

Лекция 10 — Задача минимизации

Содержание

1. Нормальные формы
 1. Пререквизиты
 2. Нормальные формы
 1. Определение
 2. Классификация
 3. Получение нормальных форм
 4. Бинарная диаграмма решений
2. Задача минимизации
 1. Мотивация
 2. Алгебраические методы минимизации
 3. Карты Карно
 1. Диаграммы Вейча
 2. Общие представления
 3. Минимизация при помощи карт Карно
 4. Многомерные карты Карно
 5. Don't-care условие
 4. Метод Петрика и алгоритм Куайна — Мак-Класки

”— Это Красти Краб?

— Нет, это Патрик!”

— С. Р. Петрик

Нормальные формы

Пререквизиты

Определение

Литерал — переменная или её отрицание.

Любое булево выражение составляется из литералов и простых бинарных операций.

ⓘ Определение

Конъюнкция литералов называется **конъюнктом** (также *cube*).

Конъюнкт истинен тогда и только тогда, когда истинен каждый из входящих в него литералов.

Дизъюнкция литералов называется **дизъюнктом** (также *clause*).

Дизъюнкт истинен тогда и только тогда, когда истинен хотя бы один из входящих в него литералов.

ⓘ Определение

Для любого набора из n булевых переменных:

- **Минтерм** m — максимальный конъюнкт, включающий в себя все переменные. Каждый минтерм принимает истинное значение только при одной комбинации значений переменных.
- **Макстерм** M — максимальный дизъюнкт, включающий в себя все переменные. Каждый макстерм принимает ложное значение только при одной комбинации значений переменных.

🔥 Замечание

Минтермы и макстермы можно индексировать: i -й из них будет выражаться через i -ю строку в таблице истинности:

i	x	y	z	m_i	M_i
0	0	0	0	$\bar{x} \wedge \bar{y} \wedge \bar{z}$	$x \vee y \vee z$
1	0	0	1	$\bar{x} \wedge \bar{y} \wedge z$	$x \vee y \vee \bar{z}$
2	0	1	0	$\bar{x} \wedge y \wedge \bar{z}$	$x \vee \bar{y} \vee z$
3	0	1	1	$\bar{x} \wedge y \wedge z$	$x \vee \bar{y} \vee \bar{z}$
4	1	0	0	$x \wedge \bar{y} \wedge \bar{z}$	$\bar{x} \vee y \vee z$
5	1	0	1	$x \wedge \bar{y} \wedge z$	$\bar{x} \vee y \vee \bar{z}$
6	1	1	0	$x \wedge y \wedge \bar{z}$	$\bar{x} \vee \bar{y} \vee z$
7	1	1	1	$x \wedge y \wedge z$	$\bar{x} \vee \bar{y} \vee \bar{z}$

$M_0 = x \vee y \vee z$, поскольку $M_0 = 0$ только при $x = 0, y = 0, z = 0$!

ⓘ Определение

Импликанта функции f — функция g такая, что

$$g(x_1, x_2, \dots, x_n) = 1 \implies f(x_1, x_2, \dots, x_n) = 1$$

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

Исходная функция

x	y	z	$g(x, y, z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Импликанта

Простые импликанты

Импликанта называется *простой* (*prime*), если:

- она является кубом;
- она не поглощается ни одной другой импликантой, являющейся кубом.

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

Исходная функция

x	y	z	$g_1(x, y, z)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

$g_1(x, y, z) = \bar{x} \wedge \bar{y} \wedge \bar{z}$
— не простая импликанта
(поглощается импликантой g_2)

x	y	z	$g_2(x, y, z)$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

$g_2(x, y, z) = \bar{x} \wedge \bar{y}$
— простая импликанта

Ядерные импликанты

Импликанта называется *ядерной* (*essential*), если:

- она является простой;
- существует минтерм исходной функции, который возможно покрыть только этой импликантой.

