Белорусский государственный университет информатики и радиоэлектроники

Дискретная математика

Литература

1. А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. Логические основы проектирования дискретных устройств . – М.: Физматлит, 2007. – 589 с.

2. А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. Основы логического проектирования. В 3 книгах. – Минск: ОИПИ НАН Беларуси, 2004, 2004, 2006. – 226 с., 240 с., 254 с.

3. Ю.В. Поттосин. Основы теории проектирования цифровых устройств. – Saarbrücken: LAP LAMBERT Academic Publishing, 2011. – 336 с.

4. Электронные учебно-методические комплексы «Дискретная математика» и «Теория проектирования цифровых устройств и систем». – БГУИР. – 2012.

Основные понятия теории множеств

a является *элементом* множества M: $a \in M$.

a не принадлежит M: $a \notin M$ или $a \mathrel{\overline{\in}} M$.

A является *подмножеством* множества B: $A \subseteq B$.

$A = B$, если $A \subseteq B$ и $B \subseteq A$.

$\varnothing$ – пустое множество. $\varnothing \subseteq M$ для любого M.

$\varnothing$ и M – *несобственные* подмножества множества M.

Если $A \subseteq B$ и $A \neq B$, то A – *собственное* подмножество множества B, $A \subset B$.

2^M – множество всех подмножеств множества M, *булеан*.

Среди элементов булеана 2^M находятся $\varnothing$ и M.

Основные понятия теории множеств

$|M|$ – *мощность* множества M (число элементов).

$2^{|M|}$ – мощность булеана множества M.

Мощность бесконечного множества выражается через соответствие.

Если $|A| = |B|$, то между множествами A и B можно установить *взаимно однозначное соответствие*.

Для бесконечных множеств отношение равномощности устанавливается путем нахождения взаимно однозначного соответствия между их элементами.

Основные понятия теории множеств

Примеры бесконечных множеств:

N = {1, 2, … } – множество натуральных чисел;

Z = { … , – 2, – 1, 0, 1, 2, … } – множество целых чисел

R – множество действительных чисел (рациональные и иррациональные числа).

Множества, равномощные с множеством N, называются *счетными*.

Множество P положительных рациональных чисел счетно.

Множество всех действительных чисел отрезка [0, 1] несчетно. Это *континуум*.

Булеан бесконечного счетного множества также не является счетным множеством.

Способы задания множеств

Перечисление элементов: $A = \{a_1, a_2, \dots, a_n\}$.

Указание свойств элементов: $M = \{x \,/\, x = 2^k,\ k \in \mathrm{N}\}$ – множество натуральных степеней двоек.

Индуктивный способ: бесконечное множество $M = \{1, 2, 4, 8, 16, \dots\}$ задается следующим образом:
1) $1 \in M$; 2) если $m \in M$, то $2m \in M$.

Алгебраический способ.

Визуальное представление множеств (диаграммы Эйлера–Венна).

Булевы векторы. Вводится универсальное множество U (*универсум*). Если $U = \{a, b, c, d, e\}$ и $M = \{a, b, d\}$. Тогда M задается вектором 11010.

$\varnothing$ и U задаются векторами 00000 и 11111

Операции над множествами

Объединение множеств A и B:

$$A \cup B = \{x \,/\, x \in A \text{ или } x \in B\}.$$

Пересечением множеств A и B:

$$A \cap B = \{x \,/\, x \in A \text{ и } x \in B\}.$$

Разность множеств A и B:

$$A \setminus B = \{x \,/\, x \in A \text{ и } x \notin B\}.$$

Сумма или *симметрическая разность* множеств A и B:

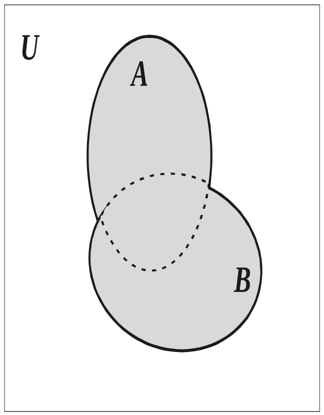
$$A + B = \{x \,/\, (x \in A \text{ и } x \notin B) \text{ или } (x \in B \text{ и } x \notin A)\}.$$

Дополнение множества A:

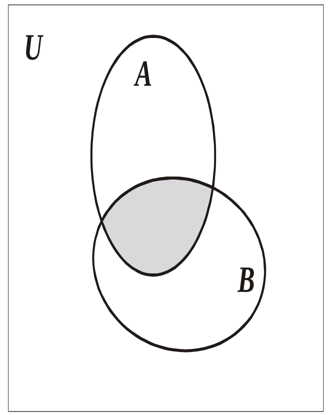
$$\overline{A} = \{x \,/\, x \in U \text{ и } x \notin A\}.$$

Операции над множествами

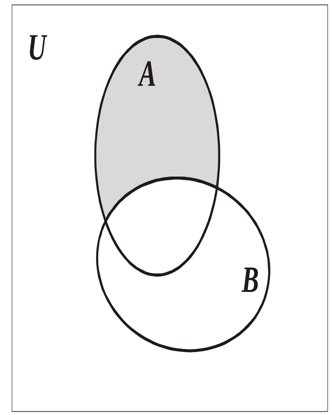
Диаграммы Эйлера-Венна



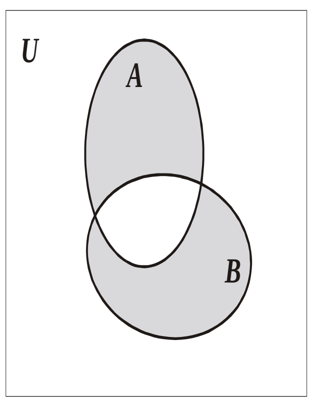
$A \cup B$



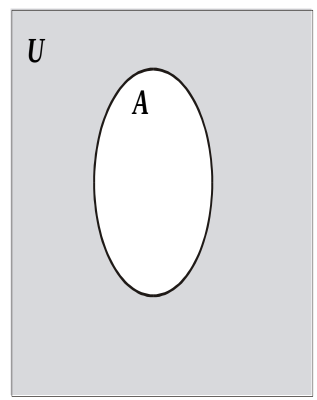
$A \cap B$



$A \setminus B$



$A + B$



$\overline{A}$

Операции над множествами

Некоторые операции выражаются через другие:

$$A + B = (A \cap \bar{B}) \cup (\bar{A} \cap B) = (A \cup B) \setminus (A \cap B);$$

$$\bar{A} = U \setminus A;$$

$$A \setminus B = A \cap \bar{B}.$$

Операции $\bar{\ }$, $\cap$ и $\cup$ составляют булеву алгебру множеств.

Основные законы булевой алгебры множеств ($\cup$, $\cap$, $\overline{\ }$)

Коммутативность:

$A \cup B = B \cup A$; $\qquad$ $A\ B = B\ A$.

Ассоциативность:

$A \cup (B \cup C) = (A \cup B) \cup C$; $\qquad$ $A\ (B\ C) = (A\ B)\ C$.

Дистрибутивность:

$A\ (B \cup C) = A\ B \cup A\ C$; $\qquad$ $A \cup B\ C = (A \cup B)\ (A \cup C)$.

Идемпотентность:

$A \cup A = A$; $\qquad$ $A\ A = A$.

Законы де Моргана:

$\overline{A \cup B} = \bar{A}\ \bar{B}$; $\qquad$ $\overline{AB} = \bar{A} \cup \bar{B}$.

Основные законы булевой алгебры множеств ($\cup$, $\cap$, $\overline{\ }$)

Законы операций с константами ($\varnothing$ *и* U):

$A \cup \varnothing = A;$ $\qquad$ $A\ U = A;$

$A \cup U = U;$ $\qquad$ $A\ \varnothing = \varnothing;$

$A \cup \overline{A} = U;$ $\qquad$ $A\ \overline{A} = \varnothing.$

Закон двойного дополнения:

$$\overline{\overline{A}} = A.$$

Принцип двойственности.

Основные законы булевой алгебры множеств ($\cup$, $\cap$, $\bar{\ }$)

Вывод формулы $A \cup B\,C = (A \cup B)\,(A \cup C)$:

По закону дистрибутивности пересечения:

$$(A \cup B)\,(A \cup C) = AA \cup BA \cup AC \cup BC.$$

Используем константу U и закон идемпотентности:

$$A\,A = A = A\,U;$$

$$AA \cup BA \cup AC \cup BC = A\,U \cup BA \cup AC \cup BC.$$

Выносим за скобки A и используем формулу $A \cup U = U$:

$$A\,U \cup BA \cup AC \cup BC = A\,(U \cup B \cup C) \cup B\,C = A \cup B\,C.$$

Отношения

$A \times B$ – декартово произведение множеств A и B.

$A \times B = \{(a, b) / a \in A, b \in B\}$

Если $A = \{a, b, c\}$ и $B = \{l, m\}$, то

$A \times B = \{(a, l), (b, l), (c, l), (a, m), (b, m), (c, m)\}$.

$\mathsf{R} \times \mathsf{R} = \mathsf{R}^2$ – множество координат точек на плоскости.

Обобщение:

$A_1 \times A_2 \times \ldots \times A_n = \{(a_1, a_2, \ldots, a_n) / a_1 \in A_1, a_2 \in A_2, \ldots, a_n \in A_n\}$.

Отношения:

унарное	$R \subseteq A$;
бинарное	$R \subseteq A_1 \times A_2$;
тернарное	$R \subseteq A_1 \times A_2 \times A_3$;
n-арное	$R \subseteq A_1 \times A_2 \times \ldots \times A_n$.

Бинарные отношения (соответствия)

$R \subseteq A \times B$.

$(a, b) \in R$ можно записывать как $a\ R\ b$ – a и b находятся в отношении R.

Пример:

$a\ R\ b$ – a есть делитель b.

Пусть $A = \{1, 2, 3\}$ и $B = \{1, 2, 3, 4, 5, 6\}$. Тогда

$R = \{(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 2), (2, 4), (2, 6), (3, 3), (3, 6)\}$.

Представления бинарных отношений

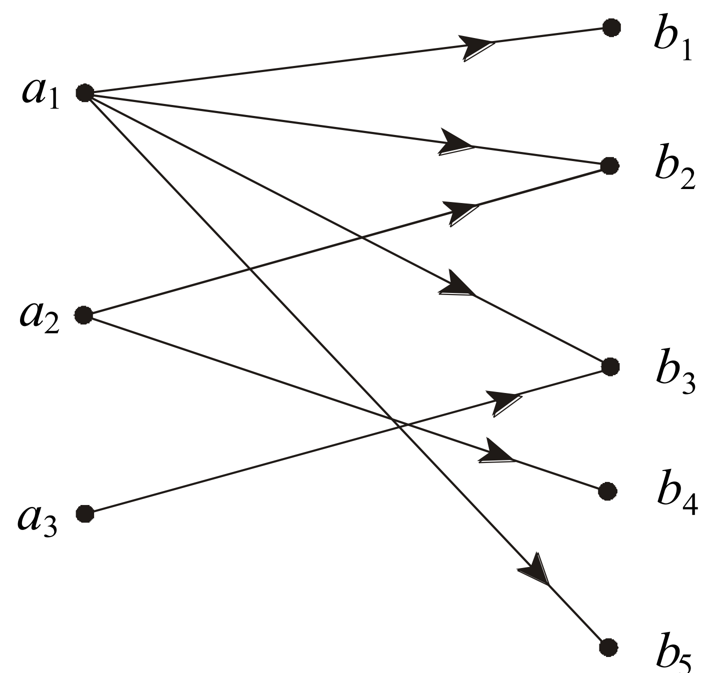
Отношение R между элементами множеств $A = \{a_1, a_2, a_3\}$ и $B = \{b_1, b_2, b_3, b_4, b_5\}$:

$R = \{(a_1, b_1), (a_1, b_2), (a_1, b_3), (a_1, b_5), (a_2, b_2), (a_2, b_4), (a_3, b_3)\}$.

Матричное представление

$$\begin{array}{cc} \begin{array}{ccccc} b_1 & b_2 & b_3 & b_4 & b_5 \end{array} & \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} & \begin{array}{c} a_1 \\ a_2 \\ a_3 \end{array} \end{array}$$

Графическое представление



Бинарные отношения

Обратное отношение R^{-1} для отношения $R \subseteq A \times B$:

$R^{-1} = \{(b, a) \,/\, (a, b) \in R\}$.

$$\boldsymbol{M}(R) = \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}, \quad \boldsymbol{M}(R^{-1}) = \boldsymbol{M}^{T}(R) = \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}.$$

Функциональные отношения

$R \subseteq A \times B$ является *функциональным отношением*, если $|\{(a, b) \,/\, (a, b) \in R, b \in B\}| \leq 1$ для каждого $a \in A$.

С ним связана *функция* $f : A \to B$.

Используется запись $f(x) = y$ для $(x, y) \in R$.

x – *аргумент*.

y – *значение функции* f.

$$\begin{array}{cc} \begin{array}{ccc} b & d & e \end{array} & \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} & \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \end{array}$$

$f(a) = f(c) = b;$

$f(b) = f(e) = d;$

$f(d) = e.$

Если R^{-1} для функционального R, также функциональное, то R – *взаимно однозначное отношение*.

Бинарные отношения на множестве

$R \subseteq A \times A$.

Возможные свойства:

рефлексивность: если $a = b$, то $a\ R\ b$;

иррефлексивность: если $a\ R\ b$, то $a \neq b$;

симметричность: если $a\ R\ b$, то $b\ R\ a$;

антисимметричность: если $a\ R\ b$ и $b\ R\ a$, то $a = b$;

транзитивность: если $a\ R\ b$ и $b\ R\ c$, то $a\ R\ c$;

дихотомия: если $a \neq b$, то либо $a\ R\ b$, либо $b\ R\ a$.

Бинарные отношения на множестве

Типы бинарных отношений:

Отношение *эквивалентности* рефлексивно, симметрично и транзитивно (равносильность формул, подобие геометрических фигур и т. п.). *Классы эквивалентности*.

Отношение *совместимости* рефлексивно и симметрично (близость чисел, знакомство людей и т. п.).

Отношение *нестрогого порядка* рефлексивно, антисимметрично и транзитивно ($\leq$, $\geq$, $\subseteq$, $\supseteq$).

Отношение *строгого порядка* иррефлексивно, антисимметрично и транзитивно ($<$, $>$, $\subset$, $\supset$).

Порядок *полный* (*линейный*), порядок *частичный*.

Лексикографический порядок.

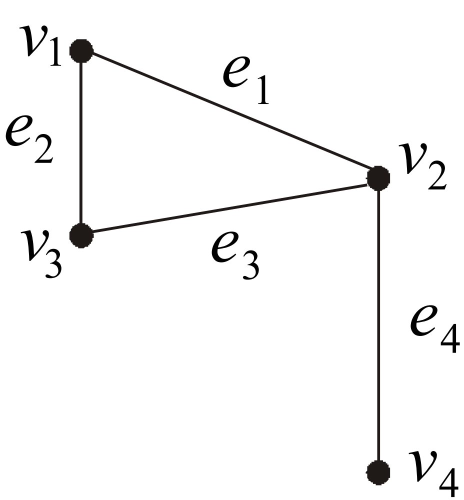
Основные понятия теории графов

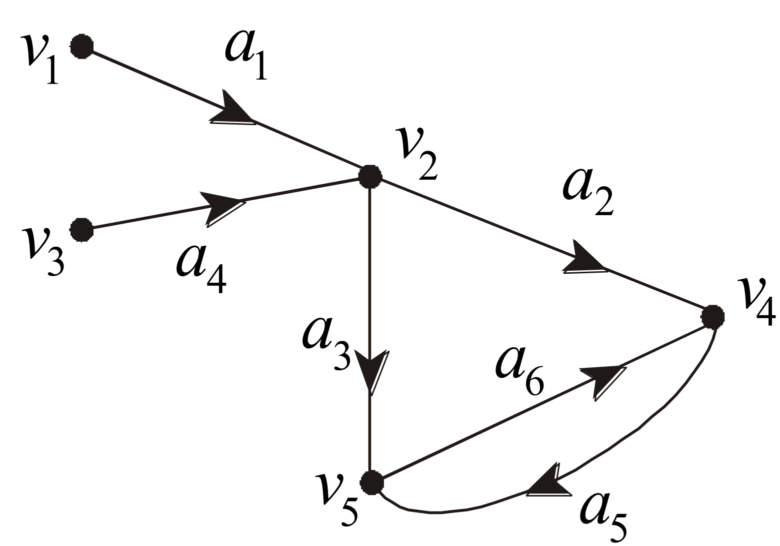
$G = (V, E)$,

V – непустое множество *вершин,*

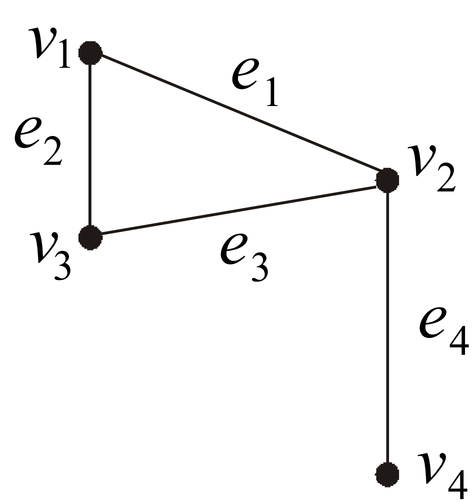
E – произвольное множество *ребер* – пар (v_i, v_j) элементов из V, т. е. $v_i \in V$, $v_j \in V$, $E \subseteq V^2$.

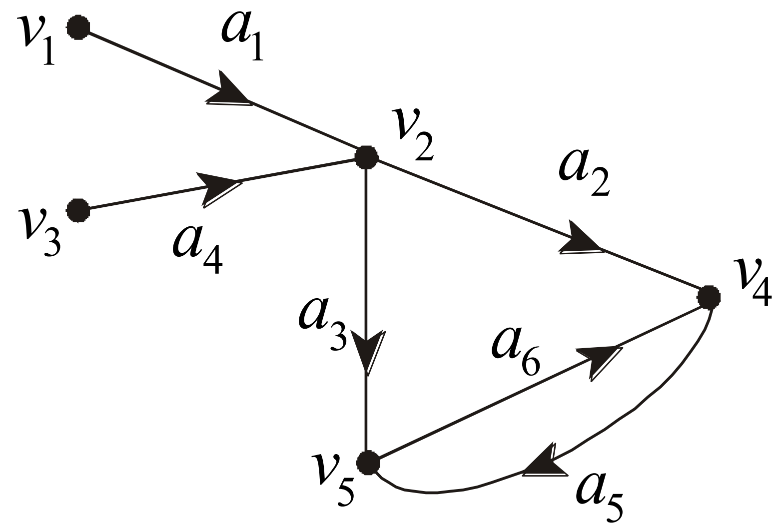
Неориентированный | *Ориентированный* (*орграф*)





Основные понятия теории графов





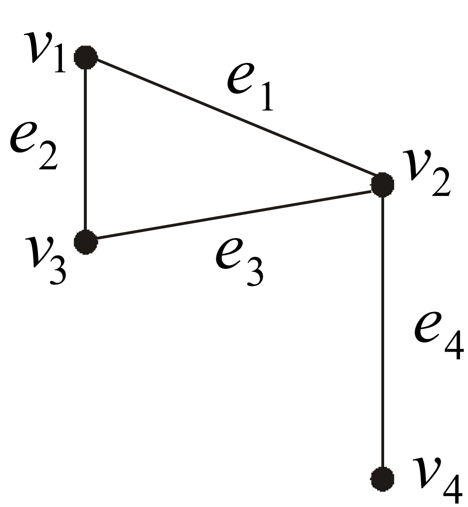
Ребро $e_2 = v_1v_3$ имеет *концы* v_1 и v_3.

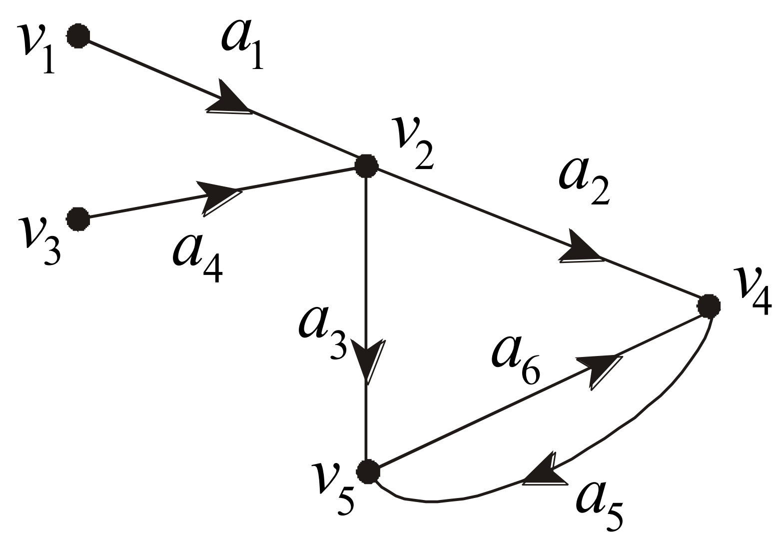
Ориентированное ребро (*дуга*) $a_4 = (v_3, v_2)$ имеет начало v_3 и *конец* v_2 (дуга a_4 исходит из v_3 и заходит в v_2).

В неориентированном графе v и ребро e *инцидентны*, если v – один из концов ребра e.

В орграфе v и *дуга* a *инцидентны*, если v – либо начало, либо конец дуги a.

Основные понятия теории графов





В неориентированном графе две вершины *смежны*, если они инцидентны одному и тому же ребру.

Окрестность $N(v)$ вершины v – множество всех вершин, смежных v.

$|N(v)| = d(v)$ – *степень* вершины v.

В орграфе: *полуокрестность исхода* $N^+(v)$: *полуокрестность захода* $N^-(v)$. *Полустепени* $d^+(v)$, $d^-(v)$.

Основные понятия теории графов

Графы *конечные*, графы *бесконечные*.

Граф $G = (V, E)$ *пустой*, если $E = \varnothing$.

Неориентированный граф *полный*, если любые две его вершины смежны. K_n – полный n-вершинный граф.

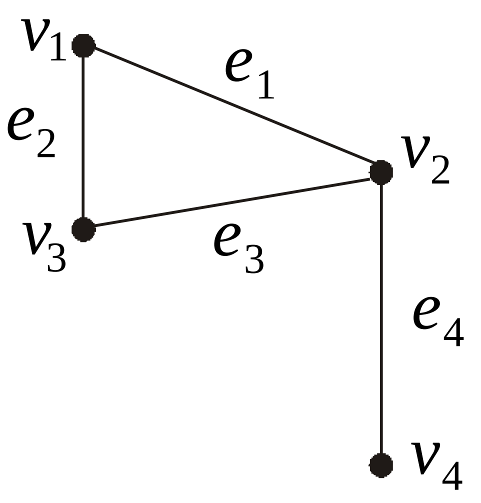
$\overline{G} = (V, \overline{E})$ – *дополнение* графа $G = (V, E)$,

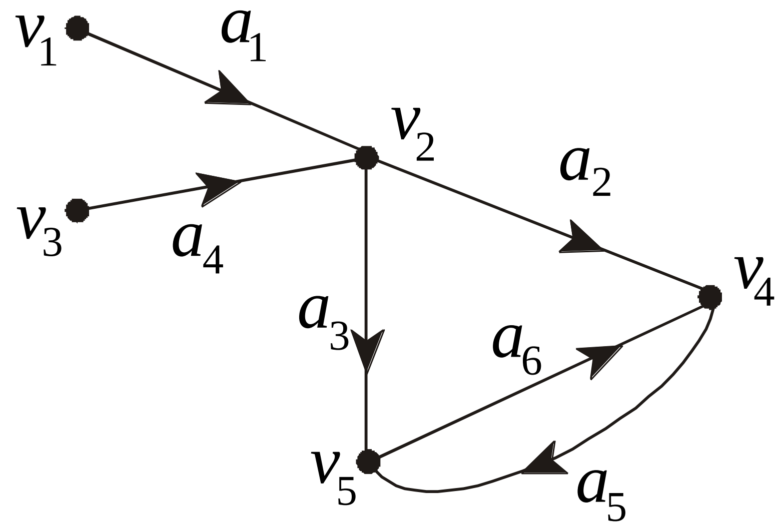
$\overline{E} = U \setminus E$, где U – множество ребер полного графа с множеством вершин V.

Двудольный граф $G = (V', V'', E)$ – для любого $e = xy \in E$: $x \in V'$, $y \in V''$.

Дерево – связный граф без циклов.

Матричные представления графа



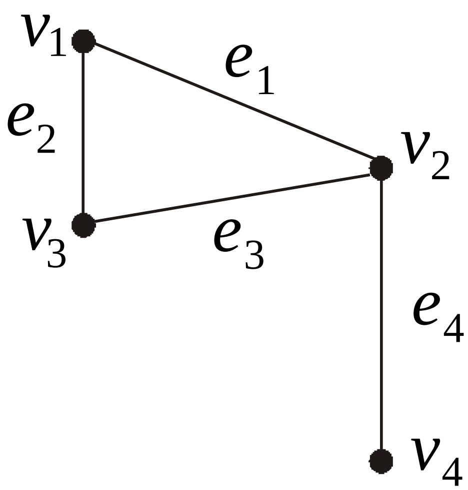


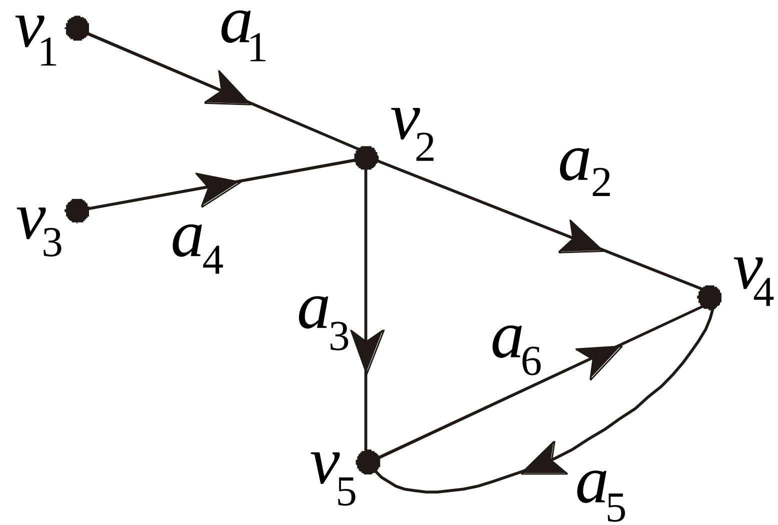
Матрица смежности

$$
\begin{array}{c}
\begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\
\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}
\end{array}
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}
$$

$$
\begin{array}{c}
\begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\
\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}
\end{array}
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
$$

Матричные представления графа





Матрица инцидентности

$$
\begin{array}{cccccc}
e_1 & e_2 & e_3 & e_4 & & \\
\left[\begin{array}{c} 1 \\ 1 \\ 0 \\ 0 \end{array}\right.
& \begin{array}{c} 1 \\ 0 \\ 1 \\ 0 \end{array}
& \begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array}
& \left.\begin{array}{c} 0 \\ 1 \\ 0 \\ 1 \end{array}\right]
& \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}
\end{array}
$$

$$
\begin{array}{c}
\begin{array}{cccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{array} \\
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & -1 & 0 & 0 & 1 & -1 \\
0 & 0 & -1 & 0 & -1 & 1
\end{bmatrix}
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\end{array}
$$

Части графа

$H = (W, F)$ – *подграф* графа $G = (V, E)$, если $W \subseteq V$, $F \subseteq E$.

$H = (W, F)$ – *остовный* подграф, если $W = V$.

$H = (W, F)$ – *подграф, порожденный множеством* W, если F содержит все ребра, оба конца которых принадлежат W.

$v_1, e_1, v_2, e_2, \ldots, e_k, v_{k+1}$ – *маршрут*, $e_i = v_i v_{i+1}$, $i = 1, 2, \ldots, k$.

Длина маршрута – количество ребер.

Цепь – маршрут, все ребра которого различны.

Простая цепь – цепь, все вершины которой различны.

Расстояние между вершинами – длина кратчайшей цепи.

Части графа

Маршрут $v_1, e_1, v_2, e_2, \dots , e_k, v_1$ – *циклический*.

Цикл – циклическая цепь. *Простой цикл*.

Граф *связный*, если любые две его вершины связаны цепью.

Компонента связности графа – связный подграф, не содержащийся ни в каком другом его связном подграфе.

В орграфе:

$v_1, a_1, v_2, a_2, \dots , a_k, v_{k+1}$ – маршрут, если $a_i = (v_i, v_{i+1})$.

Путь – маршрут, где все вершины различны.

$v_1, a_1, v_2, a_2, \dots , a_k, v_1$ – *контур*.

Вершина v_j *достижима* из v_i, если имеется путь из v_i в v_j.

Орграф *сильно связный*, если любая вершина достижима из любой вершины.

Обобщения графов

Мультиграф – граф, в котором любые две вершины могут быть связаны любым количеством ребер (допускает кратные ребра).

Взвешенный граф – вершины и/или ребра снабжаются весами в виде действительных чисел.

Смешанный граф – наряду с элементами ориентированного графа (дугами) имеются элементы неориентированного графа (ребра).

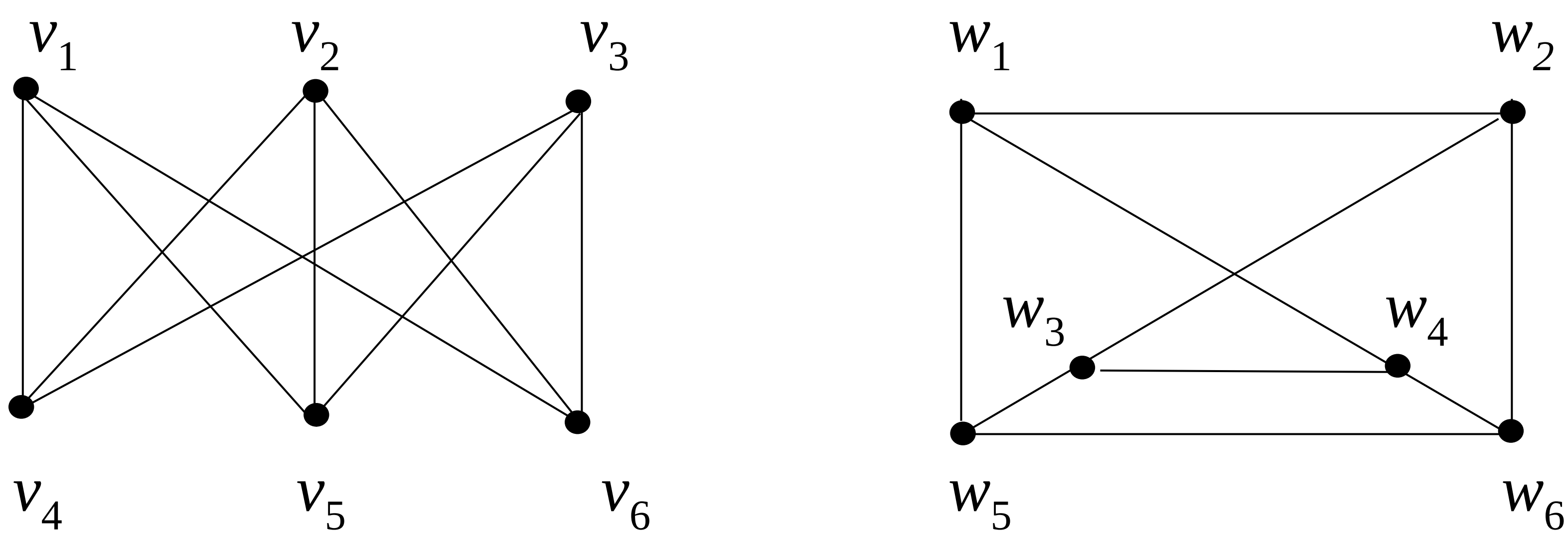
Гиперграф. Если ребром графа является пара вершин, то ребром гиперграфа может быть любое непустое подмножество множества вершин.

От гиперграфа можно перейти к двудольному графу, долями которого являются множество вершин и множество ребер гиперграфа, а ребра показывают принадлежность вершин гиперграфа его ребрам.

Изоморфизм графов

Два графа $G = (V, E)$ и $H = (W, F)$ *изоморфны*, если между их множествами вершин имеется взаимно однозначное соответствие, сохраняющее отношение смежности.

$\varphi : V \to W$, $\psi : E \to F$, и если $\varphi(v_i) = w_k$ и $\varphi(v_j) = w_l$, то $\psi(v_i v_j) = w_k w_l$.



$\varphi(v_1) = w_1$, $\varphi(v_2) = w_3$, $\varphi(v_3) = w_6$, $\varphi(v_4) = w_4$, $\varphi(v_5) = w_5$, $\varphi(v_6) = w_2$.

$\psi(v_1 v_4) = w_1 w_4$, $\psi(v_1 v_5) = w_1 w_5$, $\psi(v_1 v_6) = w_1 w_2$, $\psi(v_2 v_4) = w_3 w_4$, $\psi(v_2 v_5) = w_3 w_5$, $\psi(v_2 v_6) = w_2 w_3$, $\psi(v_3 v_4) = w_4 w_6$, $\psi(v_3 v_5) = w_5 w_6$, $\psi(v_3 v_6) = w_2 w_6$.

Изоморфизм графов

Канонизация графа

Величина *a инвариантна* относительно преобразования T, если она не меняет свое значение при преобразовании T.

a называется *инвариантой* относительно T. В нашем случае T – *перенумерация* вершин.

Инварианты графа:

число вершин, число ребер, число компонент связности…

Инварианты вершины:

степень, полустепени, число вершин, отстоящих от данной вершины на определенном расстоянии…
Канонизация графа заключается в упорядочении его вершин по значениям инвариант.

Пусть для вершин графа имеется система инвариант $\alpha_1, \alpha_2, \ldots, \alpha_p$. Считаем, что задано отношение частичного порядка $\prec$ на множестве вершин графа $V = \{v_1, v_2, \ldots, v_n\}$, такое, что $v_i \prec v_j$, если $\alpha_k(v_i) < \alpha_k(v_j)$ для некоторого $k \in \{1, 2, \ldots, p\}$ и $\alpha_l(v_i) = \alpha_l(v_j)$ для всех $l < k$.

Изоморфизм графов

Канонизация графа

Полная канонизация графа достигается, когда порядок оказывается полным и строгим.

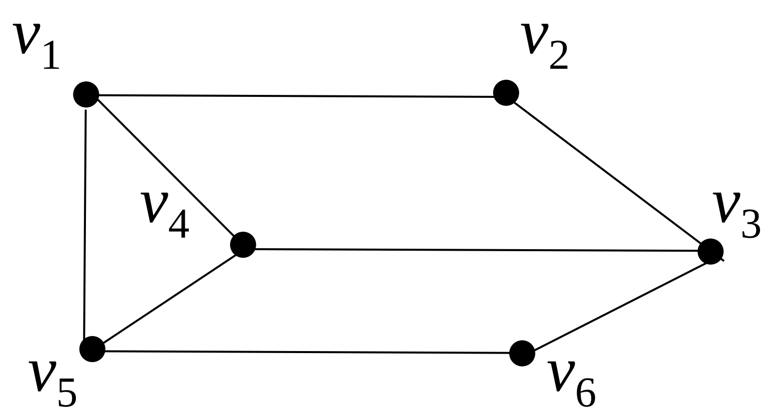
Разобьем множество V вершин графа G на подмножества $S_1, S_2, \ldots , S_m$, число m которых равно числу различных степеней вершин и в каждом из которых присутствуют вершины с одинаковой степенью.

Инварианта вершины $v_i \in V$ – вектор размерности m, компоненты которого соответствуют множествам $S_1, S_2, \ldots , S_m$ и значением j-й компоненты является число вершин из множества S_j, смежных с v_i.

Если в одном и том же S_k ($k = 1, 2, \ldots , m$) оказались вершины с различными векторами, то разобьем это S_k так, чтобы в каждом из получившихся множеств оставались вершины с одинаковыми векторами, соответственно увеличив размерность векторов и придав их компонентам новые значения.

Изоморфизм графов

Канонизация графа



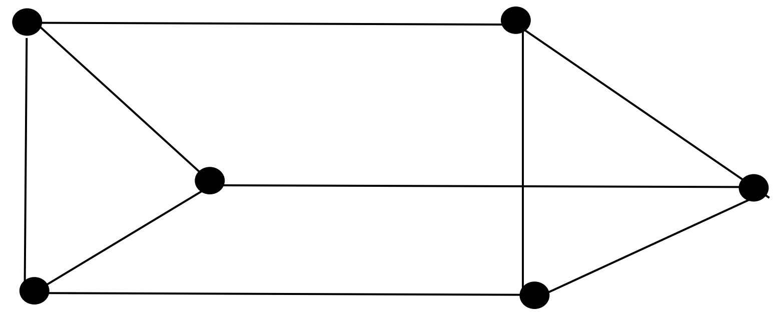
		α
S_1	v_2	2
	v_6	2
	v_1	3
S_2	v_3	3
	v_4	3
	v_5	3

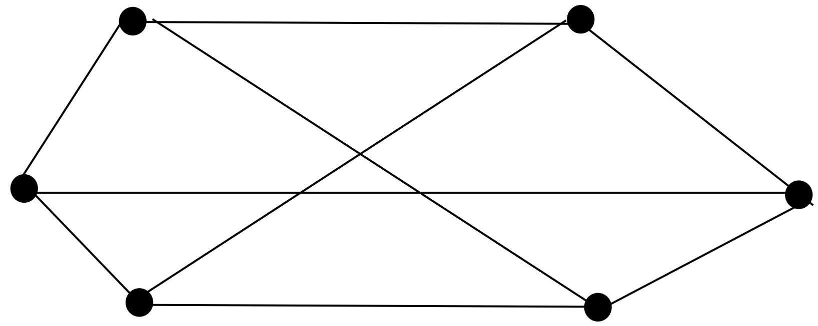
		α_1	α_2
S_1	v_2	0	2
	v_6	0	2
	v_1	1	2
S_2	v_3	2	1
	v_4	0	3
	v_5	1	2

		α_1	α_2	α_3	α_4
S_1	v_2	0	0	1	1
	v_6	0	0	1	1
S_2	v_4	0	0	2	1
S_3	v_1	1	1	1	0
	v_5	1	1	1	0
S_4	v_3	2	1	0	0

Изоморфизм графов

Пример однородных неизоморфных графов





Циклы и разрезы

Цикломатическое число графа

Дерево – это связный граф, число ребер которого на единицу меньше числа вершин.

Дерево – это связный граф, не имеющий циклов.

Дерево – это граф, в котором каждая пара вершин связана одной и только одной цепью.

Лес – граф, каждая компонента связности которого является деревом.

Пусть G – неориентированный граф с n вершинами, m ребрами и p компонентами связности.

Остовное дерево – остовный подграф в виде дерева связного графа (p = 1).

Число ребер в остовном дереве $n - 1$.

Число ребер в остовном лесе $n - p$.

$\nu(G) = m - n + p$ – *цикломатическое число*

$\rho(G) = n - p$ – *коцикломатическое число*.

Циклы и разрезы

Базис циклов

T – остовное дерево связного графа $G = (V, E)$.

Добавление одного ребра из E к T приводит к появлению точно одного простого цикла.

$m - n + 1$ – число таких циклов в графе G. Оно совпадает с $\nu(G)$.

Эти циклы независимы (каждый из них имеет ребро, не принадлежащее никакому другому).

Фундаментальные циклы. Они составляют *базис циклов* графа G.

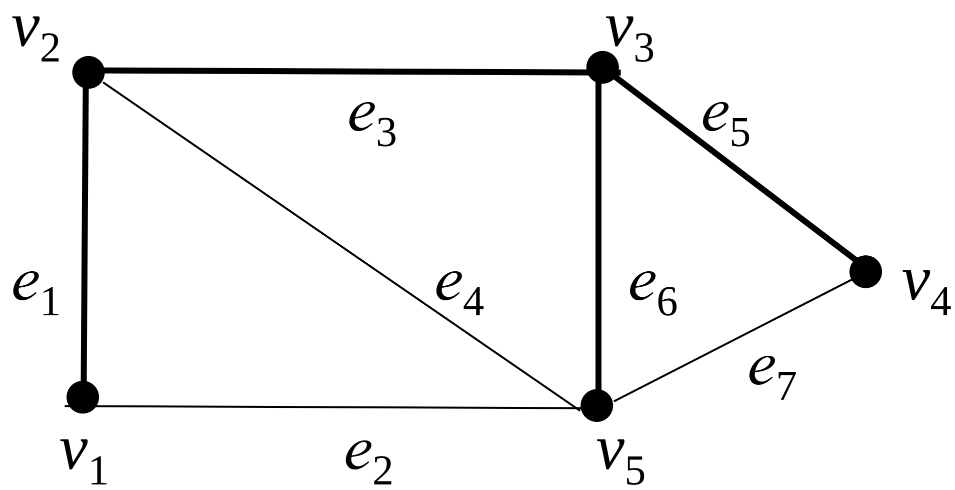
Любой цикл, не принадлежащий базису, может быть выражен в виде линейной комбинации фундаментальных циклов.

Всякий цикл графа G представим m-мерным булевым вектором, в котором i-я компонента имеет значение 1 или 0 в зависимости от того, принадлежит или нет i-е ребро данному циклу. Любой цикл можно выразить как покомпонентную сумму по модулю 2 векторов, представляющих фундаментальные циклы.

Сумма по модулю 2: $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$.

Циклы и разрезы

Базис циклов



Фундаментальные циклы:

$v_1, e_1, v_2, e_3, v_3, e_6, v_5, e_2, v_1$; v_2, v_3, v_5; v_3, v_4, v_5.

$e_1 e_2 e_3 e_4 e_5 e_6 e_7$

0 0 1 **1** 0 1 0

0 0 0 0 1 1 **1**

<u>1 **1** 1 0 0 1 0</u>

1 1 0 1 1 1 1

$e_1 e_2 e_3 e_4 e_5 e_6 e_7$

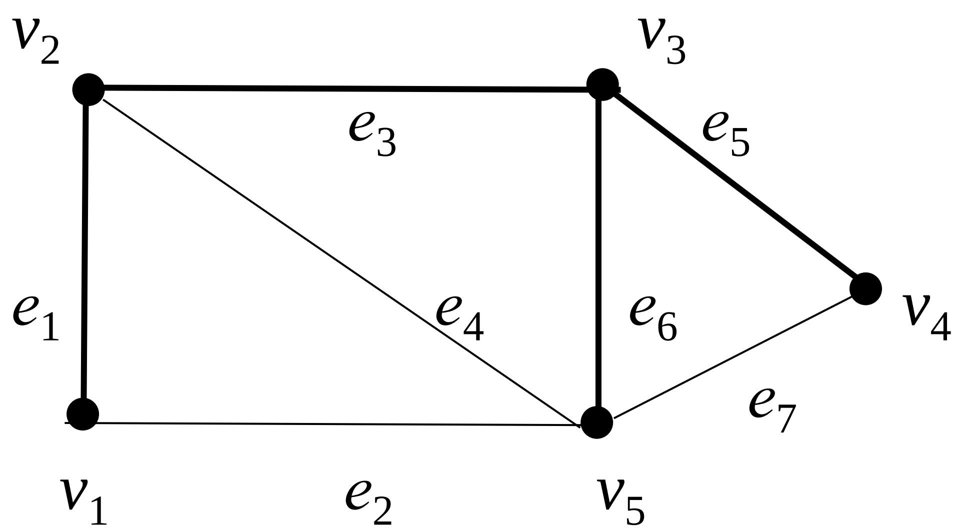
0 0 1 **1** 0 1 0

<u>0 0 0 0 1 1 **1**</u>

0 0 1 1 1 0 1

Циклы и разрезы

Базис разрезов

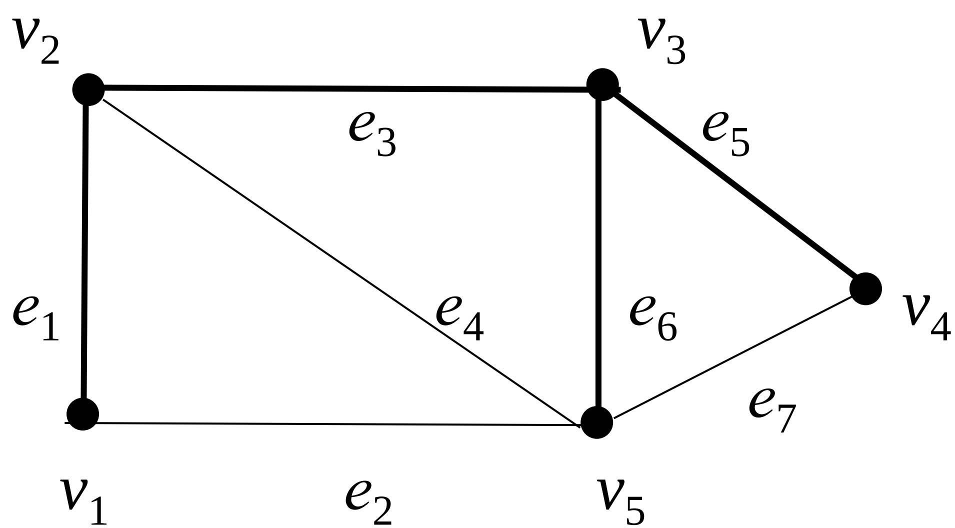


Разрез графа – множество ребер, удаление которых увеличивает число компонент связности.

