_j для добавления в M аналогично предыдущему шагу;
   - этот процесс продолжается до тех пор, пока не будет исчерпано множество D_3. Из способа построения M видно, что так как D_3 конечно, то этот процесс рано или поздно завершится.

4) По множеству M, которое покрывает D_3, но не обязательно является наименьшим по мощности, строится множество S следующим образом:
   - $S = \varnothing$;
   - для всех элементов D_3 вычисляется их «вес» относительно множества M: $p(d) = |\{x \in M | x \in d\}|$, то есть, количество элементов из M, входящих в данный дизъюнкт;
   - для всех $d \in D_3$, таких что $p(d) = 1$ (то есть дизъюнкты, покрытые только одним элементом из M), соответствующая переменная $x \in M$, такая что $x \in d$, добавляется в множество S;
   - (*) обозначим $D(S) = \{d \in D_3 | \exists x \in S,\text{ такое что } x \in d\}$, тогда если $D_3 \backslash D(S) \neq \varnothing$ (то есть S не покрывает полностью D_3), то переходим к следующему шагу;
   - добавляем в S переменную из $M \backslash S$, которая встречается наибольшее количество раз в $D_3 \backslash D(S)$, и переходим к (*);
   - этот процесс продолжается до тех пор, пока не будет исчерпано множество D_3.
5) S разбивается на непересекающиеся по дизъюнкциям подмножества S_i следующим образом:
   - на шаге i: $S_i = \varnothing$;
   - выбирается первый элемент из S, не входящий в предыдущие S_j $(j < i)$, (на первом шаге выбирается любой элемент множества S) и добавляется в S_i;
   - для элемента $s_{i_1} \in S_i$ определяется множество $D_3^i = D(s_{i_1}) = \{d \in D_3 | s_{i_1} \in d\}$;
   - (**) для каждого элемента множества D_3^i осуществляется проверка, не входят ли туда другие элементы из S. Если существует $s \in S$, $\forall s_{i_j} \in S_i \; s_{i_j} \neq s$, такое что для какого-то $d \in D_3^i$ $s \in d$, то в D_3^i добавляются элементы множества $D(s) = \{d \in D_3 | s \in d\}$, а элемент s добавляется к множеству S_i;
   - процесс повторяется до тех пор, пока для всех дизъюнкций $d \in D_3^i$ не будет выполнено условие $\forall\, x \in d$, таких что $x \in S$, $x \in S_i$ (то есть все элементы S, входящие в D_3^i, уже учтены в S_i);

- осуществляется проверка, входят ли другие переменные множества D_3^i в $D_3 \backslash D_3^i$. Если $\exists d \in D_3^i \exists x \in d$, $x \notin S$, такое что $x \in D_3 \backslash D_3^i$, то для всех таких x элементы множества $D(x) = \{d \in D_3 | x \in d\}$ добавляем к множеству D_3^i и переходим к шагу (**);
- если же $\forall d \in D_3^i \, \forall x \in d \;\; \nexists d' \in D_3 \backslash D_3^i$, такого что $x \in d'$, то переходим к шагу i+1;
- процесс формирования множеств S_i продолжается до тех пор, пока не будет исчерпано всё множество S. Так как S конечно, то построение разбиения на каком-то шаге обязательно закончится.

6) В соответствии с разбиением множества S на S_i множество дизъюнкций от 3 переменных D_3 оказывается разбито на D_3^i, такие что каждое D_3^i зависит от своего набора переменных, которые не встречаются в других множествах разбиения. Далее проверка выполнимости формулы осуществляется независимо для каждого такого подмножества дизъюнкций. Для каждого S_i осуществляется перебор значений переменных, входящих в S_i.

7) При фиксированном наборе значений переменных из S_i проверяется выполнимость конъюнкции формул, зависящих не более чем от двух переменных. Пусть D_1 — множество конъюнктов, входящих в исходную формулу и зависящих ровно от одной переменной; D_2 — множество конъюнктов, входящих в исходную формулу и зависящих ровно от двух переменных; D_1^i, D_2^i — множества конъюнктов, входящих в формулу D_3^i и полученных в результате подстановки фиксированного набора значений из S_i, зависящие от одной и двух переменных соответственно. Тогда на для каждого S_i проверяется выполнимость формулы

$$\mathop{\&}_{q=1}^{|D_1|} d_q^1 \mathop{\&}_{j=1}^{|D_2|} d_j^2 \mathop{\&}_{k=1}^{|D_1^i|} d_k^{i_1} \mathop{\&}_{p=1}^{|D_2^i|} d_p^{i_2}, \text{ где } d_j^l \in D_l,\ d_j^{i_l} \in D_l^i,\ l \in \{1, 2\}.$$

Задача о выполнимости такой формулы тривиальным образом сводится к задаче о 2-выполнимости КНФ, которая решается за полиномиальное время с помощью метода резолюций.

