1. Обратный и прямой вывод в задачах синтеза программ, вывода из хорновских формул и замыкания функциональных зависимостей

Определение 1. *Вычислительная модель по Тьюгу – это совокупность A величин (параметров), характеризующих некоторую предметную область, и некоторое множество предложений (аксиом) вычислимости F, описывающих законы этой области. Будем рассматривать простые предложения вычислимости вида*

$$a_1, a_2, \ldots, a_r \longrightarrow_f b_1, b_2, \ldots, b_m,$$

означающие, что функция (программа) f по входным величинам $a_1, a_2, \ldots, a_r$ вычисляет значения величин $b_1, b_2, \ldots, b_m$. Далее мы будем считать, что $m = 1$.

В вычислительной модели (A, F) по величинам $x_1, ..., x_n$ можно вычислить величину y, если существует линейная программа P с входными переменными $x_1, ..., x_n$ выходной переменной y и командами присваивания вида $b := f(a_1, ..., a_r)$, соответствующими предложениям F.

Вычислительная задача — это четверка $Q = (A, F, X, y)$, где (A, F) – вычислительная модель, $X \subseteq A$ и $y \in A$. Вычислительная задача Q *разрешима,* если в модели (A, F) можно по X вычислить y. соответствующая программа P называется *решением* Q.

Определение 2. *Пусть A — это множество пропозициональных (булевых) переменных. Хорновская (H-) формула — это формула вида*

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

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 ($F \models \phi$).

Определение 3. *Схема отношения R — множество атрибутов. Пусть r — отношение (таблица) со схемой R, X и Y — подмножества R. Отношение r удовлетворяет функциональной (F-) зависимости $X \to Y$, если любые два кортежа r, имеющие одинаковые X-координаты, имеют также одинаковые Y-координаты.*

Множество F-зависимостей F влечет за собой F-зависимость $X \to Y$ ($F \models X \to Y$), если каждое отношение, удовлетворяющее всем зависимостям из F, удовлетворяет также зависимости $X \to Y$.

Теорема 1. *Пусть $Q = (A, F, X, y)$ — вычислительная задача. Тогда следующие утверждения эквивалентны:*
(1) задача Q разрешима;
(2) формула $\phi = \bigwedge_{x \in X} \to y$ является логическим следствием формул F;
(3) множество зависимостей F влечет F-зависимость $X \to y$.

Нас будут интересовать алгоритмы проверки разрешимости вычислительных задач вида $Q = (A, F, X, y)$. (Эти же алгоритмы решают сответствующие задачи для функциональных зависимостей и для Хорновских формул).

Алгоритм поиска вывода от цели (обратная волна)

Идея: в процессе построения программы (вывода) для очередной величины a (текущей цели) ищем первую еще не использованную для ее получения аксиому в F и заменяем a на ее левую часть и т.д.

Будем считать, что аксиомы F занумерованы числами от 1 до N. Основная структура данных — стек недостигнутых подцелей S, ячейки которого могут

содержать элементы вида (a, i), где $a \in A$, i — номер правила из F, с помощью которого можно вычислить a, или скобку $'['$, означающую начало вывода для стоящей левее подцели. Кроме того, в переменной G будут храниться достигнутые подцели (выведенные атрибуты).

АЛГОРИТМ ОВ

```
    BEGIN
1   ВТОЛК(S,(y,0)); G := X;
2   P := ∅; /* программа
3   WHILE  S ≠ ∅ AND y ∉ G DO
4         BEGIN  T :=ВЕРХ(S);
5         IF  T =' ['       /* цель достигнута
6         THEN ВЫТОЛК(S);
7               (a,i) :=ВЕРХ(S);    * аксиома i подошла для a
8               Пусть аксиома i = x_1,...,x_k →_{f_i} a;
9               P := "P; a := f_i(x_1,...,x_k)";  G := G ∪ {a};
10              ВЫТОЛК(S)
11        ELSE
12              Пусть T = (a,i);
13              Найти первое j > i такое, что аксиома j = y_1,...,y_m → a  ∈ F AND
                ни для какого y_k,  (1 ≤ k ≤ m) в стеке S нет пары (y_k, i_k) с i_k > 0;
14              IF нет такого j       /* вывод неудачен
15              THEN WHILE T ≠' [' DO ВЫТОЛК(S) ENDDO;
16                   ВЫТОЛК(S);     /* на поиск след. аксиомы
17              ELSE BEGIN ВЫТОЛК(S);
18                   ВТОЛК(S,(a,j));    /* очередная цель
19                   ВТОЛК(S,' [');
20                   FOR k = 1 TO m DO       /* добавление подцелей
21                      IF (y_k ∉ G))        /* y_k еще не вычислен
22                      THEN ВТОЛК(S,(y_k,0)) ENDIF
23                   ENDDO
24        ENDIF ENDIF
25        ENDDO;
26  IF y ∈ G THEN result := "да" ELSE result := "нет" ENDIF
    END
```