Под разрезом будем понимать *минимальный* разрез, т.е. такой, что при удалении из него любого ребра он перестает быть разрезом. *Фундаментальный разрез* содержит одно и только одно ребро e, принадлежащее остовному дереву T. Кроме e, он содержит все ребра, не принадлежащие T, но входящие в фундаментальные циклы, содержащие e.

Циклы и разрезы

Базис разрезов

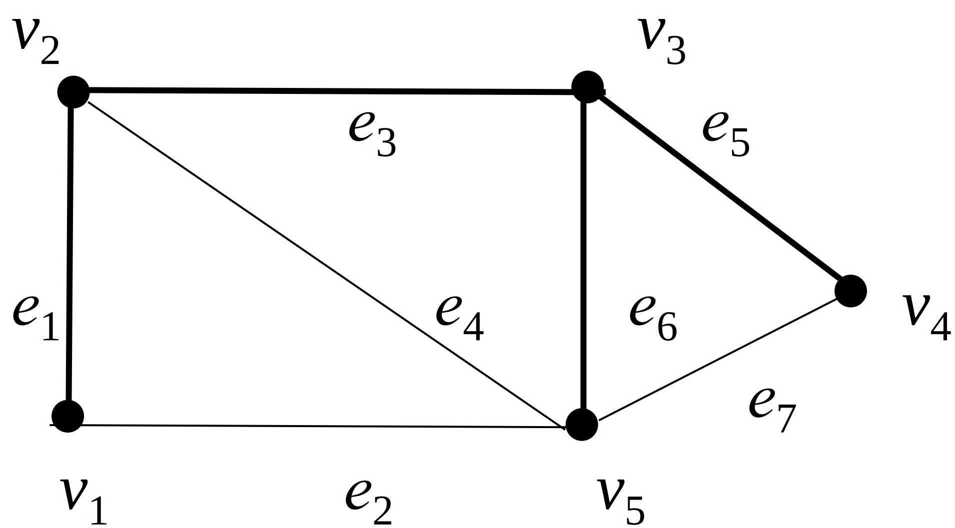


Базис разрезов – множество фундаментальных разрезов.

$$
\begin{array}{ccccccc}
e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \\
0 & 1 & \mathbf{1} & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & \mathbf{1} & 1 \\
\hline
0 & 0 & 1 & 0 & 0 & 1 & 1
\end{array}
$$

Циклы и разрезы

Матрицы циклов и разрезов



Матрица фундаментальных циклов

$$\begin{array}{ccccccc} e_2 & e_4 & e_7 & e_1 & e_3 & e_5 & e_6 \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Матрица фундаментальных разрезов

$$\begin{array}{ccccccc} e_2 & e_4 & e_7 & e_1 & e_3 & e_5 & e_6 \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Циклы и разрезы

Матрицы циклов и разрезов

Матрица фундаментальных циклов

$$\begin{array}{ccccccc} e_2 & e_4 & e_7 & e_1 & e_3 & e_5 & e_6 \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Матрица фундаментальных разрезов

$$\begin{array}{ccccccc} e_2 & e_4 & e_7 & e_1 & e_3 & e_5 & e_6 \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{array}{ccccccc} e_2 & e_4 & e_7 & e_1 & e_3 & e_5 & e_6 \end{array}$$
$$\left[\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ - & - & - & - & - & - & - \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

Доминирующие множества графа

S – *доминирующее* множество ($S \subseteq V$), если $S \cup N(S) = V$, где $N(S) = \bigcup_{v \in S} N(v)$.

Если S – доминирующее множество некоторого графа G, то всякое $S' \supseteq S$ также является доминирующим.

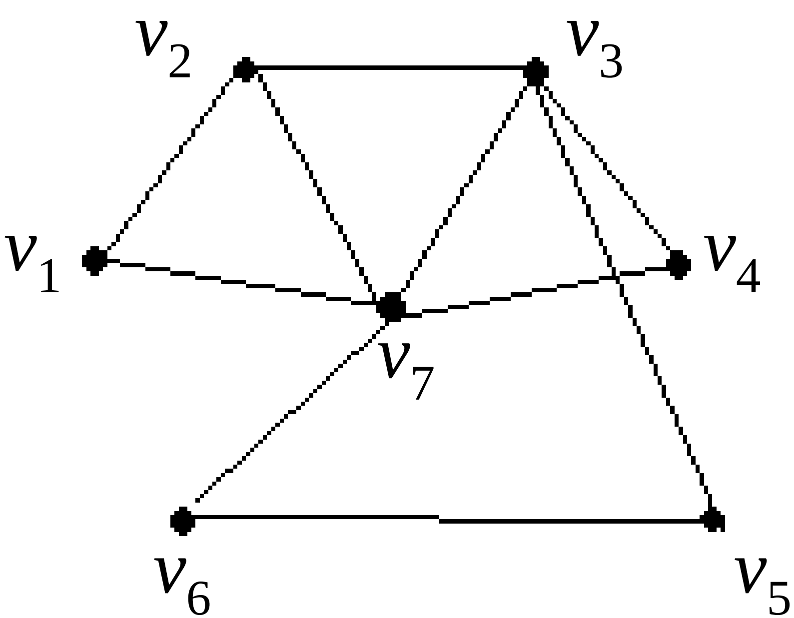
Минимальное доминирующее множество – ни одно его собственное подмножество не является доминирующим.

Наименьшее доминирующее множество – имеет наименьшую мощность $\beta(G)$.

$\beta(G)$ – *число доминирования* графа G.

Задача о ферзях.

Доминирующие множества графа



$$
\begin{array}{c c}
\begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{array} & \\
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 1
\end{bmatrix} &
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array}
\end{array}
$$

Строка v_i матрицы представляет множество $\{v_i\} \cup N(v_i)$.

Минимальные доминирующие множества: $\{v_1, v_3, v_5\}$, $\{v_1, v_3, v_6\}$, $\{v_1, v_4, v_5\}$, $\{v_1, v_4, v_6\}$, $\{v_2, v_3, v_5\}$, $\{v_2, v_3, v_6\}$, $\{v_2, v_4, v_5\}$, $\{v_2, v_4, v_6\}$, $\{v_3, v_7\}$, $\{v_5, v_7\}$ и $\{v_6, v_7\}$.

Независимые множества графа

S – *независимое* множество ($S \subseteq V$), если $S \cap N(S) = \varnothing$.

Если S – независимое множество некоторого графа G, то всякое $S' \subseteq S$ также является независимым.

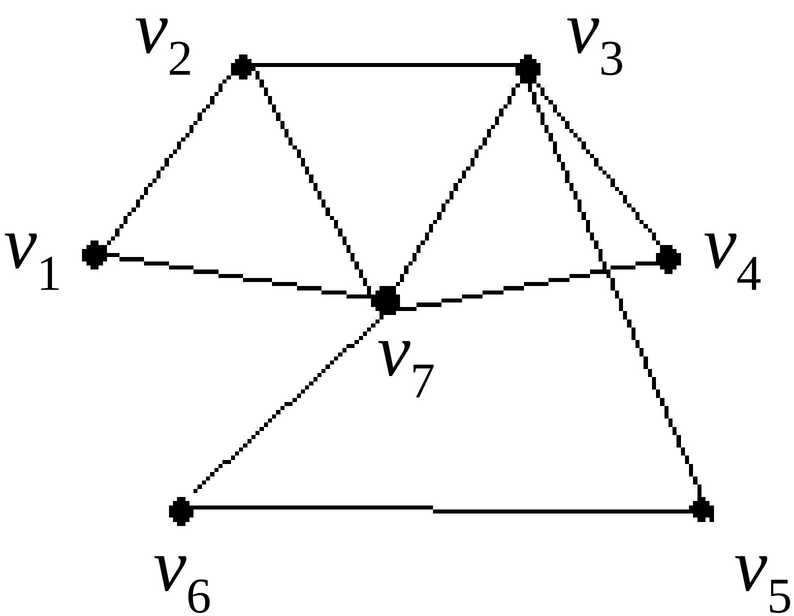
Максимальное независимое множество – не является собственным подмножеством ни одного независимого множества.

Наибольшее независимое множество – имеет наибольшую мощность $\alpha(G)$.

$\alpha(G)$ – *число независимости* графа G.

Задача о ферзях.

Независимые множества графа



$$\begin{array}{ccccccc|c} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 1 & v_1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & v_2 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & v_3 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & v_4 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & v_5 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & v_6 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & v_7 \end{array}$$

Максимальные независимые множества:

$\{v_1, v_3, v_6\}$, $\{v_1, v_4, v_5\}$, $\{v_1, v_4, v_6\}$, $\{v_2, v_4, v_5\}$, $\{v_2, v_4, v_6\}$, $\{v_5, v_7\}$

Независимые множества графа

Нахождение всех максимальных независимых множеств

$$\begin{array}{cc} \begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{array} & \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} & \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array} \end{array}$$

$V = \{v_1, v_2, \ldots , v_n\}$ – множество вершин графа $G = (V, E)$.

$G_1, G_2, \ldots , G_n$ – последовательность порожденных подграфов: $G_i = (V_i, E_i)$, где $V_i = \{v_1, v_2, \ldots , v_i\}$ $(i = 1, 2, \ldots , n)$.

$S^i = \{S_1^i, S_2^i, \ldots , S_{k_i}^i\}$ – совокупность всех максимальных независимых множеств графа G_i.

К каждому S_j^i $(j = 1, 2, \ldots , k_i)$ применяется формула

$$S' = (S_j^i \setminus N(v_{i+1})) \cup \{v_{i+1}\}.$$

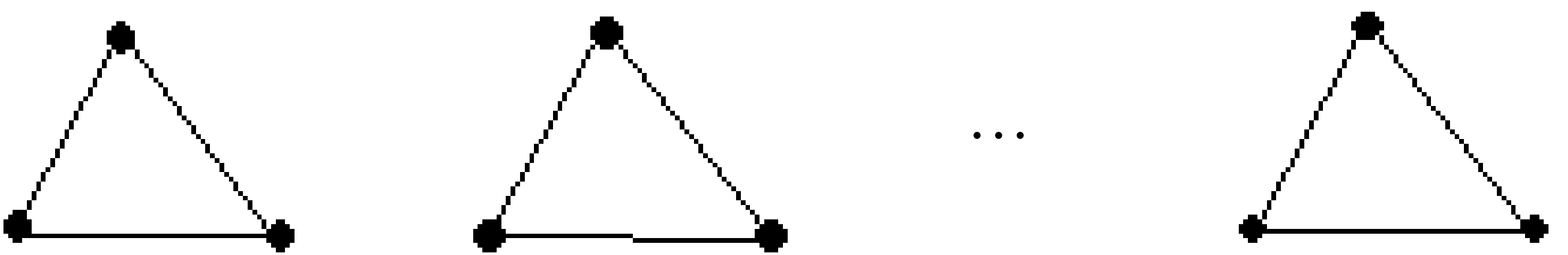
Независимые множества графа

Сколько всего может быть максимальных независимых множеств в графе с n вершинами?

$$2 \cdot 3^{k-1}, \text{ если } n = 3k - 1;$$

$$3 \cdot 3^{k-1}, \text{ если } n = 3k;$$

$$4 \cdot 3^{k-1}, \text{ если } n = 3k + 1.$$

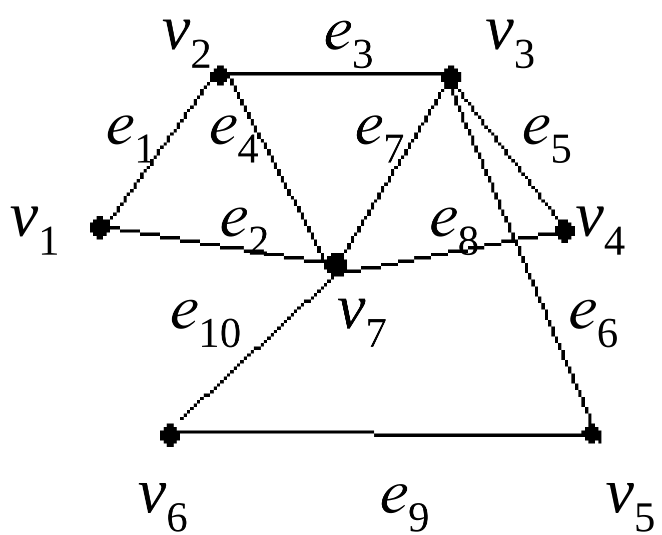


Независимые множества графа

Нахождение наибольшего независимого множества.

Вершинное покрытие графа $G = (V, E)$ – множество $B \subseteq V$ такое, что каждое ребро из E инцидентно хотя бы одной вершине из B.

Если B – наименьшее вершинное покрытие, то $V \setminus B$ – наибольшее независимое множество.

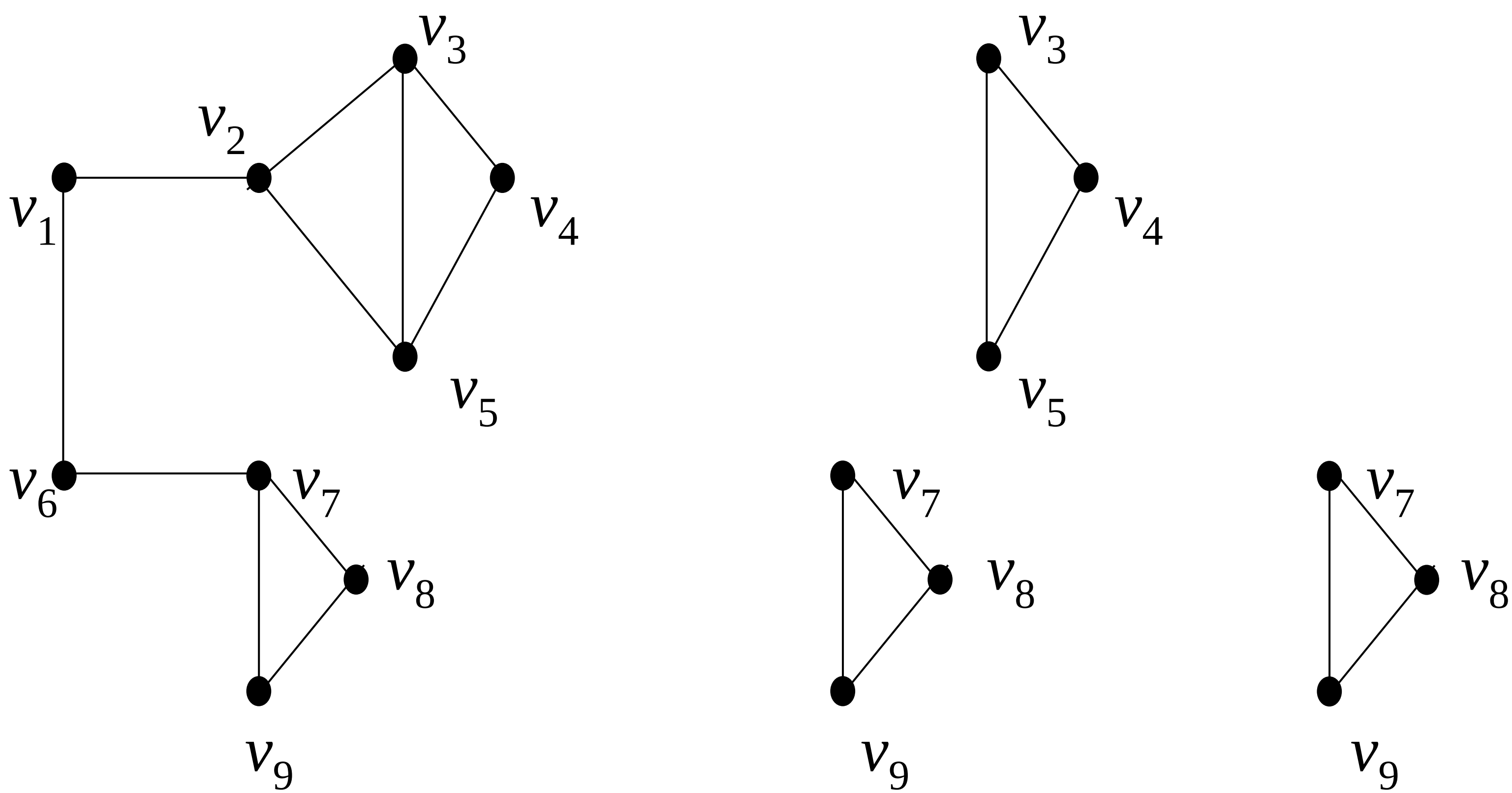


$$
\begin{array}{c}
\begin{array}{cccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} \end{array} \\
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1
\end{bmatrix}
\end{array}
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array}
$$

$B = \{v_1, v_3, v_5, v_7\}, \qquad V \setminus B = \{v_2, v_4, v_6\}.$

Независимые множества графа

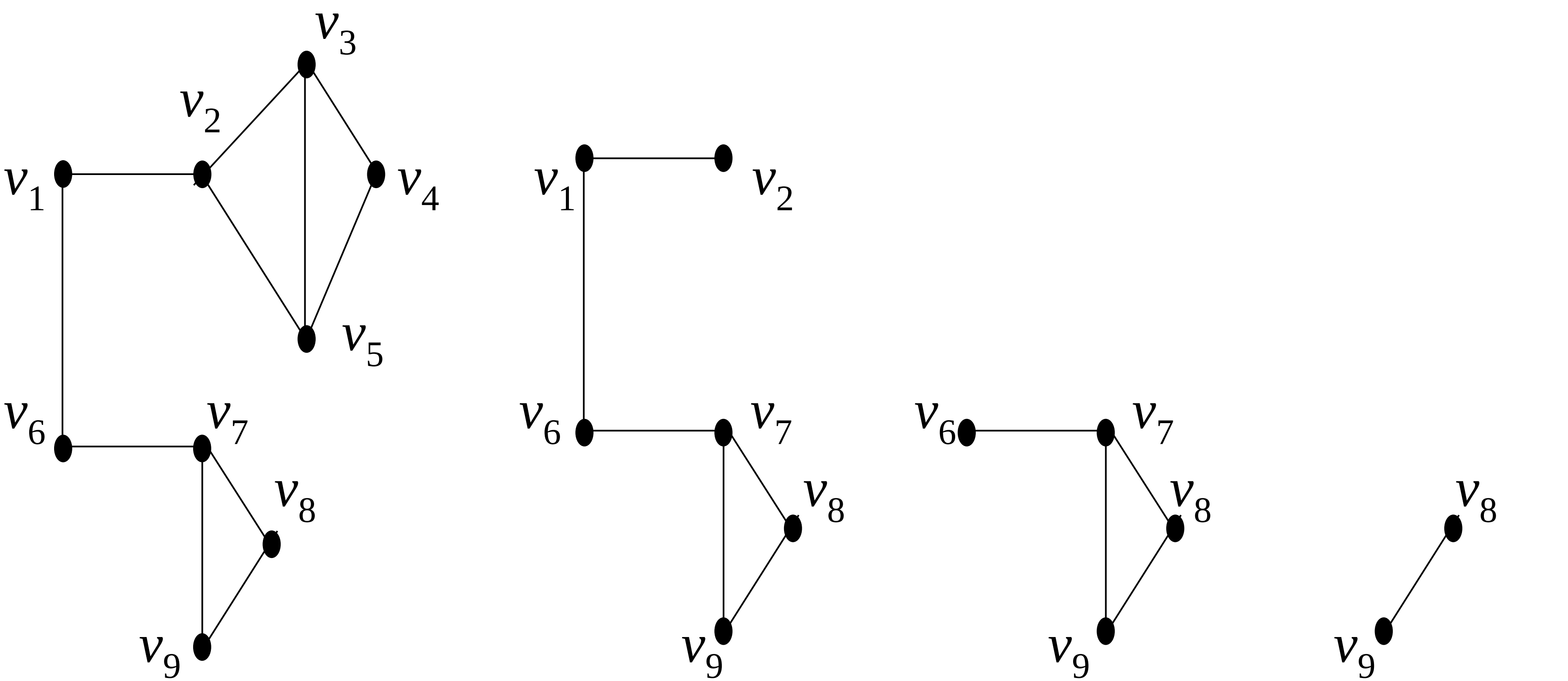
Нахождение наибольшего независимого множества. «Жадный» алгоритм.



$\{v_1\}$ $\{v_1, v_3\}$ $\{v_1, v_3, v_7\}$

Независимые множества графа

Нахождение наибольшего независимого множества. «Жадный» алгоритм.



$\{v_4\}$ $\{v_2, v_4\}$ $\{v_2, v_4, v_6\}$ $\{v_2, v_4, v_6, v_8\}$

Раскраска графа

Раскраска графа $G = (V, E)$ – такое разбиение множества V на непересекающиеся подмножества $V_1, V_2, \dots, V_k$, что никакие две вершины из любого V_i ($i = 1, 2, \dots, k$) не смежны.

Задача: раскрасить вершины графа G в минимальное число цветов.

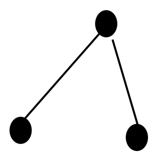
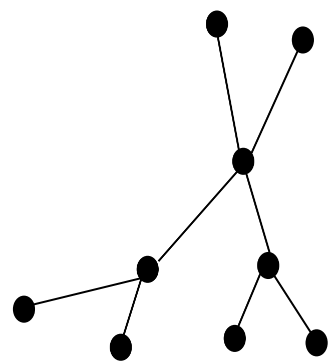
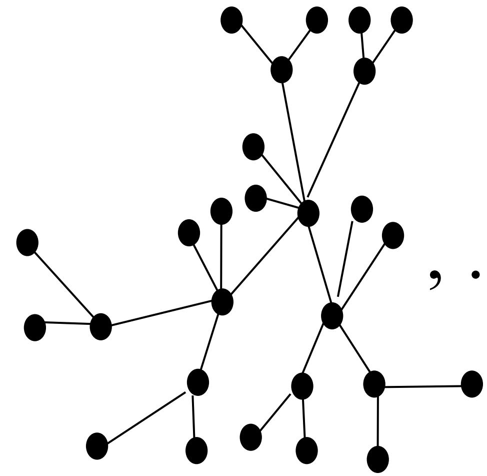
$\gamma(G)$ – *хроматическое число* графа G (минимум k).

Любое V_i ($i = 1, 2, \dots, k$) – независимое множество.

Раскраска $V_1, V_2, \dots, V_k$ – совокупность независимых множеств.

Раскраска графа

Неточность «жадного» алгоритма видна на последовательности:

 ,  ,  , …

Граф G является k-*хроматическим*, если $\gamma(G) = k$.

Т е о р е м а К ё н и г а. Непустой граф является бихроматическим тогда и только тогда, когда он не содержит циклов нечетной длины.

Раскраска графа

Метод раскраски графа

k – число задействованных цветов;

A – множество еще не раскрашенных вершин;

$B_1, B_2, \dots, B_k$ – совокупность подмножеств множества вершин V, такая, что B_i ($i = 1, 2, \dots, k$) содержит те и только те вершины из множества A, которые нельзя раскрасить в i-й цвет.

1. Имеется вершина $v \in A$, такая, что $v \in B_i$ для всех $i = 1, 2, \dots, k$.

v красится в $(k + 1)$-й цвет, удаляется из множества A и из всех B_i. Формируется B_{k+1} и $k := k + 1$. Если таких вершин несколько, из них выбирается та, для которой B_{k+1} имеет максимальную мощность.

2. Имеется вершина $v \in A$ и цвет i, такие, что $v \notin B_i$ и $N(v) \cap A \subseteq B_i$.

v красится в i-й цвет, удаляется из A и из всех B_j.

В остальных случаях выбираются цвет i и вершина v из A такие, что $v \notin B_i$ и приращение $\Delta|B_i|$ минимально среди всех пар v, Bi ($v \in A$, $i = 1, 2, \dots, k$). Вершина v красится в i-й цвет и удаляется из A из всех B_j.

Раскраска графа

Метод раскраски графа

$$\begin{array}{c}
\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\end{array}
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}$$

1. Имеется вершина $v \in A$, такая, что $v \in B_i$ для всех $i = 1, 2, \dots, k$.
2. Имеется вершина $v \in A$ и цвет i такие, что $v \notin B_i$ и $N(v) \cap A \subseteq B_i$

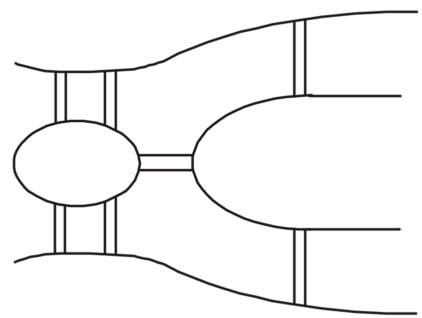
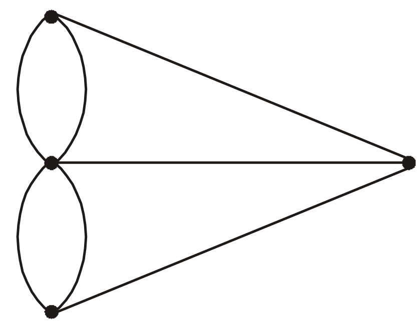
Цвет	Вершины	B_i
1	1	{2,6,8,10}
1	1	{6,8,10}
2	2	{4,5}
1	1	{6,10}
2	2,8	{4,5}
1	1,7	{3,6,10}
2	2,8	{4,5}
1	1,7	{3,10}
2	2,6,8	{4,5,9}
1	1,7	{10}
2	2,3,6,8	{4,5,9}
1	1,7	$\varnothing$
2	2,3,6,8,10	{4,5,9}
1	1,4,7	{5,9}
2	2,3,6,8,10	{5,9}

Результат: {1,4,7}, {2,3,6,8,10}, {5}, {9}. Точный алгоритм дает {1,3,4}, {2,8,9}, {5,6,7,10}.

Обходы графа

Эйлеровы цепи и циклы

Задача о кёнигсбергских мостах (1736 г.)

Эйлеров цикл содержит все ребра графа.

Эйлерова цепь.

Т е о р е м а Э й л е р а. Связный неориентированный граф имеет эйлеров цикл тогда и только тогда, когда степени всех его вершин четны. В связном неориентированном графе существует эйлерова цепь тогда и только тогда, когда он имеет не более двух вершин с нечетной степенью.

Обходы графа

Эйлеровы цепи и циклы

Эйлеров граф – граф, имеющий эйлеров цикл.

Алгоритм Флёри:

1. Идем из некоторой вершины по ребру и удаляем каждое пройденное ребро, помещая его в получаемую последовательность. Начальная вершина выбирается произвольно.

2. Отправляясь из очередной вершины, никогда не идем по ребру, удаление которого делает граф несвязным.

Задача китайского почтальона:

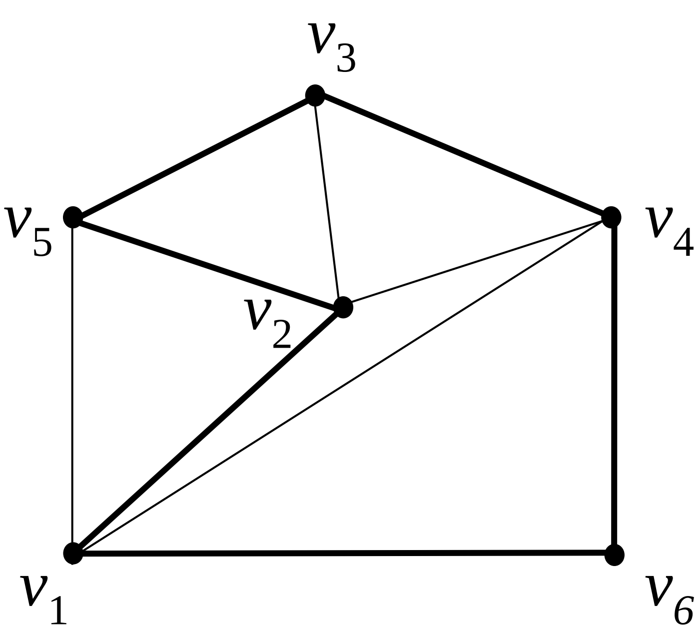
Каждому ребру e_i графа G приписывается положительный вес $c(e_i)$ (расстояние). Требуется найти маршрут, проходящий через каждое ребро графа G по крайней мере один раз и такой, что сумма величин $n_i c(e_i)$ минимальна, где n_i – число прохождений ребра e_i.

Обходы графа

Гамильтоновы цепи и циклы

Цикл называется *гамильтоновым*, если он проходит каждую вершину графа ровно один раз.

Гамильтонова цепь – цепь, проходящая каждую вершину графа ровно один раз. Граф, содержащий гамильтонов цикл, называется *гамильтоновым графом*.



v_1: v_2, v_4, v_5, v_6;
v_2: v_1, v_3, v_4, v_5;
v_3: v_2, v_4, v_5;
v_4: v_1, v_2, v_3, v_6;
v_5: v_1, v_2, v_3;
v_6: v_1, v_4.

$(v_1, v_2, v_3, v_4, v_6)$
(v_1, v_2, v_3, v_4)
(v_1, v_2, v_3)
(v_1, v_2, v_3, v_5)
(v_1, v_2, v_4)
$(v_1, v_2, v_4, v_3, v_5)$
(v_1, v_2, v_4, v_6)
$(v_1, v_2, v_5, v_3, v_4, v_6, v_1)$

Обходы графа

Кратчайшие пути в графе

Ребра графа $G = (V, E)$ взвешены действительными положительными числами. Вес ребра $e = v_i v_j$ будем считать его длиной $l(e) = l(v_i v_j)$.

Найти цепь C минимальной длины, соединяющую вершины v_1 и v_n в графе G, т. е. такую цепь, для которой величина $\sum_{e \in C} l(e)$ минимальна.

Алгоритм Форда

Каждой вершине $v_i \in V$ припишем индекс $\lambda(v_i)$. При этом положим $\lambda(v_1) = 0$ и $\lambda(v_i) = +\infty$ для $i \neq 1$.

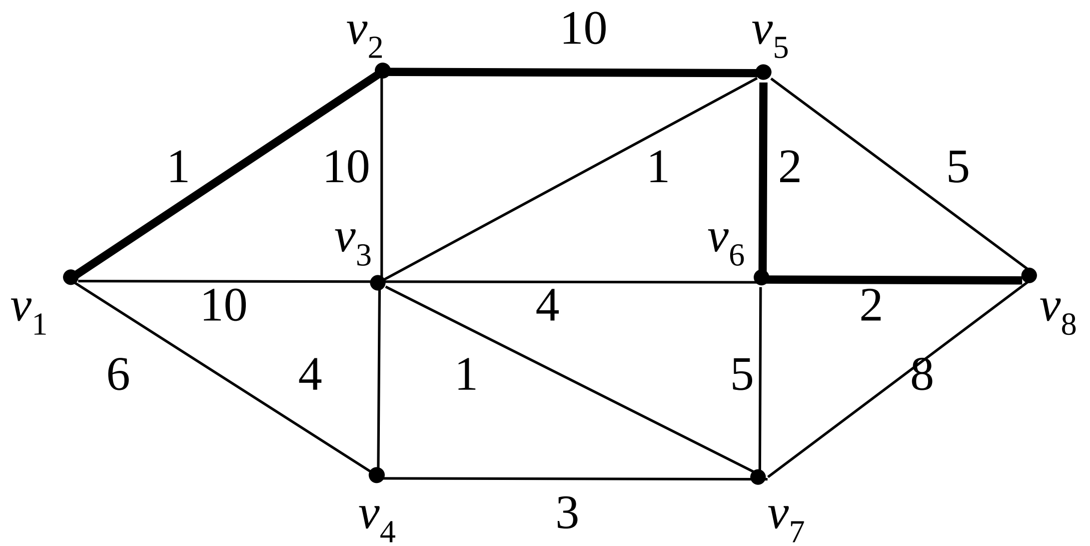
На каждом шаге находим такое ребро $v_i v_j$, что $\lambda(v_i) - \lambda(v_j) > l(v_i v_j)$, и индекс $\lambda(v_i)$ заменяем на $\lambda'(v_i) = \lambda(v_j) + l(v_i v_j)$.

Шаги повторяются, пока находятся ребра, для которых выполняется неравенство $\lambda(v_i) - \lambda(v_j) > l(v_i v_j)$.

Путь строим, начиная с v_n и двигаясь обратно к v_1. После вершины v_i выбираем вершину v_j чтобы выполнялось $\lambda(v_i) - \lambda(v_j) = l(v_i v_j)$.

Обходы графа

Кратчайшие пути в графе



v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
0	∞	∞	∞	∞	∞	∞	∞
	1	10	6	11	14	9	16
					13		15

Планарные графы

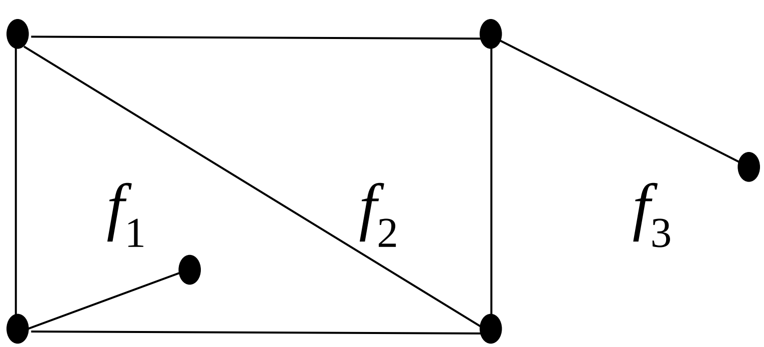
Граф *укладывается* на некоторой поверхности, если его можно так нарисовать на этой поверхности, что никакие два ребра не будут иметь общей точки, кроме, возможно, общей вершины.

Граф *планарный*, если его можно уложить на плоскости.

Плоский граф – граф, уложенный на плоскости.

Грань плоского графа – область плоскости, ограниченная ребрами, любые две точки которой могут быть соединены линией, не пересекающей ребра графа.

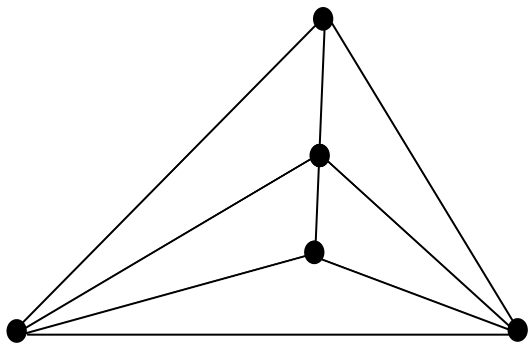
Грани *внешняя* и *внутренние.*

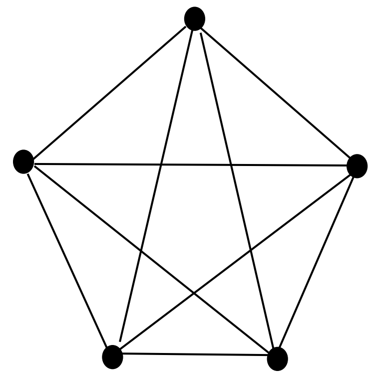


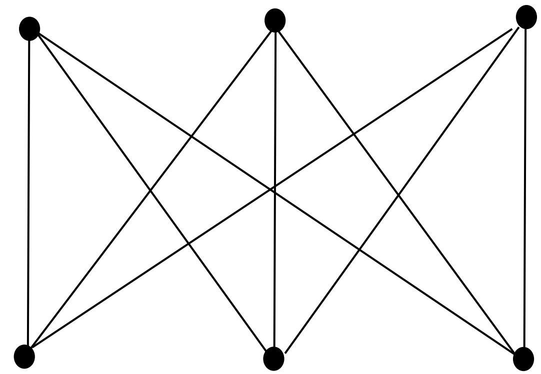
Т е о р е м а Э й л е р а. Для всякого связного плоского графа, имеющего n вершин, m ребер и f граней, имеет место соотношение $n - m + f = 2$.
(Формула Эйлера)

Планарные графы

Максимальный планарный граф и простейшие непланарные графы





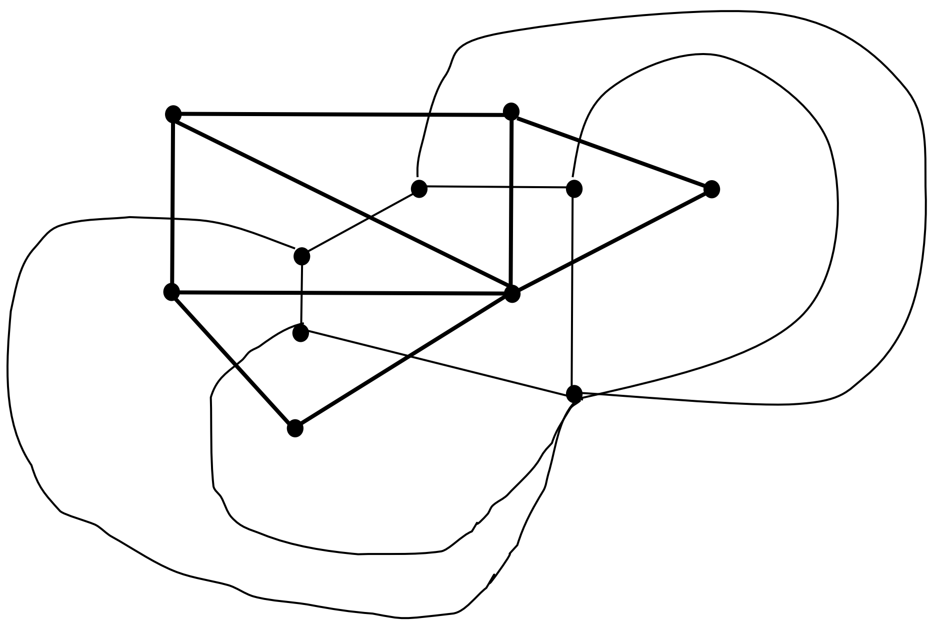


Т е о р е м а П о н т р я г и н а – К у р а т о в с к о г о. Необходимым и достаточным условием непланарности графа является любое из следующих условий: 1) в графе можно выделить пять вершин, каждая из которых связана цепью с любой другой из них, причем все эти цепи не пересекаются по ребрам; 2) в графе можно выделить два множества, состоящие из трех вершин каждое, так, что каждая вершина одного множества связана цепью со всеми вершинами другого множества, причем все эти цепи не пересекаются по ребрам.

Планарные графы

Раскраска планарных графов (раскраска карт)

Плоский граф и его двойственный граф



Раскраска граней плоского графа, при которой соседние грани раскрашиваются в различные цвета, эквивалентна раскраске вершин его двойственного графа.

Гипотеза четырех красок доказана с помощью ЭВМ в 1976 г.

1 482 конфигурации. 2 000 ч машинного времени.

Комбинаторные задачи и методы комбинаторного поиска

Три типа комбинаторных задач:

Задачи подсчета – сколько конфигураций определенного вида?

Перечислительные задачи – получение всех конструкций определенного вида.

Оптимизационные комбинаторные задачи – получение конструкции, обладающей оптимальным значением некоторого параметра среди всех конструкций данного вида.

Комбинаторные задачи и методы комбинаторного поиска

Задачи подсчета:

Число размещений (разместить n предметов по m ящикам)

$$U(m, n) = m^n.$$

Число перестановок

$$P(n) = n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1 = n!.$$

Число размещений без повторений

$$A(m, n) = m \cdot (m-1) \cdot \ldots \cdot (m-n+1) = \frac{m!}{(m-n)!}$$

Число сочетаний

$$C(m,n) = \frac{A(m,n)}{n!} = \frac{m!}{(m-n)!n!}$$

Комбинаторные задачи и методы комбинаторного поиска

Особенности оптимизационных комбинаторных задач

Решение комбинаторной задачи сводится зачастую к *полному перебору* различных вариантов.

Велика зависимость трудоемкости задачи от размера области возможных решений.

Множество, среди элементов которого отыскивается решение, всегда конечно. Реализовав полный перебор, либо найдем решение, либо убедимся в том, что решения нет.

Комбинаторные задачи и методы комбинаторного поиска

Вычислительная сложность оптимизационных задач

Трудоемкость алгоритма оценивается функцией $f(n)$, где n – натуральное число, выражающее объем исходных данных.

$f(n) = O(g(n))$, если найдется такая константа c, что $f(n) \leq cg(n)$ для любого $n \geq 0$, где $g(n)$ – некоторая конкретная функция от n.

$O(1)$ – трудоемкость не зависит от объема исходных данных.

$O(n)$ – алгоритм *линейный*.

$O(n^b)$ – алгоритм *полиномиальный*.

$g(n) = 2^n$ – алгоритм обладает *неполиномиальной*, или *экспоненциальной*, сложностью.

Комбинаторные задачи и методы комбинаторного поиска

Связь трудоемкости алгоритма с максимальным размером задачи, решаемой за единицу времени

Временна́я сложность	Максимальный размер задачи		
	1 с	1 мин	1 ч
n	1000	6×10^4	$3{,}6 \times 10^6$
$n \log n$	140	4893	$2{,}0 \times 10^5$
n^2	31	244	1897
n^3	10	39	153
2^n	9	15	21

Комбинаторные задачи и методы комбинаторного поиска

Связь размера задачи, решаемой за заданное время, с быстродействием вычислительной машины

Временна́я сложность	Максимальный размер задачи	
	до ускорения	после ускорения
n	s_1	$10\ s_1$
$n \log n$	s_2	$\approx 10\ s_2$
n^2	s_3	$3{,}16\ s_3$
n^3	s_4	$2{,}15\ s_4$
2^n	s_5	$s_5 + 3{,}3$

Комбинаторные задачи и методы комбинаторного поиска

Комбинаторный поиск представляется как обход *дерева поиска*.

Вершины соответствуют ситуациям, возникающим в процессе поиска, ребра – отдельным шагам процесса.

Корень дерева – вершина, соответствующая начальной ситуации.

Выделение корня придает дереву ориентацию, при которой все пути ведут из корня в остальные вершины.

Некоторые ситуации соответствуют решениям.

Задача о кратчайшем покрытии

Дано:

$A = \{a_1, a_2, \ldots, a_n\}$; $B_1, B_2, \ldots, B_m$; $B_i \subseteq A$, $i = 1, 2, \ldots, m$, причем $B_1 \cup B_2 \cup \ldots \cup B_m = A$.

Требуется выделить $B_{i_1}, B_{i_2}, \ldots, B_{i_k}$ так, чтобы выполнялось

$$B_{i_1} \cup B_{i_2} \cup \ldots \cup B_{i_k} = A$$

при минимальном k.

Матричная формулировка: требуется найти наименьшее множество строк заданной булевой матрицы, чтобы каждый ее столбец имел единицу хотя бы в одной строке из этого множества.

Задача о кратчайшем покрытии

Для матрицы

$$
\begin{array}{c c}
\begin{array}{cccccccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \end{array} & \\
\begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}
&
\begin{array}{c} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \\ B_9 \end{array}
\end{array}
$$

$\{B_4, B_6, B_7\}$ – кратчайшее покрытие.

Задача о кратчайшем покрытии

Алгоритмы решения

«Жадный» алгоритм повторяет операцию: выбор строки с максимальным числом единиц, включение ее в решение и удаление ее и покрываемых ею столбцов из матрицы. Для матрицы

$$\begin{array}{cc} \begin{matrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{matrix} & \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} & \begin{matrix} B_1 \\ B_2 \\ B_3 \end{matrix} \end{array}$$

находит покрытие $\{B_1, B_2, B_3\}$, хотя кратчайшее – $\{B_2, B_3\}$.

Минимаксный алгоритм повторяет операцию: выбор столбца с минимальным числом единиц, выбор покрывающей его строки с максимальным числом единиц, включение ее в решение и удаление ее и покрываемых ею столбцов из матрицы.

Точный алгоритм совершает обход дерева поиска.

