Хорновские формулы и задача получения продукции

Определим вначале один интересный класс булевых формул – Хорновские формулы.

Определение 1. *Пусть A — это множество логических (булевых) переменных. Хорновская (H-) формула — это формула вида*

$$a_1 \wedge a_2 \wedge \ldots \wedge a_r \to b,$$

где $a_i, b \in A$. H-формула ϕ является следствием множества H-формул F, если на всяком наборе значений переменных из A, на котором истинны все формулы из F, истинна и ϕ (будем это обозначать как $F \models \phi$).

Нетрудно понять, что H-формула ϕ является следствием множества H-формул F тогда и только тогда, когда формула

$$\Big(\bigwedge_{\psi \in F} \psi \Big) \ \to \ \phi \qquad\qquad (*)$$

является истинной на всех наборах значений переменных (т.е. тождественно истинной).

Задача 1. *Докажите это утверждение.*

Вообще говоря, проблема проверки по булевой формуле ее тождественной истинности является весьма сложной. Известный нам метод такой проверки с помощью построения таблицы значений на всех наборах переменных практически не работает уже для формул с несколькими десятками переменных. В то же время во многих практических задачах число логических параметров исчисляется сотнями. Оказывается, что для установления тождественной истинности формул вида (*) или, что то же самое, для задачи проверки условия $F \models \phi$ для H-формул имеется простой и очень эффективный алгоритм, позволяющий ее решать для формул с сотнями и тысячами переменных. Эта задача находит приложения во многих областях информатики: базах данных, автоматическом синтезе программ, автоматическом доказательстве теорем и др. Прежде, чем изложить алгоритм ее решения, мы в качестве одного из интересных приложений H-формул рассмотрим задачу о возможности производства заданной продукции (набора товаров) из некоторого множества исходных продуктов (товаров, сырья).

Пусть задано некоторое множество $A = \{a_1, a_2, \ldots, a_N\}$ имен товаров (продуктов, сырья и т.п.) и имеется некоторое множество F технологических процессов (производств), описывающих возможности получения одних продуктов из других. Каждый технологический процесс $t \in F$ задается множеством $L_t \subseteq A$ исходных продуктов (входов) этого процесса и результирующим продуктом $b_t \in A$, т.е. процесс t позволяет из исходных продуктов L_t получить продукт b_t - его выход. Будем задавать технологический процесс в виде $t: \ L_t \ \to b_t$. Продукт, полученный в одном процессе, может далее использоваться в других процессах. *Задача получения продукции* состоит в том, чтобы выяснить по заданному набору исходных продуктов $X \subseteq A$ и результирующему продукту $y \in A$ можно ли с помощью технологических процессов из F получить выход y по входным продуктам из X. (Можно обобщить эту задачу и рассматривать возможность получения по X некоторого множества результирующих продуктов $Y \subseteq A$.)

Пример 1. Пусть $A = \{$ *дерево, клей, гвозди, кирпич, стекло, окна, полы, стены, крыша, столы* $\}$. Множество технологических процессов $F = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ задается соответствующими множествами входов и выходов.

t_1 : { *дерево, клей, гвозди*} $\rightarrow$ *столы*
t_2 : { *дерево, гвозди*} $\rightarrow$ *полы*
t_3 : { *дерево, клей, стекло*} $\rightarrow$ *окна*
t_4 : { *стены, полы, крыша*} $\rightarrow$ *дача*
t_5 : { *кирпич, окна, дерево*} $\rightarrow$ *стены*
t_6 : { *дерево, гвозди*} $\rightarrow$ *столы*
Рассмотрим для этой системы технологических процессов задачу получения *дачи* по исходному множеству *{дерево, клей, гвозди, стекло, кирпич, крыша}*. Нетрудно понять, что эта задача решается положительно с помощью следующей цепочки процессов: $t_3; t_5; t_2; t_4$. Действительно, в t_3 получаются *окна*, которые используются в t_5 для получения *стен*, в t_2 производятся *полы*, а затем произведенные ранее *стены, полы, крыша* используются в t_4 для получения результата *дача*.

Подчеркнем, что мы абстрагируемся от количественных оценок исходных и производимых продуктов и считаем, что они всегда даются на входе и производятся в количестве, достаточном для обеспечения "сырьем"всех запускаемых процессов.

Построим формальную модель задачи о производстве с помощью булевых формул.