Теорема 2. *Алгоритм ОВ корректен, т. е. задача Q разрешима $\Leftrightarrow$ result = "да"и программа P правильно вычисляет y.*

Пример Рассмотрим множество атрибутов $A = \{a, b, c\} \cup \{a_i, b_i | 1 \le i \le n\}$ и следующую систему аксиом F:

$a_1 \to a;\ b_1 \to a;$

$a_i \to a_{i-1};\ b_i \to a_{i-1};\quad 2 \le i \le n$

$a_i \to b_{i-1};\ b_i \to b_{i-1};\quad 2 \le i \le n$

Задача. *Докажите, что для решения вычислительной задачи $Q = (A, F, \{c\}, a)$ алгоритм ОВ затратит не менее $O(2^n)$ шагов.*

Алгоритм поиска вывода от данных (прямая волна)

Пусть F - множество функциональных зависимостей, X — исходное множество атрибутов. Определим замыкание X с помощью F как

$Cl(X, F) = \{y |\ y \in A\ \&\ F \models X \to y\}$.

Алгоритм ЗАМЫКАНИЕ(X,F)

1. $OLD := \emptyset$; $NEW := X$;
2. WHILE $NEW \neq OLD$ DO
3. BEGIN $OLD := NEW$;
4. FOR EACH $(W \to Z) \in F$ DO
5. IF $W \subseteq NEW$ THEN $NEW := NEW \cup Z$;
6. END;
7. return(NEW).

Алгоритм ПрямаяВолна(X,Y,F)

1. $Z :=$ ЗАМЫКАНИЕ(X, F);
2. IF $Y \subseteq Z$ THEN return("ДА") ELSE return("НЕТ").

Теорема 3. *Алгоритм ЗАМЫКАНИЕ(X,F) возвращает множество $Cl(X, F)$, а алгоритм ПрямаяВолна(X,Y,F) выдает ответ "ДА"тогда и только тогда, когда $F \models X \to Y$.*

Время работы обоих алгоритмов не превосходит $O(n^2)$, где n — размер их входов.

Алгоритм БыстроеЗамыкание(X,F)

Структуры данных: СЧЕТ[j] — число неизвестных атрибутов в правой части правила j, СПИСОК[a] — список номеров правил, в правую часть которых входит атрибут a, множества атрибутов NEW, UPDATE и ADD реализуются булевскими массивами длины $|A|$, единицы которых объединены в списки.

I) Инициализация:

1. FOR EACH зависимости $j = W \to Z \in F$ DO
2. BEGIN СЧЕТ[j] := $|W|$;
3. FOR EACH $a \in A$
4. DO СПИСОК[a] :=СПИСОК[a] $\cup \{j\}$;
5. END;
6. $NEW := X$; $UPDATE := X$;

II) Вычисление:

7. WHILE $UPDATE \neq \emptyset$ DO
8. BEGIN выбрать $a \in UPDATE$;
9. $UPDATE := UPDATE \setminus \{a\}$;
10. FOR EACH $j = W \to Z \in$ СПИСОК[a] DO
11. BEGIN СЧЕТ[j] := СЧЕТ[j] $- 1$;
12. IF СЧЕТ[j] $= 0$
13. THEN BEGIN
14. $ADD := Z \setminus NEW$;
15. $NEW := NEW \cup ADD$;
16. $UPDATE := UPDATE \cup ADD$
17. END
18. END
19. END ;
20. return(NEW).

Теорема 4. *Алгоритм БыстроеЗамыкание(X,F) строит замыкание $Cl(X, F)$ за линейное время.*