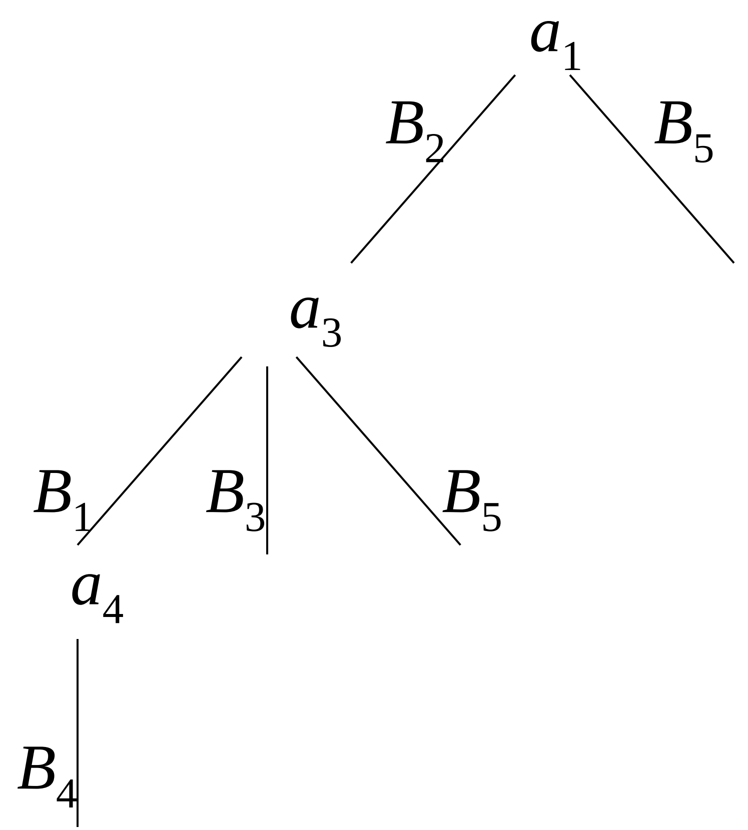
Задача о кратчайшем покрытии

Обход дерева поиска

$$
\begin{array}{c}
\begin{array}{cccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{array} \\
\begin{bmatrix}
0 & 1 & 1 & 0 & 1 & 0 \\
1 & 1` & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 1
\end{bmatrix}
\end{array}
\begin{array}{c} \\ B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{array}
$$

$$
\begin{array}{c}
\begin{array}{cc} a_3 & a_4 \end{array} \\
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
0 & 1 \\
1 & 0
\end{bmatrix}
\end{array}
\begin{array}{c} \\ B_1 \\ B_3 \\ B_4 \\ B_5 \end{array}
$$

$$
\begin{array}{c} a_4 \\ [1]\ B_4 \end{array}
$$



Одно из покрытий (не обязательно кратчайшее) – B_1, B_2, B_4.

Задача о кратчайшем покрытии

Обход дерева поиска. Правила редукции

1. Если столбец *k* имеет единицы везде, где имеет единицы столбец *l*, то столбец *k* можно удалить.

$$
\begin{array}{cc}
\begin{matrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{matrix} & \\
\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1^{\backprime} & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} & \begin{matrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{matrix}
\end{array}
\Rightarrow
\begin{array}{cc}
\begin{matrix} a_1 & a_2 & a_3 & a_4 \end{matrix} & \\
\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} & \begin{matrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{matrix}
\end{array}
$$

2. Если строка *i* имеет единицы везде, где имеет единицы строка *j*, то строку *j* можно удалить.

$$
\begin{array}{cc}
\begin{matrix} a_1 & a_2 & a_3 & a_4 \end{matrix} & \\
\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} & \begin{matrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{matrix}
\end{array}
\Rightarrow
\begin{array}{cc}
\begin{matrix} a_1 & a_2 & a_3 & a_4 \end{matrix} & \\
\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} & \begin{matrix} B_1 \\ B_2 \\ B_4 \\ B_5 \end{matrix}
\end{array}
$$

Булевы функции

$x_1, x_2, \ldots, x_n$ – *булевы переменные*, принимают значения из $\{0, 1\}$.

$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ – n-компонентный *булев вектор*.

n – *длина* вектора, или *размерность*.

2^n – число всех различных векторов, состоящих из констант 0 и 1 и образующих *булево пространство* .

$f : \{0, 1\}^n \to \{0, 1\}$ – *булева функция*.

$M = \{0, 1\}^n$ – область определения, $\{0, 1\}$ – область значений.

M_f^1 – область, где функция f принимает значение 1.
M_f^0 – область, где функция f принимает значение 0.
M_f^1 – *характеристическое множество* функции f.

Булевы функции

Способы задания

Таблица истинности:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Булевы функции

Способы задания

Матричный способ – перечисление элементов M_f^1:

$$\begin{array}{ccc} x_1 & x_2 & x_3 \\ \end{array}$$
$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Более компактное задание:

$$\begin{array}{ccc} x_1 & x_2 & x_3 \\ \end{array}$$
$$\begin{bmatrix} - & - & 1 \\ 0 & 1 & - \end{bmatrix}$$

Булевы функции

Способы задания

Матричный способ (интервальный) – троичная матрица:

$$\begin{array}{ccc} x_1 & x_2 & x_3 \end{array}$$
$$\begin{bmatrix} - & - & 1 \\ 0 & 1 & - \end{bmatrix}$$

Булевы векторы $\boldsymbol{a} = (a_1, a_2, \dots , a_n)$ и $\boldsymbol{b} = (b_1, b_2, \dots , b_n)$ находятся в отношении $\leq$ ($\boldsymbol{a} \leq \boldsymbol{b}$, $\boldsymbol{a}$ меньше $\boldsymbol{b}$), если $a_i \leq b_i$ для любого $i = 1, 2, \dots , n$, в противном случае они *несравнимы*. При этом считается, что $0 \leq 1$.

Интервал булева пространства – множество векторов, среди которых есть минимальный и максимальный векторы, а также все векторы, меньшие максимального и большие минимального.

Интервал представляется *троичным вектором*.

Булевы функции

Способы задания

Векторное задание – булев вектор, компоненты которого соответствуют наборам значений аргументов.

Функция

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

задается вектором (0 1 1 1 0 1 0 1).

Булевы функции

Алгебраический способ задания булевых функций.

Карты Карно.

Функции *полностью определенные*.

Функции *не полностью определенные*, или *частичные*.

Частичная булева функция делит булево пространство на три части: M_f^1, M_f^0 и M_f^-.

Обычно задаются M_f^1 и M_f^0.

Булевы функции

Векторная форма задания булевой функции позволяет легко определить число булевых функций от n переменных – это число всех 2^n-компонентных булевых векторов.

2^{2^n} – число булевых функций от n переменных.

Функция $f(x_1, x_2, \ldots, x_n)$ *существенно зависит* от аргумента x_i, если

$$f(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \neq f(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n).$$

x_i – *существенный аргумент*.

В противном случае x_i – *несущественный,* или *фиктивный аргумент*.

Булевы функции

Элементарные булевы функции и алгебраические формы

Элементарные булевы функции – функции от одной и двух переменных.

Булевы функции от одной переменной

x	0	1
$f_0 = 0$ – константа 0	0	0
$f_1 = x$	0	1
$f_2 = \overline{x}$ – отрицание x, или *инверсия*	1	0
$f_3 = 1$ – константа 1	1	1

Из таблицы видно, что $\overline{0} = 1$ и $\overline{1} = 0$.

Булевы функции

Булевы функции от двух переменных

x_1	0	0	1	1
x_2	0	1	0	1
$f_0 = 0$ – константа 0	0	0	0	0
$f_1 = x_1 \wedge x_2$ – конъюнкция	0	0	0	1
f_2 – отрицание импликации	0	0	1	0
$f_3 = x_1$	0	0	1	1
f_4 – отрицание обратной импликации	0	1	0	0
$f_5 = x_2$	0	1	0	1
$f_6 = x_1 \oplus x_2$ – сложение по модулю 2	0	1	1	0
$f_7 = x_1 \vee x_2$ - дизъюнкция	0	1	1	1

Булевы функции

Булевы функции от двух переменных

x_1	0	0	1	1
x_2	0	1	0	1
$f_8 = x_1 \uparrow x_2$ – стрелка Пирса	1	0	0	0
$f_9 = x_1 \sim x_2$ – эквиваленция	1	0	0	1
$f_{10} = \bar{x}_2$	1	0	1	0
$f_{11} = x_2 \to x_1$ – обратная импликация	1	0	1	1
$f_{12} = \bar{x}_1$	1	1	0	0
$f_{13} = x_1 \to x_2$ – импликация	1	1	0	1
$f_{14} = x_1 \mid x_2$ – штрих Шеффера	1	1	1	0
$f_{15} = 1$ – константа 1	1	1	1	1

Все представленные операции составляют *алгебру логики*.

Булевы функции

Алгебраическое задание

Любая булева функция от любого числа аргументов может быть представлена формулой алгебры логики.

Формулу, содержащую более чем одну операцию, можно рассматривать как *суперпозицию* элементарных функций, т.е. использование одних функций в качестве аргументов других функций.

Определение *формулы*:

1) каждый символ переменной есть формула;

2) если A и B – формулы, то формулами являются $\overline{A}$ и $(A * B)$, где $*$ – любая операция алгебры логики;

3) других формул нет.

Приоритеты: 1) $\overline{}$; 2) $\wedge$; 3) $\vee$ и $\oplus$; 4) $\sim$ и $\rightarrow$.

Булевы функции

Вычисление по формуле

$$f(x_1, x_2, x_3) = \overline{(x_1 \vee x_2)x_3} \to \overline{x}_1 \vee x_2x_3$$

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
$x_1 \vee x_2$	0	0	1	1	1	1	1	1
$(x_1 \vee x_2)\, x_3$	0	0	0	1	0	1	0	1
$\overline{(x_1 \vee x_2)x_3}$	1	1	1	0	1	0	1	0
$\overline{x}_1$	1	1	1	1	0	0	0	0
x_2x_3	0	0	0	1	0	0	0	1
$\overline{x}_1 \vee x_2x_3$	1	1	1	1	0	0	0	1
$f(x_1, x_2, x_3)$	1	1	1	1	0	1	0	1

$A = B$ – *равносильность* формул A и B.

Булева алгебра

Булева алгебра содержит только три операции: $\bar{\ }$, $\wedge$ и $\vee$.

Основные законы булевой алгебры:

Коммутативность:

$x \vee y = y \vee x$; $\quad$ $x\,y = y\,x$.

Ассоциативность:

$x \vee (y \vee z) = (x \vee y) \vee z$; $\quad$ $x\,(y\,z) = (x\,y)\,z$.

Дистрибутивность:

$x\,(y \vee z) = x\,y \vee x\,z$; $\quad$ $x \vee y\,z = (x \vee y)\,(x \vee z)$.

Идемпотентность:

$x \vee x = x$; $\quad$ $x\,x = x$.

Булева алгебра

Основные законы булевой алгебры:

Законы де Моргана:

$\overline{x \vee y} = \bar{x}\,\bar{y}$; $\overline{xy} = \bar{x} \vee \bar{y}$.

Законы операций с константами:

$x \vee 0 = x$; $x\,1 = x$;

$x\,0 = 0$; $x \vee 1 = 1$;

$x \vee \bar{x} = 1$; $x\,\bar{x} = 0$.

Закон двойного отрицания:

$$\bar{\bar{x}} = x\ .$$

Принцип двойственности.

Булева алгебра

Вывод формул $x \vee x\,y = x$ и $x\,(x \vee y) = x$:

$$x \vee x\,y = x\,1 \vee x\,y = x\,(1 \vee y) = x1 = x;$$

$$x\,(x \vee y) = x\,x \vee x\,y = x \vee x\,y = x.$$

Все операции алгебры логики можно выразить через булевы операции:

$$x \oplus y = x\,\bar{y} \vee \bar{x}\,y;$$

$$x \sim y = \bar{x}\,\bar{y} \vee x\,y;$$

$$x \rightarrow y = \bar{x} \vee y.$$

Преобразование формулы алгебры логики в булеву формулу:

$$((x \rightarrow y) \vee (x \oplus z))\,\bar{y} =$$

$$= (\bar{x} \vee y \vee x\,\bar{z} \vee \bar{x}\,z)\,\bar{y} = (\bar{x} \vee y \vee \bar{z})\,\bar{y} = \bar{x}\,\bar{y} \vee \bar{y}\,\bar{z}.$$

Интерпретации алгебры логики

Булева алгебра множеств:

Константам 1 и 0 соответствуют множества U и $\varnothing$.

Операциям $\bar{\ }$, $\wedge$ и $\vee$ соответствуют $\bar{\ }$, $\cap$ и $\cup$

Алгебра событий, используемая в теории вероятностей:

Операции отрицания ($\neg$), объединения ($\cup$) и пересечения ($\cap$).

$A \cap B$ или AB – *произведение независимых событий*,

$A \cup B$ – *сумма несовместимых событий*.

Исчисление высказываний:

$\bar{a}$ – «не a».

$a \wedge b$ – «a и b».

$a \vee b$ – «a или b».

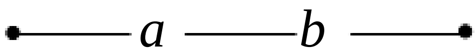
$a \oplus b$ – «a либо b».

$a \sim b$ – «a равносильно b».

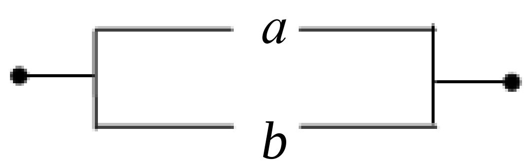
$a \rightarrow b$ – «если a, то b».

Интерпретации алгебры логики

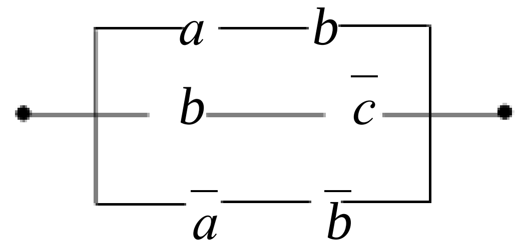
Алгебра переключательных схем:



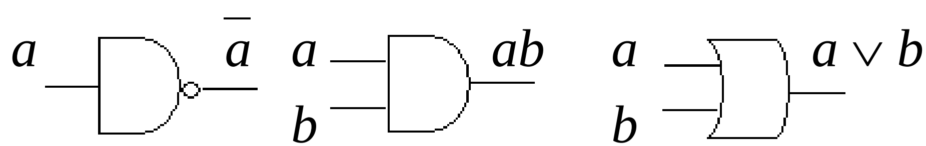
$a \cdot b$

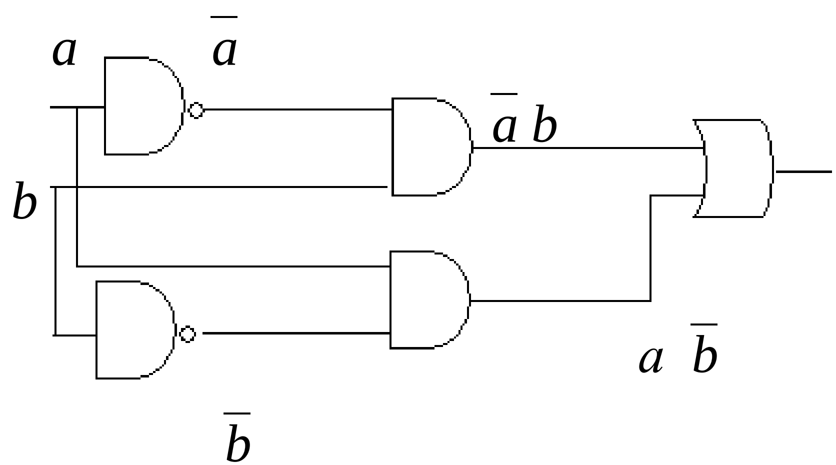


$a + b$



$a\, b + b\, \bar{c} + \bar{a}\, \bar{b}$





$\bar{a}\, b \vee a\, \bar{b} = a \oplus b$

Булевы функции. Операции над характеристическими множествами

Если $f = f_1 \wedge f_2$, то $M_f^1 = M_{f_1}^1 \cap M_{f_2}^1$;

если $f = f_1 \vee f_2$, то $M_f^1 = M_{f_1}^1 \cup M_{f_2}^1$;

если $f = f_1 \oplus f_2$, то $M_f^1 = (M_{f_1}^1 \cap M_{f_2}^0) \cup (M_{f_1}^0 \cap M_{f_2}^1)$;

если $f = f_1 \to f_2$, то $M_f^1 = M_{f_1}^0 \cup M_{f_2}^1$;

если $f = f_1 \sim f_2$, то $M_f^1 = (M_{f_1}^1 \cap M_{f_2}^1) \cup (M_{f_1}^0 \cap M_{f_2}^0)$;

если $f_1 = \overline{f_2}$, то $M_{f_1}^1 = M_{f_2}^0$.

Нормальные формы

Дизъюнктивные нормальные формы

x_i и $\overline{x}_j$ – *литералы*.

Обозначим a^σ, где $a^\sigma = \begin{cases} a, \text{ если } \sigma = 1 \\ \overline{a}, \text{ если } \sigma = 0 \end{cases}$.

$K_i = x_{i_1}^{\sigma_1} x_{i_2}^{\sigma_2} \ldots x_{i_r}^{\sigma_r}$ – *элементарная конъюнкция*, r – ее *ранг*.

$x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_n^{\sigma_n}$ – *полная элементарная конъюнкция*.

$\bigvee\limits_{i=1}^{m} K_i$ – *дизъюнктивная нормальная форма (ДНФ)*.

Пример: $x_1 \overline{x}_2 \vee x_2 x_3 x_4 \vee \overline{x}_1 x_3$.

Нормальные формы

Дизъюнктивное разложение Шеннона

Теорема Шеннона. Любая булева функция $f(x_1, x_2, \ldots, x_n)$ при любом m ($1 \le m \le n$) может быть представлена в следующем виде:

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{\sigma_1, \sigma_2, \ldots, \sigma_m} x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_m^{\sigma_m} f(\sigma_1, \sigma_2, \ldots, \sigma_m, x_{m+1}, \ldots, x_n),$$

где дизъюнкция берется по всевозможным 2^m наборам значений переменных $x_1, x_2, \ldots, x_m$.

$f(\sigma_1, \sigma_2, \ldots, \sigma_m, x_{m+1}, \ldots, x_n)$ – *коэффициент разложения*.

При $m = 1$ для любого $i = 1, 2, \ldots, n$:

$$f(x_1, x_2, \ldots, x_n) = x_i f(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) \vee$$
$$\vee\ \overline{x}_i f(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n).$$

При $m = n$: $f(x_1, x_2, \ldots, x_n) = \bigvee_{\sigma_1, \sigma_2, \ldots, \sigma_n} x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_n^{\sigma_n} f(\sigma_1, \sigma_2, \ldots, \sigma_n)$.

Нормальные формы

Совершенная дизъюнктивная нормальная форма (СДНФ):

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{\sigma_1, \sigma_2, \ldots, \sigma_n} x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_n^{\sigma_n} \; f(\sigma_1, \sigma_2, \ldots, \sigma_n).$$

Получение СДНФ по таблице истинности:

$x\,y\,z$	$f(x, y, z)$
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	0
1 1 1	0

Выделить наборы $(\sigma_1, \sigma_2, \ldots, \sigma_n)$, на которых функция принимает значение 1, и для каждого из них ввести в СДНФ полную элементарную конъюнкцию, где любая переменная x_i присутствует с отрицанием, если $\sigma_i = 0$, и без отрицания, если $\sigma_i = 1$.

$f(x, y, z) = \bar{x}\, y\, \bar{z} \vee x\, \bar{y}\, \bar{z} \vee x\, \bar{y}\, z.$

СДНФ – *каноническая форма*. $x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_n^{\sigma_n}$ – *конституент единицы*.

Нормальные формы

Конъюнктивные нормальные формы

$D_i = x_{i_1}^{\sigma_1} \vee x_{i_2}^{\sigma_2} \vee \ldots \vee x_{i_r}^{\sigma_r}$ – *элементарная дизъюнкция*, r – ее *ранг*.

$x_1^{\sigma_1} \vee x_2^{\sigma_2} \vee \ldots \vee x_n^{\sigma_n}$ – *полная элементарная дизъюнкция*.

$\bigwedge\limits_{i=1}^{m} D_i$ – *конъюнктивная нормальная форма* (КНФ).

Пример: $(x_2 \vee \overline{x}_3 \vee x_4)(x_1 \vee \overline{x}_2)$.

Конъюнктивное разложение: $f(x_1, x_2, \ldots, x_n) =$

$$= \bigwedge_{\sigma_1, \sigma_2, \ldots, \sigma_m} (x_1^{\overline{\sigma}_1} \vee x_2^{\overline{\sigma}_2} \vee \ldots \vee x_m^{\overline{\sigma}_m} \vee f(\sigma_1, \sigma_2, \ldots, \sigma_m, x_{m+1}, \ldots, x_n)).$$

Совершенная конъюнктивная нормальная форма (СКНФ):

$$f(x_1, x_2, \ldots, x_n) = \bigwedge_{\sigma_1, \sigma_2, \ldots, \sigma_n} (x_1^{\overline{\sigma}_1} \vee x_2^{\overline{\sigma}_2} \vee \ldots \vee x_n^{\overline{\sigma}_n} \vee f(\sigma_1, \sigma_2, \ldots, \sigma_n)).$$

Нормальные формы

Конъюнктивные нормальные формы

Получение СКНФ по таблице истинности:

$x\,y\,z$	$f(x,y,z)$
000	0
001	0
010	1
011	0
100	1
101	1
110	0
111	0

Выделить наборы (σ_1, σ_2, … , σ_n), на которых функция принимает значение 0, и для каждого из них ввести в СКНФ полную элементарную дизъюнкцию, где любая переменная x_i присутствует с отрицанием, если $\sigma_i = 1$, и без отрицания, если $\sigma_i = 0$.

$$f(x, y, z) = (x \vee y \vee z)(x \vee y \vee \bar{z})(x \vee \bar{y} \vee \bar{z})(\bar{x} \vee \bar{y} \vee z)(\bar{x} \vee \bar{y} \vee \bar{z}).$$

СКНФ – *каноническая форма*.

$x_1^{\sigma_1} \vee x_2^{\sigma_2} \vee \ldots \vee x_n^{\sigma_n}$ – *конституент нуля*.

Функциональная полнота

$\{f_1, f_2, \ldots, f_m\}$ – *функционально полная*, или просто *полная* система, если любая булева функция может быть представлена в виде суперпозиции функций из этого множества. Другое название – *базис*.

Минимальный базис.

$\{\bar{\ }, \wedge, \vee\}$ – полная система по теореме Шеннона.

$\{\bar{\ }, \wedge\}$ и $\{\bar{\ }, \vee\}$ – минимальные базисы.

$$a \vee b = \overline{\overline{a \vee b}} = \overline{\bar{a} \wedge \bar{b}}, \qquad a \wedge b = \overline{\overline{a \wedge b}} = \overline{\bar{a} \vee \bar{b}}.$$

Система $\{\wedge, \vee\}$ не является полной.

Функциональная полнота

$\{|\}$ и $\{\uparrow\}$ – полные системы. $a \,|\, b = \overline{a \wedge b}$; $a \uparrow b = \overline{a \vee b}$.

$$\bar{a} = a \,|\, a; \quad a \wedge b = \overline{\overline{a \wedge b}} = \overline{a \,|\, b} = (a \,|\, b) \,|\, (a \,|\, b);$$

$$\bar{a} = a \uparrow a; \quad a \wedge b = \overline{\overline{a \wedge b}} = \overline{\bar{a} \vee \bar{b}} = \bar{a} \uparrow \bar{b} = (a \uparrow a) \uparrow (b \uparrow b).$$

$\{1, \wedge, \oplus\}$ – полная система.

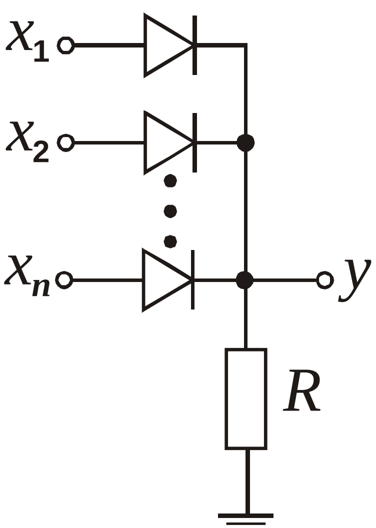
$$\bar{a} = a \oplus 1;$$

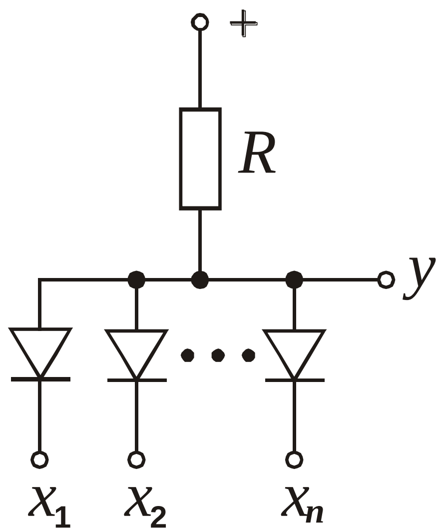
$$a \vee b = \overline{\overline{a \vee b}} = \overline{\bar{a} \wedge \bar{b}} = (a \oplus 1)(b \oplus 1) \oplus 1.$$

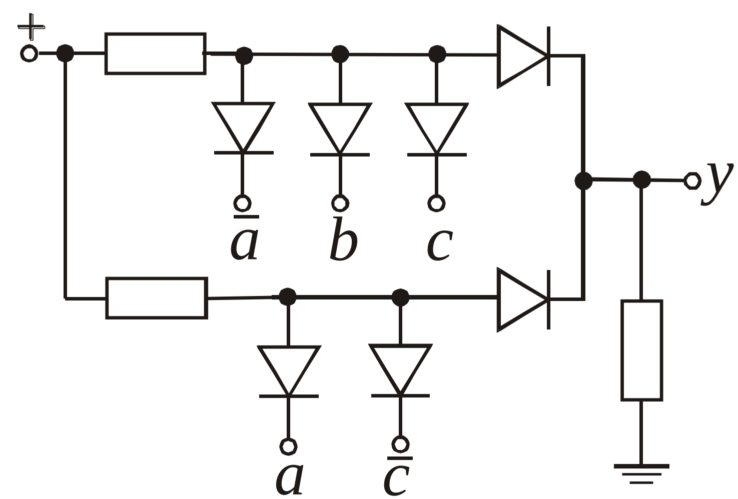
$$a \vee b = ab \oplus b \oplus a \oplus 1 \oplus 1 = ab \oplus b \oplus a.$$

Реализация булевых функций комбинационными схемами

Диодные схемы







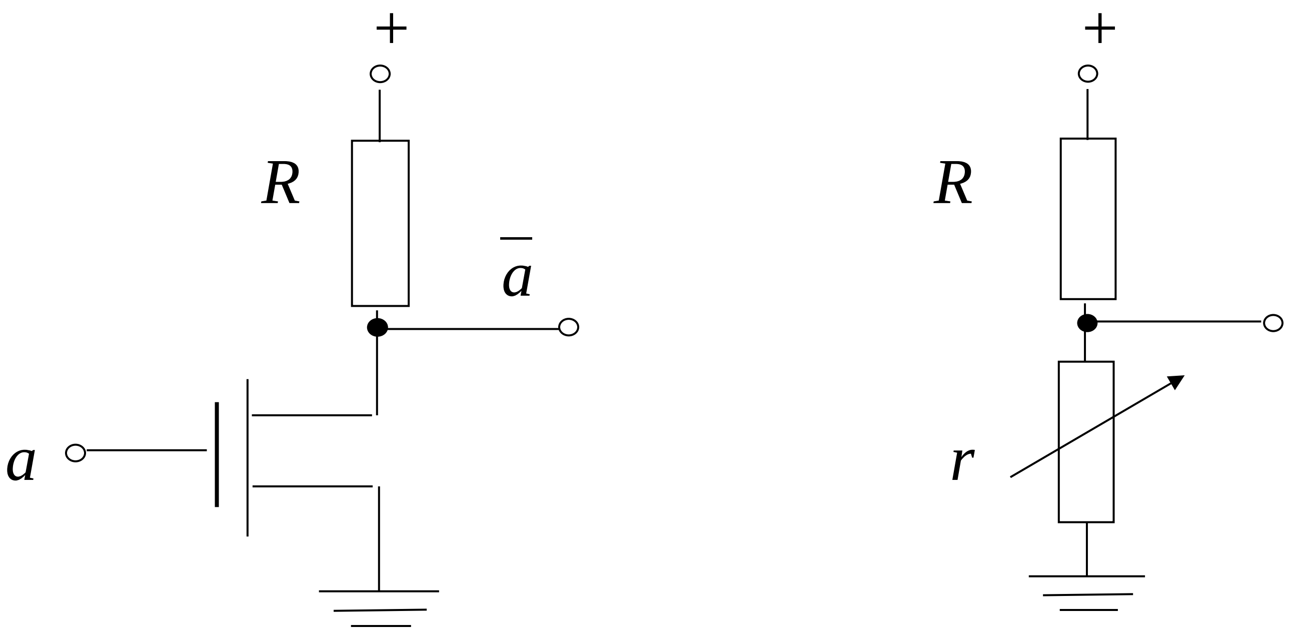
$$y = x_1 \vee x_2 \vee \ldots \vee x_n$$

$$y = x_1 x_2 \ldots x_n$$

$$y = \bar{a}\, b\, c \vee a\, \bar{c}$$

Реализация булевых функций комбинационными схемами

Транзисторные схемы

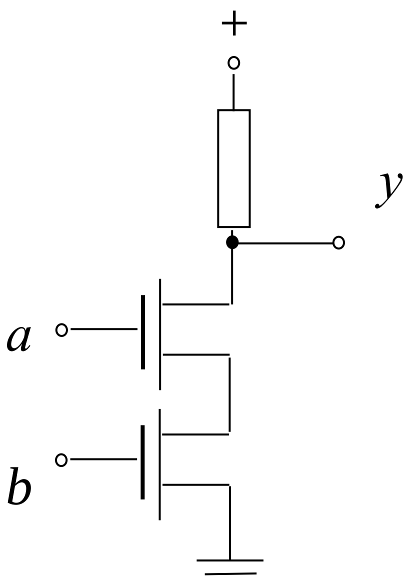


$$a = 0, \quad R << r$$

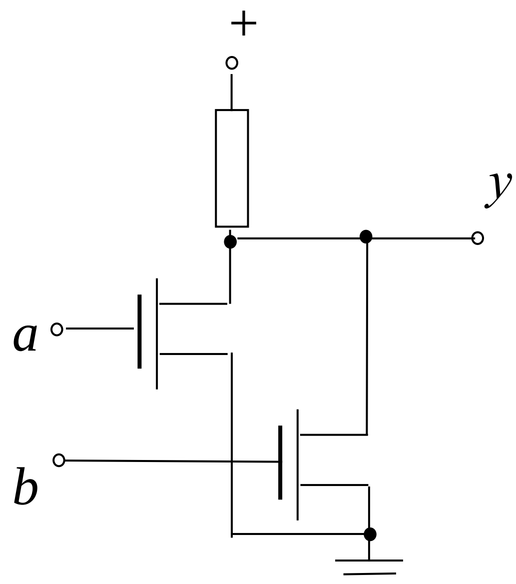
$$a = 1, \quad R >> r$$

Реализация булевых функций комбинационными схемами

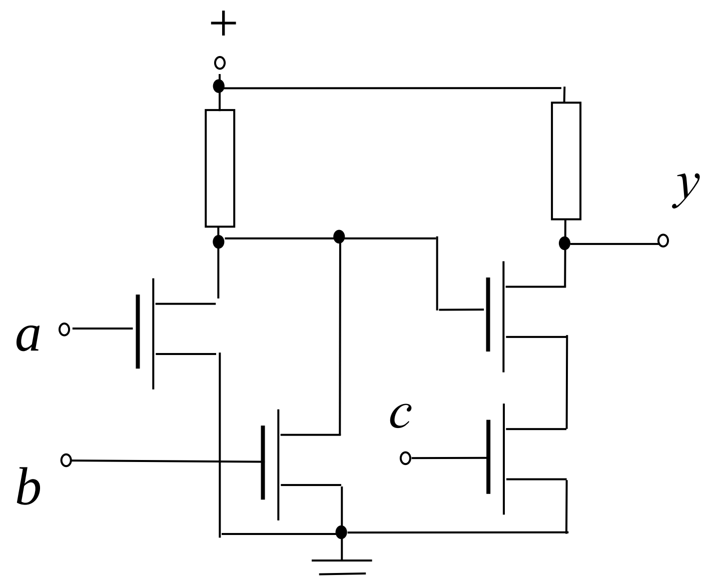
Транзисторные схемы



$$y = \overline{a \wedge b}$$



$$y = \overline{a \vee b}$$



$$y = \overline{\overline{a \vee b} \wedge c}$$

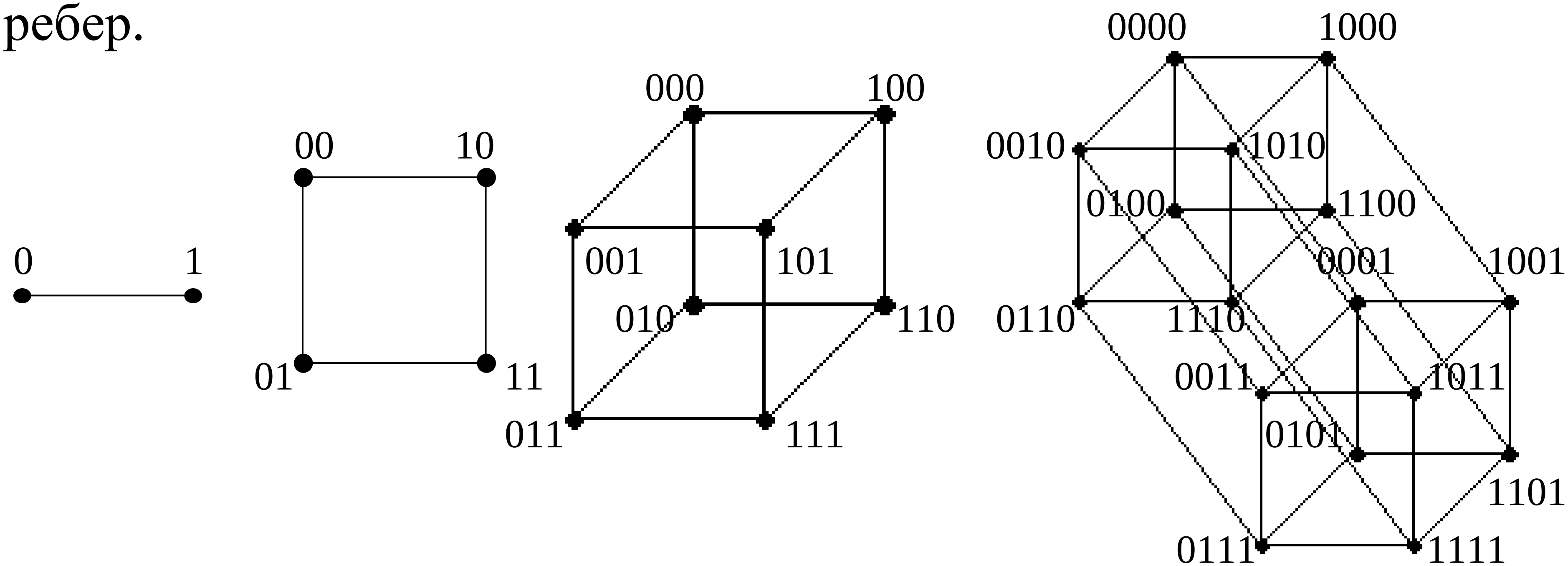
Булевы функции. Графическое представление

Два вектора являются *соседними*, если они отличаются друг от друга значением только одной компоненты.

Пример: (1 0 0 1) и (1 1 0 1).

Отношение соседства представляется графом.

Полный булев граф, или *n-мерный гиперкуб* имеет 2^n вершин и $n2^{n-1}$ ребер.



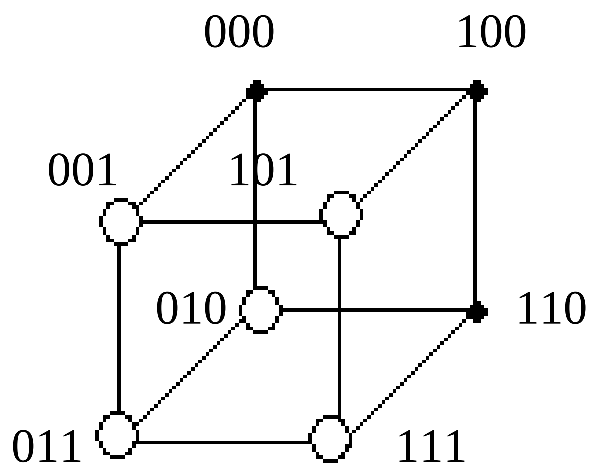
Интервал – порожденный подграф в виде $(n - k)$-мерного гиперкуба (*гипергрань*).

Булевы функции. Графическое представление

Характеристическое множество M_f^1 функции f, выражаемой одной элементарной конъюнкцией, есть интервал.

Пример: $x_1\ \bar{x}_3\, x_4$ представляется троичным вектором (1 – 0 1).

Представление булевой функции на гиперкубе:

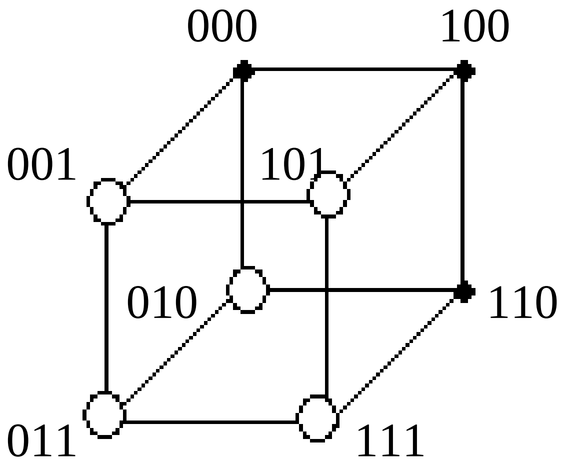


СДНФ: $\bar{x}_1\ \bar{x}_2\, x_3 \vee \bar{x}_1\, x_2\ \bar{x}_3 \vee \bar{x}_1\, x_2\, x_3 \vee x_1\ \bar{x}_2\, x_3 \vee x_1\, x_2\, x_3$

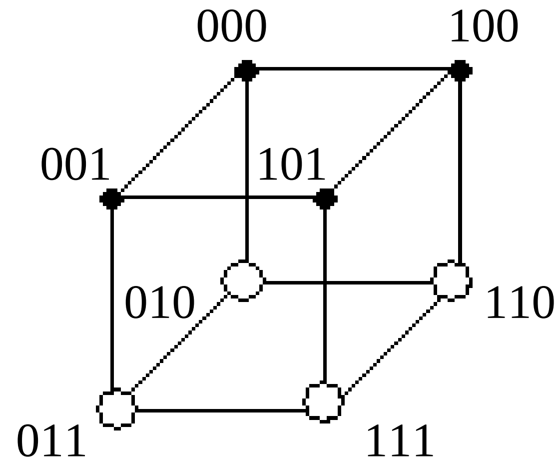
Можно задать интервалами (– – 1) и (0 1 –) или $x_3 \vee \bar{x}_1\, x_2$.

Булевы функции. Графическое представление

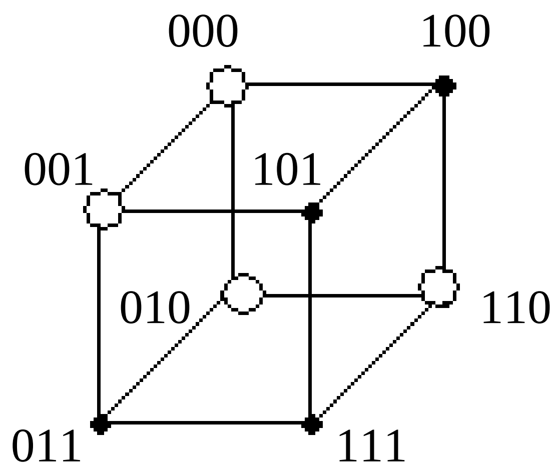
Демонстрация справедливости формул.



$$x_3 \vee \bar{x}_1 x_2 \bar{x}_3 = x_3 \vee \bar{x}_1 x_2$$



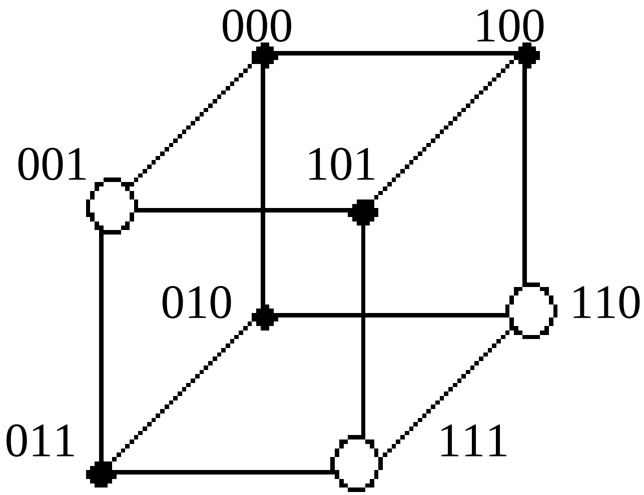
$$\bar{x}_1 x_2 \vee x_1 x_2 = x_2$$

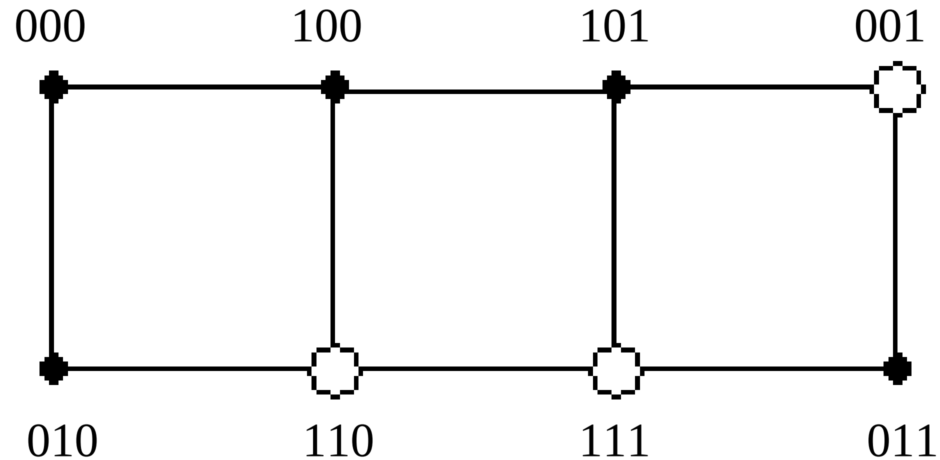


$$\bar{x}_1 \bar{x}_2 \vee x_2 \bar{x}_3 = \bar{x}_1 \bar{x}_2 \vee x_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_3$$

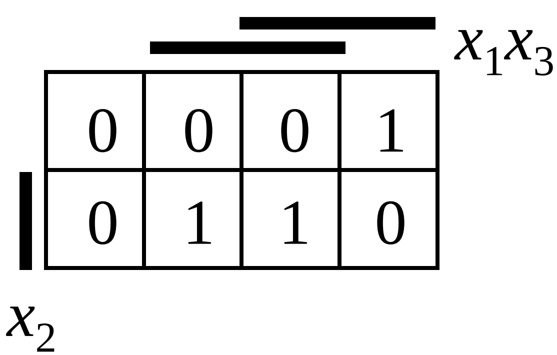
Булевы функции. Карта Карно

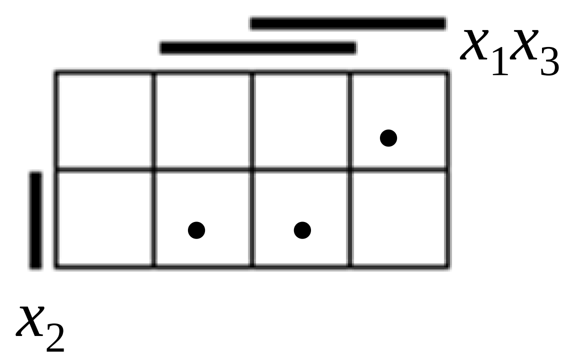
Развертка гиперкуба на плоскости:





$$f(x_1, x_2, x_3) = x_1 x_2 \vee \bar{x}_1 \bar{x}_2 x_3$$





Булевы функции. Карта Карно

Строки и столбцы карты Карно кодируются кодом Грея.

Длина кода n для кодирования N объектов должна быть такой, чтобы выполнялось $N \le 2^n$, или $n = \lceil \log_2 N \rceil$, где $\lceil a \rceil$ – ближайшее к a сверху целое число.