Будем рассматривать A как множество булевых переменных. Каждому процессу t с параметрами $L_t = \{a_1, \ldots, a_r\}$ и b_t сопоставим следующую H-формулу $\Phi(t)$:

$$a_1 \wedge a_2 \wedge ... \wedge a_r \rightarrow b_t.$$

Например, процессу t_5 из нашего примера соответствует формула $\Phi(t_5)$:

$$\textit{кирпич} \wedge \textit{окна} \wedge \ \textit{дерево} \rightarrow \textit{стены}.$$

Сохраним для множества H-формул, соответствующих процессам, обозначение F.

Справедлива следующая теорема, которая показывает, что задача о возможности получения продукции и задача о следствии из множества H-формул эквивалентны.

Теорема 1. *Для любых множества продуктов A, множества технологических процессов F, множества исходных продуктов $X \subseteq A$ и результирующего продукта $y \in A$ задача получения продукта y по входным продуктам из X с помощью процессов из F разрешима тогда и только тогда, когда $F \models \phi_X \rightarrow y$, где $\phi_X = \bigwedge_{a \in X} a$.*

Доказательство. $\Longrightarrow$ Предположим, что с помощью процессов из множества $F = \{t_1, \ldots, t_h\}$ из множества исходных продуктов X можно получить y. Пусть $\tau = t_{i_1}, \ldots, t_{i_m}$ – это последовательность процессов из F, которая приводит к получению y. Докажем, что тогда $F \models \phi_X \rightarrow y$. Рассмотрим произвольный набор значений переменных $\bar{\sigma} = (\sigma(a_1), \ldots, \sigma(a_n))$, на котором истинны все формулы из F. Если хотя бы для одной переменной $a_j \in X$ ее значение $\sigma(a_j) = 0$, то формула $\phi_X \rightarrow y$ истинна, поскольку ее левая часть ложна. Предположим теперь, что для любой переменной $a_j \in X$ ее значение $\sigma(a_j) = 1$.

Тогда индукцией по номеру i_r процесса t_{i_r} из τ покажем, что *для каждого $r = 1, \ldots, m$ значение соответствующей результирующей переменной* $\sigma(b_{i_r}) = 1$.
Действительно, при $r = 1$ из применимости процесса t_{i_1} следует, что $L_{i_1} \subseteq X$, но тогда $\sigma(a) = 1$ для любой переменной $a \in L_{i_1}$ и левая часть импликации

$\Phi(t_{i_1})$ истинна на наборе $\bar{\sigma}$. Но так как и вся формула $\Phi(t_{i_1})$ истинна на $\bar{\sigma}$, то и ее заключение b_{i_1} тоже истинно на $\bar{\sigma}$, т.е. $\sigma(b_{i_1}) = 1$. Пусть теперь для некоторого $k > 1$ $\sigma(b_{i_r}) = 1$ при $r < k$. Докажем, что и $\sigma(b_{i_k}) = 1$. Поскольку процесс t_{i_k} применим после процессов $t_{i_1}, \ldots, t_{i_{k-1}}$, то $L_{i_k} \subseteq X \cup \{b_{i_r} \mid r = 1, \ldots, k-1\}$. Тогда все переменные из L_{i_k} истинны на $\bar{\sigma}$ и, следовательно, $\sigma(b_{i_k}) = 1$. Из доказанного утверждения следует, что $\sigma(b_{i_h}) = 1$. Но так как последовательность τ приводит к выпуску y, то $b_{i_h} = y$ и, следовательно, $\sigma(y) = 1$. Но тогда формула $\phi_X \to y$ истинна на $\bar{\sigma}$ и условие $F \models \phi_X \to y$ выполнено.

$\Longleftarrow$ Предположим теперь, что выполнено условие $F \models \phi_X \to y$. Опишем построение последовательности процессов τ_i, которая приведет к производству y. Одновременно с последовательностью τ_i будем строить последовательность множеств продуктов X_i, которые можно произвести, исходя из X с помощью τ_i. Процедура построения последовательности τ_i завершается, как только в нее включается некоторый процесс с результатом y либо, когда на очередном этапе в X_i не добавляются новые элементы.