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

Исходная функция

x	y	z	$g(x, y, z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$g(x, y, z) = x \wedge y \wedge z$$

— ядерная импликанта

Нормальные формы

ⓘ Определение

Форма записи булева выражения, удовлетворяющая определенным условиям, называется его **нормальной формой**.

Если нормальная форма выражения единственна, она называется **канонической**.

Классификация нормальных форм

Отрицательная нормальная форма

Булево выражение находится в *отрицательной нормальной форме (NNF)*, если в нём отсутствуют сложные отрицания, которые можно было бы раскрыть при помощи законов де Моргана.

Дизъюнктивная нормальная форма

Булево выражение находится в *дизъюнктивной нормальной форме (DNF)*, если оно представляет собой дизъюнкцию кубов:

$$f(x_1, x_2, \dots, x_n) = c_1 \vee c_2 \vee \dots \vee c_k,$$

где c_i — произвольный конъюнкт.

Конъюнктивная нормальная форма

Булево выражение находится в *конъюнктивной нормальной форме (CNF)*, если оно представляет собой конъюнкцию дизъюнктов:

$$f(x_1, x_2, \dots, x_n) = c_1 \wedge c_2 \wedge \dots \wedge c_k,$$

где c_i — произвольный дизъюнкт.

Совершенная ДНФ

Булево выражение находится в *совершенной дизъюнктивной нормальной форме (CDNF — canonical disjunctive normal form, также SoP — sum of products)*, если оно представляет собой дизъюнкцию минтермов:

$$f(x_1, x_2, \dots, x_n) = \bigvee_{i \in I} m_i, \quad I \subseteq \{0, 1, \dots, 2^n - 1\}$$

Совершенная КНФ

Булево выражение находится в *совершенной конъюнктивной нормальной форме (CCNF — canonical conjunctive normal form, также PoS — product of sums)*, если оно представляет собой конъюнкцию макстермов:

$$f(x_1, x_2, \dots, x_n) = \bigwedge_{i \in I} M_i, \quad I \subseteq \{0, 1, \dots, 2^n - 1\}$$

Сокращённая ДНФ

Булево выражение находится в *сокращённой дизъюнктивной нормальной форме* (далее — *СДНФ*, также *BCF* — *Blake Canonical Form*), если оно представляет собой дизъюнкцию всех простых импликант.

Минимальная ДНФ

Булево выражение находится в *минимальной дизъюнктивной нормальной форме* (*Minimal DNF*), если оно представляет собой дизъюнкцию таких импликант, что исключение любой из них изменяет функцию.

Минимальная КНФ

Булево выражение находится в *минимальной конъюнктивной нормальной форме* (*Minimal CNF*), если оно представляет собой конъюнкцию таких импликант, что исключение любой из них изменяет функцию.

Получение нормальных форм

Получение ДНФ/КНФ

Чтобы получить выражение в ДНФ или КНФ, необходимо:

1. упростить сложные операции

$$x \iff y \rightarrow (x \wedge y) \vee (\bar{x} \wedge \bar{y})$$

$$x \implies y \rightarrow \bar{x} \vee y$$

$$x \oplus y = (x \wedge \bar{y}) \vee (\bar{x} \wedge y)$$

2. упростить сложные отрицания (получив NNF):

$$\overline{x \vee y} = \bar{x} \wedge \bar{y}$$

$$\overline{x \wedge y} = \bar{x} \vee \bar{y}$$

$$\overline{\bar{x}} = x$$

3. применить дистрибутивность:

$$(x \vee y) \wedge z \xrightarrow{\text{DNF}} (x \wedge z) \vee (y \wedge z)$$

$$(x \wedge y) \vee z \xrightarrow{\text{CNF}} (x \vee z) \wedge (y \vee z)$$

4. упростить получившееся выражение, используя законы идентичности и дополнения.