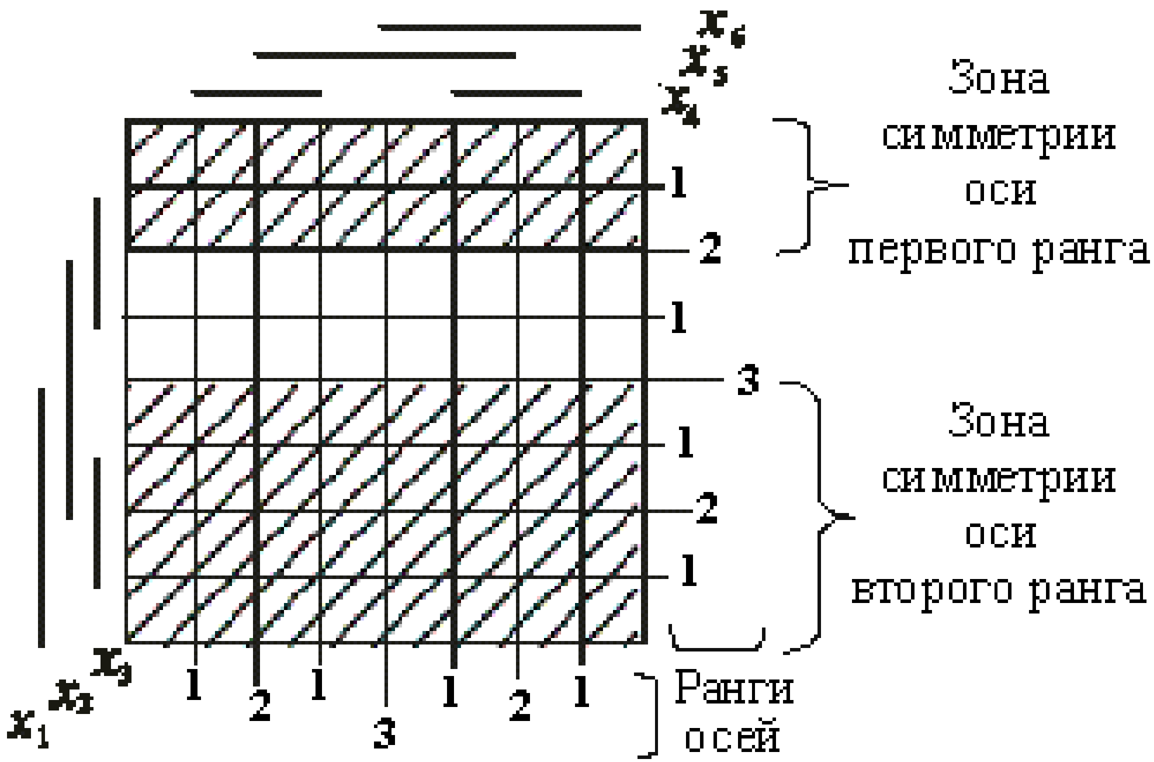
0 0 … 0 – код первого объекта.

Для получения следующего кода берется последний код и в нем меняется значение той самой правой компоненты, изменение значения которой приводит к новому коду.

000	101	Другой способ:	0	0	00	00	000
001	100		1	1	01	01	001
011				1	11	11	011
010				0	10	10	010
110						10	110
111						11	111

Булевы функции. Карта Карно

Отношение соседства элементов булева пространства представляется отношением симметрии в карте Карно.

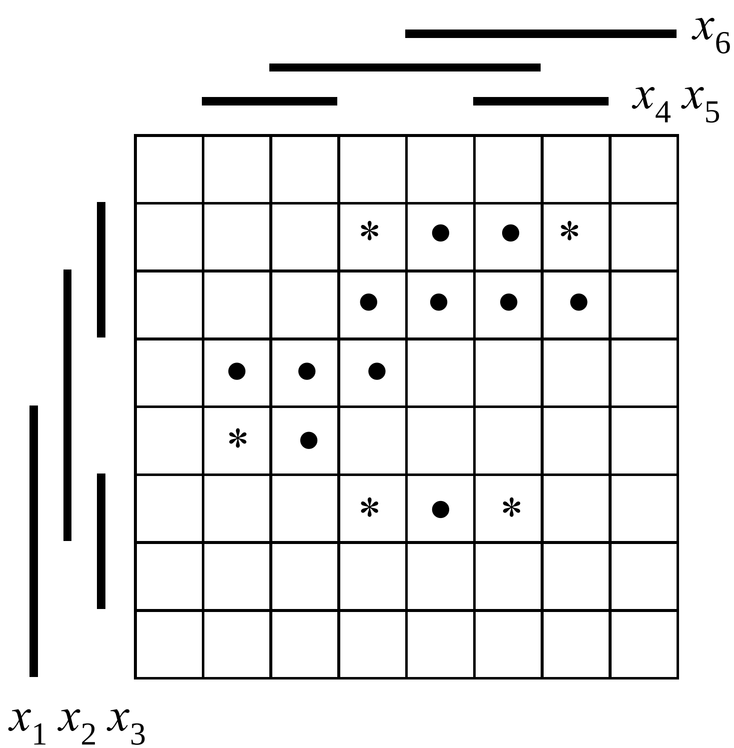


Каждая ось имеет свою *зону симметрии*, ширина которой определяется *рангом* оси.

Булевы функции. Карта Карно

Упрощение ДНФ. Поиск максимальных интервалов.

Поиск *определяющих элементов* и *обязательных интервалов.*

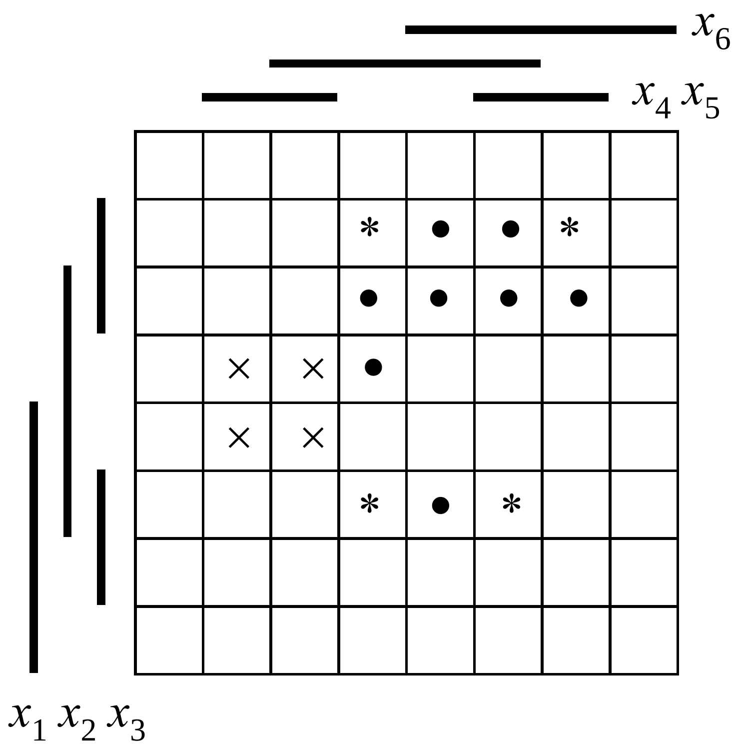


$$x_2\, \bar{x}_3\, x_4\, \bar{x}_6 \vee \bar{x}_1\, x_3\, \bar{x}_4\, x_5 \vee x_2\, x_3\, \bar{x}_4\, x_5 \vee x_2\, x_3\, x_5\, x_6 \vee \bar{x}_1\, x_3\, \bar{x}_4\, x_6 \vee$$
$$\vee\ \bar{x}_1\, x_2\, \bar{x}_3\, x_5\, \bar{x}_6\ .$$

Булевы функции. Карта Карно

Упрощение ДНФ.

Формирование элементарных конъюнкций

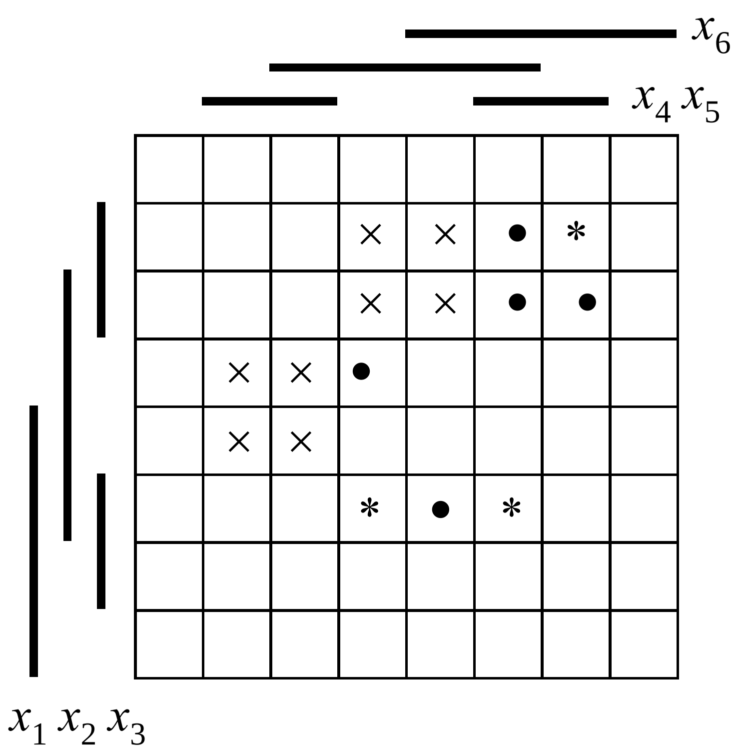


$x_2\ \overline{x}_3\ x_4\ \overline{x}_6$.

Булевы функции. Карта Карно

Упрощение ДНФ.

Формирование элементарных конъюнкций.

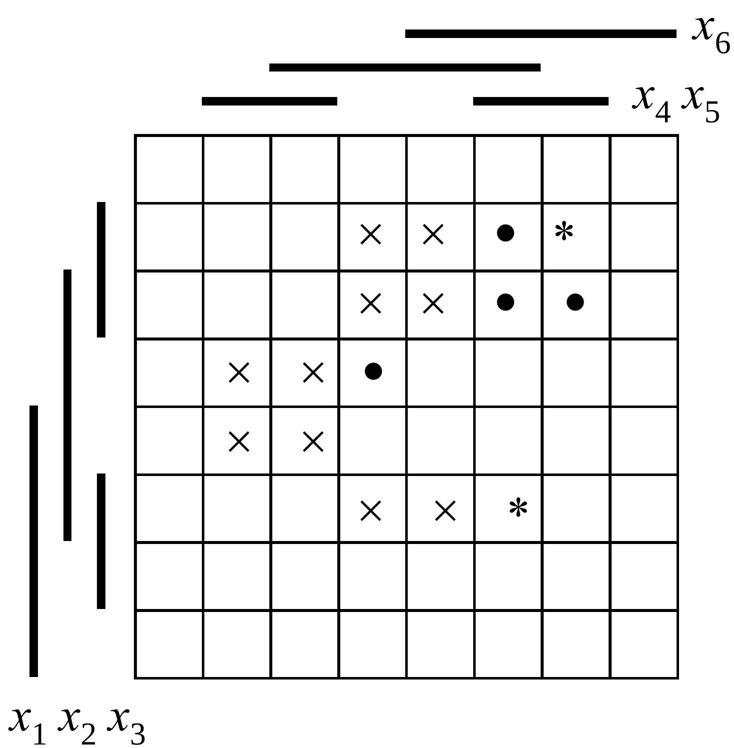


$$x_2\ \bar{x}_3\ x_4\ \bar{x}_6 \vee \bar{x}_1\ x_3\ \bar{x}_4\ x_5.$$

Булевы функции. Карта Карно

Упрощение ДНФ.

Формирование элементарных конъюнкций.

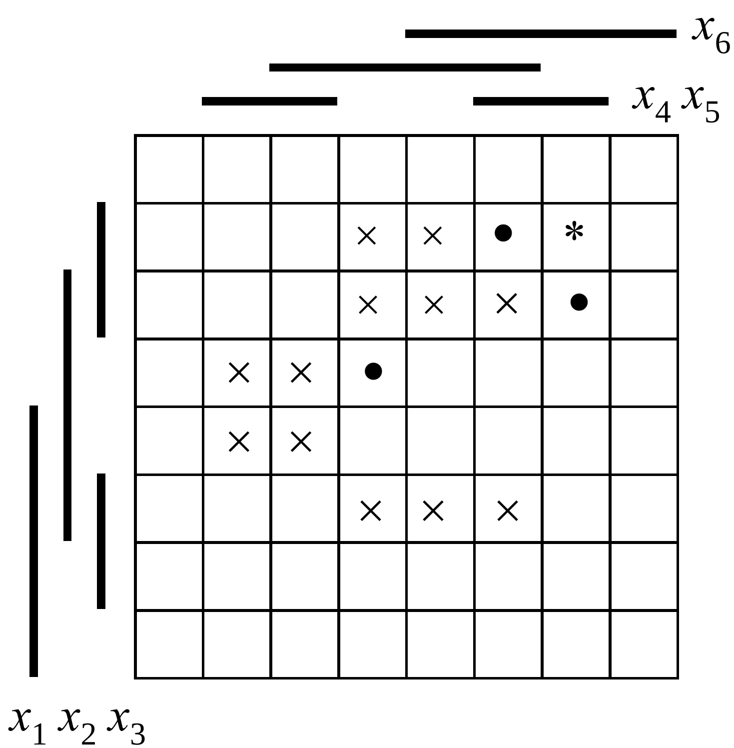


$$x_2\,\bar{x}_3\,x_4\,\bar{x}_6 \vee \bar{x}_1\,x_3\,\bar{x}_4\,x_5 \vee x_2\,x_3\,\bar{x}_4\,x_5.$$

Булевы функции. Карта Карно

Упрощение ДНФ.

Формирование элементарных конъюнкций.

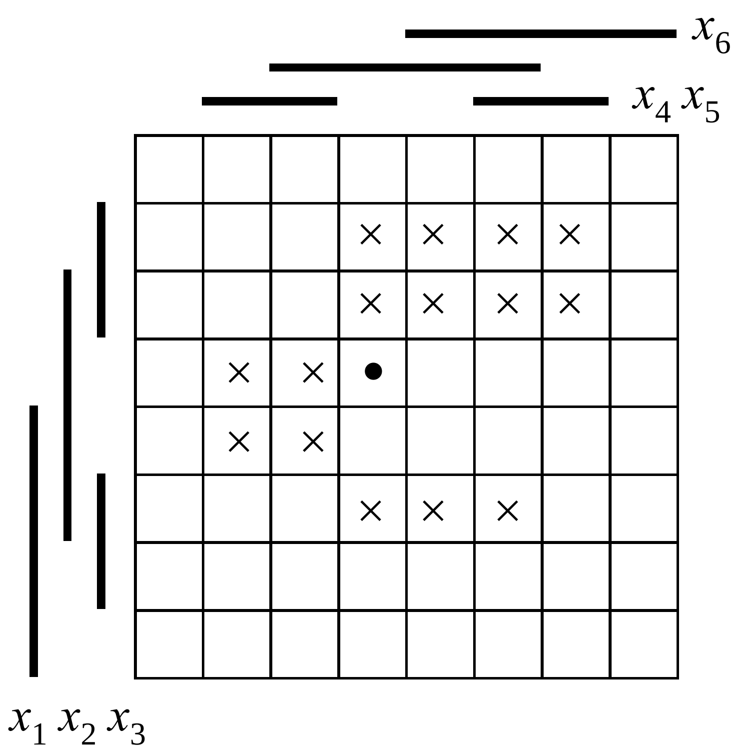


$$x_2\, \bar{x}_3\, x_4\, \bar{x}_6 \vee \bar{x}_1\, x_3\, \bar{x}_4\, x_5 \vee x_2\, x_3\, \bar{x}_4\, x_5 \vee x_2\, x_3\, x_5\, x_6.$$

Булевы функции. Карта Карно

Упрощение ДНФ.

Формирование элементарных конъюнкций.

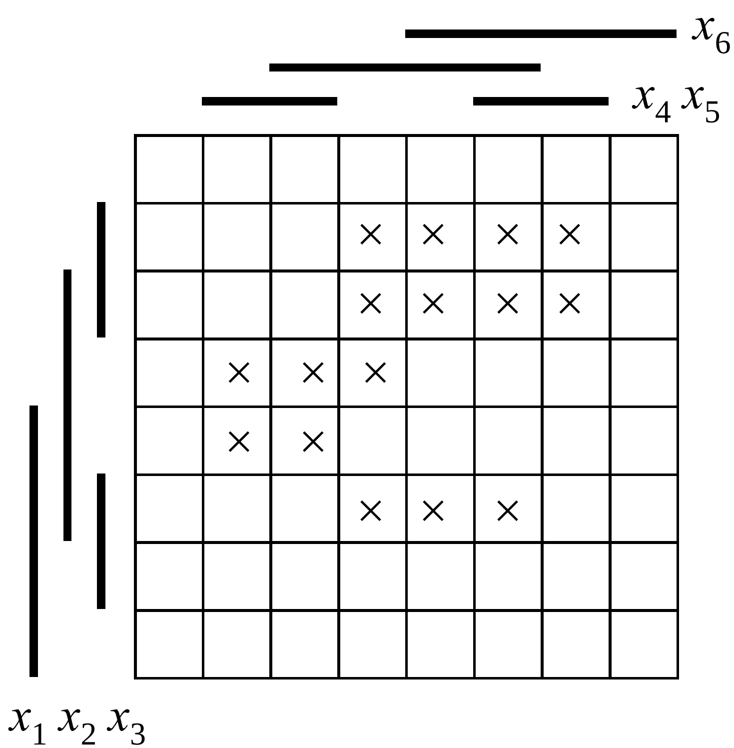


$$x_2\,\bar{x}_3\,x_4\,\bar{x}_6 \vee \bar{x}_1\,x_3\,\bar{x}_4\,x_5 \vee x_2\,x_3\,\bar{x}_4\,x_5 \vee x_2\,x_3\,x_5\,x_6 \vee \bar{x}_1\,x_3\,x_4\,x_6.$$

Булевы функции. Карта Карно

Упрощение ДНФ.

Формирование элементарных конъюнкций.

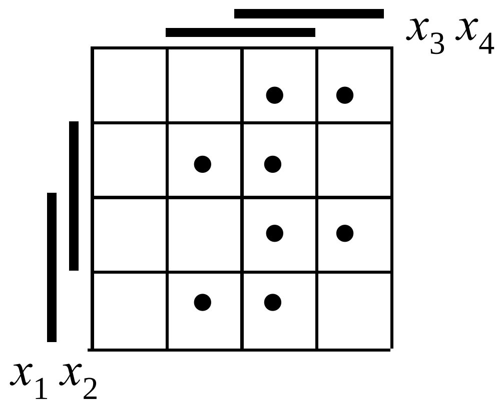


$$x_2\,\overline{x}_3\,x_4\,\overline{x}_6 \vee \overline{x}_1\,x_3\,\overline{x}_4\,x_5 \vee x_2\,x_3\,\overline{x}_4\,x_5 \vee x_2\,x_3\,x_5\,x_6 \vee \overline{x}_1\,x_3\,x_4\,x_6 \vee$$
$$\vee\ \overline{x}_1\,x_2\,\overline{x}_3\,x_5\,\overline{x}_6\ .$$

Булевы функции. Карта Карно

Упрощение ДНФ.

«Жадный» способ не устраняет избыточность:



$f(x_1, x_2, x_3, x_4) = x_3\, x_4 \vee \bar{x}_1\, \bar{x}_2\, x_4 \vee \bar{x}_1\, x_2\, x_3 \vee x_1\, x_2\, x_4 \vee x_1\, \bar{x}_2\, x_3.$

$x_3\, x_4$ – избыточная конъюнкция.

Троичные векторы и матрицы

Вектор (0 – 1 0 – 1) задает {(0 0 1 0 0 1), (0 0 1 0 1 1), (0 1 1 0 0 1), (0 1 1 0 1 1)} – интервал булева пространства.

Интервал – характеристическое множество элементарной конъюнкции. Например, вектор (0 – 1 0 – 1) представляет конъюнкцию $\bar{x}_1 x_3 \bar{x}_4 x_6$.

Тогда всякую троичную матрицу (строками которой являются троичные векторы) можно считать представлением ДНФ некоторой булевой функции.

Троичные векторы и матрицы

Отношения на множестве троичных векторов

Ортогональность. Векторы ***u*** и ***v*** ортогональны по *i*-й компоненте, если и только если *i*-я компонента имеет 0 в одном из них и 1 – в другом. Троичные векторы ортогональны, если они ортогональны хотя бы по одной компоненте. Пример: (0 – 1 0 – 1) и (0 1 0 – 1 0).

Симметрично, иррефлексивно.

Пересечение. Если векторы ***u*** и ***v*** неортогональны, то они находятся в отношении пересечения. Пример: (0 – 1 0 – 1) и (0 0 1 – 1–).

Рефлексивно, симметрично.

Смежность. Векторы ***u*** и ***v***, ортогональные только по одной компоненте, являются смежными. Им соответствуют смежные элементарные конъюнкции. Пример: (0 – 1 0 – 1) и (0 1 0 – 1 –).

Симметрично, иррефлексивно.

Троичные векторы и матрицы

Отношения на множестве троичных векторов

Соседство. Векторы ***u*** и ***v*** являются соседними, если по некоторой *i*-й компоненте они ортогональны, а значения остальных одноименных компонент совпадают.

Пример: (0 – 1 0 – 1) и (0 – 1 0 – 0).

Симметрично, иррефлексивно.

Поглощение. Вектор ***u*** поглощает вектор ***v***, если и только если все компоненты вектора ***u***, значения которых отличны от «–», совпадают с одноименными компонентами вектора ***v***. Интервал, представляемый вектором ***v***, является подмножеством интервала, представляемого вектором ***u***.

Пример: (0 – 1 0 – –) поглощает (0 – 1 0 – 0).

Рефлексивно, транзитивно.

Троичные векторы и матрицы

Эквивалентность матриц

Троичная матрица $\boldsymbol{U}$ *эквивалентна* булевой матрице $\boldsymbol{W}$, если каждая из строк матрицы $\boldsymbol{W}$ поглощается хотя бы одной строкой матрицы $\boldsymbol{U}$, а любой вектор, не совпадающий ни с одной из строк матрицы $\boldsymbol{W}$, не поглощается ни одной строкой матрицы $\boldsymbol{U}$.

Троичные матрицы $\boldsymbol{U}$ и $\boldsymbol{V}$ *эквивалентны*, если существует булева матрица, эквивалентная обеим матрицам $\boldsymbol{U}$ и $\boldsymbol{V}$. Бинарное отношение эквивалентности матриц рефлексивно, симметрично и транзитивно.

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & - \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} - & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & - \\ - & 1 & 0 \end{bmatrix}$$

Троичные векторы и матрицы

Операции над троичными векторами

Склеивание соседних строк:

$$\begin{bmatrix} 0 & 1 & - & - & 1 & 0 & 0 & 1 \\ 0 & 1 & - & - & 1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & - & - & 1 & - & 0 & 1 \end{bmatrix}$$

Поглощение:

$$\begin{bmatrix} 1 & - & 0 & 0 & 1 & - & 1 & - \\ 1 & 1 & 0 & 0 & 1 & - & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & - & 0 & 0 & 1 & - & 1 & - \end{bmatrix}$$

Обобщенное склеивание смежных строк:

$$\begin{bmatrix} 1 & - & 0 & - & - & 1 & 0 & 0 \\ 1 & 1 & - & - & 1 & 0 & - & 0 \end{bmatrix} = \begin{bmatrix} 1 & - & 0 & - & - & 1 & 0 & 0 \\ 1 & 1 & - & - & 1 & 0 & - & 0 \\ 1 & 1 & 0 & - & 1 & - & 0 & 0 \end{bmatrix}$$

Разложение строки по i-й компоненте.

$$\begin{bmatrix} 1 & 1 & 0 & - & 1 & - & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & - & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & - & 0 & 0 \end{bmatrix}$$

Анализ троичной матрицы на вырожденность

Троичная матрица ***U*** является *вырожденной*, если не существует троичного вектора, ортогонального каждой строке матрицы ***U***.

Совокупность интервалов, представляемая вырожденной матрицей, покрывает все булево пространство.

Функция, ДНФ которой представляется вырожденной матрицей, является константой 1.

Для заданной троичной матрицы ***U*** требуется найти троичный вектор ***v***, ортогональный каждой ее строке, или убедиться в том, что такого вектора не существует.

Вектор ***v*** в этом случае представляет набор значений аргументов, обращающий в нуль функцию, задаваемую матрицей ***U***.

Анализ троичной матрицы на вырожденность

Троичный вектор, имеющий k компонент со значением «–», представляет множество 2^k булевых векторов. Любой из этих булевых векторов *покрывается* данным троичным вектором.

$$\begin{bmatrix} 1 & - & - \\ - & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Вектор (0, 0, –) ортогонален обеим строкам троичной матрицы. Следовательно, матрица не вырожденная.

Ни один из покрываемых вектором (0, 0, –) двух булевых векторов (0, 0, 0) и (0, 0, 1) не является строкой булевой матрицы.

Решить задачу о вырожденности троичной матрицы можно простым перебором всех 2^n различных булевых векторов.

Следует использовать более эффективный *редукционный метод*, опирающийся на комбинаторный поиск .

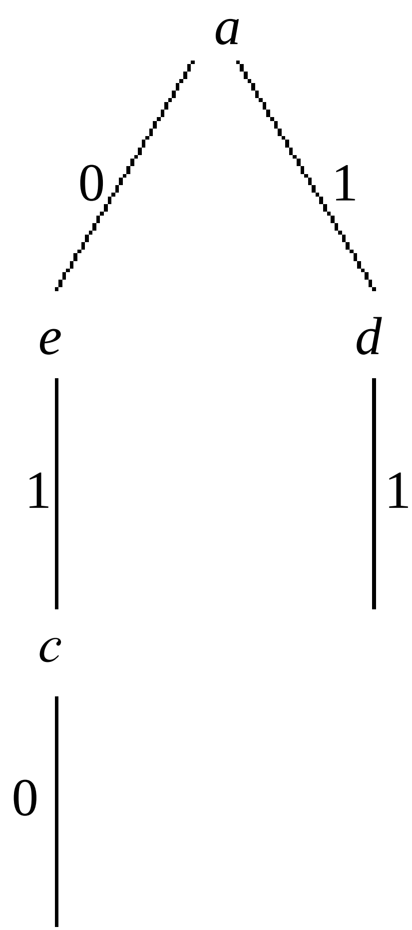
Анализ троичной матрицы на вырожденность

Комбинаторный поиск

$$\boldsymbol{T} = \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \left[\begin{array}{ccccc} 0 & - & - & - & 0 \\ - & - & 1 & 0 & 0 \\ 1 & - & - & 0 & - \\ 0 & 0 & 0 & - & - \\ - & - & 1 & - & 1 \\ 0 & 1 & - & - & 1 \\ 1 & - & - & 1 & - \end{array}\right] \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}$$

$$a = 0 \quad \begin{array}{c} \begin{array}{cccc} b & c & d & e \end{array} \\ \left[\begin{array}{cccc} - & - & - & 0 \\ - & 1 & 0 & 0 \\ 0 & 0 & - & - \\ - & 1 & - & 1 \\ 1 & - & - & 1 \end{array}\right] \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 4 \\ 5 \\ 6 \end{array}$$

$$e = 1 \quad \begin{array}{c} \begin{array}{ccc} b & c & d \end{array} \\ \left[\begin{array}{ccc} 0 & 0 & - \\ - & 1 & - \\ 1 & - & - \end{array}\right] \end{array} \begin{array}{c} \\ 4 \\ 5 \\ 6 \end{array} \qquad c = 0 \quad \begin{array}{c} b \\ \left[\begin{array}{c} 0 \\ 1 \end{array}\right] \end{array} \begin{array}{c} \\ 4 \\ 6 \end{array}$$



$$a = 1 \quad \begin{array}{c} \begin{array}{cccc} b & c & d & e \end{array} \\ \left[\begin{array}{cccc} - & 1 & 0 & 0 \\ - & - & 0 & - \\ - & 1 & - & 1 \\ - & - & 1 & - \end{array}\right] \end{array} \begin{array}{c} \\ 2 \\ 3 \\ 5 \\ 7 \end{array}$$

$$d = 1 \quad \begin{array}{c} \begin{array}{cc} c & e \end{array} \\ \left[\begin{array}{cc} 1 & 1 \\ - & - \end{array}\right] \end{array} \begin{array}{c} \\ 5 \\ 7 \end{array}$$

$$\boldsymbol{w} = (1 \quad - \quad - \quad 1 \quad -)$$

$$\boldsymbol{w} = (0 \quad - \quad 0 \quad - \quad 1)$$

Анализ троичной матрицы на вырожденность

Правила редукции

Правило 1. Из матрицы ***T*** удаляются столбцы, содержащие только «–».

Правило 2. Из матрицы ***T*** удаляются строки, ортогональные вектору ***w***, а затем столбцы, которым соответствуют компоненты вектора ***w*** со значением 0 или 1.

Правило 3. Если в матрице ***T*** имеется строка, где лишь одна компонента обладает значением, отличным от «–», то соответствующей компоненте вектора ***w*** приписывается инверсное значение.

Правило 4. Если в матрице ***T*** существует столбец, не содержащий значения 0 (или 1), то это значение приписывается соответствующей компоненте вектора ***w***.

Правило расщепления предписывает перебор значений 0 и 1 некоторой компоненты вектора ***w***.

Правило нахождения решения. Если непосредственно после удаления строки по правилу 2 матрица становится пустой, ***w*** – искомый вектор.

Правило возврата. Если матрица ***T*** становится пустой после удаления столбца или она содержит строку без значений 0 и 1, то следует возвратиться к последней точке ветвления.

Правило прекращения поиска. Если при полном обходе дерева поиска вектор ***v*** найти не удалось, то это свидетельствует о вырожденности матрицы ***U***.

Локальные упрощения ДНФ

Дизъюнктивная нормальная форма *безызбыточна*, если из нее нельзя удалить ни одной элементарной конъюнкции и ни одного литерала из какой-либо конъюнкции.

Простейшие случаи подобного сокращения:

$$A\,x \vee A = A; \quad A\,\bar{x} \vee x = A \vee x; \quad A\,x \vee B\,\bar{x} \vee AB = A\,x \vee B\,\bar{x}.$$

Более сложный случай:

$$\bar{x}_1\,\bar{x}_2\,x_3 \vee x_1\,\bar{x}_2\,x_4 \vee x_1\,x_2\,x_3 \vee \bar{x}_1\,x_2\,x_4 \vee x_3\,x_4,$$

где конъюнкция $x_3\,x_4$ является избыточной.

Два вида избыточности:

$$D = k \vee D' = D', \qquad D = xk \vee D' = k \vee D'.$$

Локальные упрощения ДНФ

Удаление избыточных элементарных конъюнкций

$$D = k \vee D' = D'$$

k и D' находятся в отношении *формальной импликации*, т. е. $k \Rightarrow D'$. Функция g *имплицирует* функцию f, если f имеет значение 1 везде, где имеет значение 1 функция g.

Матрица $\boldsymbol{V}$ представляет ДНФ D', а вектор $\boldsymbol{v}$ – конъюнкцию k. Результат подстановки в D' значений переменных, обращающих k в единицу – минор матрицы $\boldsymbol{V}$, образованный строками, не ортогональными $\boldsymbol{v}$ и столбцами, соответствующими компонентам $\boldsymbol{v}$, имеющими значение «–». Если этот минор – вырожденная матрица, то k – избыточная конъюнкция.

Вектор, ортогональный всем строкам полученного минора, представляет набор значений переменных, обращающий D' в нуль.

Локальные упрощения ДНФ

Удаление избыточных элементарных конъюнкций

$$\begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{bmatrix} 0 & 1 & - & 0 & 1 & - \\ 0 & 1 & 1 & 0 & - & 0 \\ 0 & - & 0 & 0 & 1 & - \\ - & 1 & 1 & - & 1 & 1 \\ - & 1 & 0 & - & - & 1 \\ 1 & - & 1 & - & 0 & - \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} = \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & - & 0 \\ 0 & - & 0 & 0 & 1 & - \\ - & 1 & 1 & - & 1 & 1 \\ - & 1 & 0 & - & - & 1 \\ 1 & - & 1 & - & 0 & - \end{bmatrix} \end{array} \begin{array}{c} \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}$$

Результат подстановки значений $x_1 = 0$, $x_2 = 1$, $x_4 = 0$, $x_5 = 1$:

$$\begin{array}{c} \begin{array}{cc} x_3 & x_6 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & - \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \text{ – вырожденная матрица.}$$

Локальные упрощения ДНФ

Удаление избыточных литералов

$$D = xk \vee D' = k \vee D'.$$

$$k \vee D' = k(x \vee \bar{x}) \vee D' = xk \vee D' \vee \bar{x}k = D \vee \bar{x}k.$$

Литерал x в выражении $xk \vee D'$ является избыточным, если конъюнкция $\bar{x}k$ является избыточной в выражении $D \vee \bar{x}k$. Следовательно, задача определения избыточности литерала в ДНФ сводится к предыдущей задаче – задаче определения избыточности элементарной конъюнкции.

Надо построить минор, образованный столбцами, где i-я строка имеет значения «–», и строками, не ортогональными вектору, полученному из i-й строки заменой нуля (или единицы) в j-м столбце на противоположное значение. Если полученный минор оказался вырожденной матрицей, то замена нуля (или единицы) на «–» возможна.

Локальные упрощения ДНФ

Удаление избыточных литералов

$$\begin{array}{cc} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} & \\ \begin{bmatrix} 0 & 1 & 1 & 0 & - & 0 \\ 0 & - & 0 & 0 & 1 & - \\ - & 1 & 1 & - & 1 & 1 \\ - & 1 & 0 & - & - & 1 \\ 1 & - & 1 & - & - & 0 \end{bmatrix} & \begin{array}{c} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \end{array} = \begin{array}{cc} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} & \\ \begin{bmatrix} - & 1 & 1 & 0 & - & 0 \\ 0 & - & 0 & 0 & 1 & - \\ - & 1 & - & - & 1 & 1 \\ - & 1 & 0 & - & - & 1 \\ 1 & - & 1 & - & - & 0 \end{bmatrix} & \begin{array}{c} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \end{array}$$

$$\begin{array}{cc} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} & \\ \begin{bmatrix} 1 & 1 & 1 & 0 & - & 0 \end{bmatrix} & 2 \end{array}$$

Результат подстановки значений $x_1 = 1$, $x_2 = 1$, $x_3 = 1$, $x_4 = 0$, $x_6 = 0$:

$$\begin{array}{cc} x_5 & \\ \begin{bmatrix} - \end{bmatrix} & 6 \end{array}$$ – вырожденная матрица.

Минимизация ДНФ

Метод Квайна-МакКласки

Кратчайшая ДНФ имеет минимум элементарных конъюнкций.

Минимальная ДНФ имеет минимум литералов.

Функция g *имплицирует* функцию f, т. е. $g \Rightarrow f$, если f имеет значение 1 везде, где это значение имеет g.

g – *импликанта* функции f. Всякая элементарная конъюнкция, входящая в ДНФ функции f, является импликантой функции f.

Простая импликанта – это импликанта в виде элементарной конъюнкции, которая перестает быть импликантой при удалении любого литерала.

Максимальный интервал – характеристическое множество простой импликанты.

Сокращенная ДНФ функции f – дизъюнкция всех простых импликант функции f.

Минимизация ДНФ

Метод Квайна-МакКласки требует представление заданной булевой функции в виде совершенной ДНФ.

Процесс минимизации состоит из двух этапов:

1) нахождение сокращенной ДНФ;

2) выделение из множества простых импликант минимального подмножества, составляющего ДНФ заданной функции.

На этапе 1формируется последовательность C_0, C_1, … , C_k, где C_i – множество конъюнкций ранга $n - i$, полученных путем простого склеивания конъюнкций из множества C_{i-1}. Если удалить все поглощаемые конъюнкции, то останутся только все простые импликанты.

На этапе 2 решается задача о покрытии: элементы множества M^1 покрываются максимальными интервалами.

Минимизация ДНФ

Метод Квайна-МакКласки

Этап 1: получение сокращенной ДНФ

$$C_0 = \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array} \begin{array}{c} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{array}, \quad C_1 = \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 0 & 0 & - & 0 \\ - & 0 & 0 & 0 \\ 0 & 0 & 1 & - \\ 1 & 0 & 0 & - \\ 0 & - & 1 & 1 \\ - & 0 & 1 & 1 \\ 1 & 0 & - & 1 \\ - & 1 & 1 & 1 \\ 1 & - & 1 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ \\ \\ \\ * \\ * \\ \\ * \\ * \end{array}, \quad C_2 = \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} - & - & 1 & 1 \end{bmatrix} \end{array}.$$

Минимизация ДНФ

Метод Квайна-МакКласки

Этап 2: решение задачи о покрытии.

Заданы множество $A = M^1$ и совокупность подмножеств $B_1, B_2, \dots, B_m$ множества A в виде матриц

$$\begin{array}{c c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\
\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} & \begin{array}{c} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{array}
\end{array}
\quad \text{и} \quad
\begin{array}{c c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\
\begin{bmatrix} 0 & 0 & - & 0 \\ - & 0 & 0 & 0 \\ 0 & 0 & 1 & - \\ 1 & 0 & 0 & - \\ 1 & 0 & - & 1 \\ - & - & 1 & 1 \end{bmatrix} & \begin{array}{c} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \end{array}
\end{array}.$$

Требуется выделить минимум подмножеств B_i, покрывающих все множество A.

Минимизация ДНФ

Метод Квайна-МакКласки

Этап 2

Задача в матричной форме: в следующей матрице выбрать минимальное количество строк так, чтобы каждый столбец имел единицу хотя бы в одной из них.

$$
\begin{array}{cc}
\begin{array}{cccccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \end{array} & \\
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 1
\end{bmatrix} &
\begin{array}{c} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \end{array}
\end{array}
$$

B_1, B_4 и B_6 – решение.

Минимизация ДНФ

Метод Квайна-МакКласки

Решением примера является матрица

$$\begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 0 & 0 & - & 0 \\ 1 & 0 & 0 & - \\ - & - & 1 & 1 \end{bmatrix} \end{array}.$$

В алгебраической форме: $\bar{x}_1 \bar{x}_2 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_3 x_4$.

Минимизация ДНФ

Метод Квайна-МакКласки

Поиск *определяющих элементов* и *обязательных интервалов.*

Для элемента m_i достаточно найти в M^1 все соседние с ним элементы, а затем построить содержащий их *минимальный поглощающий интервал* U, представляемый вектором $\boldsymbol{u}$.

Для элементов $m_1, m_2, \ldots, m_k$ вектор $\boldsymbol{u}$получается следующим образом: если i-я компонента во всех векторах $m_1, m_2, \ldots, m_k$ имеет значение 0, то $\boldsymbol{u}$ имеет в этой компоненте 0, если 1, то 1. Если i-я компонента имеет различные значения в этих векторах, то i-я компонента вектора $\boldsymbol{u}$ имеет значение «–».

Если все элементы полученного таким образом интервала U принадлежат M^1, то он является максимальным в M^1 и притом обязательным, а m_i является определяющим элементом. В противном случае U не содержится целиком в M^1, а m_i не является определяющим ни для какого интервала.

Минимизация ДНФ

Метод Квайна-МакКласки

Поиск *определяющих элементов* и *обязательных интервалов.*

Чтобы определить, содержится ли интервал U во множестве M^1, достаточно для матрицы, представляющей множество M^1, построить минор, определяемый столбцами, где вектор $\boldsymbol{u}$ имеет значение «–», и строками, не ортогональными вектору $\boldsymbol{u}$. Число строк в этом миноре не превышает 2^p, где p – число компонент вектора $\boldsymbol{u}$, имеющих значение «–». Очевидно, интервал U целиком содержится в M^1 тогда и только тогда, когда число строк в этом миноре равно 2^p.

Минимизация ДНФ

Метод Квайна-МакКласки. Поиск *обязательных интервалов*.

0 1 0 0 0*

0 0 1 1 0

1 0 0 0 1

0 1 0 1 0*

1 1 0 0 0*

0 1 1 1 0

1 1 0 1 0*

1 1 0 0 1

1 0 1 0 1

1 0 1 1 0

1 1 0 1 1

1 1 1 1 0

1 1 1 0 1

0 1 0 0 0

0 1 0 1 0

1 1 0 0 0

– 1 0 – 0

Минимизация ДНФ

Метод Квайна-МакКласки. Поиск *обязательных интервалов*.

0 1 0 0 0*	0 1 0 0 0	0 0 1 1 0
0 0 1 1 0*	0 1 0 1 0	0 1 1 1 0
1 0 0 0 1	1 1 0 0 0	1 0 1 1 0
0 1 0 1 0*	———	———
1 1 0 0 0*	– 1 0 – 0	– – 1 1 0
0 1 1 1 0*		
1 1 0 1 0*		
1 1 0 0 1		
1 0 1 0 1		
1 0 1 1 0*		
1 1 0 1 1		
1 1 1 1 0*		
1 1 1 0 1		

Минимизация ДНФ

Метод Квайна-МакКласки. Поиск *обязательных интервалов*.

0 1 0 0 0*	0 1 0 0 0	0 0 1 1 0	1 0 0 0 1
0 0 1 1 0*	0 1 0 1 0	0 1 1 1 0	1 1 0 0 1
1 0 0 0 1*	1 1 0 0 0	1 0 1 1 0	1 0 1 0 1
0 1 0 1 0*	———	———	———
1 1 0 0 0*	– 1 0 – 0	– – 1 1 0	1 – – 0 1
0 1 1 1 0*			
1 1 0 1 0*			
1 1 0 0 1*			
1 0 1 0 1*			
1 0 1 1 0*			
1 1 0 1 1			
1 1 1 1 0*			
1 1 1 0 1*			

Минимизация ДНФ

Метод Квайна-МакКласки. Поиск *обязательных интервалов*.

0 1 0 0 0*
0 0 1 1 0*
1 0 0 0 1*
0 1 0 1 0*
1 1 0 0 0*
0 1 1 1 0*
1 1 0 1 0*
1 1 0 0 1*
1 0 1 0 1*
1 0 1 1 0*
1 1 0 1 1*
1 1 1 1 0*
1 1 1 0 1*

0 1 0 0 0	0 0 1 1 0	1 0 0 0 1	1 1 0 1 1
0 1 0 1 0	0 1 1 1 0	1 1 0 0 1	1 1 0 1 0
1 1 0 0 0	1 0 1 1 0	1 0 1 0 1	1 1 0 0 1
– 1 0 – 0	– – 1 1 0	1 – – 0 1	1 1 0 – –

Обязательные интервалы покрывают всё M^1.

Решение:

– 1 0 – 0
– – 1 1 0
1 – – 0 1
1 1 0 – –

$$x_2 \bar{x}_3 \bar{x}_5 \vee x_3 x_4 \bar{x}_5 \vee x_1 \bar{x}_4 x_5 \vee x_1 x_2 \bar{x}_3$$

Минимизация ДНФ

Метод Квайна-МакКласки

В предыдущем примере только один обязательный интервал.

0 0 0 0	0 0 0 0	0 0 1 0	1 0 0 0	0 0 1 1
0 0 1 0	0 0 1 0	0 0 0 0	0 0 0 0	0 0 1 0
1 0 0 0	1 0 0 0	0 0 1 1	1 0 0 1	0 1 1 1
0 0 1 1	– 0 – 0	0 0 – –	– 0 0 –	1 0 1 1
1 0 0 1				– – 1 –
0 1 1 1	1 0 0 1	0 1 1 1		
1 0 1 1	1 0 0 0	0 0 1 1		
1 1 1 1	1 0 1 1	1 1 1 1		
	1 0 – –	– – 1 1		

Минимизация ДНФ

Метод Квайна-МакКласки

В предыдущем примере только один обязательный интервал.

			1	2	3	4
0 0 0 0						
0 0 1 0	1 0 0 0 0*	0 0 – 0 +	1	1		
1 0 0 0	2 0 0 1 0*	– 0 0 0	1		1	
0 0 1 1	3 1 0 0 0*	0 0 1 –		1		
1 0 0 1	4 1 0 0 1*	1 0 0 – +			1	1
0 1 1 1	– – 1 1	1 0 – 1				1
1 0 1 1		– – 1 1				
1 1 1 1						

Решение: 0 0 – 0

1 0 0 –

– – 1 1

Минимизация ДНФ

Метод Блейка-Порецкого

Функция задается в произвольной ДНФ. Если преобразовать $x_1 x_2 x_3 x_5 \vee x_2 x_3 x_4 x_5 \vee \bar{x}_1$ в СДНФ, то получим 18 конъюнкций.

Применение обобщенного склеивания: $x_1 x_2 x_3 x_5 \vee x_2 x_3 x_4 x_5 \vee \bar{x}_1 =$
$= x_1 x_2 x_3 x_5 \vee x_2 x_3 x_4 x_5 \vee \bar{x}_1 \vee x_2 x_3 x_5 = \bar{x}_1 \vee x_2 x_3 x_5$.

Процесс минимизации состоит из двух этапов:

1) нахождение сокращенной ДНФ;

2) выделение из множества простых импликант минимального подмножества, составляющего ДНФ заданной функции.

На этапе 1 выполняются операции обобщенного склеивания и простого поглощения:

$$A x \vee B \bar{x} = A x \vee B \bar{x} \vee A B \quad \text{и} \quad A \vee A B = A.$$

Минимизация ДНФ

Метод Блейка-Порецкого

Этап 1: получение сокращенной ДНФ:

$$
\begin{array}{cccc}
x_1 & x_2 & x_3 & x_4
\end{array}
\qquad
\begin{array}{cccc}
x_1 & x_2 & x_3 & x_4
\end{array}
$$

$$
\begin{bmatrix}
1 & 1 & - & 0 \\
1 & 0 & - & 1 \\
0 & 0 & 0 & - \\
0 & - & 1 & 0 \\
- & 1 & 1 & 1
\end{bmatrix}
\begin{array}{c}
1 \\ 2 \\ 3 \\ 4 \\ 5
\end{array}
\Rightarrow
\begin{bmatrix}
1 & 1 & - & 0 \\
1 & 0 & - & 1 \\
0 & 0 & 0 & - \\
0 & - & 1 & 0 \\
- & 1 & 1 & 1 \\
- & 0 & 0 & 1
\end{bmatrix}
\begin{array}{c}
1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 2,3
\end{array}
$$

Минимизация ДНФ

Метод Блейка-Порецкого

Этап 1: получение сокращенной ДНФ:

$$
\begin{array}{cccc}
x_1 & x_2 & x_3 & x_4
\end{array}
\qquad\qquad
\begin{array}{cccc}
x_1 & x_2 & x_3 & x_4
\end{array}
$$

$$
\begin{bmatrix}
1 & 1 & - & 0 \\
1 & 0 & - & 1 \\
0 & 0 & 0 & - \\
0 & - & 1 & 0 \\
- & 1 & 1 & 1
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
1 & 1 & - & 0 \\
1 & 0 & - & 1 \\
0 & 0 & 0 & - \\
0 & - & 1 & 0 \\
- & 1 & 1 & - \\
- & 0 & 0 & 1 \\
0 & 0 & - & 0 \\
1 & - & 1 & 1
\end{bmatrix}
$$

Минимизация ДНФ

Метод Блейка-Порецкого

Этап 2: решение задачи о покрытии.