Пусть $X_0 = X$. Положим $\tau_1 = \{t \in F \mid L_t \subseteq X_0\}$ и $X_1 = X_0 \cup \{b_t \mid t \in \tau_1\}$. Пусть уже определены τ_i и X_i. Положим $\tau'_{i+1} = \{t \in (F \setminus \tau_i) \mid L_t \subseteq X_i\}$, $X_{i+1} = X_i \cup \{b_t \mid t \in \tau'_{i+1}\}$ и $\tau_{i+1} = \tau_i \tau'_{i+1}$ (процессы внутри τ'_{i+1} упорядочиваются в произвольном порядке). Эта процедура построения τ_i и X_i завершается на таком i, для которого впервые $y \in X_i$ или $X_i = X_{i+1}$. Понятно, что в первом случае последовательность процессов τ_i приводит к производству y. Покажем, что второй случай не может быть причиной остановки. Действительно, предположим, что процедура завершилась после этапа i и $X_i = X_{i+1}$, но $y \notin X_i$. Покажем, что тогда существует набор значений переменных $\bar{\sigma} = (\sigma(a_1), \ldots, \sigma(a_n))$, на котором все формулы из F истинны, а формула $\phi_X \to y$ ложна. Положим $\sigma(a) = 1$ при $a \in X_i$ и $\sigma(a) = 0$ при $a \notin X_i$. Ясно, что для каждого $a \in X$ значение $\sigma(a) = 1$, а $\sigma(y) = 0$, т.е. формула $\phi_X \to y$ на наборе $\bar{\sigma}$ ложна. Каждая формула $\Phi(t)$ для $t \in \tau_i$ истинна, поскольку $b_t \in X_i$ и, следовательно, $\sigma(b_t) = 1$. Ложной могла бы оказаться лишь такая формула $\Phi(t)$, у которой заключение $b_t \notin X_i$. Но для такого процесса t обязательно имеется продукт $a \in L_t$, который не входит в X_i (иначе бы b_t попало в X_{i+1} и процедура не остановилась на i-ом этапе). Для этого a значение $\sigma(a) = 0$. Но тогда условие импликации $\Phi(t)$ ложно на $\bar{\sigma}$, а вся формула $\Phi(t)$ на нем истинна. Таким образом, мы пришли к противоречию, которое показывает, что $y \in X_i$ и процесс τ_i приводит к производству y. $\Box$

Задача 2. *Докажите, что последовательность процессов τ_i в доказательстве теоремы 1 определена корректно, т.е. все исходные продукты каждого процесса в этой последовательности имеются перед его запуском.*

Можно ли построить эффективную процедуру, проверяющую разрешимость задачи о продукции или (что то же самое) задачи о следствии для H-формул? Процедура, описанная во второй части доказательства теоремы 1, является основой для следующего алгоритма, который строит множество всех продуктов, которые можно получить из исходных с помощью заданной системы технологических процессов.

Алгоритм поиска решения от данных (прямой поиск или прямая волна)

Пусть F - множество технологических процессов, X — исходное множество продуктов. Определим замыкание X с помощью F как

$Cl(X, F) = \{y \mid (y \in A) \ \ и \ \ (F \models \phi_X \to y)\}$.

В приведенном ниже алгоритме переменные СТАРЫЕ и НОВЫЕ – это множества продуктов (истинных булевых переменных).

Алгоритм ЗАМЫКАНИЕ(X,F)

1. СТАРЫЕ := $\emptyset$; НОВЫЕ := X;
2. **ПОКА** НОВЫЕ $\neq$ СТАРЫЕ **ВЫПОЛНЯТЬ**
3. { СТАРЫЕ := НОВЫЕ;
4. **ДЛЯ КАЖДОГО** процесса $t \in F$ **ВЫПОЛНЯТЬ**
5. **ЕСЛИ** $L_t \subseteq$ НОВЫЕ **ТО** НОВЫЕ := НОВЫЕ $\cup\{b_t\}$;
6. **ВСЕ_ПОКА**;
7. **вернуть**(НОВЫЕ).

Алгоритм ПрямаяВолна(X,y,F)

1. $Z :=$ ЗАМЫКАНИЕ(X, F);
2. **ЕСЛИ** $y \subseteq Z$ **ТО вернуть**("ДА") **ИНАЧЕ вернуть**("НЕТ").

Теорема 2. *Алгоритм ЗАМЫКАНИЕ(X,F) возвращает множество* $Cl(X,F)$, *а алгоритм ПрямаяВолна(X,y,F) выдает ответ "ДА"тогда и только тогда, когда* $F \models \phi_X \to y$.