Получение СДНФ

Чтобы получить выражение в совершенной дизъюнктивной форме, необходимо:

1. найти все столбцы таблицы истинности, в которых функция принимает истинное значение;
2. записать соответствующие минтермы;
3. произвести их дизъюнкцию.

Получение СКНФ

Чтобы получить выражение в совершенной конъюнктивной форме, необходимо:

1. найти все столбцы таблицы истинности, в которых функция принимает ложное значение;
2. записать соответствующие макстермы;
3. произвести их конъюнкцию.

🔥 Замечание

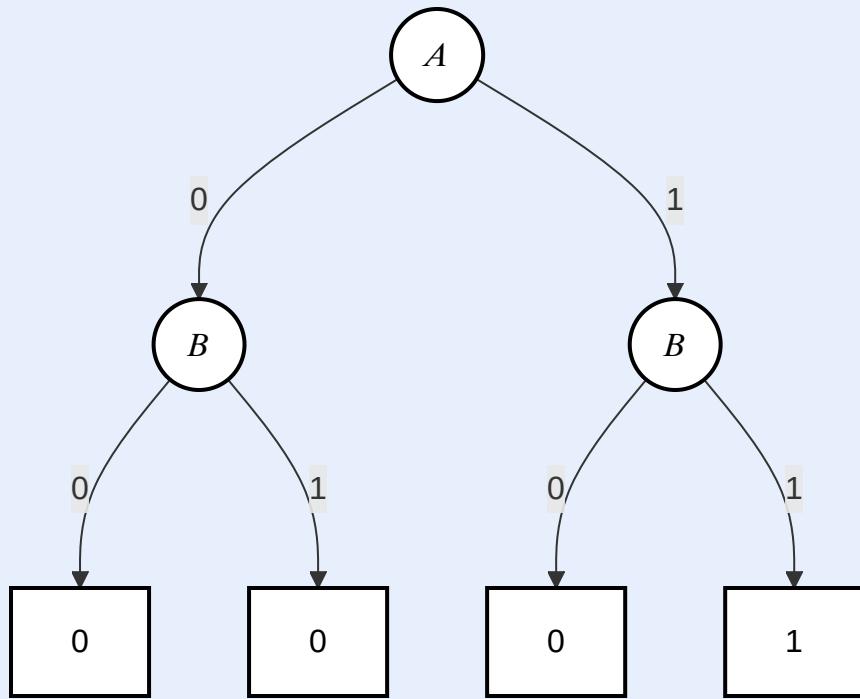
СДНФ и СКНФ функции можно получить путём последовательного применения к ней операции разложения Шеннона по каждой из переменных (в произвольном порядке), причём получившаяся форма зависит от операции, относительно которой разлагается функция.

Бинарная диаграмма решений

Если рекурсивно применять разбиение Шеннона, то можно построить двоичное дерево, отражающее результаты применения функции ко всем возможным входным наборам.

Такое дерево называется **бинарной диаграммой решений** (*BDD — Binary Decision Diagram*) и представляет собой одну из канонических форм представления функции.

Например, так выглядит бинарная диаграмма решений для функции $f(A, B) = A \wedge B$:



Задача минимизации

Мотивация

Мы всегда можем представить любое выражение в одной из совершенных форм, однако результат часто не будет являться минимальным, что в реальных задачах ведёт к избыточным расходам, которых хочется избежать.

Замечание

«Минимальность» в этом контексте может означать:

- минимальное количество литералов;
- минимальное количество конъюнктов/дизъюнктов;
- минимальное количество операций и проч.

Чаще всего мы будем апеллировать ко второй интерпретации. Иными словами, нашей задачей будет получение *минимальной* (или *сокращённой* в качестве промежуточного результата) ДНФ.

Далее будут рассмотрены основные методы решения поставленной задачи.