Поиск *ядра* и *антиядра*. Задача поиска ядра сводится к нахождению избыточных элементарных конъюнкций в ДНФ.

Конъюнкция k избыточна в $D = k \vee D'$, если $k \vee D' = D'$.

Если, подставив в D' любой набор значений переменных, обращающий k в единицу, получим $D' = 1$, то k избыточна.

$$\begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & & \\ \end{array}$$
$$\begin{bmatrix} 1 & 1 & - & 0 \\ 1 & 0 & - & 1 \\ 0 & 0 & 0 & - \\ 0 & - & 1 & 0 \\ - & 1 & 1 & - \\ - & 0 & 0 & 1 \\ 0 & 0 & - & 0 \\ 1 & - & 1 & 1 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}$$

Конъюнкция $x_2 x_3$ (строка 5) не избыточна, т.к. результат подстановки $x_2 = x_3 = 1$, представляемый матрицей

$$\begin{array}{cc} x_1 & x_4 \end{array}$$
$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{array}{c} 1 \\ 4 \\ 8 \end{array},$$

не является тождественной единицей (матрица не вырожденная).

Минимизация ДНФ

Метод Блейка-Порецкого

Этап 2: решение задачи о покрытии.
Поиск ядра и антиядра. Для каждой строки строим минор, образованный столбцами, где она имеет «–», и не ортогональными ей строками. Если минор – невырожденная матрица, то строка принадлежит ядру.

$$\begin{array}{c c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\
\begin{bmatrix} 1 & 1 & - & 0 \\ 1 & 0 & - & 1 \\ 0 & 0 & 0 & - \\ 0 & - & 1 & 0 \\ - & 1 & 1 & - \\ - & 0 & 0 & 1 \\ 0 & 0 & - & 0 \\ 1 & - & 1 & 1 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}
\end{array}$$

1) $\begin{array}{cc} x_3 & \\ \begin{bmatrix} 1 \end{bmatrix} & 5 \end{array}$

я

2) $\begin{array}{cc} x_3 & \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \begin{array}{c} 6 \\ 8 \end{array} \end{array}$

3) $\begin{array}{cc} x_4 & \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{array}{c} 6 \\ 7 \end{array} \end{array}$

4) $\begin{array}{cc} x_2 & \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{array}{c} 5 \\ 7 \end{array} \end{array}$

5) $\begin{array}{cc} \begin{array}{cc} x_1 & x_4 \end{array} & \\ \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} & \begin{array}{c} 1 \\ 4 \\ 8 \end{array} \end{array}$

я

6) $\begin{array}{cc} x_1 & \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{array}{c} 2 \\ 3 \end{array} \end{array}$

7) $\begin{array}{cc} x_3 & \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \begin{array}{c} 3 \\ 4 \end{array} \end{array}$

8) $\begin{array}{cc} x_2 & \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \begin{array}{c} 2 \\ 5 \end{array} \end{array}$

Минимизация ДНФ

Метод Блейка-Порецкого

Этап 2: решение задачи о покрытии.

$$\begin{array}{c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\
\begin{bmatrix}
1 & 1 & - & 0 \\
1 & 0 & - & 1 \\
0 & 0 & 0 & - \\
0 & - & 1 & 0 \\
- & 1 & 1 & - \\
- & 0 & 0 & 1 \\
0 & 0 & - & 0 \\
1 & - & 1 & 1
\end{bmatrix}
\end{array}
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}$$

$$\begin{array}{c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\
\begin{bmatrix}
1 & 1 & - & 0 \\
- & 1 & 1 & -
\end{bmatrix}
\end{array}
\begin{array}{c} \\ 1 \\ 5 \end{array} \text{ – ядро}$$

Антиядра нет

Элементы, не покрытые ядром:

$$\begin{array}{c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\
\begin{bmatrix}
1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0
\end{bmatrix}
\end{array}
\begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{array}$$

Минимизация ДНФ

Метод Блейка-Порецкого

Этап 2: решение задачи о покрытии.

$$\begin{array}{cc} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\ \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} & \begin{array}{c} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{array} \end{array}$$

надо покрыть интервалами

$$\begin{array}{cc} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\ \begin{bmatrix} 1 & 0 & - & 1 \\ 0 & 0 & 0 & - \\ 0 & - & 1 & 0 \\ - & 0 & 0 & 1 \\ 0 & 0 & - & 0 \\ 1 & - & 1 & 1 \end{bmatrix} & \begin{array}{c} 2 \\ 3 \\ 4 \\ 6 \\ 7 \\ 8 \end{array} \end{array}$$

$$\begin{array}{cc} \begin{array}{ccccc} a_1 & a_2 & a_3 & a_4 & a_5 \end{array} & \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} & \begin{array}{c} 2 \\ 3 \\ 4 \\ 6 \\ 7 \\ 8 \end{array} \end{array} \Rightarrow \begin{array}{cc} \begin{array}{ccccc} a_1 & a_2 & a_3 & a_4 & a_5 \end{array} & \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} & \begin{array}{c} 2 \\ 3 \\ 6 \\ 7 \end{array} \end{array}$$

Решение:

$$\begin{array}{cc} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\ \begin{bmatrix} 1 & 0 & - & 1 \\ 0 & 0 & 0 & - \\ 0 & 0 & - & 0 \end{bmatrix} & \begin{array}{c} 2 \\ 3 \\ 7 \end{array} \end{array}$$

Минимизация ДНФ

Метод Блейка-Порецкого

Этап 2: решение задачи о покрытии.

Для получения окончательного решения к ядру

$$\begin{array}{cc} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\ \begin{bmatrix} 1 & 1 & - & 0 \\ - & 1 & 1 & - \end{bmatrix} & \begin{array}{c} 1 \\ 5 \end{array} \end{array}$$

добавляем

$$\begin{array}{cc} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\ \begin{bmatrix} 1 & 0 & - & 1 \\ 0 & 0 & 0 & - \\ 0 & 0 & - & 0 \end{bmatrix} & \begin{array}{c} 2 \\ 3 \\ 7 \end{array} \end{array}$$

Результат:

$$\begin{array}{cc} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\ \begin{bmatrix} 1 & 1 & - & 0 \\ - & 1 & 1 & - \\ 1 & 0 & - & 1 \\ 0 & 0 & 0 & - \\ 0 & 0 & - & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{array} \end{array}$$

В алгебраической форме:

$$x_1 x_2 \overline{x}_4 \vee x_2 x_3 \vee x_1 \overline{x}_2 x_4 \vee$$
$$\vee \overline{x}_1 \overline{x}_2 \overline{x}_3 \vee \overline{x}_1 \overline{x}_2 \overline{x}_4$$

Минимизация ДНФ

Метод Блейка-Порецкого

$P(\boldsymbol{U})$ – совокупность непустых подмножеств множества номеров строк матрицы $\boldsymbol{U}$ такая, что для каждого элемента множества M^1 имеется подмножество в $P(\boldsymbol{U})$, из номеров всех строк матрицы $\boldsymbol{U}$, представляющих интервалы, содержащие данный элемент.

$$
\begin{array}{cc}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\
\left[\begin{array}{cccc}
1 & 1 & - & 0 \\
1 & 0 & - & 1 \\
0 & 0 & 0 & - \\
0 & - & 1 & 0 \\
- & 1 & 1 & - \\
- & 0 & 0 & 1 \\
0 & 0 & - & 0 \\
1 & - & 1 & 1
\end{array}\right] &
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}
\end{array}
$$

x_4

x_3

3,7	4,7		3,6
	4,5	5	
1	1,5	5,8	
		2,8	2,6

$x_1\,x_2$

$P(\boldsymbol{U}) = \{(1), (5), (1,5), (2,6), (2,8), (3,6), (3,7), (4,5), (4,7), (5,8)\}$

Задача: Выбрать минимум строк из $\boldsymbol{U}$ так, чтобы каждый элемент из $P(\boldsymbol{U})$ содержал номер хотя бы одной из этих строк.

Минимизация ДНФ

Метод Блейка-Порецкого

У т в е р ж д е н и е. Если из матриц $\boldsymbol{U}_1$ и $\boldsymbol{U}_2$ построить матрицу $\boldsymbol{U}$, приписав столбцы матрицы $\boldsymbol{U}_2$ к столбцам $\boldsymbol{U}_1$, то множество $P(\boldsymbol{U})$ можно получить, взяв за его элементы всевозможные непустые пересечения элементов $P(\boldsymbol{U}_1)$ с элементами $P(\boldsymbol{U}_2)$.

$$
\begin{array}{cc}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\
\left[\begin{array}{cccc}
1 & 1 & - & 0 \\
1 & 0 & - & 1 \\
0 & 0 & 0 & - \\
0 & - & 1 & 0 \\
- & 1 & 1 & - \\
- & 0 & 0 & 1 \\
0 & 0 & - & 0 \\
1 & - & 1 & 1
\end{array}\right] &
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}
\end{array}
$$

$P(\boldsymbol{U}_1) = \{(1,2,5,6,8), (3,4,5,6,7)\};$

$P(\boldsymbol{U}_2) = \{(1,5,8), (4,5), (2,6,8), (3,4,6,7)\};$

$P(\boldsymbol{U}_3) = \{(1), (2,6), (3,6,7), (1,5,8), (4,5), (2,8), (4,7)\};$

$P(\boldsymbol{U}_4) = P(\boldsymbol{U}) = \{(1), (5), (1,5), (2,6), (2,8), (3,6), (3,7), (4,5), (4,7), (5,8)\}$

Задача: Выбрать минимум строк из $\boldsymbol{U}$ так, чтобы каждый элемент из $P(\boldsymbol{U})$ содержал номер хотя бы одной из этих строк.

Минимизация ДНФ

Метод Блейка-Порецкого

Правила редукции:

1. Если $A \in P(\boldsymbol{U})$, $B \in P(\boldsymbol{U})$ и $A \subseteq B$, то B удаляется из $P(\boldsymbol{U})$.
2. Если номер i присутствует только в тех элементах множества $P(\boldsymbol{U})$, где присутствует номер k, то i удаляется отовсюду.

$$\begin{array}{c c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\
\begin{bmatrix}
1 & 1 & - & 0 \\
1 & 0 & - & 1 \\
0 & 0 & 0 & - \\
0 & - & 1 & 0 \\
- & 1 & 1 & - \\
- & 0 & 0 & 1 \\
0 & 0 & - & 0 \\
1 & - & 1 & 1
\end{bmatrix} &
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}
\end{array}$$

$P(\boldsymbol{U})$ = {(1), (5), (1,5), (2,6), (2,8), (3,6), (3,7), (4,5), (4,7), (5,8)}.

Строки 1 и 5 составляют ядро.

Результат применения правила 1:

{(2,6), (2,8), (3,6), (3,7), (4,7)}.

Результат применения правила 2:

{(2,6), (2), (3,6), (3,7), (7)}.

Для получения окончательного решения надо к обязательным строкам 1, 2, 5 и 7 добавить строку 3 либо строку 6.

Минимизация ДНФ

Метод Блейка-Порецкого

Решения

$$\begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & & \\ \end{array}$$

$$\begin{array}{c c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\ \left[\begin{array}{cccc} 1 & 1 & - & 0 \\ 1 & 0 & - & 1 \\ 0 & 0 & 0 & - \\ 0 & - & 1 & 0 \\ - & 1 & 1 & - \\ - & 0 & 0 & 1 \\ 0 & 0 & - & 0 \\ 1 & - & 1 & 1 \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \end{array}$$

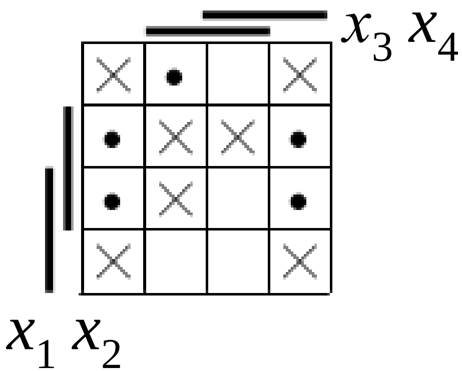
$$\begin{array}{c c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\ \left[\begin{array}{cccc} 1 & 1 & - & 0 \\ 1 & 0 & - & 1 \\ 0 & 0 & 0 & - \\ - & 1 & 1 & - \\ 0 & 0 & - & 0 \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 3 \\ 5 \\ 7 \end{array} \end{array}$$

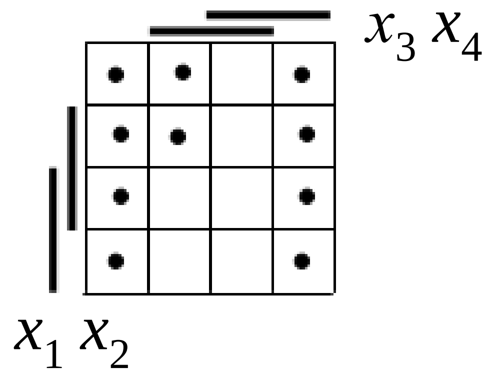
$$\begin{array}{c c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\ \left[\begin{array}{cccc} 1 & 1 & - & 0 \\ 1 & 0 & - & 1 \\ - & 1 & 1 & - \\ - & 0 & 0 & 1 \\ 0 & 0 & - & 0 \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 5 \\ 6 \\ 7 \end{array} \end{array}$$

Минимизация не полностью определенных булевых функций

M^1, M^0 и M^- – области, где соответственно функция имеет значения 1, 0 и не определена.

Достаточно задать два подмножества, например, M^1 и M^0.





Результат минимизации: $\bar{x}_1 \bar{x}_4 \vee \bar{x}_3$

Минимизация не полностью определенных булевых функций

Отношение *реализации*: функция g реализует функцию f $(g \rangle f)$,
если $M^1{}_f \subseteq M^1{}_g$ и $M^0{}_f \subseteq M^0{}_g$.

Постановка задачи:

Для функции f найти минимальную (или кратчайшую) ДНФ среди всех ДНФ всех функций g, удовлетворяющих условию $f \langle g$.

Число вариантов доопределения: $2^{|M_f^-|}$.

Выделим два из них – $f_{\min}$ и $f_{\max}$:

$$M^1_{f_{\min}} = M^1_f, \qquad M^0_{f_{\min}} = M^0_f \cup M^-_f;$$
$$M^1_{f_{\max}} = M^1_f \cup M^-_f, \qquad M^0_{f_{\max}} = M^0_f.$$

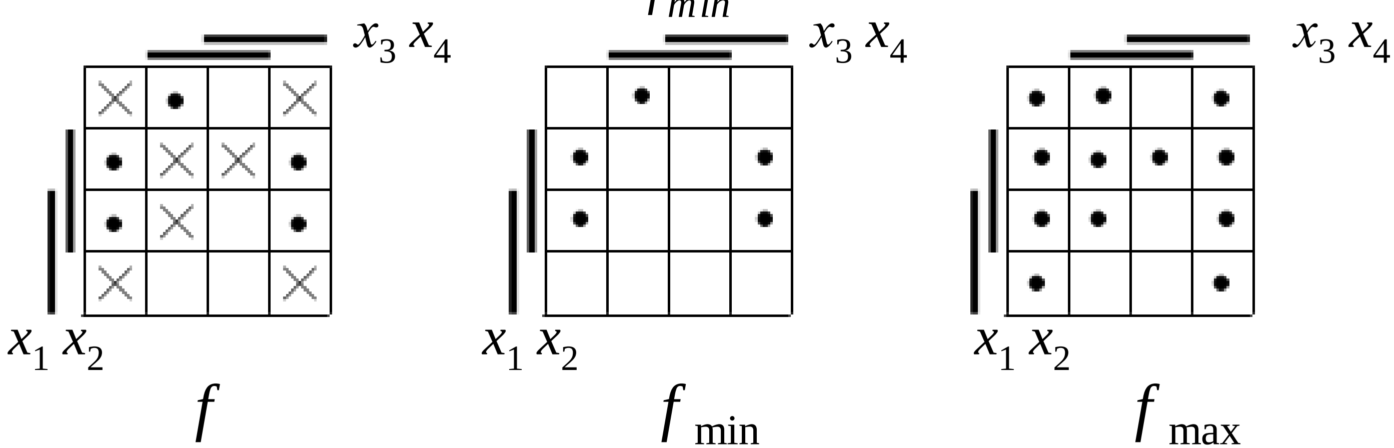
Минимизация не полностью определенных булевых функций

Распространение метода Квайна-МакКласки

Этапы:

1) нахождение множества всех максимальных интервалов для $f_{\max}$;

2) покрытие ими элементов из $M^1_{f_{min}}$.



Элементы $\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \\ \end{array}$ $\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$ надо покрыть интервалами $\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \\ \end{array}$ $\begin{bmatrix} 0 & - & - & 0 \\ - & 1 & - & 0 \\ - & - & 0 & - \\ 0 & 1 & - & - \end{bmatrix}$.

Минимизация не полностью определенных булевых функций

Распространение метода Квайна-МакКласки

Элементы $\begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{array}$ надо покрыть интервалами $\begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 0 & - & - & 0 \\ - & 1 & - & 0 \\ - & - & 0 & - \\ 0 & 1 & - & - \end{bmatrix} \end{array}$.

Матрица покрытия:

$$\begin{array}{c} \begin{array}{ccccc} a_1 & a_2 & a_3 & a_4 & a_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ B_1 \\ B_2 \\ B_3 \\ B_4 \end{array}$$

Результат:

$$\begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 0 & - & - & 0 \\ - & - & 0 & - \end{bmatrix} \end{array}$$

$$f = \bar{x}_1 \bar{x}_4 \vee \bar{x}_3,$$

Минимизация слабо определенной функции

Если $|M^1| + |M^0| << |M^-|$, то применение описанного метода потребует формирования большого количества бесполезных интервалов.

Интервально поглощаемое множество – множество элементов из M^1, для которых существует интервал, содержащий все эти элементы и не пересекающийся с множеством M^0.

Максимальное интервально поглощаемое множество – не содержится в качестве собственного подмножества в другом интервально поглощаемом множестве.

Этапы:

1) получение всех максимальных интервально поглощаемых множеств;
2) получение кратчайшего покрытия ими элементов множества M^1;
3) максимальное расширение выбранных интервалов.

Минимизация слабо определенной функции

Этап 1: получение всех максимальных интервально поглощаемых множеств.

Используется лексикографический перебор.

Для проверки, является ли $M^1{}_i \subseteq M^1$ интервально поглощаемым множеством, надо построить *минимальный покрывающий интервал* для $M^1{}_i$, т. е. наименьший по мощности интервал, содержащий все элементы множества $M^1{}_i$, и затем проверить, не пересекается ли он с множеством M^0.

Нахождение минимального покрывающего интервала для $M^1{}_i$: если значения одноименных компонент булевых векторов, принадлежащих $M^1{}_i$, совпадают, то это значение присваивается соответствующей компоненте получаемого троичного вектора, а если нет, то данная компонента принимает значение «–».

Для (1 0 1 1 1 0 0), (0 0 1 0 1 1 0) и (1 0 1 1 0 1 0) получим (– 0 1 – – – 0),

Минимизация слабо определенной функции

Этап 1: получение всех максимальных интервально поглощаемых множеств.

$$M^1 = \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}, \quad M^0 = \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}.$$

Элементы – 1, 2; интервал – (– 0 1 – – 0). Не пересекается с M^0, следовательно, {1, 2} – интервально поглощаемое множество.

Элементы – 1, 2, 3; интервал – (– – – – – 0). Пересекается с M^0.

Максимальные множества:		Интервалы:
	{1, 2, 4}	(– 0 1 – – –)
	{1, 3}	(– – – 1 1 0)
	{1, 5}	(– 0 – – 1 –)
	{2, 4, 5}	(– 0 – 0 – –)
	{3, 5}	(0 – 0 – 1 –)

Минимизация слабо определенной функции

Этап 2: получение кратчайшего покрытия элементов множества M^1 максимальными интервально поглощаемыми множествами.

Максимальные множества:	$\{1, 2, 4\}$	Интервалы: $(-\ 0\ 1\ -\ -\ -)$
	$\{1, 3\}$	$(-\ -\ -\ 1\ 1\ 0)$
	$\{1, 5\}$	$(-\ 0\ -\ -\ 1\ -)$
	$\{2, 4, 5\}$	$(-\ 0\ -\ 0\ -\ -)$
	$\{3, 5\}$	$(0\ -\ 0\ -\ 1\ -)$

Кратчайшее покрытие: $\{1, 2, 4\}$, $\{3, 5\}$.

Кратчайшая ДНФ:
$$\begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array}$$
$$\begin{bmatrix} - & 0 & 1 & - & - & - \\ 0 & - & 0 & - & 1 & - \end{bmatrix}.$$

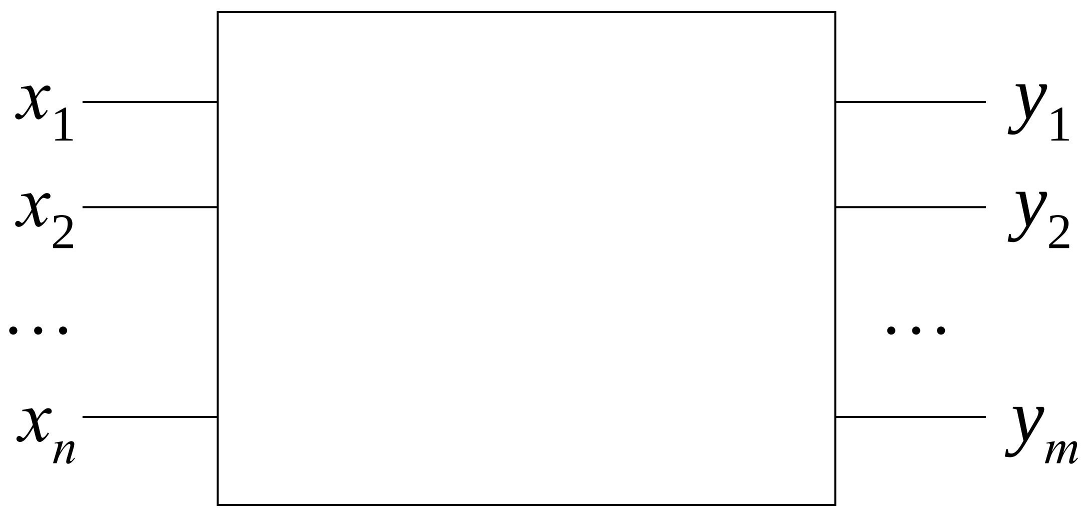
Этап 3: максимальное расширение выбранных интервалов.

$$\begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array}$$
$$\begin{bmatrix} - & 0 & 1 & - & - & - \\ - & - & 0 & - & 1 & - \end{bmatrix} \qquad \bar{x}_2 x_3 \vee \bar{x}_3 x_5$$

Модель комбинационной схемы

Система булевых функций – функциональная модель комбинационной схемы

$$
\begin{aligned}
y_1 &= f_1(x_1, x_2, \ldots, x_n);\\
y_2 &= f_2(x_1, x_2, \ldots, x_n);\\
&\ldots\\
y_m &= f_m(x_1, x_2, \ldots, x_n).
\end{aligned}
$$



Способы задания системы булевых функций

Матрица аргументов

Матрица функций

$$\begin{array}{cc} & \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \boldsymbol{X} = & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}; \end{array} \qquad \begin{array}{cc} & \begin{array}{ccc} y_1 & y_2 & y_3 \end{array} \\ \boldsymbol{Y} = & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$$

Функции полностью определенные и не полностью определенные.

Способы задания системы булевых функций

- Интервальное задание

- $$\boldsymbol{X} = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & - \\ 1 & 0 & - & 1 & 0 & 1 \\ 1 & - & 1 & 0 & - & 0 \\ 0 & 1 & 0 & 1 & 0 & - \\ 1 & 0 & 1 & - & 1 & 1 \\ 0 & 1 & - & 1 & 0 & 1 \end{bmatrix} \end{array}; \qquad \boldsymbol{Y} = \begin{array}{c} \begin{matrix} y_1 & y_2 & y_3 \end{matrix} \\ \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array}$$

- Ортогональным строкам матрицы ***Y*** должны соответствовать ортогональные строки матрицы ***X***.

Способы задания системы булевых функций

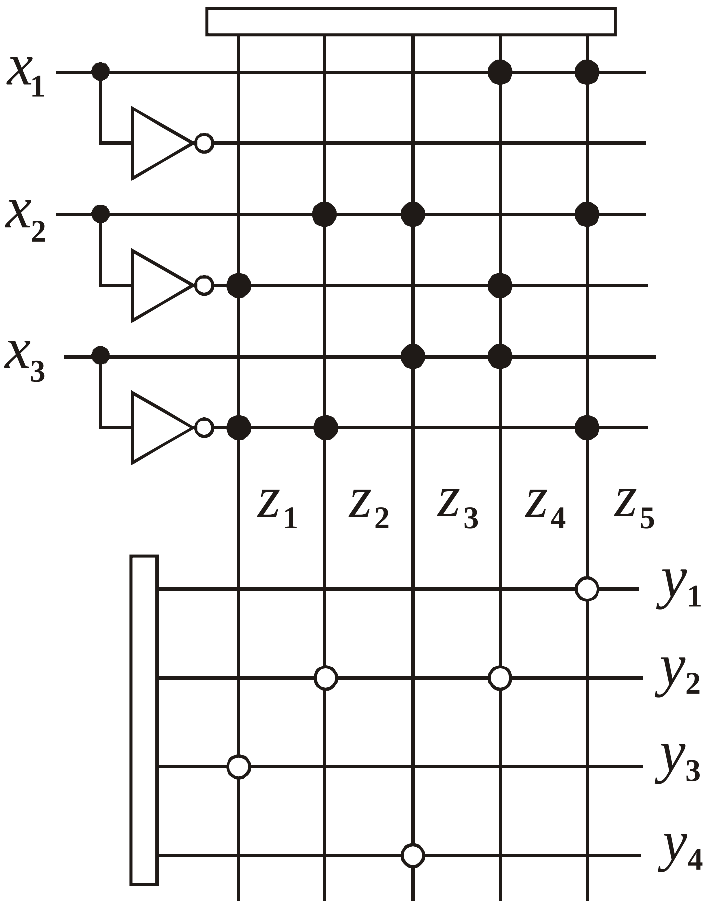
- Задание системы ДНФ

- $$\boldsymbol{X} = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 \end{matrix} \\ \begin{bmatrix} 1 & 0 & - & 1 \\ - & 1 & 0 & - \\ 1 & - & 1 & 0 \\ 0 & - & - & 0 \\ - & 0 & 1 & - \end{bmatrix} \end{array}; \qquad \boldsymbol{Y} = \begin{array}{c} \begin{matrix} y_1 & y_2 & y_3 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \end{array}$$

- $y_1 = x_1 \bar{x}_2 x_4 \vee \bar{x}_1 \bar{x}_4 \vee \bar{x}_2 x_3;$
- $y_2 = x_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_4;$
- $y_3 = x_1 \bar{x}_2 x_4 \vee x_2 \bar{x}_3 \vee x_1 x_3 \bar{x}_4.$

Реализация булевых функций комбинационными схемами

Программируемая логическая матрица (ПЛМ)



И

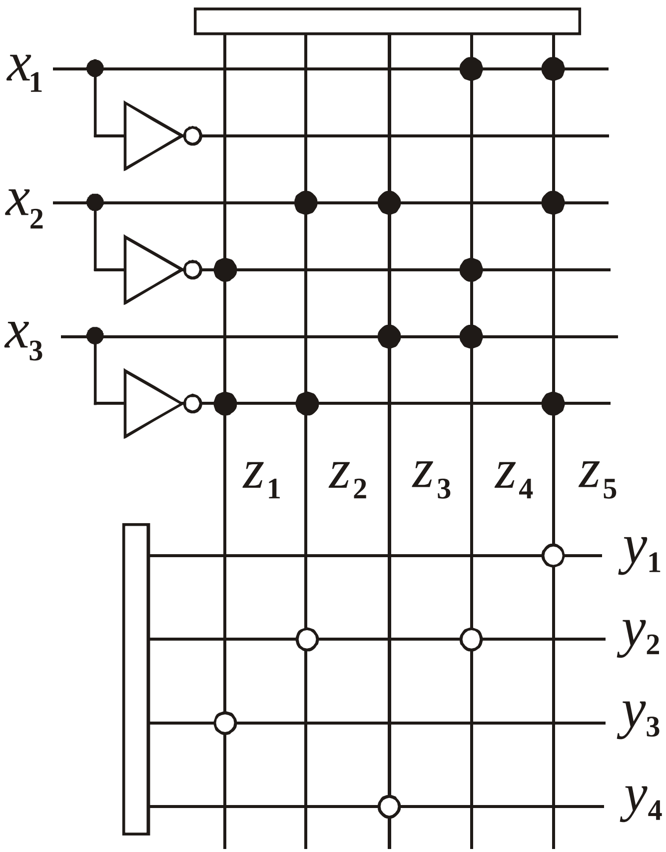
$$z_1 = \bar{x}_2\,\bar{x}_3,$$
$$z_2 = x_2\,\bar{x}_3,$$
$$z_3 = x_2\,x_3,$$
$$z_4 = x_1\,\bar{x}_2\,x_3,$$
$$z_5 = x_1\,x_2\,\bar{x}_3$$

ИЛИ

$$y_1 = x_1\,x_2\,\bar{x}_3;$$
$$y_2 = x_2\,\bar{x}_3 \vee x_1\,\bar{x}_2\,x_3;$$
$$y_3 = \bar{x}_2\,\bar{x}_3;$$
$$y_4 = x_2\,x_3.$$

Реализация булевых функций комбинационными схемами

Программируемая логическая матрица (ПЛМ)



И

$$\begin{array}{ccc} x_1 & x_2 & x_3 \\ \end{array}$$
$$\begin{bmatrix} - & 0 & 0 \\ - & 1 & 0 \\ - & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

ИЛИ

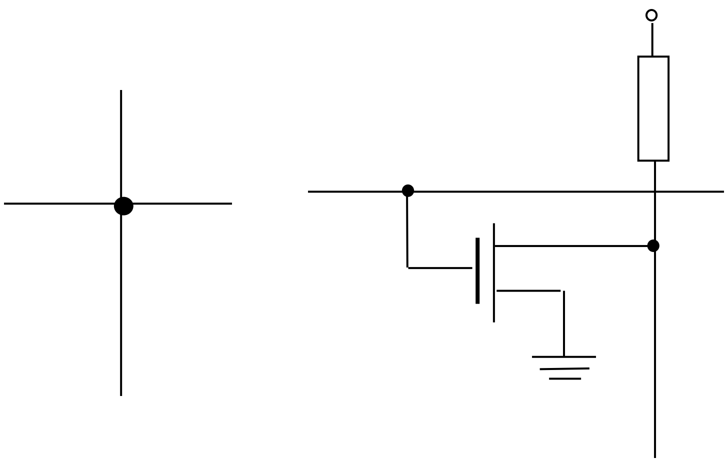
$$\begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \\ \end{array}$$
$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Реализация булевых функций комбинационными схемами.

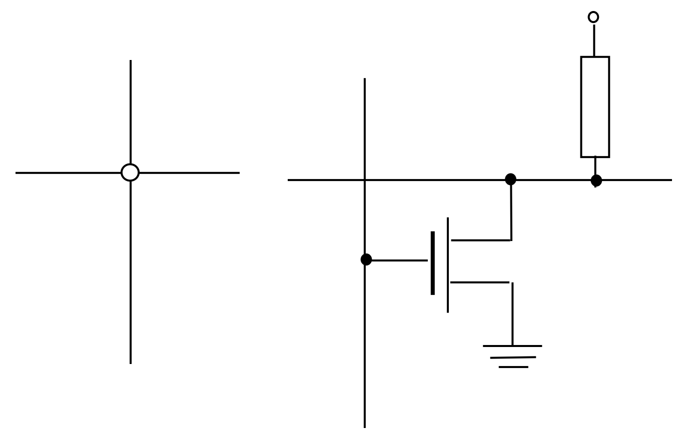
Программируемая логическая матрица (ПЛМ)

Способы транзисторных соединений

В матрице И



В матрице ИЛИ

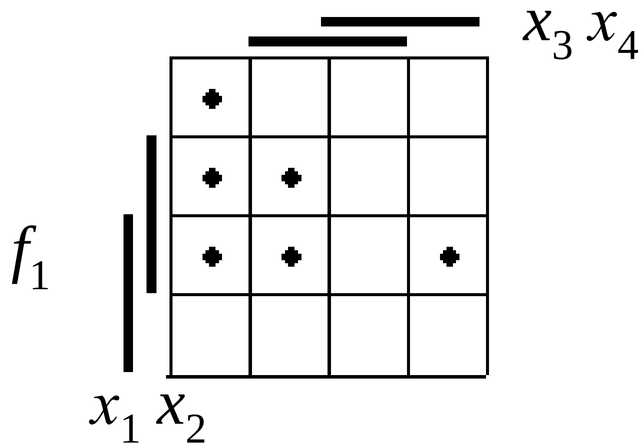


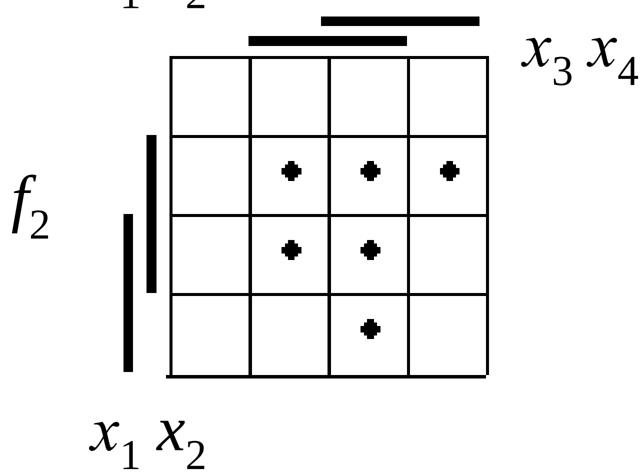
$y_2 = x_2\,\bar{x}_3 \vee x_1\,\bar{x}_2\,x_3$ фактически реализуется как $\overline{\overline{x_2 \vee \bar{x}_3} \vee \overline{x_1 \vee \bar{x}_2 \vee x_3}}$

$\overline{\bar{x}_2\; x_3 \vee \bar{x}_1\; x_2\; \bar{x}_3}$

Минимизация системы ДНФ

Для заданной системы булевых функций получить такую систему ДНФ, в которой общее число различных элементарных конъюнкций было бы минимальным.





Результат раздельной минимизации:

$$
\begin{array}{cccc}
x_1 & x_2 & x_3 & x_4 \\
\end{array}
\qquad\qquad
\begin{array}{cc}
f_1 & f_2
\end{array}
$$

$$
\begin{bmatrix}
0 & - & 0 & 0 \\
- & 1 & - & 0 \\
1 & 1 & 0 & - \\
0 & 1 & - & 1 \\
- & 1 & 1 & - \\
1 & - & 1 & 1
\end{bmatrix}
\begin{array}{c}
1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6
\end{array}
\qquad
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0 \\
0 & 1 \\
0 & 1 \\
0 & 1
\end{bmatrix}
$$

Результат совместной минимизации:

$$
\begin{array}{cccc}
x_1 & x_2 & x_3 & x_4 \\
\end{array}
\qquad\qquad
\begin{array}{cc}
f_1 & f_2
\end{array}
$$

$$
\begin{bmatrix}
0 & - & 0 & 0 \\
- & 1 & 1 & 0 \\
1 & 1 & 0 & - \\
0 & 1 & - & 1 \\
1 & - & 1 & 1
\end{bmatrix}
\begin{array}{c}
1 \\ 2 \\ 3 \\ 4 \\ 5
\end{array}
\qquad
\begin{bmatrix}
1 & 0 \\
1 & 1 \\
1 & 0 \\
0 & 1 \\
0 & 1
\end{bmatrix}
$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Пусть F – некоторая система булевых функций. Элементарная конъюнкция является *обобщенной простой импликантой* для $F' \subseteq F$, если она является импликантой для любой функции из F' и перестает быть импликантой при удалении из нее любого литерала и не является импликантой ни для какой функции, не принадлежащей F'.

Этапы минимизации:

1) нахождение множества всех обобщенных простых импликант для заданной системы;
2) выделение из этого множества минимального подмножества, удовлетворяющего условию, что для всякой функции, принадлежащей заданной системе, можно из этого подмножества получить задающую ее ДНФ.

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 1: получение сокращенной системы ДНФ:

Матрица аргументов Матрица функций Результаты склеивания

$$
\begin{array}{cccccc}
x_1 & x_2 & x_3 & x_4 & & \\
\left[\begin{array}{c} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{array}\right. &
\begin{array}{c} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{array} &
\begin{array}{c} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{array} &
\left.\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{array}\right] &
\begin{array}{c} 1^* \\ 2^* \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}
\end{array}
\qquad
\begin{array}{cc}
f_1 & f_2 \\
\left[\begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array}\right. &
\left.\begin{array}{c} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{array}\right]
\end{array}
$$

$$
\begin{array}{cccc}
x_1 & x_2 & x_3 & x_4 \\
\left[0\right. & - & 0 & \left.0\right]
\end{array}
\qquad
\begin{array}{cc}
f_1 & f_2 \\
\left[1\right. & \left.0\right]
\end{array}
$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 1: получение сокращенной системы ДНФ:

Матрица аргументов Матрица функций Результаты склеивания

$$
\begin{array}{cccc}
x_1 & x_2 & x_3 & x_4
\end{array}
\qquad\qquad
\begin{array}{cc}
f_1 & f_2
\end{array}
$$

$$
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1
\end{bmatrix}
\begin{array}{c}
1* \\ 2* \\ 3* \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10
\end{array}
\qquad
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 1 \\
0 & 1 \\
1 & 1 \\
1 & 0 \\
0 & 1 \\
0 & 1 \\
0 & 1
\end{bmatrix}
$$

$$
\begin{array}{cccc}
x_1 & x_2 & x_3 & x_4
\end{array}
\qquad
\begin{array}{cc}
f_1 & f_2
\end{array}
$$

$$
\begin{bmatrix}
0 & - & 0 & 0 \\
- & 1 & 0 & 0
\end{bmatrix}
\qquad
\begin{bmatrix}
1 & 0 \\
1 & 0
\end{bmatrix}
$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 1: получение сокращенной системы ДНФ:

Матрица аргументов

$$
\begin{array}{cccc}
x_1 & x_2 & x_3 & x_4
\end{array}
$$

$$
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1
\end{bmatrix}
\begin{array}{c}
1* \\ 2* \\ 3* \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10
\end{array}
$$

Матрица функций

$$
\begin{array}{cc}
f_1 & f_2
\end{array}
$$

$$
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 1 \\
0 & 1 \\
1 & 1 \\
1 & 0 \\
0 & 1 \\
0 & 1 \\
0 & 1
\end{bmatrix}
$$

Результаты склеивания

$$
\begin{array}{cccc}
x_1 & x_2 & x_3 & x_4
\end{array}
\qquad
\begin{array}{cc}
f_1 & f_2
\end{array}
$$

$$
\begin{bmatrix}
0 & - & 0 & 0 \\
- & 1 & 0 & 0 \\
0 & 1 & - & 0
\end{bmatrix}
\qquad
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0
\end{bmatrix}
$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 1: получение сокращенной системы ДНФ:

Матрица аргументов

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \end{array}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} 1* \\ 2* \\ 3* \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix}$$

Матрица функций

$$\begin{array}{cc} f_1 & f_2 \end{array}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Результаты склеивания

$$\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \qquad \begin{array}{cc} f_1 & f_2 \end{array}$$

$$\begin{bmatrix} 0 & - & 0 & 0 \\ - & 1 & 0 & 0 \\ 0 & 1 & - & 0 \\ 1 & 1 & - & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 1: получение сокращенной системы ДНФ:

Матрица аргументов

$$\begin{array}{cc} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} & \begin{array}{c} 1* \\ 2* \\ 3* \\ 4 \\ 5 \\ 6 \\ 7* \\ 8 \\ 9 \\ 10 \end{array} \end{array}$$

Матрица функций

$$\begin{array}{c} \begin{array}{cc} f_1 & f_2 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{array}$$

Результаты склеивания

$$\begin{array}{cc} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \begin{array}{cc} f_1 & f_2 \end{array} \\ \begin{bmatrix} 0 & - & 0 & 0 \\ - & 1 & 0 & 0 \\ 0 & 1 & - & 0 \\ 1 & 1 & - & 0 \\ 1 & 1 & 0 & - \end{bmatrix} & \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \end{array}$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 1: получение сокращенной системы ДНФ:

Матрица аргументов

$$\begin{array}{c c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1
\end{bmatrix} &
\begin{array}{c} 1* \\ 2* \\ 3* \\ 4* \\ 5 \\ 6* \\ 7* \\ 8 \\ 9 \\ 10 \end{array}
\end{array}$$

Матрица функций

$$\begin{array}{c}
\begin{array}{cc} f_1 & f_2 \end{array} \\
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 1 \\
0 & 1 \\
1 & 1 \\
1 & 0 \\
0 & 1 \\
0 & 1 \\
0 & 1
\end{bmatrix}
\end{array}$$

Результаты склеивания

$$\begin{array}{c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\
\begin{bmatrix}
0 & - & 0 & 0 \\
- & 1 & 0 & 0 \\
0 & 1 & - & 0 \\
1 & 1 & - & 0 \\
1 & 1 & 0 & - \\
- & 1 & 1 & 0
\end{bmatrix}
\end{array}
\quad
\begin{array}{c}
\begin{array}{cc} f_1 & f_2 \end{array} \\
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 1
\end{bmatrix}
\end{array}$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 1: получение сокращенной системы ДНФ:

Матрица аргументов Матрица функций Результаты склеивания

$$\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \qquad \begin{array}{cc} f_1 & f_2 \end{array}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{array}{c} 1* \\ 2* \\ 3* \\ 4* \\ 5 \\ 6* \\ 7* \\ 8* \\ 9 \\ 10 \end{array} \qquad \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \qquad \begin{array}{cc} f_1 & f_2 \end{array}$$

$$\begin{bmatrix} 0 & - & 0 & 0 \\ - & 1 & 0 & 0 \\ 0 & 1 & - & 0 \\ 1 & 1 & - & 0 \\ 1 & 1 & 0 & - \\ - & 1 & 1 & 0 \\ 0 & 1 & 1 & - \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 1: получение сокращенной системы ДНФ:

Матрица аргументов   Матрица функций   Результаты склеивания

$$\begin{array}{cc} \begin{matrix} x_1 & x_2 & x_3 & x_4 \end{matrix} & \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} & \begin{matrix} 1* \\ 2* \\ 3* \\ 4* \\ 5* \\ 6* \\ 7* \\ 8* \\ 9 \\ 10 \end{matrix} \end{array} \qquad \begin{array}{c} \begin{matrix} f_1 & f_2 \end{matrix} \\ \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{array}$$

$$\begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 \end{matrix} \\ \begin{bmatrix} 0 & - & 0 & 0 \\ - & 1 & 0 & 0 \\ 0 & 1 & - & 0 \\ 1 & 1 & - & 0 \\ 1 & 1 & 0 & - \\ - & 1 & 1 & 0 \\ 0 & 1 & 1 & - \\ 0 & 1 & - & 1 \end{bmatrix} \end{array} \qquad \begin{array}{c} \begin{matrix} f_1 & f_2 \end{matrix} \\ \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{array}$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 1: получение сокращенной системы ДНФ:

Матрица аргументов Матрица функций Результаты склеивания

$$
\begin{array}{c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1
\end{bmatrix}
\end{array}
\begin{array}{c}
\\
1* \\ 2* \\ 3* \\ 4* \\ 5* \\ 6* \\ 7* \\ 8* \\ 9 \\ 10*
\end{array}
\qquad
\begin{array}{c}
\begin{array}{cc} f_1 & f_2 \end{array} \\
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 1 \\
0 & 1 \\
1 & 1 \\
1 & 0 \\
0 & 1 \\
0 & 1 \\
0 & 1
\end{bmatrix}
\end{array}
\qquad
\begin{array}{c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\
\begin{bmatrix}
0 & - & 0 & 0 \\
- & 1 & 0 & 0 \\
0 & 1 & - & 0 \\
1 & 1 & - & 0 \\
1 & 1 & 0 & - \\
- & 1 & 1 & 0 \\
0 & 1 & 1 & - \\
0 & 1 & - & 1 \\
1 & 1 & 1 & -
\end{bmatrix}
\end{array}
\quad
\begin{array}{c}
\begin{array}{cc} f_1 & f_2 \end{array} \\
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 1 \\
0 & 1 \\
0 & 1 \\
0 & 1
\end{bmatrix}
\end{array}
$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 1: получение сокращенной системы ДНФ:

Матрица аргументов   Матрица функций   Результаты склеивания

$$
\begin{array}{cc}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1
\end{bmatrix}
&
\begin{array}{c}
1* \\ 2* \\ 3* \\ 4* \\ 5* \\ 6* \\ 7* \\ 8* \\ 9* \\ 10*
\end{array}
\end{array}
\qquad
\begin{array}{c}
\begin{array}{cc} f_1 & f_2 \end{array} \\
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 1 \\
0 & 1 \\
1 & 1 \\
1 & 0 \\
0 & 1 \\
0 & 1 \\
0 & 1
\end{bmatrix}
\end{array}
\qquad
\begin{array}{c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\
\begin{bmatrix}
0 & - & 0 & 0 \\
- & 1 & 0 & 0 \\
0 & 1 & - & 0 \\
1 & 1 & - & 0 \\
1 & 1 & 0 & - \\
- & 1 & 1 & 0 \\
0 & 1 & 1 & - \\
0 & 1 & - & 1 \\
1 & 1 & 1 & - \\
- & 1 & 1 & 1
\end{bmatrix}
\end{array}
\qquad
\begin{array}{c}
\begin{array}{cc} f_1 & f_2 \end{array} \\
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 1 \\
0 & 1 \\
0 & 1 \\
0 & 1 \\
0 & 1
\end{bmatrix}
\end{array}
$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 1: получение сокращенной системы ДНФ:

Матрица аргументов

$$\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \\ \end{array}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} 1* \\ 2* \\ 3* \\ 4* \\ 5* \\ 6* \\ 7* \\ 8* \\ 9* \\ 10* \end{matrix}$$

Матрица функций

$$\begin{array}{cc} f_1 & f_2 \end{array}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Результаты склеивания

$$\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \qquad \begin{array}{cc} f_1 & f_2 \end{array}$$

$$\begin{bmatrix} 0 & - & 0 & 0 \\ - & 1 & 0 & 0 \\ 0 & 1 & - & 0 \\ 1 & 1 & - & 0 \\ 1 & 1 & 0 & - \\ - & 1 & 1 & 0 \\ 0 & 1 & 1 & - \\ 0 & 1 & - & 1 \\ 1 & 1 & 1 & - \\ - & 1 & 1 & 1 \\ 1 & - & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 1: получение сокращенной системы ДНФ:

Матрица аргументов Матрица функций Результаты склеивания

$$\begin{array}{cc}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1
\end{bmatrix} &
\begin{array}{c} 1* \\ 2* \\ 3* \\ 4* \\ 5* \\ 6* \\ 7* \\ 8* \\ 9* \\ 10* \end{array}
\end{array}
\qquad
\begin{array}{c}
\begin{array}{cc} f_1 & f_2 \end{array} \\
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 1 \\
0 & 1 \\
1 & 1 \\
1 & 0 \\
0 & 1 \\
0 & 1 \\
0 & 1
\end{bmatrix}
\end{array}$$

$$\begin{array}{ccc}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & & \begin{array}{cc} f_1 & f_2 \end{array} \\
\begin{bmatrix}
0 & - & 0 & 0 \\
- & 1 & 0 & 0 \\
0 & 1 & - & 0 \\
1 & 1 & - & 0 \\
1 & 1 & 0 & - \\
- & 1 & 1 & 0 \\
0 & 1 & 1 & - \\
0 & 1 & - & 1 \\
1 & 1 & 1 & - \\
- & 1 & 1 & 1 \\
1 & - & 1 & 1
\end{bmatrix} &
\begin{array}{c} \\ * \\ * \\ * \\ \\ \\ * \\ \\ * \\ * \\ \\ \end{array} &
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 1 \\
0 & 1 \\
0 & 1 \\
0 & 1 \\
0 & 1 \\
0 & 1
\end{bmatrix}
\end{array}$$

$$\begin{bmatrix}
- & 1 & - & 0 \\
- & 1 & 1 & -
\end{bmatrix}
\qquad
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}$$

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 2: решение задачи о покрытии.

$$\begin{array}{c c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} & \\
\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}
\end{array}
\qquad
\begin{array}{c}
\begin{array}{cc} f_1 & f_2 \end{array} \\
\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}
\end{array}$$

Сокращенная система ДНФ:

Интервалы Характеристики

$$\begin{array}{c}
\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\
\begin{bmatrix} 0 & - & 0 & 0 \\ 1 & 1 & 0 & - \\ - & 1 & 1 & 0 \\ 0 & 1 & - & 1 \\ 1 & - & 1 & 1 \\ - & 1 & - & 0 \\ - & 1 & 1 & - \end{bmatrix}
\end{array}
\qquad
\begin{array}{c}
\begin{array}{cc} f_1 & f_2 \end{array} \\
\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}
\end{array}$$

Пары $(1, f_1)$, $(2, f_1)$, $(3, f_1)$, $(4, f_1)$, $(6, f_1)$, $(7, f_1)$, $(4, f_2)$, $(5, f_2)$, $(6, f_2)$, $(8, f_2)$, $(9, f_2)$, $(10, f_2)$ надо покрыть парами (интервал, характеристика).