8) Если для какого-то S_i была установлена невыполнимость, то искомая формула невыполнима. Если же для каждого S_i соответствующие формулы выполнимы, то искомая формула выполнима.

Теорема 2. *Для множества S, построенного на шаге 4 алгоритма, выполнено $S = \min\limits_{|M|}\{M|D_3\backslash D(M) = \varnothing\}$.*

Доказательство. Допустим, что утверждение теоремы не верно и существует множество S', такое что $D_3\backslash D(S') = \varnothing$ и $|S'| < |S|$. Значит, из соображений размерности, существует подмножество D_3, которое покрыто одним элементом S' и как минимум двумя элементами S. Искомый элемент S' не может лежать одновременно и в S, так как иначе рассматриваемое множество можно было бы покрыть меньшим количеством элементов S, что противоречит предположению.

Введем обозначения для этих элементов: пусть $x_1 \in S'$, $x_2, x_3 \in S$, $x_1 \neq x_2$, $x_1 \neq x_3$, $x_2 \neq x_3$.

Так как $x_1 \notin S$, то для этой переменной возможны следующие варианты:

- x_1 на шаге 3 алгоритма не попала в исходное множество M;
- x_1 на шаге 4 алгоритма не попала в множество S.

1) Предположим, элемент $x_1 \in S'$ не попал в M. Это значит, что на каком-то шаге алгоритма оказалось, что x_1 встречается реже, чем x_2 (x_3), или что x_1 встречается столько же раз, сколько x_2 (x_3), а элемент M был выбран случайным образом. В любом случае, так как в $D(x_1)$ встречаются и x_2, и x_3, то существуют $d_2, d_3 \in D_3\backslash D(x_1)$, такие что $x_2 \in d_2$, $x_3 \in d_3$.
   Так как S' покрывает D_3, то дизъюнкты, где встречаются x_2 и x_3 (в том числе d_2, d_3) тоже должны покрываться какими-то элементами S'. Возможны два случая:
   - множество $(D(x_2)\bigcup D(x_3))\backslash D(x_1)$ покрывается несколькими элементами S', тогда одно и то же множество $D(x_2)\bigcup D(x_3)$ покрывается двумя элементами множества S либо как минимум тремя элементами множества S';

- множество $(D(x_2) \bigcup D(x_3)) \setminus D(x_1)$ покрывается одним элементом S', тогда обозначим этот элемент x_4. Если бы $x_4 \in M$, то в $D_3 \setminus D(M)$ элемент x_1 входил бы чаще, чем x_2 и x_3, тогда бы включили x_1 в M, а это не так по предположению. Значит, $x_4 \notin M$, а следовательно $x_4 \notin S$. Итак, x_2 и x_3 встречаются только вместе с x_1 или x_4, значит $|D(x_1) \bigcup D(x_4)| \geqslant |D(x_2) \bigcup D(x_3)|$. Если $|D(x_1) \bigcup D(x_4)| > |D(x_2) \bigcup D(x_3)|$, то должны были бы включить x_1 и x_4 в M, но $x_1, x_4 \notin M$. Значит, $|D(x_1) \bigcup D(x_4)| = |D(x_2) \bigcup D(x_3)|$, но тогда одно и то же множество $D(x_2) \bigcup D(x_3)$ покрывается двумя элементами S' либо двумя элементами S.

Таким образом, приходим к противоречию с предположением, что $|S'| < |S|$. Значит, $x_1 \in M$.

2) Предположим, что элемент x_1 на шаге 4 алгоритма не включили в S. Следовательно, x_1 не может входить в дизъюнкции $d \in D_3$ с $p(d) = 1$ (иначе должны были бы включить в S). Таким образом, для всех $d \in D(x_1)$ $p(d) \geqslant 2$. Выходит, что x_1 могли не включить в S по следующим причинам:

   - все элементы $D(x_1)$ покрываются такими элементами S, которые покрывают дизъюнкции с $p(d) = 1$. То есть, в частности, x_2, x_3 покрывают такие дизъюнкции. Однако эти дизъюнкции должны одновременно покрываться какими-то элементами S'. Так как в таких дизъюнкциях с единичным весом уже есть по одному элементу из M (это элементы, входящие в S), то элементы S', покрывающие то же множество дизъюнкций, не могут входить в множество M. Значит, количество элементов S', покрывающих дизъюнкции с единичным весом, не может превышать количество соответствующих элементов S (это следует из пункта 1 настоящей теоремы). Таким образом, одно и то же множество $D(x_2) \bigcup D(x_3)$ оказывает покрыто двумя элементами множества S либо как минимум тремя элементами множества S'.
   - множество $D(x_1)$ на каком-то шаге оказалось покрыто элементами M, которые встречаются чаще, чем x_1. После до-

бавления элементов, покрывающих дизъюнкции с единичным весом, в S, хотя бы один из элементов x_2 или x_3 должен был остаться в $D(x_1)$. Допустим, x_2 входит в $D(x_1)$, и остальные элементы $D_(x_1)$ уже покрыты другими элементами S. Тогда либо элемент x_2 встречается чаще, чем x_1, и тогда одно и то же множество $D(x_2)$ оказывается покрыто одним элементов S и как минимум двумя элементами S', либо x_2 встречается столько же раз, сколько x_1, и тогда $D(x_2)$ покрыто одинаковым количеством элементов S' и S.

Таким образом, пришли к противоречию с тем, что $|S'| < |S|$ и $x_1 \in S$, что противоречит предположению.

Значит, действительно, множество S, построенное на шаге 4 алгоритма является наименьшим по мощности множеством, покрывающим D_3. Теорема доказана.

5. Оценка сложности алгоритма

Утверждение 2. *Сложность алгоритма решения задачи о 3–F-выполнимости булевых формул составляет* $\left(1 + \sum\limits_{i=1}^{k} 2^{|S_i|}\right) poly(|x|)$, *где* $|x|$ *— длина входа.*

Доказательство. Если обозначить количество дизъюнктов в КНФ F-формулы за m, то длину входной информации алгоритма можно оценить как $|x| \leqslant Cm$, где $C = const$. Шаги 1 и 2 алгоритма выполняются за полиномиальное от m время. Количество переменных D_3 линейно зависит от m, поэтому 3 шаг алгоритма также выполняется за полиномиальное время. Количество элементов множеств M и S ограничивается общим количеством переменных в D_3, поэтому 4 и 5 шаги алгоритма выполняются за полиномиальное от m время. Перебор значений переменных из S_i занимает $2^{|S_i|}$ шагов, а решение соответствующей подзадачи о 2-выполнимости методом резолюций полиномиально. Утверждение доказано.

Следствие 1. *В случае если для всех* $i = 1 \dots k$ $|S_i| \leqslant \log_2(poly(m))$ *сложность алгоритма будет полиномиальной величиной.*

Рассмотрим самый худший случай, когда S не разбивается на непересекающиеся подмножества и S насчитывает ровно m_1 элементов, тогда оценка сложности должна была бы выглядеть так $(2^{m_1}+1)poly(m)$. Величина m_1 достигает максимального значения, когда она равна количеству всех задействованных дизъюнкций от трех переменных. Однако в этом случае все переменные, входящие в дизъюнкции, должны быть различными (иначе m_1 можно было бы сократить). Значит, S непременно разобьется на непересекающиеся подмножества, и сложность алгоритма составит $2m \cdot poly(m)$. Отсюда можно сделать вывод, что в любом случае (даже когда условия полиномиальности задачи из Следствия 1 не выполнены) сложность алгоритма оказывается лучше, чем сложность полного перебора значений переменных, хотя и может оставаться экспоненциальной.

В соответствии с разработанным алгоритмом была написана программа на языке C, позволяющая по заданному множеству булевых функций определить, выполнима ли F-формула, являющаяся конъюнкцией от этих функций, и в случае выполнимости выдающая выполняющий набор. С помощью данной программы были получены практические результаты, полностью отвечающие доказанным теоретическим оценкам.

Автор работы выражает признательность В. А. Носову за научное руководство.

Список литературы

[1] Алексеев В. Б., Носов В. А. NP-полные задачи и их полиномиальные варианты. Обзор // Обозрение прикладной и промышленной математики. Т. 4, вып. 2. 1997. С. 165–193.

[2] Гизунов С. А., Носов В. А. Сложность распознавания классов Шеффера // Вестник МГУ. Сер. 1. 1995.

[3] Dubois O. On the r, s-satisfiability problems and a conjecture of Tovey // Discrete Appl. Math. 1990. V. 26. P. 51–60.

[4] Schaefer T. J. The complexity of satisfiability problems // Proceedings of the 10th ACM Symposium on Theory of Computing. 1978. P. 216–226.