Алгебраическая минимизация

Самый наивный способ минимизировать булевые выражения — применение аксиом и законов булевой алгебры:

$$\begin{aligned} f &= ABC \vee A\bar{B}\bar{C} \vee \bar{A}BC \vee \bar{A}\bar{B}C \\ &= AB(C \vee \bar{C}) \vee \bar{A}C(B \vee \bar{B}) && \text{дистрибутивность} \\ &= AB \wedge 1 \vee \bar{A}C \wedge 1 && \text{дополнение} \\ &= AB \vee \bar{A}C && \text{идентичность} \end{aligned}$$

Метод получения СДНФ из совершенной ДНФ путём последовательного применения закона склеивания называется **алгоритмом Куайна**.

Карты Карно

Общие представления

ⓘ Определение

Код Грея — бинарная кодировка, в которой последовательные значения различаются ровно на один бит:

000	110
001	111
011	101
010	100

ⓘ Определение

Таблица размерностью $2^{\lfloor \frac{n}{2} \rfloor} \times 2^{\lceil \frac{n}{2} \rceil}$, полученная из таблицы истинности булевой функции $f(x_1, x_2, \dots, x_n)$ путём переупорядочения ячеек в соответствии с кодом Грея, называется **картой Карно**:

Карта Карно
для четырёх переменных

D			
A	0	0	1
	0	1	0
	0	1	1
	1	1	0
			1
C			

⌚ Замечание

Код Грея используется для упорядочения значений переменных, составляющих строку или столбец. Так, пары значений переменных A и B в приведенном примере составляют следующую последовательность:

00
01
11
10

Соответственно, все переменные в первой строке будут иметь в своём составе конъюнкту \overline{AB} , во второй — $\overline{A}B$ и.т.д.

⚠ Важно!

Карта Карно имеет структуру *тора*: ячейки, принадлежащие её первому и последнему столбцу или первой и последней строке, также считаются соседними. В частности, углы карты также соседствуют (но не диагональные!)

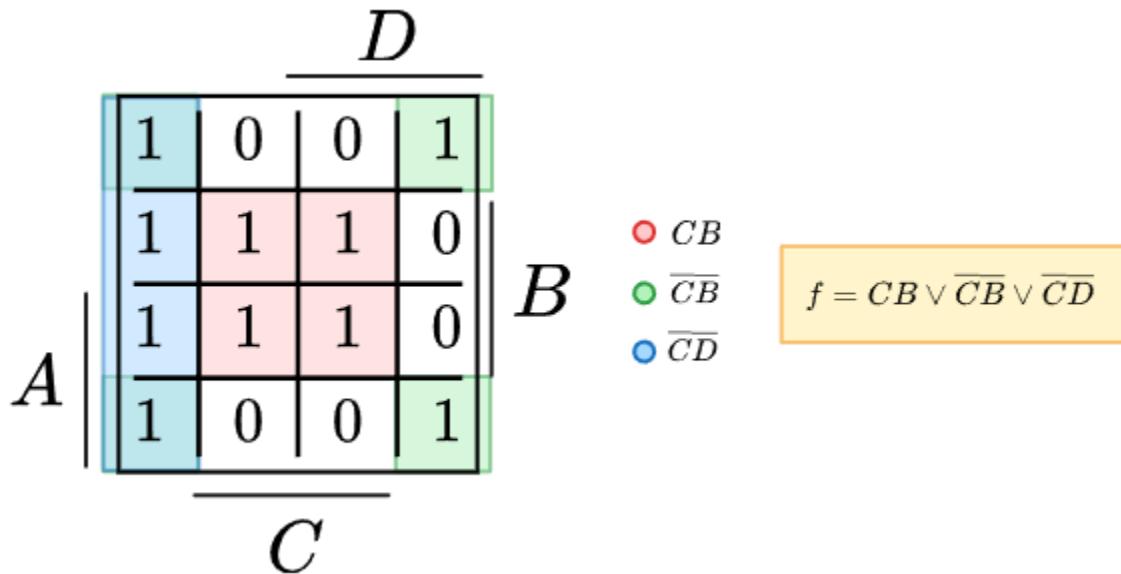
Минимизация при помощи карт Карно

Структура карты Карно позволяет перейти от алгебраической минимизации к визуальному распознаванию паттернов: соседние ячейки в ней соответствуют минтермам, к которым применим закон склеивания; более того, легко показать, что этот закон применим к целым областям ячеек.