Минимизация системы ДНФ

Распространение метода Квайна-МакКласки

Этап 2: решение задачи о покрытии.

$$
\begin{array}{cccc}
x_1 & x_2 & x_3 & x_4
\end{array}
\quad
\begin{array}{cc}
f_1 & f_2
\end{array}
\quad
\begin{array}{cccccccccccc}
(1,f_1) & (2,f_1) & (3,f_1) & (4,f_1) & (6,f_1) & (7,f_1) & (4,f_2) & (5,f_2) & (6,f_2) & (8,f_2) & (9,f_2) & (10,f_2)
\end{array}
$$

$$
\begin{bmatrix}
0 & - & 0 & 0 \\
1 & 1 & 0 & - \\
- & 1 & 1 & 0 \\
0 & 1 & - & 1 \\
1 & - & 1 & 1 \\
- & 1 & - & 0 \\
- & 1 & 1 & -
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
1 & 1 \\
0 & 1 \\
0 & 1 \\
1 & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 1 & & & & & & & & & & \\
 & & 1 & & & 1 & & & & & & \\
 & & & 1 & 1 & & 1 & & 1 & & & \\
 & & & & & & & 1 & & 1 & & \\
 & & & & & & & & & & 1 & 1 \\
 & 1 & 1 & 1 & 1 & & & & & & & \\
 & & & & & & 1 & & 1 & 1 & & 1
\end{bmatrix}
$$

Строки 1-я, 2-я 4-я и 5-я сразу вносятся в решение. Оставшиеся непокрытыми столбцы покрывает строка 3-я.

Результат:

$$f_1(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 \vee x_2 x_3 \bar{x}_4;$$

$$f_2(x_1, x_2, x_3, x_4) = \bar{x}_1 x_2 x_4 \vee x_1 x_3 x_4 \vee x_2 x_3 \bar{x}_4.$$

Минимизация системы слабо определенных функций

$F = \{f_1, f_2, \ldots , f_m\}$ – система слабо определенных булевых функций. Любая f_i задана с помощью множеств $M^1{}_i$ и $M^0{}_i$.

Получение кратчайшей системы ДНФ для системы F сводится к нахождению такого минимального множества интервалов булева пространства M, чтобы каждое из множеств $M^1{}_i$ покрывалось теми из них, которые не пересекаются с множеством $M^0{}_i$.

Интервально поглощаемое множество – такое множество элементов вида (m_j, f_k), где $m_j \in M^1{}_k$, что существует интервал пространства M, который для каждой пары (m_j, f_k) из этого множества содержит m_j и не пересекается с множеством $M^0{}_k$.

Максимальное интервально поглощаемое множество – не содержится в качестве собственного подмножества в другом интервально поглощаемом множестве.

Минимизация системы слабо определенных функций

Этапы:

1) получение всех максимальных интервально поглощаемых множеств;
2) получение кратчайшего покрытия ими всех элементов вида (m_j, f_k);
3) максимальное расширение выбранных интервалов.

Всякое интервально поглощаемое множество является декартовым произведением $M_p \times F_p$, где M_p – множество некоторых элементов булева пространства, а F_p – множество функций, принимающих значение 1 на этих элементах.

Минимизация системы слабо определенных функций

Этап 1: получение всех максимальных интервально поглощаемых множеств.

$$
\boldsymbol{X} = \begin{array}{c c}
\begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} & \\
\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\end{array}, \qquad
\boldsymbol{Y} = \begin{array}{c}
\begin{array}{ccc} f_1 & f_2 & f_3 \end{array} \\
\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{array}.
$$

$\{1\}$, $\{f_1, f_2\}$; $(0\ 1\ 0\ 1\ 0)$;
$\{1, 3, 5\}$, $\{f_2\}$; $(0\ 1\ –\ –\ –)$;
$\{1, 4\}$, $\{f_1\}$; $(0\ –\ –\ 1\ 0)$;
$\{2, 3\}$, $\{f_2, f_3\}$; $(–\ –\ 1\ –\ 1)$;
$\{2, 3, 5\}$, $\{f_2\}$; $(–\ –\ –\ –\ 1)$;
$\{2, 6\}$, $\{f_3\}$; $(1\ –\ 1\ –\ –)$;
$\{3, 6\}$, $\{f_3\}$; $(–\ 1\ 1\ 0\ –)$.

Минимизация системы слабо определенных функций

Этап 2: получение кратчайшего покрытия пар $(1, f_1)$, $(4, f_1)$, $(1, f_2)$, $(2, f_2)$, $(3, f_2)$, $(5, f_2)$, $(2, f_3)$, $(3, f_3)$, $(6, f_3)$ максимальными интервально поглощаемыми множествами.

		$(1, f_1)$	$(4, f_1)$	$(1, f_2)$	$(2, f_2)$	$(3, f_2)$	$(5, f_2)$	$(2, f_3)$	$(3, f_3)$	$(6, f_3)$
$\{1\}$	$\{f_1, f_2\}$	1		1						
$\{1, 3, 5\}$	$\{f_2\}$			1		1	1			
$\{1, 4\}$	$\{f_1\}$	1	1							
$\{2, 3\}$	$\{f_2, f_3\}$				1	1		1	1	
$\{2, 3, 5\}$	$\{f_2\}$				1	1	1			
$\{2, 6\}$	$\{f_3\}$							1		1
$\{3, 6\}$	$\{f_3\}$								1	1

$(\{1, 4\}, \{f_1\})$, $(\{1, 3, 5\}, \{f_2\})$, $(\{2, 3\}, \{f_2, f_3\})$, $(\{2, 6\}, \{f_3\})$ составляют кратчайшее покрытие.

Минимизация системы слабо определенных функций

Этап 3: максимальное расширение выбранных интервалов.

$$\boldsymbol{X} = \begin{array}{c} x_1\quad x_2\quad x_3\quad x_4\quad x_5 \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}, \qquad \boldsymbol{Y} = \begin{array}{c} f_1\quad f_2\quad f_3 \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}.$$

$$\begin{array}{l} (\{1, 3, 5\},\{f_2\}); \\ (\{1, 4\}, \{f_1\}); \\ (\{2, 3\}, \{f_2, f_3\}); \\ (\{2, 6\}, \{f_3\}). \end{array} \begin{array}{c} x_1\quad x_2\quad x_3\quad x_4\quad x_5 \\ \begin{bmatrix} 0 & 1 & - & - & - \\ 0 & - & - & 1 & 0 \\ - & - & 1 & - & 1 \\ 1 & - & 1 & - & - \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} x_1\quad x_2\quad x_3\quad x_4\quad x_5 \\ \begin{bmatrix} 0 & 1 & - & - & - \\ - & - & - & 1 & 0 \\ - & - & 1 & - & 1 \\ 1 & - & - & - & - \end{bmatrix} \end{array} \begin{array}{c} f_1\quad f_2\quad f_3 \\ \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

Декомпозиция булевых функций

Задача состоит в том, чтобы представить заданную функцию в виде суперпозиции более простых функций.

Примером суперпозиции вида $f(\boldsymbol{x}) = \varphi(g_1, g_2, \dots, g_m)$, где $g_i = g_i(\boldsymbol{z}_i)$, а $\boldsymbol{z}_i$ – булев вектор, составленный из компонент вектора $\boldsymbol{x}$ $(i = 1, 2, \dots, m)$, является дизъюнктивная нормальная форма, где в качестве функций g_i выступают элементарные конъюнкции, а в качестве φ – дизъюнкция.

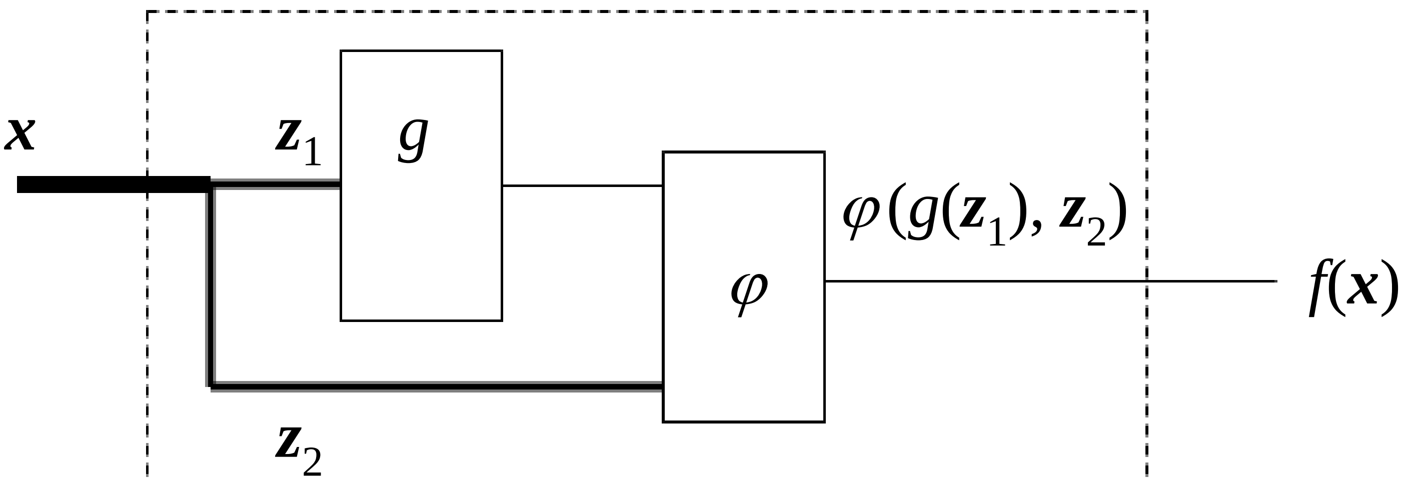
Двухблочная разделительная декомпозиция

$f(\boldsymbol{x}) = \varphi(g(\boldsymbol{z}_1), \boldsymbol{z}_2)$, где $\boldsymbol{x}$ – вектор, компонентами которого являются переменные из множества X, $\boldsymbol{z}_1$ и $\boldsymbol{z}_2$ – векторы, компоненты которых составляют непустые множества Z_1 и Z_2 соответственно, причем $Z_1 \cup Z_2 = X$, $Z_1 \cap Z_2 = \varnothing$.

Декомпозиция булевых функций

Двухблочная разделительная декомпозиция

$f(\boldsymbol{x}) = \varphi(g(\boldsymbol{z}_1), \boldsymbol{z}_2)$,



Декомпозиция нетривиальная, когда $1 < |Z_1| < |X|$.

Декомпозиция булевых функций

Двухблочная разделительная декомпозиция

Необходимое и достаточное условие существования нетривиальной двухблочной разделительной декомпозиции для заданной функции при заданном разбиении множества аргументов выражается через *карту декомпозиции*.

Это двумерная таблица, строки которой кодируются значениями $\boldsymbol{z}_1$, а столбцы – значениями $\boldsymbol{z}_2$. На пересечении строки и столбца – значение $f(\boldsymbol{x}) = f(\boldsymbol{z}_1, \boldsymbol{z}_2)$.

У т в е р ж д е н и е: Полностью определенная булева функция $f(\boldsymbol{x})$ допускает двухблочную разделительную декомпозицию в форме $f(\boldsymbol{x}) = \varphi(g(\boldsymbol{z}_1), \boldsymbol{z}_2)$ тогда и только тогда, когда в карте декомпозиции для (Z_1, Z_2) имеется не более двух различных значений строк.

Декомпозиция булевых функций

Двухблочная разделительная декомпозиция

Пусть функция $f(x_1, x_2, x_3, x_4, x_5)$, для которой надо получить суперпозицию $f(x_1, x_2, x_3, x_4, x_5) = \varphi(g(x_1, x_2, x_3), x_4, x_5)$, задана картой декомпозиции

x_1, x_2, x_3 \ x_4, x_5	00	01	10	11	
0 0 0	1	1	0	0	1
0 0 1	0	1	1	1	0
0 1 0	0	1	1	1	0
0 1 1	0	1	1	1	0
1 0 0	0	1	1	1	0
1 0 1	0	1	1	1	0
1 1 0	1	1	0	0	1
1 1 1	1	1	0	0	1

$$g(x_1, x_2, x_3) = \bar{x}_1\ \bar{x}_2\ \bar{x}_3 \vee$$
$$\vee\ x_1\, x_2;$$

Задание функции $\varphi(g, x_4, x_5)$:

g \ x_4, x_5	00	01	10	11
0	0	1	1	1
1	1	1	0	0

$$\varphi(g, x_4, x_5) = g\ \bar{x}_4 \vee \bar{g}\, x_4 \vee$$
$$\vee\ \bar{x}_4\, x_5.$$

Декомпозиция булевых функций

Двухблочная разделительная декомпозиция не полностью определенных булевых функций

Для заданной не полностью определенной функции $f(\boldsymbol{x})$ и разбиения множества аргументов (Z_1, Z_2) найти суперпозицию $\varphi(g(\boldsymbol{z}_1), \boldsymbol{z}_2)$, реализующую $f(\boldsymbol{x})$.

Не полностью определенная булева функция допускает двухблочную разделительную декомпозицию, если ее можно доопределить до функции, для которой существует искомая суперпозиция.

Карта декомпозиции частичной функции

x_1, x_2 \ x_3, x_4	00	01	10	11
0 0	0	0	1	–
0 1	1	0	–	1
1 0	1	0	0	1
1 1	–	0	–	0

Карта декомпозиции доопределения функции

x_1, x_2 \ x_3, x_4	00	01	10	11
0 0	0	0	1	0
0 1	1	0	0	1
1 0	1	0	0	1
1 1	0	0	1	0

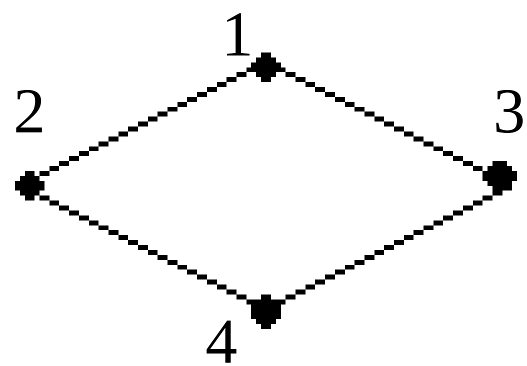
$$f(x_1, x_2, x_3, x_4) = \\ = \varphi(g(x_1, x_2), x_3, x_4)$$

Декомпозиция булевых функций

Двухблочная разделительная декомпозиция не полностью определенных булевых функций

Строки карты декомпозиции можно рассматривать как троичные векторы. Граф ортогональности строк карты декомпозиции:

$$\begin{array}{c}\begin{bmatrix} 0 & 0 & 1 & - \\ 1 & 0 & - & 1 \\ 1 & 0 & 0 & 1 \\ - & 0 & - & 0 \end{bmatrix}\end{array} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}$$



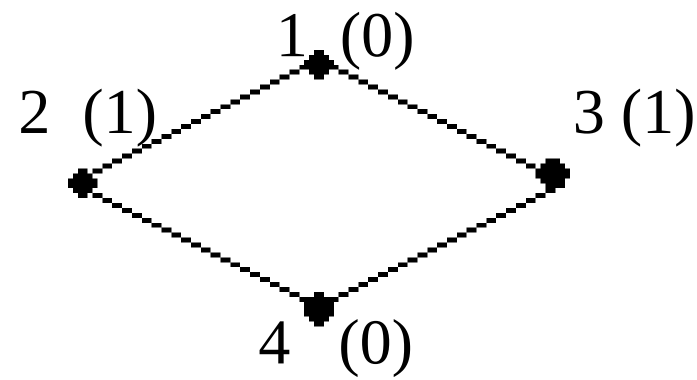
У т в е р ж д е н и е: Не полностью определенная булева функция допускает двухблочную разделительную декомпозицию, если и только если граф ортогональности строк ее карты декомпозиции является бихроматическим.

Декомпозиция булевых функций

Двухблочная разделительная декомпозиция не полностью определенных булевых функций

$$f(x_1, x_2, x_3, x_4) = \varphi(g(x_1, x_2), x_3, x_4)$$

x_3	0	0	1	1		
x_4	0	1	0	1	$x_1 x_2$	g
	0	0	1	–	0 0	0
	1	0	–	1	0 1	1
	1	0	0	1	1 0	1
	–	0	–	0	1 1	0



x_3	0	0	1	1	
x_4	0	1	0	1	g
	0	0	1	0	0
	1	0	0	1	1

$$g = \bar{x}_1 x_2 \vee x_1 \bar{x}_2$$

$$\varphi = g(x_3 x_4 \vee \bar{x}_3 \bar{x}_4) \vee \bar{g}\, x_3 \bar{x}_4$$

Декомпозиция булевых функций

Многоблочные разделительные декомпозиции

Задается разбиение множества X аргументов исходной функции $f(\boldsymbol{x})$ на более чем два подмножества: $X = Z_1 \cup Z_2 \cup \ldots \cup Z_m$, $Z_i \cap Z_j = \varnothing$, $i \neq j$.

Последовательная разделительная декомпозиция. Сначала получается суперпозиция

$$f(\boldsymbol{x}) = \varphi(g(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{m-1}), \mathbf{z}_m),$$

затем разлагается функция g:

$$g(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{m-1}) = \varphi'(g'(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{m-2}), \mathbf{z}_{m-1})$$

и т. д. В результате получаем

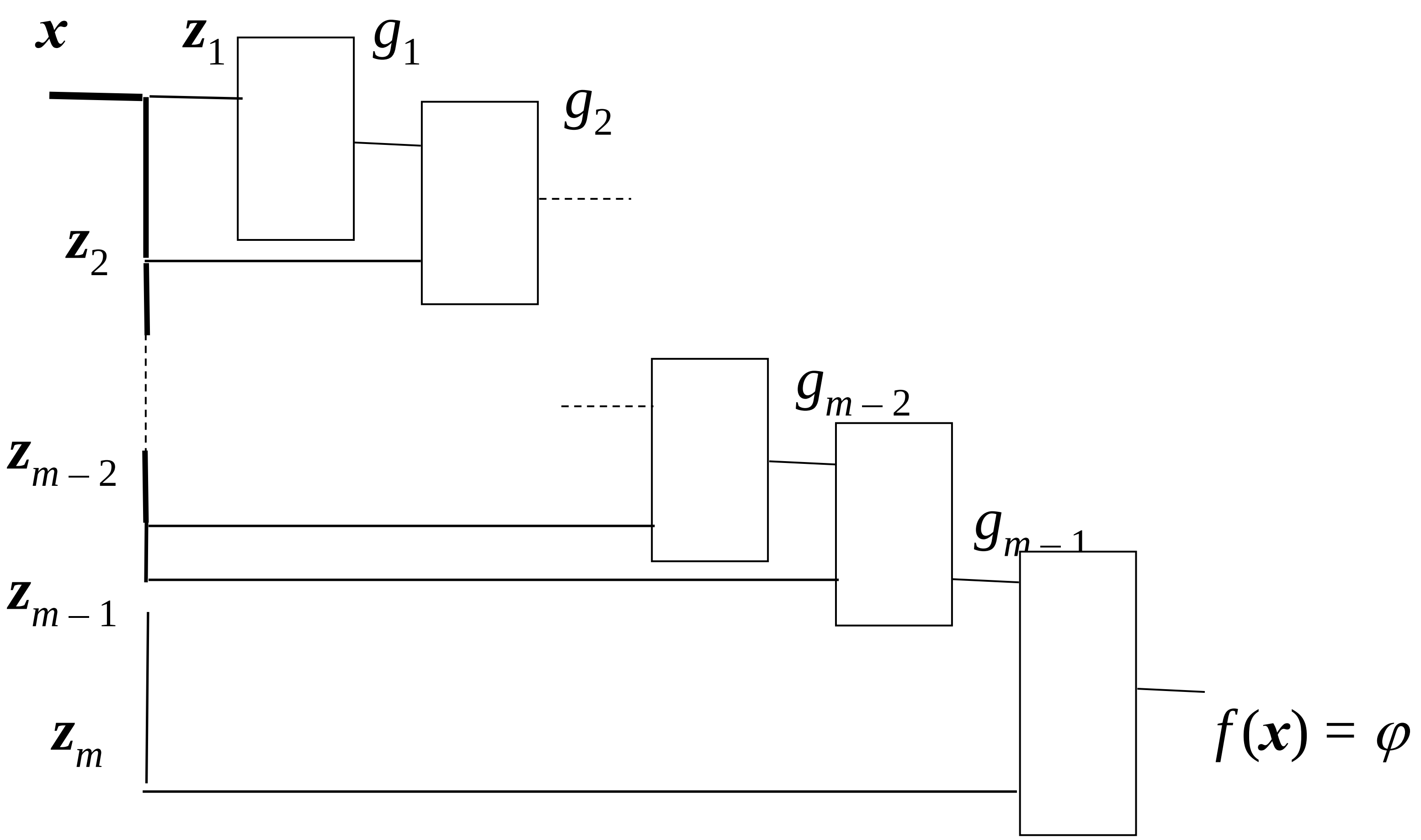
$$f(\boldsymbol{x}) = \varphi(g_{m-1}(g_{m-2}(\ldots(g_1(\mathbf{z}_1), \mathbf{z}_2)\ldots), \mathbf{z}_{m-1}), \mathbf{z}_m).$$

Декомпозиция булевых функций

Многоблочные разделительные декомпозиции

Последовательная разделительная декомпозиция.

$$f(\boldsymbol{x}) = \varphi(g_{m-1}(g_{m-2}(\ldots(g_1(\boldsymbol{z}_1), \boldsymbol{z}_2)\ldots), \boldsymbol{z}_{m-1}), \boldsymbol{z}_m)$$



Декомпозиция булевых функций

Многоблочные разделительные декомпозиции

Последовательная разделительная декомпозиция.

Пример: $f(x_1, x_2, x_3, x_4) = \varphi(g_2(g_1(x_1, x_2), x_3), x_4)$

x_1	x_2	x_3	x_4
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$Z_1 = \{x_1, x_2\}$

$Z_2 = \{x_3\}$

$Z_3 = \{x_4\}$

			x_4	
x_1	x_2	x_3	0	1
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

$\varphi'(g, x_4)$

	x_4	
g	0	1
0	0	1
1	1	0

$g(x_1, x_2, x_3) = x_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3,$ $\qquad \varphi'(g, x_4) = g \bar{x}_4 \vee \bar{g} x_4.$

Декомпозиция булевых функций

Многоблочные разделительные декомпозиции

Последовательная разделительная декомпозиция.

Пример: $f(x_1, x_2, x_3, x_4) = \varphi(g_2(g_1(x_1, x_2), x_3), x_4)$

$$g(x_1, x_2, x_3) = x_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3, \qquad \varphi'(g, x_4) = g\, \bar{x}_4 \vee \bar{g}\, x_4.$$

x_1	x_2	x_3
1	0	0
0	1	0

$Z_1 = \{x_1, x_2\}$

$Z_2 = \{x_3\}$

x_1	x_2	$x_3 = 0$	$x_3 = 1$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	0

g_1	$x_3 = 0$	$x_3 = 1$
0	0	0
1	1	0

$$f(x_1, x_2, x_3, x_4) = \varphi(g_2(g_1(x_1, x_2), x_3), x_4) = g_2\, \bar{x}_4 \vee \bar{g}_2\, x_4;$$

$$g_2(g_1, x_3) = g_1\, \bar{x}_3;$$

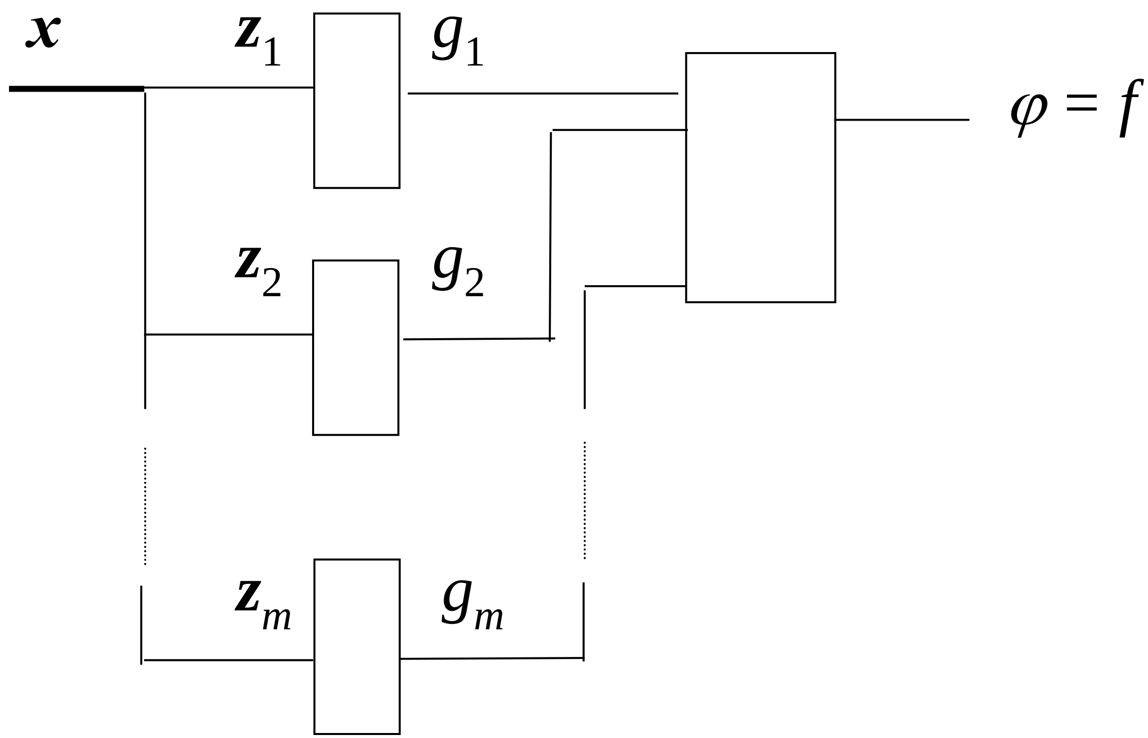
$$g_1(x_1, x_2) = x_1 \bar{x}_2 \vee \bar{x}_1 x_2\, .$$

Декомпозиция булевых функций

Многоблочные разделительные декомпозиции

Параллельная разделительная декомпозиция по разбиению $(Z_1, Z_2, \ldots, Z_m)$ приводит функцию $f(\boldsymbol{x})$ к виду

$$f(\boldsymbol{x}) = \varphi(g_1(\mathbf{z}_1), g_2(\mathbf{z}_2), \ldots, g_m(\mathbf{z}_m)).$$



Декомпозиция булевых функций

Многоблочные разделительные декомпозиции

Параллельная разделительная декомпозиция по разбиению $(Z_1, Z_2, \dots, Z_m)$

У т в е р ж д е н и е. Булева функция $f(\boldsymbol{x})$ допускает параллельную разделительную декомпозицию вида

$$f(\boldsymbol{x}) = \varphi(g_1(\mathbf{z}_1), g_2(\mathbf{z}_2), \dots, g_m(\mathbf{z}_m)).$$

тогда и только тогда, когда она допускает двухблочные разделительные декомпозиции вида

$$f(\boldsymbol{x}) = \varphi_1(g_1(\mathbf{z}_1), \mathbf{z}_2, \dots, \mathbf{z}_m);$$
$$f(\boldsymbol{x}) = \varphi_2(\mathbf{z}_1, g_2(\mathbf{z}_2), \dots, \mathbf{z}_m);$$
$$\dots$$
$$f(\boldsymbol{x}) = \varphi_m(\mathbf{z}_1, \mathbf{z}_2, \dots, g_m(\mathbf{z}_m)).$$

Декомпозиция булевых функций

Многоблочные разделительные декомпозиции

Параллельная разделительная декомпозиция

Пусть функция $f(\boldsymbol{x})$ задана с помощью таблицы (картой декомпозиции):

$x_1x_2x_3$ \ $x_4x_5x_6$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1	g_1
0 0 0	1	0	1	1	1	0	1	0	1
0 0 1	1	0	1	1	1	0	1	0	1
0 1 0	0	1	0	0	0	1	0	1	0
0 1 1	0	1	0	0	0	1	0	1	0
1 0 0	0	1	0	0	0	1	0	1	0
1 0 1	0	1	0	0	0	1	0	1	0
1 1 0	0	1	0	0	0	1	0	1	0
1 1 1	1	0	1	1	1	0	1	0	1

Существуют нетривиальные разложения

$$f(\boldsymbol{x}) = \varphi_1(g_1(x_1, x_2, x_3), x_4, x_5, x_6),$$
$$f(\boldsymbol{x}) = \varphi_2(x_1, x_2, x_3, g_2(x_4, x_5, x_6)).$$

Декомпозиция булевых функций

Многоблочные разделительные декомпозиции

Параллельная разделительная декомпозиция

$x_1x_2x_3$ \ $x_4x_5x_6$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1	g_1
0 0 0	1	0	1	1	1	0	1	0	1
0 0 1	1	0	1	1	1	0	1	0	1
0 1 0	0	1	0	0	0	1	0	1	0
0 1 1	0	1	0	0	0	1	0	1	0
1 0 0	0	1	0	0	0	1	0	1	0
1 0 1	0	1	0	0	0	1	0	1	0
1 1 0	0	1	0	0	0	1	0	1	0
1 1 1	1	0	1	1	1	0	1	0	1

$f(\boldsymbol{x})$ допускает декомпозицию вида $f(x) = \varphi(g_1, g_2)$.

$$g_1(x_1, x_2, x_3) = \bar{x}_1\,\bar{x}_2 \vee x_1 x_2 x_3.$$

Декомпозиция булевых функций

Многоблочные разделительные декомпозиции

Параллельная разделительная декомпозиция

g_1 \ $x_4x_5x_6$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0	0	1	0	0	0	1	0	1
1	1	0	1	1	1	0	1	0
g_2	1	0	1	1	1	0	1	0

$$g_1(x_1, x_2, x_3) = \bar{x}_1\ \bar{x}_2 \vee x_1 x_2 x_3.$$

$$g_2(x_4, x_5, x_6) = \bar{x}_6 \vee \bar{x}_4 x_5$$

$$\varphi(g_1, g_2) = \bar{g}_1\ \bar{g}_2 \vee g_1\ g_2$$

Декомпозиция булевых функций

Двухблочная неразделительная декомпозиция

$f(\boldsymbol{x}) = \varphi(g(\boldsymbol{z}_1), \boldsymbol{z}_2)$

Компонентами векторов $\boldsymbol{x}$, $\boldsymbol{z}_1$ и $\boldsymbol{z}_2$ служат булевы переменные из множеств X, Z_1 и Z_2, соответственно, причем $Z_1 \cup Z_2 = X$, $Z_1 \cap Z_2 \neq \varnothing$.

Естественные ограничения: $|Z_1| < |X|$ и $|Z_2| + 1 < |X|$.

Задача сводится к задаче разделительной декомпозиции не полностью определенной булевой функции.

Неопределенность получается при построении карты декомпозиции.

Декомпозиция булевых функций

Двухблочная неразделительная декомпозиция

$$\begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \\ \end{array}$$
$$\begin{bmatrix} - & 0 & 1 & 1 & - \\ 0 & 1 & 1 & 0 & - \\ 1 & 1 & - & 0 & 0 \\ - & 1 & 0 & 0 & 1 \\ - & 0 & 0 & - & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

– интервальное задание функции $f(x_1, x_2, x_3, x_4, x_5)$.

Карта декомпозиции для $f(x_1, x_2, x_3, x_4, x_5) = \varphi(g(x_1, x_2, x_3), x_2, x_4, x_5)$:

$x_1x_2x_3$ \ $x_2x_4x_5$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0	1	0	1	0	–	–	–	–
0 0 1	0	0	1	1	–	–	–	–
0 1 0	–	–	–	–	1	1	0	0
0 1 1	–	–	–	–	1	1	0	0
1 0 0	1	0	1	0	–	–	–	–
1 0 1	0	0	1	1	–	–	–	–
1 1 0	–	–	–	–	1	1	0	0
1 1 1	–	–	–	–	1	0	0	0

Декомпозиция булевых функций

Двухблочная неразделительная декомпозиция

$x_1x_2x_3$ \ $x_2x_4x_5$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0	1	0	1	0	–	–	–	–
0 0 1	0	0	1	1	–	–	–	–
0 1 0	–	–	–	–	1	1	0	0
0 1 1	–	–	–	–	1	1	0	0
1 0 0	1	0	1	0	–	–	–	–
1 0 1	0	0	1	1	–	–	–	–
1 1 0	–	–	–	–	1	1	0	0
1 1 1	–	–	–	–	1	0	0	0

У т в е р ж д е н и е. Булева функция $f(\boldsymbol{x})$ допускает двухблочную неразделительную декомпозицию вида $f(\boldsymbol{x}) = \varphi(g(\boldsymbol{z}_1), \boldsymbol{z}_2)$ тогда и только тогда, когда граф ортогональности строк карты декомпозиции для (Z_1, Z_2) является бихроматическим.

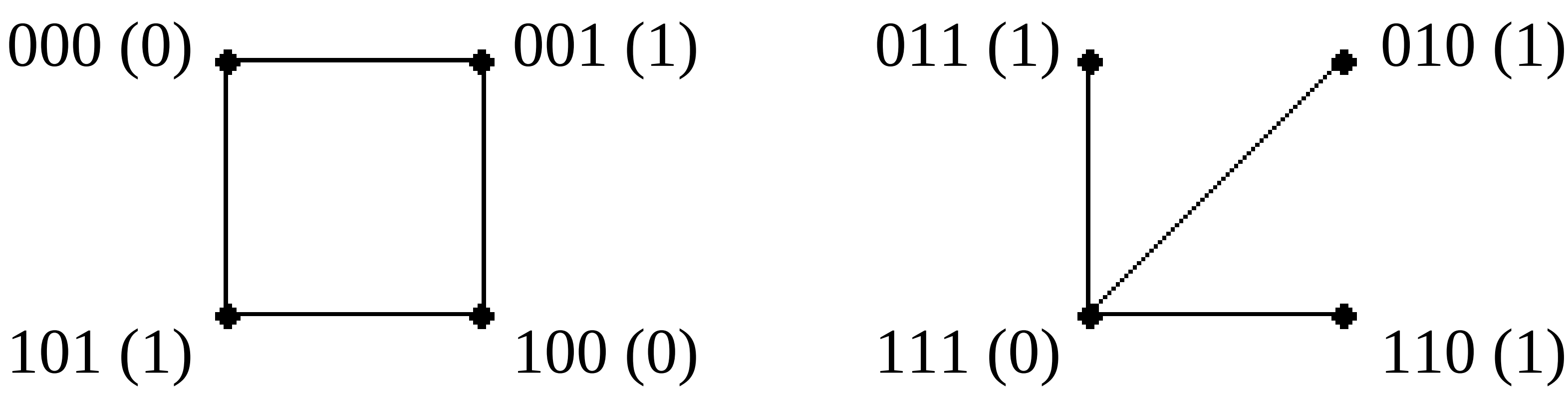
Утверждение справедливо как для полностью, так и для не полностью определенных функций.

Декомпозиция булевых функций

Двухблочная неразделительная декомпозиция

$x_1x_2x_3$ \ $x_2x_4x_5$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0	1	0	1	0	–	–	–	–
0 0 1	0	0	1	1	–	–	–	–
0 1 0	–	–	–	–	1	1	0	0
0 1 1	–	–	–	–	1	1	0	0
1 0 0	1	0	1	0	–	–	–	–
1 0 1	0	0	1	1	–	–	–	–
1 1 0	–	–	–	–	1	1	0	0
1 1 1	–	–	–	–	1	0	0	0

Граф ортогональности строк карты декомпозиции:



Декомпозиция булевых функций

Двухблочная неразделительная декомпозиция

$x_1x_2x_3$ \ $x_2x_4x_5$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1	g
0 0 0	1	0	1	0	–	–	–	–	0
0 0 1	0	0	1	1	–	–	–	–	1
0 1 0	–	–	–	–	1	1	0	0	1
0 1 1	–	–	–	–	1	1	0	0	1
1 0 0	1	0	1	0	–	–	–	–	0
1 0 1	0	0	1	1	–	–	–	–	1
1 1 0	–	–	–	–	1	1	0	0	1
1 1 1	–	–	–	–	1	0	0	0	0

Задание функции $\varphi(g, x_2, x_4, x_5)$:

g \ $x_2x_4x_5$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0	1	0	1	0	1	0	0	0
1	0	0	1	1	1	1	0	0

$$g(x_1, x_2, x_3) = \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \vee x_2 \bar{x}_3;$$

$$\varphi(g, x_2, x_4, x_5) = g\, x_2 \bar{x}_4 \vee g\, \bar{x}_2 x_4 \vee \bar{g}\, \bar{x}_4 \bar{x}_5 \vee \bar{g}\, \bar{x}_2 \bar{x}_5.$$

Декомпозиция булевых функций

Декомпозиция системы булевых функций

Задана система $f_1, f_2, \ldots, f_m$ не полностью определенных булевых функций от общего множества аргументов $X = \{x_1, x_2, \ldots, x_n\}$.

Для заданной пары подмножеств (Z_1, Z_2) множества X, такой, что $X = Z_1 \cup Z_2$, требуется найти суперпозиции

$$f_i(\boldsymbol{x}) = \varphi_i(g_1(\mathbf{z}_1), g_2(\mathbf{z}_1), \ldots, g_k(\mathbf{z}_1), \mathbf{z}_2), \quad i = 1, 2, \ldots, m,$$

где $\mathbf{z}_1$и $\mathbf{z}_2$ – векторы, компонентами которых служат переменные из множеств Z_1 и Z_2, причем $k < |Z_1|, \quad k + |Z_2| < n$, и не важно, пересекаются или нет подмножества Z_1 и Z_2.

У т в е р ж д е н и е. Система не полностью определенных булевых функций $f_1, f_2, \ldots, f_m$ допускает декомпозицию в указанной форме тогда и только тогда, когда хроматическое число графа ортогональности строк ее карты декомпозиции для (Z_1, Z_2) не превышает 2^k.

Декомпозиция булевых функций

Декомпозиция системы булевых функций

Пример. Слабо определенные функции (на наборах, не присутствующих в матрице, значения функций не определены).

$$\begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array}, \quad \begin{array}{c} \begin{array}{ccc} y_1 & y_2 & y_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}$$

Надо получить $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\varphi}(\boldsymbol{w}, \boldsymbol{z}_2)$, $\boldsymbol{w} = \boldsymbol{g}(\boldsymbol{z}_1)$,

где $\boldsymbol{z}_1 = (x_1, x_2, x_3, x_4)$, $\boldsymbol{z}_2 = (x_5, x_6)$.