Задача 3. *Докажите теорему 2.*

Рассмотрим работу алгоритма ЗАМЫКАНИЕ(X,F) на задаче получения *дачи* по исходному множеству *{дерево, клей, гвозди, стекло, кирпич, крыша}* из Примера 1. В следующей таблице показаны шаги алгоритма, на которых изменяются значения переменных СТАРЫЕ и НОВЫЕ. В первом столбце этой таблицы указан номер соответствующей строки алгоритма, после строки 5 в скобках указан тот процесс, который приводит к увеличению множества НОВЫЕ.

Стр.	СТАРЫЕ	НОВЫЕ
1	$\emptyset$	*дерево, клей, гвозди, кирпич, крыша, стекло, столы*
3	*дерево, клей, гвозди, кирпич, крыша, стекло*	*дерево, клей, гвозди, кирпич, крыша стекло*
5 (t_1)	— " —	*дерево, клей, гвозди, кирпич, крыша, стекло, столы*
5 (t_2)	— " —	*дерево, клей, гвозди, кирпич, крыша, стекло, столы, полы*
5 (t_3)	— " —	*дерево, клей, гвозди, кирпич, крыша, стекло, столы, полы, окна*
5 (t_5)	— " —	*дерево, клей, гвозди, кирпич, крыша, стекло, столы, полы, окна , стены*
3	*дерево, клей, гвозди, кирпич, крыша, стекло, столы, полы, окна, стены*	*дерево, клей, гвозди, кирпич, крыша, стекло, столы, полы, окна, окна, стены*
5 (t_4)	— " —	*дерево, клей, гвозди, кирпич, крыша,стекло, столы, полы, окна , стены, дача*
3	*дерево, клей, гвозди, кирпич, крыша, стекло, столы, полы, окна , стены, дача*	*дерево, клей, гвозди, кирпич, крыша, стекло, столы, полы, окна, стены, дача*

Задача 4. *Алгоритм ПрямаяВолна(X,y,F) позволяет ответить на вопрос о возможности производства y из исходных продуктов X с помощью процессов F, но в случае положительного ответа не строит последовательность процессов, приводящую к y. Измените алгоритм ЗАМЫКАНИЕ(X,F) так, чтобы по его результату для любого продукта $a \in Cl(X,F)$ можно было построить последовательность процессов, приводящую к a.*

Задача 5. *Назовем* сложным технологическим процессом (или производством) *такой процесс t, который по набору исходных продуктов L_t производит некоторое множество продуктов B_t (а не один продукт b_t). Обобщите алгоритм ЗАМЫКАНИЕ(X,F) так, чтобы он строил замыкание X относительно системы сложных технологических процессов F.*

Недостатком алгоритма ЗАМЫКАНИЕ(X,F) является то, что на каждой итерации основного цикла в строках 2-6 в строке 4 перебираются все процессы, даже уже отработавшие, а в строке 5 на вхождение в НОВЫЕ проверяются все элементы L_t даже те, вхождение которых в НОВЫЕ уже было установлено на предыдущих итерациях. Ниже приведен более эффективный алгоритм для построения замыкания. На этапе инициализации в нем для каждого процесса t подсчитывается число элементов в L_t и помещается в ячейку массива СЧЕТ$[t]$, кроме того, для каждого $a \in A$ создается список СПИСОК$[a]$ (номеров) процессов, во входы (левые части) которых входит продукт a. Далее в основной части алгоритма для каждого продукта a из множества НОВЫЕ, куда вначале помещается X, и каждого процесса t, в условие которого входит a, из СЧЕТ$[t]$ вычитается 1. Если СЧЕТ$[t]$ становится равным 0, это означает, что все продукты из L_t уже получены и тогда его результат b_t добавляется в НОВЫЕ, если его там ранее не было. Для поиска таких t используется СПИСОК$[a]$. Во множестве ОБНОВА хранятся уже полученные продукты из НОВЫЕ, которые еще не использовались для запуска новых процессов. Множества продуктов НОВЫЕ и ОБНОВА реализуются булевскими массивами длины $|A|$, единицы которых объединены в списки.