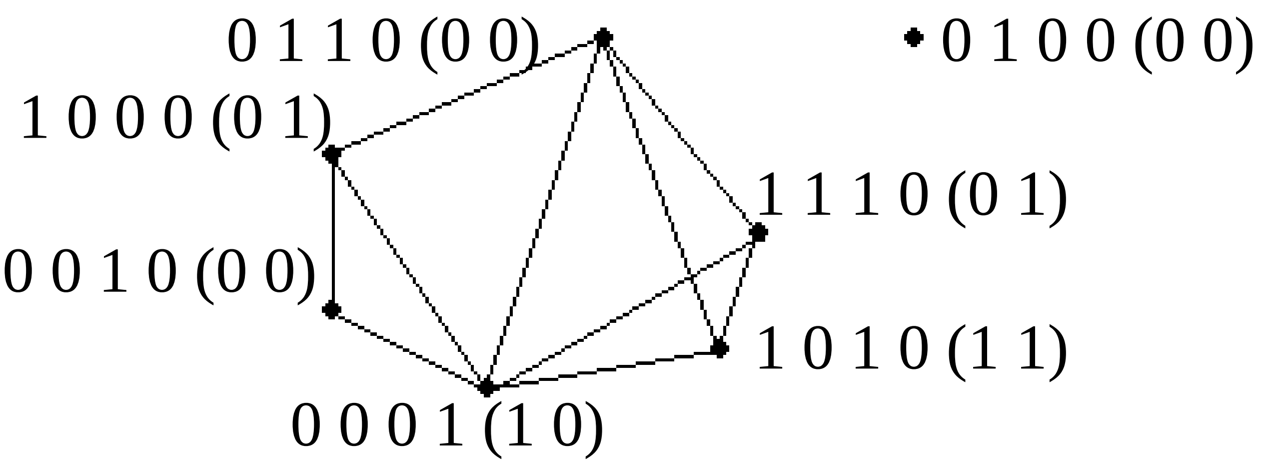
Декомпозиция булевых функций

Декомпозиция системы булевых функций

Пример. Карта декомпозиции ($\mathbf{z}_1 = (x_1, x_2, x_3, x_4)$, $\mathbf{z}_2 = (x_5, x_6)$):

$x_1\, x_2\, x_3\, x_4$ \ $x_5\, x_6$	0 0	0 1	1 1	1 0
1 0 0 0	0 1 1	1 0 0	– – –	– – –
0 1 0 0	– – –	– – –	0 1 0	– – –
0 1 1 0	0 0 1	– – –	– – –	1 0 1
0 0 0 1	– – –	1 0 1	– – –	1 1 0
1 1 1 0	– – –	– – –	– – –	0 1 1
0 0 1 0	– – –	0 0 1	– – –	– – –
1 0 1 0	– – –	– – –	– – –	1 0 0

Граф ортогональности строк карты декомпозиции раскрашивается четырьмя цветами:



Декомпозиция булевых функций

Декомпозиция системы булевых функций

Пример. $\boldsymbol{y} = \boldsymbol{f}(x_1, x_2, x_3, x_4, x_5, x_6) = \boldsymbol{\varphi}(w_1, w_2), x_5, x_6)$.

$x_1\ x_2\ x_3\ x_4$ \ $x_5\ x_6$	0 0	0 1	1 1	1 0	$w_1\ w_2$
1 0 0 0	0 1 1	1 0 0	– – –	– – –	0 1
0 1 0 0	– – –	– – –	0 1 0	– – –	0 0
0 1 1 0	0 0 1	– – –	– – –	1 0 1	0 0
0 0 0 1	– – –	1 0 1	– – –	1 1 0	1 0
1 1 1 0	– – –	– – –	– – –	0 1 1	0 1
0 0 1 0	– – –	0 0 1	– – –	– – –	0 0
1 0 1 0	– – –	– – –	– – –	1 0 0	1 1

Задание системы булевых функций $\boldsymbol{y} = \boldsymbol{\varphi}(\boldsymbol{w}, \boldsymbol{z}_2)$:

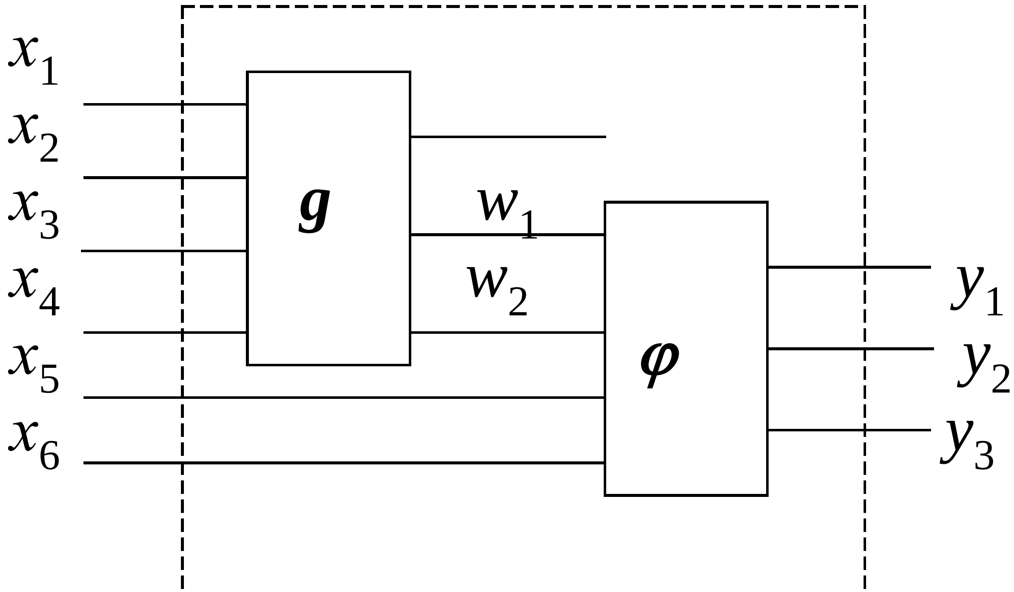
$x_5\ x_6$: 0 0	0 1	1 1	1 0	$w_1\ w_2$
0 1 1	1 0 0	– – –	0 1 1	0 1
0 0 1	0 0 1	0 1 0	1 0 1	0 0
– – –	1 0 1	– – –	1 1 0	1 0
– – –	– – –	– – –	1 0 0	1 1

Декомпозиция булевых функций

Декомпозиция системы булевых функций

Пример. $\boldsymbol{y} = \boldsymbol{f}(x_1, x_2, x_3, x_4, x_5, x_6) = \boldsymbol{\varphi}(w_1, w_2), x_5, x_6)$.

$w_1 = x_1 \bar{x}_2 x_3 \vee x_4$;
$w_2 = x_1$;
$y_1 = \bar{w}_2 x_5 \bar{x}_6 \vee w_1 \vee w_2 x_6$;
$y_2 = x_5 x_6 \vee \bar{w}_1 w_2 \bar{x}_6 \vee w_1 \bar{w}_2 x_5$;
$y_3 = \bar{w}_1 \bar{x}_6 \vee \bar{w}_2 \bar{x}_5$.



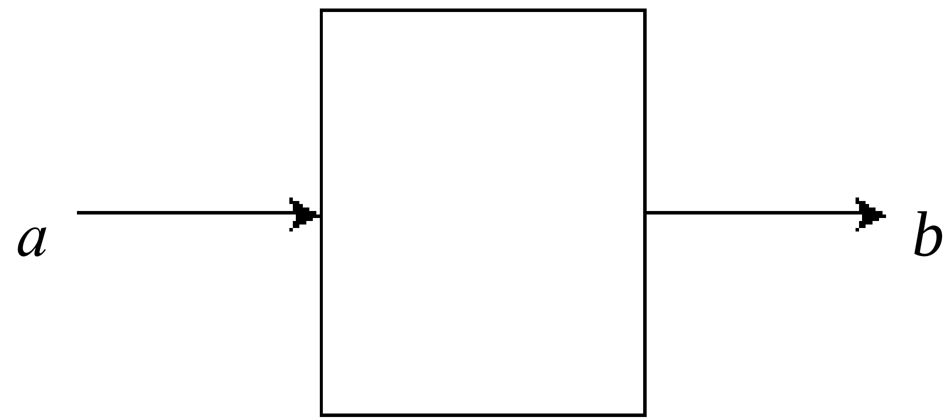
Конечный автомат. Типы

Комбинационный автомат, или **комбинационная схема**

$A = \{a_1, a_2, \ldots, a_\alpha\}$ – множество входных символов, или входной алфавит;

$B = \{b_1, b_2, \ldots, b_\beta\}$ – множество выходных символов, или выходной алфавит;

Вход (переменная a)	Выход (переменная b)
a_1	b_{i_1}
a_2	b_{i_2}
…	…
a_α	b_{i_α}



$$\Phi : A \to B$$

Конечный автомат. Типы

Автомат с памятью или **последовательностный автомат**

$A = \{a_1, a_2, \dots, a_\alpha\}$ – множество входных символов;

$B = \{b_1, b_2, \dots, b_\beta\}$ – множество выходных символов.

$$a_1\ a_1\ a_3\ a_2\ a_2\ a_4\ a_3, \qquad \alpha = 4;$$

$$b_2\ b_3\ b_2\ b_1\ b_3\ b_2\ b_1. \qquad \beta = 3.$$

$Q = \{q_1, q_2, \dots, q_\gamma\}$ – множество состояний.

Поведение можно описать последовательностью строк вида

$$(a_i, q_j) \to (q_s, b_t).$$

$\Psi : A \times Q \to Q$ – функция переходов;

$\Phi : A \times Q \to B$ – функция выходов.

Конечный автомат. Автомат Мили, автомат Мура ($\Phi : Q \to B$).

Конечный автомат. Типы

$\Psi(a, q) = q^+$, q^+ – состояние в которое переходит автомат из состояния q, если на вход его подан символ a.

$\Phi(a, q) = b$, b – выходной символ, выдаваемый автоматом в состоянии q при поступлении на его вход символа a.

Для автомата Мура $\Phi(q) = b$.

Полный автомат, частичный автомат.

Реализации:

Синхронный автомат – моменты времени, когда определяется следующее состояние, зафиксированы.

Асинхронный автомат – эти моменты определяются изменением входного сигнала.

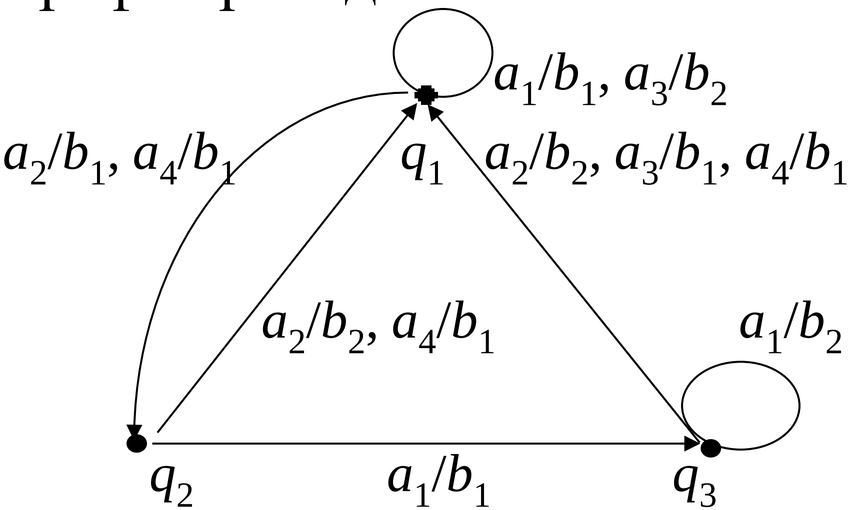
Требование прямого перехода: если $\Psi(a, q_i) = q_j$, то $\Psi(a, q_j) = q_j$.

Представления автомата

Таблица переходов и выходов

	a_1	a_2	a_3	a_4
q_1	q_1,b_1	q_2,b_1	q_1,b_2	q_2,b_1
q_2	q_3,b_1	q_1,b_2	-,-	q_1,b_1
q_3	q_3,b_2	q_1,b_2	q_1,b_1	q_1,b_1

Граф переходов



Матрица переходов

$$
\begin{array}{c}
\\ q_1 \\ q_2 \\ q_3
\end{array}
\begin{array}{c}
\begin{array}{ccc} \quad q_1 \quad & \quad q_2 \quad & q_3 \end{array} \\
\begin{bmatrix}
a_1/b_1,\ a_3/b_2 & a_2/b_1,\ a_4/b_1 & - \\
a_2/b_2,\ a_4/b_1 & - & a_1/b_1 \\
a_2/b_3,\ a_3/b_1,\ a_4/b_1 & - & a_1/b_2
\end{bmatrix}
\end{array}
$$

Связь между моделями Мили и Мура

Задан автомат Мура $M = (A, B, Q, \Psi, \Phi)$ и требуется получить эквивалентный ему автомат Мили $M^* = (A^*, B^*, Q^*, \Psi^*, \Phi^*)$.

$A^* = A$ и $B^* = B$. Положим $Q^* = Q$ и $\Psi^* = \Psi$.

Пусть $\Psi(a, q) = q'$ и $\Phi(q') = b$, где $q, q' \in Q$, $a \in A$ и $b \in B$.

Положим $\Phi^*(a, q) = b$.

	a_1	a_2	a_3	Φ
q_1	q_3	q_2	q_2	0
q_2	q_1	q_4	q_3	1
q_3	q_2	q_2	q_1	0
q_4	q_3	q_4	q_4	1

	a_1	a_2	a_3
q_1	q_3,0	q_2,1	q_2,1
q_2	q_1,0	q_4,1	q_3,0
q_3	q_2,1	q_2,1	q_1,0
q_4	q_3,0	q_4,1	q_4,1

Связь между моделями Мили и Мура

Задан автомат Мили $M = (A, B, Q, \Psi, \Phi)$ и требуется получить эквивалентный ему автомат Мура $M^* = (A^*, B^*, Q^*, \Psi^*, \Phi^*)$.

$A^* = A$ и $B^* = B$. Определим Q^*:

Рассмотрим все такие пары вида (q, b), где $q \in Q$, $b \in B$, что для каждой (q, b) имеется такая пара (a, q'), что $\Psi(a, q') = q$ и $\Phi(a, q') = b$ ($a \in A$, $q' \in Q$).

Каждой паре (q, b) поставим в соответствие состояние $q^* \in Q^*$.

Определим функции Ψ^* и Φ^*:

$$\Psi^*(a, q^*) = \Psi^*(a, (q, b)) = (\Psi(a, q), \Phi(a, q));$$

$$\Phi^*(q^*) = \Phi^*((q, b)) = b.$$

Связь между моделями Мили и Мура

Если автомат имеет состояние, в которое он никогда не переходит, то всякому такому состоянию ставится в соответствие состояние автомата Мура, переходы из него определяются аналогично, а выходной символ при нем не определен.

	a_1	a_2	a_3	a_4
q_1	$–,b_1$	q_2,b_1	$–,–$	q_2,b_1
q_2	q_3,b_1	$–,b_2$	$q_3,–$	q_2,b_1
q_3	q_3,b_2	$–,–$	q_2,b_1	q_2,b_1

			a_1	a_2	a_3	a_4	Φ^*
q_1	–	q^*_1	q^*_6	q^*_2	–	q^*_2	–
q_2,b_1	–	q^*_2	q^*_3	q^*_7	q^*_5	q^*_2	b_1
q_3,b_1	–	q^*_3	q^*_4	–	q^*_2	q^*_2	b_1
q_3,b_2	–	q^*_4	q^*_4	–	q^*_2	q^*_2	b_2
$q_3,–$	–	q^*_5	q^*_4	–	q^*_2	q^*_2	–
$–,b_1$	–	q^*_6	–	–	–	–	b_1
$–,b_2$	–	q^*_7	–	–	–	–	b_2

Автомат с абстрактным состоянием. Булев автомат

Автомат с абстрактным состоянием

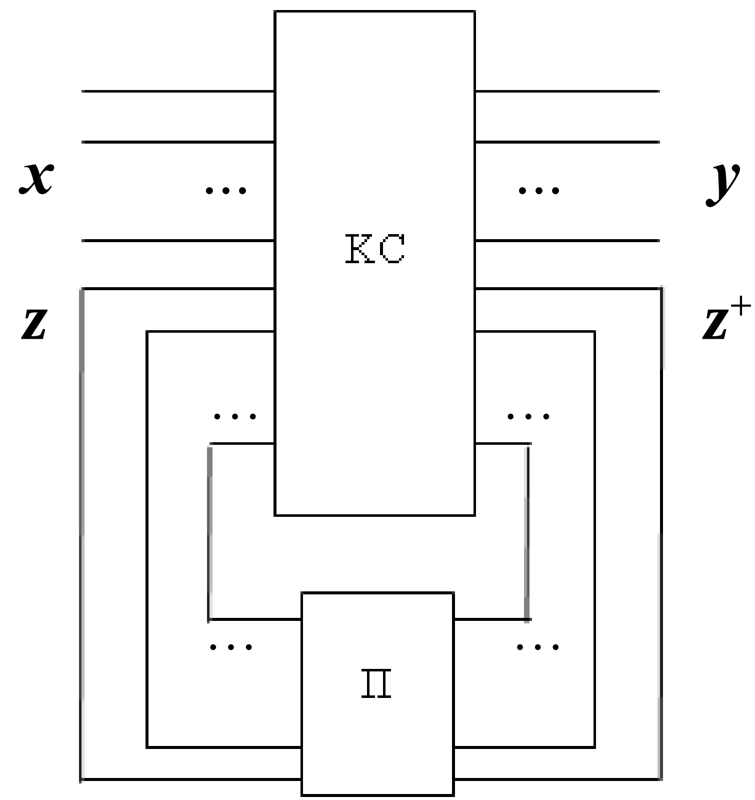
$$q^+ = \psi(x_1, x_2, \ldots, x_n; q),$$
$$y_1 = \varphi_1(x_1, x_2, \ldots, x_n; q),$$
$$y_2 = \varphi_2(x_1, x_2, \ldots, x_n; q),$$
$$\ldots$$
$$y_m = \varphi_m(x_1, x_2, \ldots, x_n; q).$$

В векторной форме:

$$q^+ = \boldsymbol{\psi}(\boldsymbol{x}, q); \qquad \boldsymbol{z}^+ = \boldsymbol{\psi}(\boldsymbol{x}, \boldsymbol{z});$$
$$\boldsymbol{y} = \boldsymbol{\varphi}(\boldsymbol{x}, q). \qquad \boldsymbol{y} = \boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{z}).$$

Булев автомат

$$z_1^{\,+} = \psi_1(x_1, x_2, \ldots, x_n; z_1, z_2, \ldots, z_k),$$
$$z_2^{\,+} = \psi_2(x_1, x_2, \ldots, x_n; z_1, z_2, \ldots, z_k),$$
$$\ldots$$
$$z_k^{\,+} = \psi_k(x_1, x_2, \ldots, x_n; z_1, z_2, \ldots, z_k),$$
$$y_1 = \varphi_1(x_1, x_2, \ldots, x_n; z_1, z_2, \ldots, z_k),$$
$$y_2 = \varphi_2(x_1, x_2, \ldots, x_n; z_1, z_2, \ldots, z_k),$$
$$\ldots$$
$$y_m = \varphi_m(x_1, x_2, \ldots, x_n; z_1, z_2, \ldots, z_k).$$



Минимизация полных автоматов

Состояние q_i и состояние q_j автомата M *эквивалентны*, если автомат M при начальном состоянии q_i и при начальном состоянии q_j под воздействием любой входной последовательности выдает одинаковые последовательности на выходе. Отношение эквивалентности рефлексивно, симметрично и транзитивно.

M_1 и M_2 *эквивалентны*, если для каждого состояния одного из них имеется хотя бы одно эквивалентное ему состояние другого.

Постановка задачи:

Для заданного автомата найти эквивалентный ему автомат, обладающий минимальным числом состояний.

Задан $M = (A, B, Q, \Psi, \Phi)$ и требуется найти $M' = (A, B, Q', \Psi', \Phi')$.

$S_1, S_2, \dots, S_m$ – классы эквивалентности, $S_i \subset Q$ ($i = 1, 2, \dots, m$).

$\{q'_1, q'_2, \dots, q'_m\} = Q'$. Если для $q^{(i)} \in S_i$ и $a \in A$ имеем $\Phi(a, q^{(i)}) = b$, где $b \in B$, то $\Phi'(a, q'_i) = b$.

Если для $q^{(i)} \in S_i$ и $a \in A$ имеем $\Psi(a, q^{(i)}) = q^{(j)}$, где $q^{(j)} \in S_j$, то $\Psi'(a, q'_i) = q'_j$.

Минимизация полных автоматов

Установление эквивалентности состояний

$\exists a \in A$: $\Phi(a, q_i) \neq \Phi(a, q_j)$ – состояния q_i и q_j *явно неэквивалентны*.

$\forall a \in A$: $[\Phi(a, q_i) = \Phi(a, q_j)$ и $\Psi(a, q_i) = \Psi(a, q_j)]$ – состояния q_i и q_j *явно эквивалентны*.

Цепью, порождаемой парой состояний $\langle q_i, q_j \rangle$ полного автомата M, называется множество C, элементами которого являются следующие пары:

1) сама пара $\langle q_i, q_j \rangle$;
2) если $\langle q_k, q_l \rangle \in C$, то все пары вида $\langle \Psi(a, q_k), \Psi(a, q_l) \rangle$, где $\Psi(a, q_k) \neq \Psi(a, q_l)$.

У т в е р ж д е н и е. Состояния q_i и q_j автомата M являются эквивалентными, если и только если в цепи C, порождаемой парой состояний $\langle q_i, q_j \rangle$, нет ни одной пары явно неэквивалентных состояний. В этом случае все пары, принадлежащие C, являются парами эквивалентных состояний.

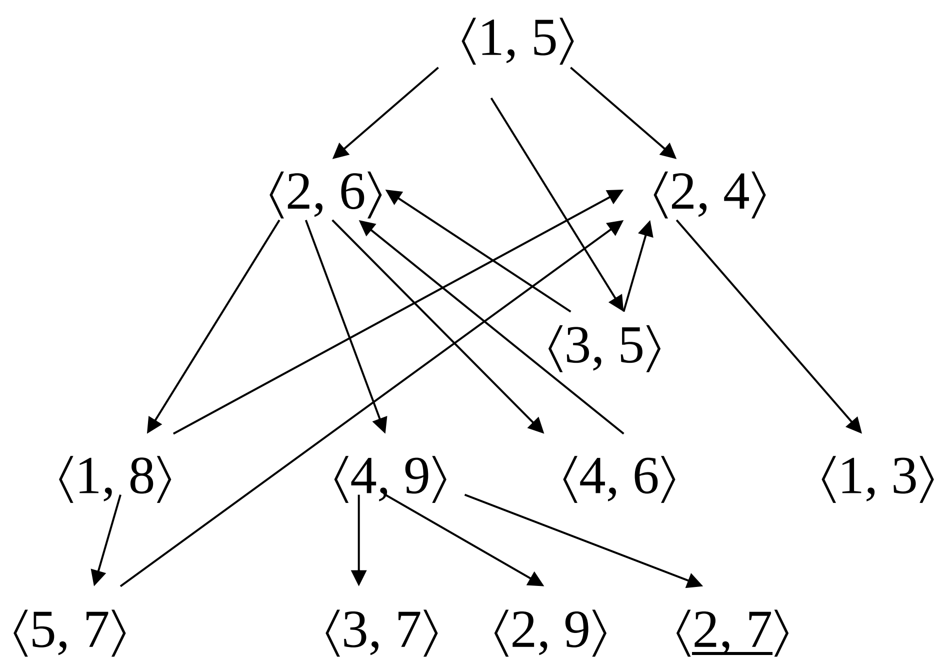
Минимизация полных автоматов

Установление эквивалентности состояний

Таблица переходов и выходов:

	a_1	a_2	a_3
1	2,1	2,0	5,0
2	1,0	4,1	4,1
3	2,1	2,0	5,0
4	3,0	2,1	2,1
5	6,1	4,0	3,0
6	8,0	9,1	6,1
7	6,1	2,0	8,0
8	4,1	4,0	7,0
9	7,0	9,1	7,1

Часть цепи, порождаемой парой $\langle 1, 5\rangle$:



Минимизация полных автоматов

Установление эквивалентности состояний

Таблица переходов и выходов:

	a_1	a_2	a_3
1	2,1	2,0	5,0
2	1,0	4,1	4,1
3	2,1	2,0	5,0
4	3,0	2,1	2,1
5	6,1	4,0	3,0
6	8,0	9,1	6,1
7	6,1	2,0	8,0
8	4,1	4,0	7,0
9	7,0	9,1	7,1

Построение матрицы эквивалентности:

2	3	4	5	6	7	8	9	
0	1	0	0	0			0	1
	0		0	0	0	0		2
		0		0			0	3
			0		0	0	0	4
				0			0	5
					0	0		6
							0	7
							0	8

Минимизация полных автоматов

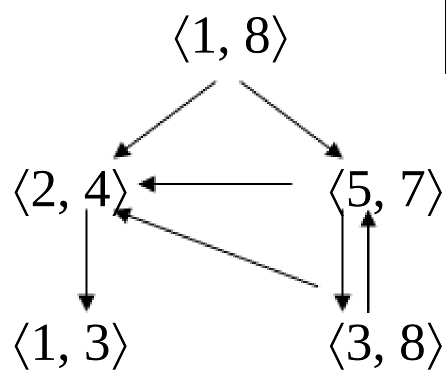
Установление эквивалентности состояний

Таблица переходов и выходов:

	a_1	a_2	a_3
1	2,1	2,0	5,0
2	1,0	4,1	4,1
3	2,1	2,0	5,0
4	3,0	2,1	2,1
5	6,1	4,0	3,0
6	8,0	9,1	6,1
7	6,1	2,0	8,0
8	4,1	4,0	7,0
9	7,0	9,1	7,1

Построение матрицы эквивалентности:

2	3	4	5	6	7	8	9	
0	1	0	0	0	0		0	1
	0		0	0	0	0		2
		0		0			0	3
			0		0	0	0	4
				0			0	5
					0	0		6
							0	7
							0	8



Минимизация полных автоматов

Установление эквивалентности состояний

Таблица переходов и выходов:

	a_1	a_2	a_3
1	2,1	2,0	5,0
2	1,0	4,1	4,1
3	2,1	2,0	5,0
4	3,0	2,1	2,1
5	6,1	4,0	3,0
6	8,0	9,1	6,1
7	6,1	2,0	8,0
8	4,1	4,0	7,0
9	7,0	9,1	7,1

Построение матрицы эквивалентности:

2	3	4	5	6	7	8	9	
0	1	0	0	0	0	1	0	1
	0	1	0	0	0	0	0	2
		0	0	0	0	1	0	3
			0	0	0	0	0	4
				0	1	0	0	5
					0	0	0	6
						0	0	7
							0	8

$\langle 2, 9\rangle$ $\langle 3, 5\rangle$ $\langle 3, 7\rangle$

$\downarrow$ $\downarrow$ $\downarrow$

$\langle 1, 7\rangle$ $\langle 2, 6\rangle$ $\langle 2, 6\rangle$

Классы эквивалентности:

- $\{1, 3, 8\}, \{2, 4\}, \{5, 7\}, \{6\}, \{9\}$

Минимизация полных автоматов

Установление эквивалентности состояний

Таблица переходов и выходов:

	a_1	a_2	a_3
1	2,1	2,0	5,0
2	1,0	4,1	4,1
3	2,1	2,0	5,0
4	3,0	2,1	2,1
5	6,1	4,0	3,0
6	8,0	9,1	6,1
7	6,1	2,0	8,0
8	4,1	4,0	7,0
9	7,0	9,1	7,1

Классы эквивалентности:

{1, 3, 8}, {2, 4}, {5, 7}, {6}, {9}

Состояния нового автомата:

1, 2, 3, 4, 5

	a_1	a_2	a_3
1	2,1	2,0	3,0
2	1,0	2,1	2,1
3	4,1	2,0	1,0
4	1,0	5,1	4,1
5	3,0	5,1	3,1

Минимизация частичных автоматов

Пусть задан частичный автомат $M = (A, B, Q, \Psi, \Phi)$.

A, B и Q – множества множества входов, выходов, состояний.

Ψ и Φ – функции переходов и выходов (определены не на всех a, q).

Входная последовательность $a_{i_1} a_{i_2} \ldots a_{i_p}$ называется *допустимой* для состояния q_{i_1} автомата M, если существует последовательность состояний $q_{i_1} q_{i_2} \ldots q_{i_p}$ такая, что значение $\Phi(a_{i_p}, q_{i_p})$ и значения $\Psi(a_{i_j}, q_{i_j})$ для $1 \le j \le p-1$ определены и $\Psi(a_{i_j}, q_{i_j}) = q_{i_{j+1}}$.

Состояние q_j автомата M_2 *реализует* состояние q_i автомата M_1, если любая входная последовательность, допустимая для q_i, допустима и для q_j, а отвечающие ей выходные последовательности, полученные от автомата M_1 при начальном состоянии q_i и от автомата M_2 при начальном состоянии q_j, совпадают везде, где выходы автомата M_1 определены.

Минимизация частичных автоматов

Автомат M_2 *реализует* автомат M_1, если для каждого состояния q_i автомата M_1 имеется по крайней мере одно состояние q_j автомата M_2, реализующее состояние q_i.

Постановка задачи: для заданного автомата $M = (A, B, Q, \Psi, \Phi)$ найти реализующий его автомат $M' = (A, B, Q', \Psi', \Phi')$ с минимальным числом состояний.

$\{q_r, q_s, \dots, q_t\} \subseteq Q$ – *непосредственно производное множество* от $\{q_i, q_j, \dots, q_k\} \subseteq Q$ *по входному символу* $a \in A$, если значения $\Psi(a, q_i), \Psi(a, q_j), \dots, \Psi(a, q_k)$ составляют $\{q_r, q_s, \dots, q_t\}$.

$Q_i \subseteq Q$ – *непосредственно производное* от $Q_j \subseteq Q$, если найдется такой $a \in A$, что Q_i – непосредственно производное по a от Q_j.

Минимизация частичных автоматов

Группировка – совокупность S подмножеств множества Q, ни одно из которых не содержится в другом. Каждое состояние входит хотя бы в одно из подмножеств.

Пример: $Q = \{q_1, q_2, q_3, q_4\}$, $S = \{\{q_1, q_2\},\{q_1, q_3, q_4\},\{q_2, q_4\}\}$.

Группировка S – *правильная*, если для каждого ее элемента S_i справедливо:

– любое непосредственно производное от него множество является подмножеством какого-то из элементов S;

– для любых q_j, $q_k \in S_i$ и для любого $a \in A$ справедливо $\Phi(a, q_j) = \Phi(a, q_k)$ всегда, когда эти значения оба определены.

*Минимальная п*равильная группировка.

Минимизация частичных автоматов

Задан $M = (A, B, Q, \Psi, \Phi)$ и требуется найти $M' = (A, B, Q', \Psi', \Phi')$.

$S_1, S_2, \ldots, S_m$ – элементы правильной группировки, $S_i \subset Q$ $(i = 1, 2, \ldots, m)$.

$\{q'_1, q'_2, \ldots, q'_m\} = Q'$. Если для $q^{(i)} \in S_i$ и $a \in A$ имеем $\Phi(a, q^{(i)}) = b$, где $b \in B$, то $\Phi'(a, q'_i) = b$. Если для всех $q^{(i)}$ из S_i значение $\Phi(a, q^{(i)})$ не определено, то значение $\Phi'(a, q'_i)$ не определено.

$\Psi(a, S_i)$ – непосредственно производное от $S_i \in S$ по a (если значение $\Psi(a, q^{(i)})$ не определено для всех $q^{(i)} \in S_i$, то $\Psi(a, S_i) = \varnothing$).

$\Psi'(a, q'_i) = q'_j$, где q'_j соответствует любому $S_j \in S$, для которого $\Psi(a, S_i) \subseteq S_j$. Если значение $\Psi(a, q^{(i)})$ не определено для всех $q^{(i)} \in S_i$, то $\Psi'(a, q'_i)$ не определено.

Минимизация частичных автоматов

Минимизация частичного автомата не сводится к минимизации полного автомата.

Варианты доопределения:

	a_1	a_2
1	1,–	2,0
2	3,0	1,0
3	2,1	1,0

	a_1	a_2
1	1,0	2,0
2	3,0	1,0
3	2,1	1,0

	a_1	a_2
1	1,1	2,0
2	3,0	1,0
3	2,1	1,0

Правильная группировка $S = \{\{1, 2\}, \{1, 3\}\}$.

Состояния нового автомата: 1, 2.

	a_1	a_2
1	1,–	2,0
2	3,0	1,0
3	2,1	1,0

	a_1	a_2
1	2,0	1,0
2	1,1	1,0

Минимизация частичных автоматов

Совместимость состояний

Состояния q_i и q_j автомата M *несовместимы*, если существует такая входная последовательность, допустимая для q_i и q_j, что заключительные выходные символы, вызываемые этой последовательностью при начальных состояниях q_i и q_j, не совпадают.

Состояния q_i и q_j автомата M *совместимы*, если они не являются несовместимыми.

$\exists a \in A$: значения $\Phi(a, q_i)$ и $\Phi(a, q_j)$ определены и $\Phi(a, q_i) \neq \Phi(a, q_j)$ – состояния q_i и q_j *явно несовместимы*.

$\forall a \in A$: $[\Phi(a, q_i) = \Phi(a, q_j)$ и $\Psi(a, q_i) = \Psi(a, q_j)]$ или хотя бы одно из этих значений не определено – состояния q_i и q_j *явно совместимы*.

Минимизация частичных автоматов

Совместимость состояний

Цепью, порождаемой парой состояний $\langle q_i, q_j \rangle$ частичного автомата M, назовем множество C, элементами которого являются следующие пары:

1) сама пара $\langle q_i, q_j \rangle$;

2) все пары вида $\langle \Psi(a, q_k), \Psi(a, q_l) \rangle$, где $\Psi(a, q_k)$ и $\Psi(a, q_l)$ определены и различны, если $\langle q_k, q_l \rangle \in C$.

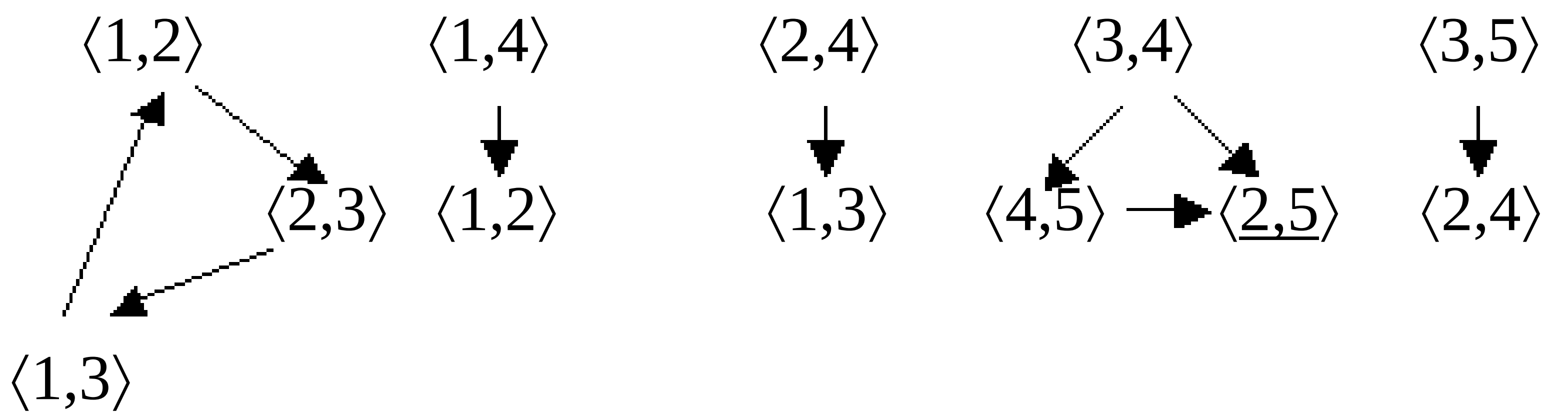
У т в е р ж д е н и е. Состояния q_i и q_j автомата M являются совместимыми, если и только если в цепи, порождаемой парой состояний $\langle q_i, q_j \rangle$, нет ни одной пары явно несовместимых состояний. В этом случае все пары, принадлежащие данной цепи, являются парами совместимых состояний.

Минимизация частичных автоматов

Совместимость состояний

	a_1	a_2	a_3
1	2,1	–,–	–,–
2	3,–	–,1	–,1
3	1,–	4,–	2,–
4	1,–	5,–	5,–
5	–,0	2,0	–,1

$$\begin{array}{cc} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\ \begin{bmatrix} & & & 0 \\ & & & 0 \\ & & & \\ & & & \end{bmatrix} & \begin{array}{c} 1 \\ 2, \\ 3 \\ 4 \end{array} \end{array} \quad \cdots \quad \begin{array}{cc} \begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\ \begin{bmatrix} 1 & 1 & 1 & 0 \\ & 1 & 1 & 0 \\ & & 0 & 1 \\ & & & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}$$



Минимизация частичных автоматов

Максимальные совместимые множества

	a_1	a_2	a_3
1	2,1	–,–	–,–
2	3,–	–,1	–,1
3	1,–	4,–	2,–
4	1,–	5,–	5,–
5	–,0	2,0	–,1

$$
\begin{array}{cc}
\begin{array}{cccc} 2 & 3 & 4 & 5 \end{array} & \\
\left[\begin{array}{cccc} 1 & 1 & 1 & 0 \\ & 1 & 1 & 0 \\ & & 0 & 1 \\ & & & 0 \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}
\end{array}
$$

Шаги:	1	2	3	4	5
	1	1,2	1,2,3	1,2,3	1,2,3
				1,2,4	1,2,4
					3,5

$S' = [S_j^i \setminus I(q_{i+1})] \cup \{q_{i+1}\}$

S_j^i - совместимое множество из полученных на *i*-м шаге;

$I(q_{i+1})$ – множество состояний, несовместимых с q_{i+1}.

Всего таких множеств может быть: $2 \cdot 3^{k-1}$, если $\gamma = 3k - 1$;

$3 \cdot 3^{k-1}$, если $\gamma = 3k$;

$4 \cdot 3^{k-1}$, если $\gamma = 3k + 1$.

γ – число состояний автомата. $m = 59\ 049$ при $\gamma = 30$.

Минимизация частичных автоматов

Нахождение минимальной правильной группировки

	a_1	a_2	a_3
1	2,1	–,–	–,–
2	3,–	–,1	–,1
3	1,–	4,–	2,–
4	1,–	5,–	5,–
5	–,0	2,0	–,1

Непосредственно производное от совместимого множества есть совместимое множество.

Совокупность всех максимальных совместимых множеств есть правильная группировка.

{1,2,3}, {1,2,4}, {3,5} – максимальные совместимые множества.

Минимальное число состояний γ' автомата, реализующего заданный автомат M с числом состояний γ, находится в границах:

$$\alpha(G) \le \gamma' \le \min(\gamma, m)$$

$\alpha(G)$ – число независимости графа G совместимости состояний;

m – число всех максимальных совместимых множеств.

Для данного примера $2 \le \gamma' \le 3$.

Минимизация частичных автоматов

Нахождение минимальной правильной группировки

	a_1	a_2	a_3
1	2,1	–,–	–,–
2	3,–	–,1	–,1
3	1,–	4,–	2,–
4	1,–	5,–	5,–
5	–,0	2,0	–,1

Для каждого совместимого множества S_i строится $T_i = \{t_{i_1}, t_{i_2}, \ldots, t_{i_k}\}$ со свойствами:

1) t_{i_j} – непосредственно производное от S_i;

2) t_{i_j} не содержится ни в S_i, ни в другом $t_{i_l} \in T_i$;

3) $|t_{i_j}| > 1$ для всех $j = 1, 2, \ldots, k$.

Если $S_g \subset S_h$, а $T_g \supseteq T_h$, то S_g можно исключить из рассмотрения.

$$\{1,2,3\} \to \varnothing$$

$$\{1,2,4\} \to \{\{1,2,3\}\}$$

$$\{3,5\} \to \{\{2,4\}\}$$

$$\{1,4\} \to \{\{1,2\}\}$$

$$\{2,4\} \to \{\{1,3\}\}$$

$$\{4\} \to \varnothing$$

$$\{5\} \to \varnothing$$

Минимизация частичных автоматов

Нахождение минимальной правильной группировки

	a_1	a_2	a_3
1	2,1	–,–	–,–
2	3,–	–,1	–,1
3	1,–	4,–	2,–
4	1,–	5,–	5,–
5	–,0	2,0	–,1

Обобщенная таблица покрытия

Совместимые множества	A					B			
	1	2	3	4	5	{1, 2, 3}	{2, 4}	{1, 2}	{1, 3}
{1, 2, 3}	×	×	×			×		×	×
{1, 2, 4}	×	×		×		•	×	×	
{3, 5}			×		×		•		
{1, 4}	×			×				•	
{2, 4}		×		×			×		•
{4}				×					
{5}					×				

Задача: найти минимальную совокупность строк со следующими свойствами.

1. Каждый столбец части A имеет знак × хотя бы в одной из них.
2. Если какая-то строка из данной совокупности имеет знак • в части B, то столбец, содержащий этот знак, имеет знак × хотя бы в одной из строк данной совокупности.

Минимизация частичных автоматов

Нахождение минимальной правильной группировки

Совместимые множества	A		B
	4	5	{2, 4}
{1, 2, 4}	×		×
{3, 5}		×	•
{1, 4}	×		
{2, 4}	×		×
{4}	×		
{5}		×	

⇒

Совместимые множества	A		B
	4	5	{2, 4}
{4}	×		
{5}		×	

Решение: {1,2,3}, {4}, {5}.

Правила редукции:

1. Если i-я строка имеет × везде, где × имеет j-я строка, а j-я строка имеет • везде, где • имеет i-я строка, то j-я строка удаляется.
2. Если i-й столбец имеет × везде, где имеет × j-й столбец из части А, то i-й столбец удаляется.
3. Если из какого-то столбца части В при включении строки в решение исчез хотя бы один •, то этот столбец переводится в часть А и из него удаляются все знаки •.
4. Если в результате удаления строк в некотором столбце части В остались только знаки •, то строки, содержащие эти знаки, удаляются.

Минимизация частичных автоматов

Нахождение минимальной правильной группировки

Совместимые множества	A					B			
	1	2	3	4	5	{1, 2, 3}	{2, 4}	{1, 2}	{1, 3}
{1, 2, 3}	×	×	×			×		×	×
{1, 2, 4}	×	×		×		●	×	×	
{3, 5}			×		×		●		
{1, 4}	×			×				●	
{2, 4}		×		×			×		●
{4}				×					
{5}					×				

Если выбрать {3, 5} для покрытия состояния 3, то

Совместимые множества	A				B		
	1	2	4	{2, 4}	{1, 2, 3}	{1, 2}	{1, 3}
{1, 2, 3}	×	×			×	×	×
{1, 2, 4}	×	×	×	×	●	×	
{1, 4}	×		×			●	
{2, 4}		×	×				●
{4}			×				

Минимизация частичных автоматов

Построение минимального автомата

Минимальная правильная группировка: {1,2,3}, {4}, {5}.

Состояния нового автомата: 1 2 3

	a_1	a_2	a_3
1	2,1	–,–	–,–
2	3,–	–,1	–,1
3	1,–	4,–	2,–
4	1,–	5,–	5,–
5	–,0	2,0	–,1

	a_1	a_2	a_3
1	1,1	2,1	1,1
2	1,–	3,–	3,–
3	–,0	1,0	–,1

Кодирование состояний синхронного автомата

Многозначные переменные a, b и q заменяем на векторные:

$$a \rightarrow \boldsymbol{x} = (x_1, x_2, \dots, x_n),$$
$$b \rightarrow \boldsymbol{y} = (y_1, y_2, \dots, y_m),$$
$$q \rightarrow \mathbf{z} = (z_1, z_2, \dots, z_k),$$

чтобы получить систему булевых функций

$$b = \Phi(a, q) \rightarrow y_i = \varphi_i(x_1, x_2, \dots, x_n, z_1, z_2, \dots, z_k), i = 1, 2, \dots, m;$$
$$q^+ = \Psi(a, q) \rightarrow z_j^+ = \psi_i(x_1, x_2, \dots, x_n, z_1, z_2, \dots, z_k), j = 1, 2, \dots, k.$$

Ограничения: $\alpha \leq 2^n$, $\beta \leq 2^m$ и $\gamma \leq 2^k$,

где α, β и γ – числа абстрактных входных и выходных символов и состояний.

$n = \lceil \log_2 \alpha \rceil$, $m = \lceil \log_2 \beta \rceil$ и $k = \lceil \log_2 \gamma \rceil$,

где $\lceil a \rceil$ означает минимальное целое число, не меньшее a.

Кодирование состояний синхронного автомата

Неравнозначность вариантов кодирования.

	0	1
1	1, 0	2, 0
2	1, 0	3, 1
3	4, 1	1, 0
4	1, 1	1, 1

	Вариант 1	Вариант 2
1	1 1	0 0
2	0 0	1 0
3	1 0	1 1
4	0 1	0 1

$$\boldsymbol{U}_1 = \begin{array}{c} \begin{array}{ccc} x & z_1 & z_2 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & - \\ 1 & 0 & - \\ 0 & - & 1 \\ - & 0 & 1 \end{bmatrix} \end{array}, \quad \boldsymbol{V}_1 = \begin{array}{c} \begin{array}{ccc} z_1^+ & z_2^+ & y \end{array} \\ \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \end{array}; \qquad \boldsymbol{U}_2 = \begin{array}{c} \begin{array}{ccc} x & z_1 & z_2 \end{array} \\ \begin{bmatrix} 1 & - & 0 \\ - & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array}, \quad \boldsymbol{V}_2 = \begin{array}{c} \begin{array}{ccc} z_1^+ & z_2^+ & y \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{array}.$$

Число различных вариантов кодирования: $\dfrac{(2^k - 1)!}{(2^k - \gamma)!k!}$,

где γ – число состояний,

k – число кодирующих переменных.

Кодирование состояний синхронного автомата

Число различных вариантов кодирования.

γ – число состояний, k – число кодирующих переменных.

$$\begin{array}{l}
\begin{array}{cccc} & z_1 & z_2 & z_3 \end{array} \\
\left.\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 \\ 3 & 0 & 1 & 0 \\ 4 & 0 & 1 & 1 \\ 5 & 1 & 0 & 0 \end{array}\right\} \gamma \\
\left.\begin{array}{cccc} 6 & 1 & 0 & 1 \\ 7 & 1 & 1 & 0 \\ 8 & 1 & 1 & 1 \end{array}\right\} 2^k - \gamma
\end{array}$$

$\dfrac{2^k!}{(2^k-\gamma)!}$ – число размещений 2^k элементов по γ позициям.