Алгоритм БыстроеЗамыкание(X,F)

```
I) Инициализация:
ДЛЯ КАЖДОГО процесса  t ∈ F   ВЫПОЛНЯТЬ
      { СЧЕТ[t] := |L_t|;
      ДЛЯ КАЖДОГО a ∈ L_t ВЫПОЛНЯТЬ
            добавить t в СПИСОК[a];
      } ;
НОВЫЕ := X;  ОБНОВА := X;
II) Вычисление:
ПОКА   ОБНОВА ≠ ∅  ВЫПОЛНЯТЬ
      { выбрать a ∈ ОБНОВА; ОБНОВА := ОБНОВА \{a};
            ДЛЯ КАЖДОГО t ∈ СПИСОК[a]  ВЫПОЛНЯТЬ
            { СЧЕТ[t] := СЧЕТ[t] − 1;
                  ЕСЛИ СЧЕТ[t] = 0
                  ТО
                     ЕСЛИ b_t ∉ НОВЫЕ
                     ТО { НОВЫЕ := НОВЫЕ ∪{b_t};
                     ОБНОВА := ОБНОВА ∪ {b_t} }
            }
      } ;
вернуть(НОВЫЕ).
```

Теорема 3. *Алгоритм БыстроеЗамыкание(X,F) строит замыкание $Cl(X,F)$.*

Отметим, что число шагов алгоритма БыстроеЗамыкание(X,F) пропорционально размеру его входа, т.е. числу продуктов в F и X или числу букв в записи формулы (*). Такие алгоритмы называются *работающими в линейное (от размера входа) время* или, просто, *линейными*. Действительно, при инициализации каждый элемент F рассматривается 2 раза, а в основном вычислении число рассматриваемых элементов и операций -1 в стр.10 также не превосходит размера входа.

Пример 2. Рассмотрим работу алгоритма **БыстроеЗамыкание** на следующем примере. Пусть $A = \{a, b, c, d, e, f, g, h\}$, а множество F состоит из следующих 6 процессов:

(1) $a, b, c, h \to d$; (3) $g, b \to e$; (5) $f, e \to d$;
(2) $b, c, d \to a$; (4) $e, f \to c$; (6) $b, f \to g$.

Пусть $X = \{b, f\}$,

Тогда при инициализации будут построен массив СЧЕТ $= [4, 3, 2, 2, 2, 2]$ и следующие списки:

СПИСОК$[a] = (1)$ СПИСОК$[e] = (4, 5)$
СПИСОК$[b] = (1, 2, 3, 6)$ СПИСОК$[f] = (4, 5, 6)$
СПИСОК$[c] = (1, 2)$ СПИСОК$[g] = (3)$
СПИСОК$[d] = (2)$ СПИСОК$[h] = (1)$

Множества ДОБАВКА и НОВЫЕ будут инициализированы в стр. 6 булевскими массивами 01000100 с 1 на местах, соответствующих продуктам b и f. Дальнейшие изменения этих структур представлены в следующей таблице.

	С	Ч	Е	Т		ДОБАВКА	НОВЫЕ
1	**2**	**3**	**4**	**5**	**6**	abcdefgh	abcdefgh
4	3	2	2	2	2	01000100	01000100
3	2	1	2	2	1	00000100	01000100
3	2	1	1	1	0	00000010	01000110
3	2	0	1	1	0	00001000	01001110
3	2	0	0	0	0	00110000	01111110
2	1	0	0	0	0	00010000	01111110
2	0	0	0	0	0	10000000	11111110
1	0	0	0	0	0	00000000	11111110

Алгоритм завершает работу, когда множество ДОБАВКА становится пустым. В этот момент результат $Cl(X, F)$ представлен множеством НОВЫЕ. В нашем примере оно равно $\{a, b, c, d, e, f, g\}$.

Задача 6. *Определите, какая последовательность процессов в Примере 2 приводит к получению a.*

Задача 7. *Измените алгоритм* **БыстроеЗамыкание** *так, чтобы по его результату для любого продукта $a \in Cl(X, F)$ можно было построить последовательность процессов, приводящую к a.*

Задача 8. *Используя алгоритм* **БыстроеЗамыкание**, *вычислить замыкание для набора исходных атрибутов $X = \{a, f\}$ и следующей системы зависимостей F :*

1) $a, b, c \to h$; 3) $e, f \to c$; 5) $g, d \to e$;
2) $a, c, d, g \to h$; 4) $f, a \to d$; 6) $d, f, a \to g$.

Определите, какая последовательность процессов приводит к получению h.