$\dfrac{2^k!}{(2^k-\gamma)!k!}$ – перестановка столбцов дает равнозначные варианты.

$\dfrac{2^k!}{(2^k-\gamma)!k!2^k}$ – инвертирование значений внутренних переменных дает равнозначный вариант.

$\dfrac{(2^k-1)!}{(2^k-\gamma)!k!} = 140$ при $\gamma = 5$, $k = 3$ и 75 675 600 при $\gamma = 10$, $k = 4$.

Кодирование состояний синхронного автомата

Метод «желательных соседств»

Каждой паре q_i, q_j приписывается вес $w_{ij} = w'_{ij} + w''_{ij}$,
где w''_{ij} – число значений a, при которых $\Psi(a, q_i) = \Psi(a, q_j)$.

Пусть w_p – число пар вида $\langle\Psi(a_s, q_p), \Psi(a_t, q_p)\rangle$, где a_s и a_t имеют соседние коды, $\Psi(a_s, q_p) = q_i$ и $\Psi(a_t, q_p) = q_j$.

Тогда $w'_{ij} = \sum_{p=1}^{\gamma} w_p$.

Желательно, чтобы коды q_i и q_j были тем ближе друг к другу, чем больше w_{ij}.

Кодирование состояний синхронного автомата

Метод «желательных соседств»

w_p – число пар вида $\langle\Psi(a_s, q_p), \Psi(a_t, q_p)\rangle$, где a_s и a_t имеют соседние коды, $\Psi(a_s, q_p) = q_i$ и $\Psi(a_t, q_p) = q_j$.

Тогда $w'_{ij} = \sum_{p=1}^{\gamma} w_p$.

Ситуация, которая учитывается при подсчете значения w'_{ij}:

$$\begin{array}{c} \begin{array}{ccccccc} x_1 & x_2 & \dots & x_n & z_1 & z_2 & \dots & z_k \end{array} \\ \begin{bmatrix} & \dots & & & & \dots & & \\ x_{s1} & x_{s2} & \dots & x_{sn} & z_{p1} & z_{p2} & \dots & z_{pk} \\ & \dots & & & & \dots & & \\ x_{t1} & x_{t2} & \dots & x_{tn} & z_{p1} & z_{p2} & \dots & z_{pk} \\ & \dots & & & & \dots & & \end{bmatrix} \end{array} \qquad \begin{array}{c} \begin{array}{cccc} z_1^+ & z_2^+ & \dots & z_k^+ \end{array} \\ \begin{bmatrix} & \dots & & \\ z_{i1} & z_{i2} & \dots & z_{ik} \\ & \dots & & \\ z_{j1} & z_{j2} & \dots & z_{jk} \\ & \dots & & \end{bmatrix} \end{array}$$

Кодирование состояний синхронного автомата

Метод «желательных соседств»

w''_{ij} – число значений a, при которых $\Psi(a, q_i) = \Psi(a, q_j)$.

Ситуация, которая учитывается при подсчете значения w''_{ij}:

$$
\begin{array}{cccccccc}
x_1 & x_2 & \ldots & x_n & z_1 & z_2 & \ldots & z_k
\end{array}
\qquad
\begin{array}{cccc}
z_1^+ & z_2^+ & \ldots & z_k^+
\end{array}
$$

$$
\begin{bmatrix}
 & \ldots & & & & \ldots & & \\
x_{u1} & x_{u2} & \ldots & x_{un} & z_{i1} & z_{i2} & \ldots & z_{ik} \\
 & \ldots & & & & \ldots & & \\
x_{u1} & x_{u2} & \ldots & x_{un} & z_{j1} & z_{j2} & \ldots & z_{jk} \\
 & \ldots & & & & \ldots & &
\end{bmatrix}
\qquad
\begin{bmatrix}
 & \ldots & & \\
z_{v1} & z_{v2} & \ldots & z_{vk} \\
 & \ldots & & \\
z_{v1} & z_{v2} & \ldots & z_{vk} \\
 & \ldots & &
\end{bmatrix}
$$

Многошаговый переход от 0-мерных гиперкубов к k-мерному.

На l-м шаге (при переходе от $(l - 1)$-мерных гиперкубов к l-мерным, $1 \le l \le k$) вводимые ребра выбираются таким образом, чтобы сумма величин w_{ij} на парах, соответствующих связываемым вершинам, была максимальной.

Кодирование состояний синхронного автомата

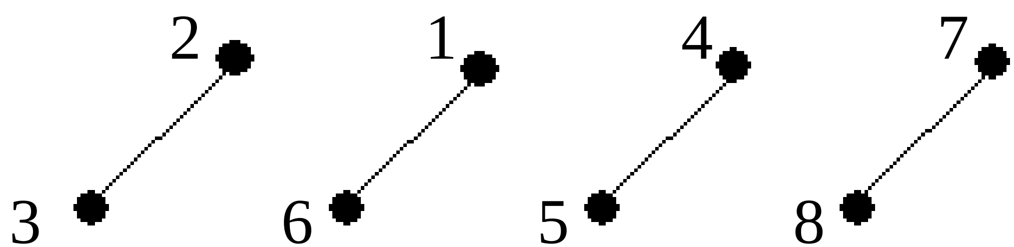
Метод «желательных соседств»

Таблица переходов

q \ $x_1 x_2$	00	01	10
1	1	2	6
2	3	2	1
3	2	3	5
4	4	5	2
5	3	5	4
6	3	2	4

Значения w_{ij}

2	3	4	5	6	7	8	
2	1	0	0	2	0	0	1
	3	1	2	2	0	0	2
		2	1	0	0	0	3
			2	0	0	0	4
				2	0	0	5
					0	0	6
						0	7

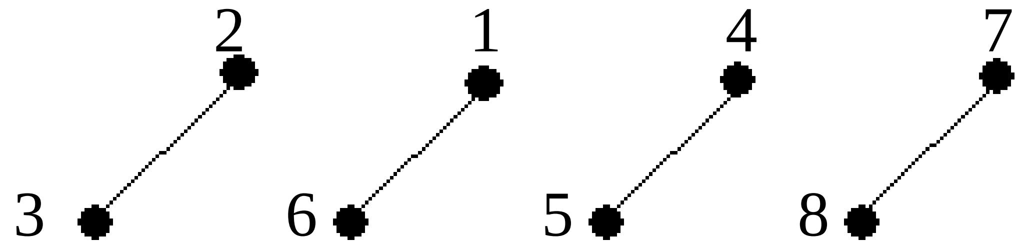


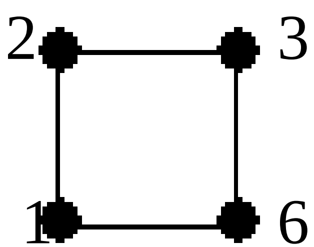
Кодирование состояний синхронного автомата

Метод «желательных соседств»

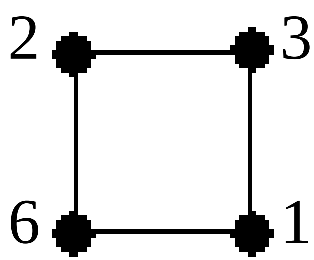
Значения w_{ij}

2	3	4	5	6	7	8	
2	1	0	0	2	0	0	1
	3	1	2	2	0	0	2
		2	1	0	0	0	3
			2	0	0	0	4
				2	0	0	5
					0	0	6
						0	7

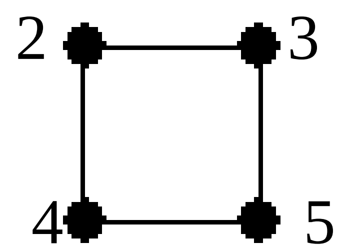




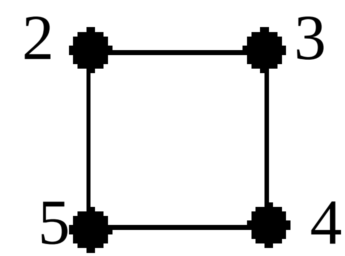
$w_{12} + w_{36} = 2$



$w_{26} + w_{13} = 3$



$w_{24} + w_{35} = 2$



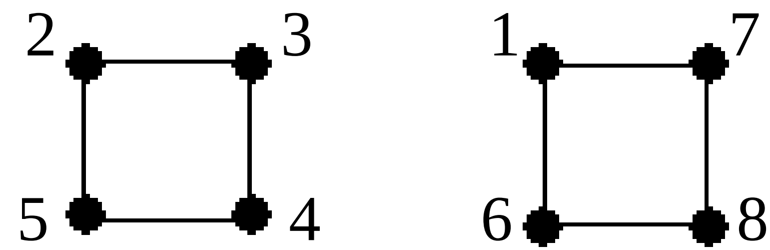
$w_{25} + w_{34} = 4$

Кодирование состояний синхронного автомата

Метод «желательных соседств»

Значения w_{ij}

2	3	4	5	6	7	8	
2	1	0	0	2	0	0	1
	3	1	2	2	0	0	2
		2	1	0	0	0	3
			2	0	0	0	4
				2	0	0	5
					0	0	6
						0	7



1 100

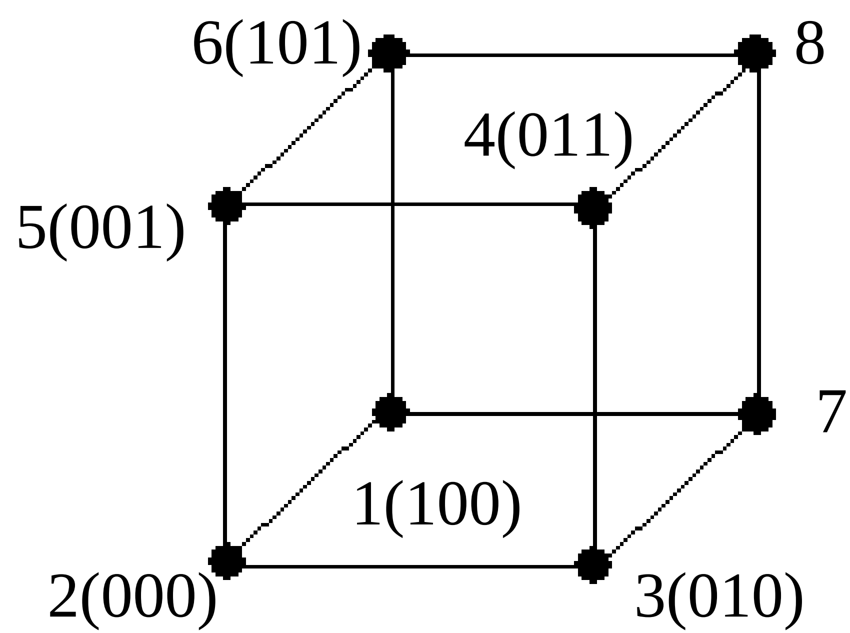
2 000

3 010

4 011

5 001

6 101



Кодирование состояний синхронного автомата

Построение функций возбуждения

q \ $x_1 x_2$	00	01	10	$z_1 z_2 z_3$
1	1	2	6	1 0 0
2	3	2	1	0 0 0
3	2	3	5	0 1 0
4	4	5	2	0 1 1
5	3	5	4	0 0 1
6	3	2	4	1 0 1

$$
\begin{array}{ccccc}
x_1 & x_2 & z_1 & z_2 & z_3
\end{array}
\qquad
\begin{array}{ccc}
z_1^+ & z_2^+ & z_3^+
\end{array}
$$

$$
\begin{bmatrix}
1 & - & - & 0 & 0 \\
- & 0 & 1 & - & 0 \\
0 & 0 & 0 & 0 & - \\
0 & 0 & - & - & 1 \\
1 & - & - & 0 & 1 \\
1 & - & - & 1 & 0 \\
- & 1 & - & 1 & 0 \\
0 & - & - & 1 & 1 \\
- & 1 & 0 & - & 1 \\
1 & - & 1 & - & -
\end{bmatrix}
\qquad
\begin{bmatrix}
1 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 1 & 1 \\
0 & 0 & 1 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 1 \\
0 & 0 & 1
\end{bmatrix}
$$

x_2 x_1

2	0	1		0
5	0	0		0
4	0	0		0
3	0	0		0
6	0	0		0
1	1	1		0

$z_1 z_2 z_3$

z_1^+

x_2 x_1

2	1	0		0
5	1	1		0
4	1	0		0
3	0	0		1
6	1	1		0
1	0	0		0

$z_1 z_2 z_3$

z_2^+

x_2 x_1

2	0	0		0
5	0	1		1
4	1	0		1
3	0	1		0
6	0	1		0
1	0	1		0

$z_1 z_2 z_3$

z_3^+

Кодирование состояний синхронного автомата

Построение функций возбуждения

$$\begin{array}{ccccc} x_1 & x_2 & z_1 & z_2 & z_3 \end{array} \qquad \begin{array}{ccc} z_1^+ & z_2^+ & z_3^+ \end{array}$$

$$\begin{bmatrix} 1 & - & - & 0 & 0 \\ - & 0 & 1 & - & 0 \\ 0 & 0 & 0 & 0 & - \\ 0 & 0 & - & - & 1 \\ 1 & - & - & 0 & 1 \\ 1 & - & - & 1 & 0 \\ - & 1 & - & 1 & 0 \\ 0 & - & - & 1 & 1 \\ - & 1 & 0 & - & 1 \\ 1 & - & 1 & - & - \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{array}{ccccc} x_1 & x_2 & z_1 & z_2 & z_3 \end{array} \qquad \begin{array}{ccc} z_1^+ & z_2^+ & z_3^+ \end{array}$$

$$\begin{bmatrix} 0 & 0 & 0 & - & 1 \\ 0 & 0 & - & 1 & - \\ - & 1 & - & 0 & 1 \\ - & 1 & - & 1 & 1 \\ - & 1 & 1 & - & 0 \\ 1 & - & 0 & - & 0 \\ - & 0 & 1 & - & - \\ - & 1 & - & 1 & 0 \\ 1 & - & - & 0 & 0 \\ 1 & - & - & 1 & 1 \\ 1 & - & 1 & - & - \\ - & 1 & 0 & - & - \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

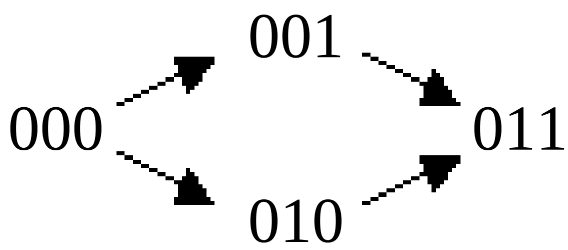
Справа функции, полученные,
если коды состояний – их номера
в двоичной системе минус единица, т.е. 1 – 000, 2 – 001, 3 – 010 …

Кодирование состояний асинхронного автомата

Ограничение: если $\Psi(a, q_i) = q_j$, то $\Psi(a, q_j) = q_j$.

Явление *состязаний* (*гонок*) элементов памяти.

	a_1	a_2	a_3	$z_1 z_2 z_3$
1	**1**	3	4	0 0 0
2	1	3	**2**	0 0 1
3	1	**3**	–	0 1 0
4	5	–	**4**	0 1 1
5	**5**	3	2	1 0 0



При изменении входного сигнала с a_1 на a_3 автомат из состояния 1 может перейти не в состояние 4, как определено таблицей, а в состояние 2.

Состязания *опасные* и *неопасные*.

Кодирование состояний, обеспечивающее отсутствие опасных состязаний, называется *противогоночным*.

Кодирование состояний асинхронного автомата

Условие отсутствия опасных состязаний

$q_i \to q_j$, $q_k \to q_l$ – *пара переходов* совершаемых при одном и том же входном сигнале ($q_j \neq q_l$).

При одновременном возбуждении элементов памяти в процессе перехода опасные состязания отсутствуют тогда и только тогда, когда для каждой пары переходов $q_i \to q_j$, $q_k \to q_l$ имеет место $U(q_i, q_j) \cap U(q_k, q_l) = \varnothing$,

где $U(q_i, q_j)$ – множество всех возможных промежуточных состояний (включая исходное и конечное), в которые автомат может попасть при реализации перехода $q_i \to q_j$.

Это эквивалентно ортогональности троичных векторов $t(q_i, q_j)$ и $t(q_k, q_l)$, представляющих соответствующие интервалы булева пространства внутренних переменных.

Если коды q_i и q_j – (0 0 0 1) и (0 1 0 1), то $t(q_i, q_j) = (0 – 0\ 1)$.

Кодирование состояний асинхронного автомата

	a_1	a_2	a_3
1	**1**	3	4
2	1	3	**2**
3	1	**3**	–
4	5	–	**4**
5	**5**	3	2

Пары переходов:

для a_1

$1 \to 1,\ 4 \to 5;$

$1 \to 1,\ 5 \to 5;$

$2 \to 1,\ 4 \to 5;$

$2 \to 1,\ 5 \to 5;$

$3 \to 1,\ 4 \to 5;$

$3 \to 1,\ 5 \to 5.$

для a_3

$1 \to 4,\ 2 \to 2;$

$1 \to 4,\ 5 \to 2;$

$2 \to 2,\ 4 \to 4;$

$4 \to 4,\ 5 \to 2.$

Для каждой пары переходов $q_i \to q_j$, $q_k \to q_l$ должна быть внутренняя переменная z_p, со значением 0 на q_i и q_j и 1 – на q_k и q_l (или наоборот).

Коды состояний должны быть ортогональны.

Кодирование состояний асинхронного автомата

	a_1	a_2	a_3
1	**1**	3	4
2	1	3	**2**
3	1	**3**	–
4	5	–	**4**
5	**5**	3	2

Достаточно рассмотреть

для a_1

$2 \to 1,\ 4 \to 5;$

$3 \to 1,\ 4 \to 5.$

для a_3

$1 \to 4,\ 5 \to 2;$

	1 2 3 4 5	
$2 \to 1,\ 4 \to 5$	0 0 1 1	z_1

	1 2 3 4 5	
$2 \to 1,\ 4 \to 5$	0 0 0 1 1	z_1
$3 \to 1,\ 4 \to 5$		

	1 2 3 4 5	
$2 \to 1,\ 4 \to 5$	0 0 0 1 1	z_1
$3 \to 1,\ 4 \to 5$	0 1 0 1	z_2
$1 \to 4,\ 5 \to 2$		

Для ортогональности:

1 2 3 4 5	
0 0 0 1 1	z_1
0 1 1 0 1	z_2
- 0 1 - -	z_3

Кодирование состояний асинхронного автомата

Получение функций возбуждения элементов памяти:

	a_1	a_2	a_3	$z_1\ z_2\ z_3$
1	**1**	3	4	0 0 –
2	1	3	**2**	0 1 0
3	1	**3**	–	0 1 1
4	5	–	**4**	1 0 –
5	**5**	3	2	1 1 –

Функции определяются на интервалах, соответствующих переходам.

Коды входных сигналов: $a_1 - 0\ 0$, $a_2 - 0\ 1$, $a_3 - 1\ 0$.

$$
\begin{array}{ccccc}
x_1 & x_2 & z_1 & z_2 & z_3
\end{array}
\qquad
\begin{array}{ccc}
z_1^+ & z_2^+ & z_3^+
\end{array}
$$

$$
\begin{bmatrix}
0 & 0 & 0 & - & - \\
0 & 0 & 1 & - & - \\
0 & 1 & - & - & - \\
1 & 0 & - & 0 & - \\
1 & 0 & - & 1 & -
\end{bmatrix}
\qquad
\begin{bmatrix}
0 & 0 & - \\
1 & 1 & - \\
0 & 1 & 1 \\
1 & 0 & - \\
0 & 1 & 0
\end{bmatrix}
$$

Кодирование состояний асинхронного автомата

Минимизация длины кода состояния

Условия отсутствия опасных состязаний можно выразить троичной матрицей, в которой столбцы соответствуют состояниям автомата, а строки – парам переходов.

	1	2	3	4	5
$2 \to 1, 4 \to 5$	0	0	–	1	1

Троичный вектор ***a*** *имплицирует* троичный вектор ***b***, если ***b*** получается из ***a*** заменой некоторых нулей или единиц значением «–» и, возможно, инвертированием полученного результата.

a = (1 0 – – 1 0 1) имплицирует ***b*** = (1 0 – – – 0 1) и
c = (0 1 – – – 1 –).

Данное отношение обладает свойствами рефлексивности и транзитивности.

Кодирование состояний асинхронного автомата

Минимизация длины кода состояния

Матрица условий представляет условия отсутствия опасных состязаний. В ней отсутствуют строки, имплицируемые другими строками.

Парам переходов $1 \to 1$, $4 \to 5$; $1 \to 1$, $5 \to 5$ и $2 \to 1$, $4 \to 5$ соответствуют следующие строки:

1	2	3	4	5
0	–	–	1	1
0	–	–	–	1
0	0	–	1	1.

Только последняя строка останется в матрице условий.

Кодирование состояний асинхронного автомата

Минимизация длины кода состояния

Троичная матрица $\boldsymbol{R}$ *имплицирует* троичную матрицу $\boldsymbol{S}$, если для каждой строки матрицы $\boldsymbol{S}$ в матрице $\boldsymbol{R}$ найдется имплицирующая ее строка.

Задача: найти *кратчайшую имплицирующую форму* матрицы условий, т. е. матрицу с минимальным числом строк, имплицирующую матрицу условий. Столбцы этой матрицы будут представлять искомые коды состояний.

Совместимое множество – множество строк матрицы условий, для которых имеется общий имплицирующий вектор.

Множество строк, в котором каждая пара строк совместима, не всегда является совместимым множеством.

Векторы $\boldsymbol{u}$ = (0 1 – – 0 1), $\boldsymbol{v}$ = (– 1 0 – 0 –) и $\boldsymbol{w}$ = (– –1 – – 1) не имеют общего имплицирующего вектора.

$\boldsymbol{u}$, $\boldsymbol{v}$ - (0 1 0 – 0 1); $\boldsymbol{u}$, $\boldsymbol{w}$ - (0 1 1 – 0 1); $\boldsymbol{v}$, $\boldsymbol{w}$ - (– 1 0 – 0 0).

Кодирование состояний асинхронного автомата

Минимизация длины кода состояния

Чтобы получить кратчайшую имплицирующую форму для троичной матрицы, надо найти все *максимальные* совместимые множества ее строк, а затем получить кратчайшее покрытие строк этими множествами.

	a_1	a_2	a_3	a_4
1	**1**	**1**	**1**	5
2	1	**2**	**2**	**2**
3	**3**	2	4	5
4	3	2	**4**	5
5	–	**5**	4	**5**

	a_1	a_2	a_3	a_4	a_5
1	**1**	**1**	**1**	5	**1**
2	1	**2**	**2**	**2**	**2**
3	**3**	2	4	5	**3**
4	3	2	**4**	5	**4**
5	–	**5**	4	**5**	**5**

Матрица условий:

$$\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 1 & - \\ 0 & 1 & 1 & - & - \\ 0 & 1 & - & 1 & - \\ - & 0 & 0 & - & 1 \\ - & 0 & - & 0 & 1 \\ 0 & - & - & 1 & 1 \\ - & 0 & - & 1 & 1 \\ 0 & 1 & - & - & 0 \\ - & 0 & 1 & - & 1 \\ - & - & 0 & 1 & - \end{bmatrix} \end{array} \begin{array}{l} \\ \left.\begin{array}{r} 1 \end{array}\right\} a_1 \\ \left.\begin{array}{r} 2 \\ 3 \\ 4 \\ 5 \end{array}\right\} a_2 \\ \left.\begin{array}{r} 6 \\ 7 \end{array}\right\} a_3 \\ \left.\begin{array}{r} 8 \\ 9 \end{array}\right\} a_4 \\ \left.\begin{array}{r} 10 \end{array}\right\} a_5 \end{array}$$

Кодирование состояний асинхронного автомата

Минимизация длины кода состояния

Матрица условий

$$
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{bmatrix}
0 & 0 & 1 & 1 & - \\
0 & 1 & 1 & - & - \\
0 & 1 & - & 1 & - \\
- & 0 & 0 & - & 1 \\
- & 0 & - & 0 & 1 \\
0 & - & - & 1 & 1 \\
- & 0 & - & 1 & 1 \\
0 & 1 & - & - & 0 \\
- & 0 & 1 & - & 1 \\
- & - & 0 & 1 & -
\end{bmatrix}
\end{array}
\begin{array}{l}
\\
\left.1\right\} \; a_1 \\
\left.\begin{array}{l} 2 \\ 3 \\ 4 \\ 5 \end{array}\right\} \; a_2 \\
\left.\begin{array}{l} 6 \\ 7 \end{array}\right\} \; a_3 \\
\left.\begin{array}{l} 8 \\ 9 \end{array}\right\} \; a_4 \\
\left.10\right\} \; a_5
\end{array}
$$

Максимальные совместимые множества

$\{1, 6, 7, 9\}$ $(0\ 0\ 1\ 1\ 1)$;

$\{2, 3, 4, 5, 8\}$ $(0\ 1\ 1\ 1\ 0)$;

$\{2, 3, 6\}$ $(0\ 1\ 1\ 1\ 1)$;

$\{2, 4, 7, 8, 10\}$ $(0\ 1\ 1\ 0\ 0)$;

$\{3, 5, 8, 9, 10\}$ $(0\ 1\ 0\ 1\ 0)$;

$\{3, 6, 10\}$ $(0\ 1\ 0\ 1\ 1)$;

$\{4, 6, 7, 10\}$ $(0\ 0\ 0\ 1\ 1)$;

$\{7, 8, 9\}$ $(0\ 1\ 0\ 0\ 0)$.

Кратчайшее покрытие

$\{1, 6, 7, 9\}$

$\{2, 3, 4, 5, 8\}$

$\{2, 4, 7, 8, 10\}$

	a_1	a_2	a_3	a_4	
1	**1**	**1**	**1**	5	0 0 0
2	1	**2**	**2**	**2**	0 1 1
3	**3**	2	4	5	1 1 1
4	3	2	**4**	5	1 1 0
5	–	**5**	4	**5**	1 0 0

Кодирование состояний асинхронного автомата

Получение функций возбуждения элементов памяти

	a_1	a_2	a_3	a_4	$z_1z_2z_3$
1	**1**	**1**	**1**	5	0 0 0
2	1	**2**	**2**	**2**	0 1 1
3	**3**	2	4	5	1 1 1
4	3	2	**4**	5	1 1 0
5	–	**5**	4	**5**	1 0 0

$$
\begin{array}{cccc}
a & z_1 & z_2 & z_3 \\
\end{array}
\quad
\begin{array}{ccc}
z_1^+ & z_2^+ & z_3^+
\end{array}
$$

$$
\begin{bmatrix}
a_1 & 0 & - & - \\
a_1 & 1 & 1 & - \\
a_2 & 0 & 0 & 0 \\
a_2 & - & 1 & - \\
a_2 & 1 & 0 & 0 \\
a_3 & 0 & 0 & 0 \\
a_3 & 0 & 1 & 1 \\
a_3 & 1 & - & - \\
a_4 & - & 0 & 0 \\
a_4 & 1 & - & - \\
a_4 & 0 & 1 & 1
\end{bmatrix},
\quad
\begin{bmatrix}
0 & 0 & 0 \\
1 & 1 & 1 \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 1 & 0 \\
1 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 1
\end{bmatrix}
$$

Кодирование состояний асинхронного автомата

Упрощение функций возбуждения элементов памяти

$$
\begin{array}{c}
\begin{array}{cccc} a & z_1 & z_2 & z_3 \end{array} \\
\begin{bmatrix}
a_1 & 0 & - & - \\
a_1 & 1 & 1 & - \\
a_2 & 0 & 0 & 0 \\
a_2 & - & 1 & - \\
a_2 & 1 & 0 & 0 \\
a_3 & 0 & 0 & 0 \\
a_3 & 0 & 1 & 1 \\
a_3 & 1 & - & - \\
a_4 & - & 0 & 0 \\
a_4 & 1 & - & - \\
a_4 & 0 & 1 & 1
\end{bmatrix}
\end{array},
\begin{array}{c}
\begin{array}{ccc} z_1^+ & z_2^+ & z_3^+ \end{array} \\
\begin{bmatrix}
0 & 0 & 0 \\
1 & 1 & 1 \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 1 & 0 \\
1 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 1
\end{bmatrix}
\end{array}
\rightarrow
\begin{array}{c}
\begin{array}{cccc} a & z_1 & z_2 & z_3 \end{array} \\
\begin{bmatrix}
a_1 & 0 & - & - \\
a_1 & 1 & - & - \\
a_2 & 0 & 0 & - \\
a_2 & - & 1 & - \\
a_2 & 1 & 0 & - \\
a_3 & 0 & 0 & - \\
a_3 & 0 & 1 & - \\
a_3 & 1 & - & - \\
a_4 & - & 0 & - \\
a_4 & 1 & - & - \\
a_4 & 0 & 1 & -
\end{bmatrix}
\end{array},
\begin{array}{c}
\begin{array}{ccc} z_1^+ & z_2^+ & z_3^+ \end{array} \\
\begin{bmatrix}
0 & 0 & 0 \\
1 & 1 & 1 \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 1 & 0 \\
1 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 1
\end{bmatrix}
\end{array}
$$

Кодирование состояний асинхронного автомата

Рассмотрение *K*-множеств

K-*множеством* называется множество состояний асинхронного автомата, переходы из которых при некотором фиксированном входном сигнале ведут в одно и то же состояние (также принадлежащее данному множеству).

Парой K-*множеств* называется два различных *K*-множества, построенных для одного и того же входного сигнала.

	a_1	a_2	a_3	a_4	a_5
1	**1**	5	**1**	5	2
2	**2**	6	**2**	–	**2**
3	1	–	4	**3**	–
4	2	5	**4**	–	6
5	2	**5**	1	**5**	–
6	7	**6**	2	3	**6**
7	**7**	5	–	3	6

Матрица условий

$$\begin{array}{c c}
\begin{array}{ccccccc} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 \end{array} & \\
\begin{bmatrix}
0 & 1 & 0 & 1 & 1 & - & - \\
0 & - & 0 & - & - & 1 & 1 \\
- & 0 & - & 0 & 0 & 1 & 1 \\
0 & 1 & - & 0 & 0 & 1 & 0 \\
0 & - & 1 & 1 & 0 & - & - \\
- & 0 & 1 & 1 & - & 0 & - \\
0 & - & 1 & - & 0 & 1 & 1 \\
0 & 0 & - & 1 & - & 1 & 1
\end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}
\end{array}$$

Кодирование состояний асинхронного автомата

Рассмотрение *K*-множеств

Матрица условий

$$\begin{array}{c} \begin{array}{ccccccc} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 \end{array} & \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & - & - \\ 0 & - & 0 & - & - & 1 & 1 \\ - & 0 & - & 0 & 0 & 1 & 1 \\ 0 & 1 & - & 0 & 0 & 1 & 0 \\ 0 & - & 1 & 1 & 0 & - & - \\ - & 0 & 1 & 1 & - & 0 & - \\ 0 & - & 1 & - & 0 & 1 & 1 \\ 0 & 0 & - & 1 & - & 1 & 1 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \end{array}$$

Совместимые множества и соответствующие векторы

$\{1, 2\}$ (0 1 0 1 1 1 1);
$\{1, 3\}$ (0 1 0 1 1 0 0);
$\{2, 3\}$ (0 0 0 0 0 1 1);
$\{2, 6\}$ (0 1 0 0 – 1 1);
$\{2, 8\}$ (0 0 0 1 – 1 1);
$\{3, 7\}$ (0 0 1 0 0 1 1),
$\{4, 6\}$ (0 1 0 0 0 1 0);
$\{5, 6\}$ (0 0 1 1 0 0 –);
$\{5, 7, 8\}$ (0 0 1 1 0 1 1).

$\{1, 2\}$, $\{1, 3\}$, $\{4, 6\}$ и $\{5, 7, 8\}$ составляют кратчайшее покрытие множества строк матрицы условий.

Кодирование состояний асинхронного автомата

Получение функций возбуждения по *K*-множествам

Матрица кодирования

$$\begin{array}{c}
\begin{array}{cccc} z_1 & z_2 & z_3 & z_4 \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1
\end{bmatrix}
\end{array}
\begin{array}{c} \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \end{array}$$

$$\begin{array}{c}
\begin{array}{ccccc} a & z_1 & z_2 & z_3 & z_4 \end{array} \\
\begin{bmatrix}
a_1 & 0 & 0 & 0 & - \\
a_1 & 1 & 1 & - & - \\
a_1 & 1 & 0 & - & 1 \\
a_2 & - & - & 0 & - \\
a_2 & 1 & - & 1 & - \\
a_3 & - & - & 0 & 0 \\
a_3 & 1 & - & 1 & - \\
a_3 & - & - & 0 & 1 \\
a_4 & - & - & 0 & 0 \\
a_4 & - & 0 & - & 1 \\
a_5 & - & - & - & 0 \\
a_5 & 1 & - & - & 1
\end{bmatrix}
\end{array}
\qquad
\begin{array}{c}
\begin{array}{cccc} z_1^+ & z_2^+ & z_3^+ & z_4^+ \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1
\end{bmatrix}
\end{array}$$

Кодирование состояний асинхронного автомата

Упрощение функций возбуждения

$$
\begin{array}{c}
\begin{array}{ccccc} a & z_1 & z_2 & z_3 & z_4 \end{array} \\
\begin{bmatrix}
a_1 & 0 & 0 & 0 & - \\
a_1 & 1 & 1 & - & - \\
a_1 & 1 & 0 & - & 1 \\
a_2 & - & - & 0 & - \\
a_2 & 1 & - & 1 & - \\
a_3 & - & - & 0 & 0 \\
a_3 & 1 & - & 1 & - \\
a_3 & - & - & 0 & 1 \\
a_4 & - & - & 0 & 0 \\
a_4 & - & 0 & - & 1 \\
a_5 & - & - & - & 0 \\
a_5 & 1 & - & - & 1
\end{bmatrix}
\end{array}
\quad
\begin{array}{c}
\begin{array}{cccc} z_1^+ & z_2^+ & z_3^+ & z_4^+ \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1
\end{bmatrix}
\end{array}
\quad \rightarrow \quad
\begin{array}{c}
\begin{array}{ccccc} a & z_1 & z_2 & z_3 & z_4 \end{array} \\
\begin{bmatrix}
a_1 & 0 & 0 & - & - \\
a_1 & - & 1 & - & - \\
a_1 & 1 & 0 & - & - \\
a_2 & - & - & 0 & - \\
a_2 & - & - & 1 & - \\
a_3 & - & - & 0 & 0 \\
a_3 & - & - & 1 & - \\
a_3 & - & - & 0 & 1 \\
a_4 & - & - & - & 0 \\
a_4 & - & - & - & 1 \\
a_5 & - & - & - & 0 \\
a_5 & - & - & - & 1
\end{bmatrix}
\end{array}
\quad
\begin{array}{c}
\begin{array}{cccc} z_1^+ & z_2^+ & z_3^+ & z_4^+ \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1
\end{bmatrix}
\end{array}
$$

Кодирование состояний асинхронного автомата

Соседнее кодирование

Устраняются любые состязания.

Каждый из прямых переходов заменяется последовательностью элементарных переходов.

Состояния размещаются по вершинам булева гиперкуба.

Переход из состояния в состояние представляется как движение по ребрам гиперкуба, из вершины в вершину.

Можно использовать процесс сборки гиперкуба, применяемый при кодировании синхронного автомата методом «желательных соседств».

Роль функции w_{ij}, задаваемой на парах состояний автомата, будет играть число K-множеств, содержащих пару состояний q_i, q_j.

Кодирование состояний асинхронного автомата

Соседнее кодирование

В гиперкубе выделяются связные подграфы, порожденные *K*-множествами. Состояния, связанные элементарным переходом, располагаются в подграфе на соседних вершинах.

Ограничение:

Подграфы, которые соответствуют *K*-множествам, определяемым одним и тем же входным сигналом, не должны пересекаться. Если это невозможно, надо увеличить размерность гиперкуба на единицу.

	a_1	a_2	a_3	a_4	a_5
1	**1**	5	**1**	5	2
2	**2**	6	**2**	–	**2**
3	1	–	4	**3**	–
4	2	5	**4**	–	6
5	2	**5**	1	**5**	–
6	7	**6**	2	3	**6**
7	**7**	5	–	3	6

$K_{11} = \{1, 3\}$, $K_{12} = \{2, 4, 5\}$, $K_{17} = \{6, 7\}$;
$K_{25} = \{1, 4, 5, 7\}$, $K_{26} = \{2, 6\}$;
$K_{31} = \{1, 5\}$, $K_{32} = \{2, 6\}$, $K_{34} = \{3, 4\}$;
$K_{43} = \{3, 6, 7\}$, $K_{45} = \{1, 5\}$;
$K_{52} = \{1, 2\}$, $K_{56} = \{4, 6, 7\}$.

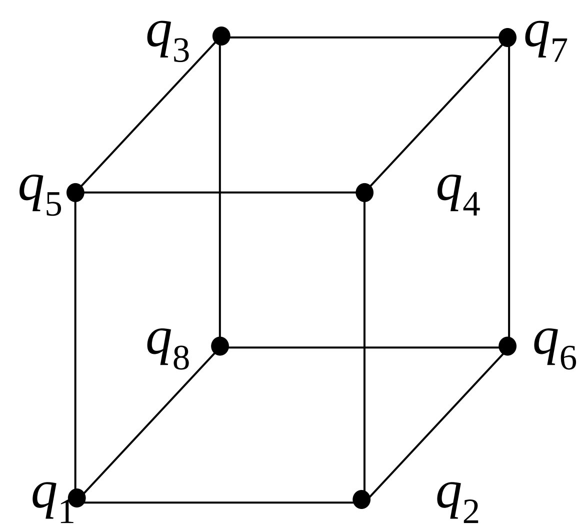
Кодирование состояний асинхронного автомата

Соседнее кодирование

	a_1	a_2	a_3	a_4	a_5
1	**1**	5	**1**	5	2
2	**2**	6	**2**	–	**2**
3	1	–	4	**3**	–
4	2	5	**4**	–	6
5	2	**5**	1	**5**	–
6	7	**6**	2	3	**6**
7	**7**	5	–	3	6

$K_{11} = \{1, 3\}$, $K_{12} = \{2, 4, 5\}$, $K_{17} = \{6, 7\}$;
$K_{25} = \{1, 4, 5, 7\}$, $K_{26} = \{2, 6\}$;
$K_{31} = \{1, 5\}$, $K_{32} = \{2, 6\}$, $K_{34} = \{3, 4\}$;
$K_{43} = \{3, 6, 7\}$, $K_{45} = \{1, 5\}$;
$K_{52} = \{1, 2\}$, $K_{56} = \{4, 6, 7\}$.

q_2	q_3	q_4	q_5	q_6	q_7	q_8	
1	1	1	3	0	1	0	q_1
	0	1	1	2	0	0	q_2
		1	0	1	1	0	q_3
			2	1	2	0	q_4
				0	1	0	q_5
					3	0	q_6
						0	q_7



Кодирование состояний асинхронного автомата

Соседнее кодирование. Выделение связных подграфов

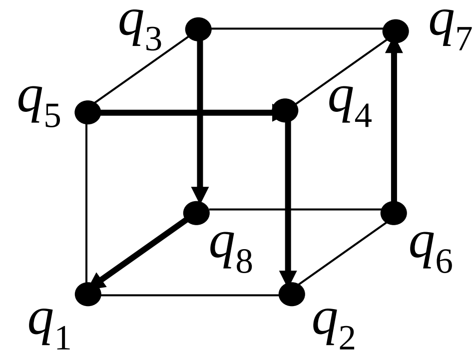
$K_{11} = \{\mathbf{1}, 3\}$, $K_{12} = \{\mathbf{2}, 4, 5\}$, $K_{17} = \{6, \mathbf{7}\}$;

$K_{25} = \{1, 4, \mathbf{5}, 7\}$, $K_{26} = \{2, \mathbf{6}\}$;

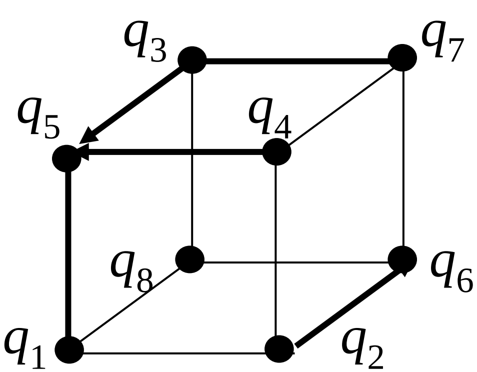
$K_{31} = \{\mathbf{1}, 5\}$, $K_{32} = \{\mathbf{2}, 6\}$, $K_{34} = \{3, \mathbf{4}\}$;

$K_{43} = \{\mathbf{3}, 6, 7\}$, $K_{45} = \{1, \mathbf{5}\}$;

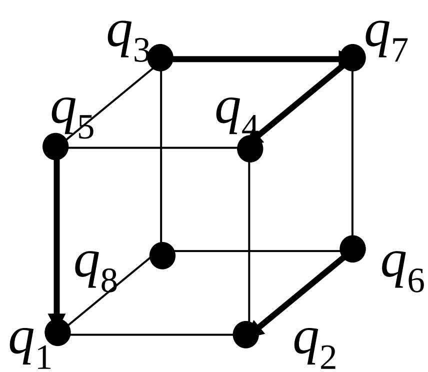
$K_{52} = \{1, \mathbf{2}\}$, $K_{56} = \{4, \mathbf{6}, 7\}$.



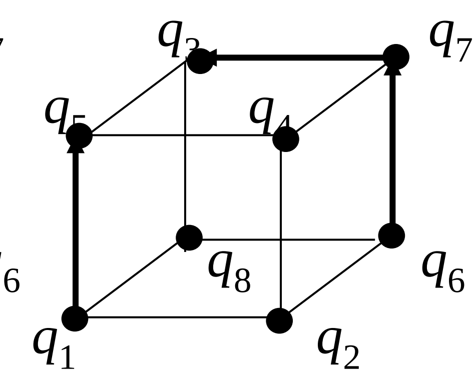
a_1



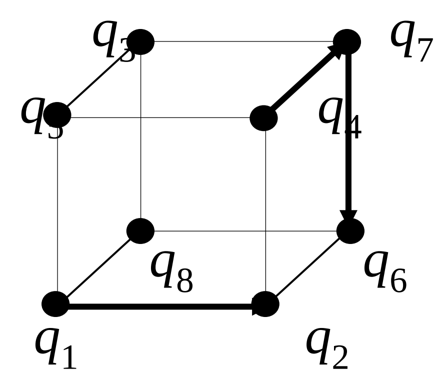
a_2



a_3



a_4



a_5

Кодирование состояний асинхронного автомата

Соседнее кодирование. Преобразование таблицы переходов

$K_{11} = \{\mathbf{1}, 3\}$, $K_{12} = \{\mathbf{2}, 4, 5\}$, $K_{17} = \{6, \mathbf{7}\}$;
$K_{25} = \{1, 4, \mathbf{5}, 7\}$, $K_{26} = \{2, \mathbf{6}\}$;
$K_{31} = \{\mathbf{1}, 5\}$, $K_{32} = \{\mathbf{2}, 6\}$, $K_{34} = \{\mathbf{3}, 4\}$;
$K_{43} = \{\mathbf{3}, 6, 7\}$, $K_{45} = \{1, \mathbf{5}\}$;
$K_{52} = \{1, \mathbf{2}\}$, $K_{56} = \{4, \mathbf{6}, 7\}$.

	a_1	a_2	a_3	a_4	a_5
q_1	$\boldsymbol{q}_1$	q_5	$\boldsymbol{q}_1$	q_5	q_2
q_2	$\boldsymbol{q}_2$	q_6	$\boldsymbol{q}_2$	–	$\boldsymbol{q}_2$
q_3	q_1	–	q_4	$\boldsymbol{q}_3$	–
q_4	q_2	q_5	$\boldsymbol{q}_4$	–	q_6
q_5	q_2	$\boldsymbol{q}_5$	q_1	$\boldsymbol{q}_5$	–
q_6	q_7	$\boldsymbol{q}_6$	q_2	q_3	$\boldsymbol{q}_6$
q_7	$\boldsymbol{q}_7$	q_5	–	q_3	q_6

	a_1	a_2	a_3	a_4	a_5
q_1	$\boldsymbol{q}_1$	q_5	$\boldsymbol{q}_1$	q_5	q_2
q_2	$\boldsymbol{q}_2$	q_6	$\boldsymbol{q}_2$	–	$\boldsymbol{q}_2$
q_3	q_8	–	q_7	$\boldsymbol{q}_3$	–
q_4	q_2	q_5	$\boldsymbol{q}_4$	–	q_7
q_5	q_4	$\boldsymbol{q}_5$	q_1	$\boldsymbol{q}_5$	–
q_6	q_7	$\boldsymbol{q}_6$	q_2	q_7	$\boldsymbol{q}_6$
q_7	$\boldsymbol{q}_7$	q_3	q_4	q_3	q_6
q_8	q_1	–	–	–	–