Склейываемые области определяются согласно следующим принципам:

1. размер области — степень двойки;
2. область имеет прямоугольную форму;
3. область может пересекать границы таблицы;
4. большие области более приоритетны.

Группы на карте Карно, состоящие только из истинных минтермов, будут соответствовать **импликантам**. Если такая группа не содержится ни в одной другой, она соответствует **простой импликанте**. Таким образом, построение СДНФ по карте Карно сводится к визуальному выделению всех простых импликант:



Для получения минимальной ДНФ после этого необходимо:

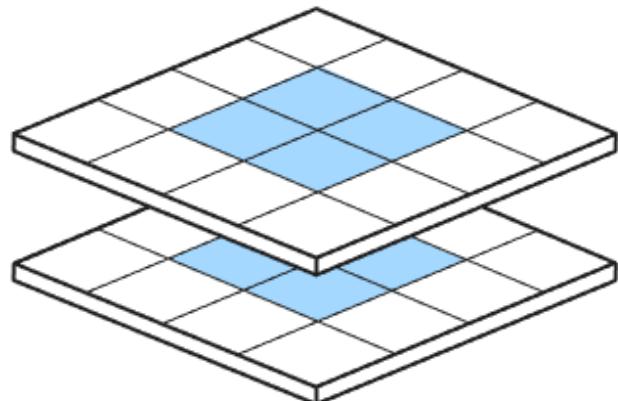
1. выделить ядерные импликанты (такие области, что какая-то из покрытых ими единиц покрыта только ими);
2. покрыть оставшиеся единицы минимальным количеством областей.

Многомерные карты Карно

При составлении карт Карно для функций от пяти и более переменных конструкция несколько усложняется — соединяющие клетки, то есть такие, что соответствующие минтермы отличаются ровно на одну переменную, теперь не обязаны соседствовать. В этом случае карту Карно можно мысленно «сворачивать» пополам, представляя её в трёхмерном виде, в результате чего склеиваемые клетки действительно будут являться соседними:

									<i>E</i>
<i>D</i>				<i>D</i>					
000	001	011	010	110	111	101	100		
1	0	0	0	1	1	0	1		
1	1	0	1	1	0	0	1		
0	0	1	1	1	1	0	0		
1	1	0	1	0	1	1	1		

A *B* *C*



Впрочем, карты Карно редко используются для минимизации функций от такого числа переменных — зачастую в картах Карно их количество ограничивается четырьмя.

Don't-care условие

При работе с булевыми функциями может возникнуть ситуация, при которой их значения при определенных наборах входных данных не имеет значения — например, если они никогда не используются или функция не вполне определена. В таблице истинности подобные комбинации помечаются \times вместо 0 или 1. При минимизации с помощью карт Карно мы можем включать незначащие мinterмы в любые группы, если необходимо, однако не можем составлять группы, целиком состоящие из них.

Метод Петрика

Метод Петрика позволяет аналитически вычислить все возможные минимальные ДНФ по СДНФ.

Алгоритм заключается в следующем:

1. Построение таблицы простых импликант

Метод Петрика работает с таблицей, где строки отвечают за простые импликанты, а столбцы — за мinterмы, покрываемые ими:

	0	1	2	3	4	5	6	7
m_1	✓	✗	✗	✓	✗	✗	✓	✓
m_2	✗	✓	✗	✗	✗	✓	✗	✗
m_3	✗	✗	✓	✓	✗	✗	✗	✓
m_4	✓	✗	✗	✗	✓	✗	✗	✓
m_5	✗	✗	✓	✗	✗	✗	✓	✗

Она также называется *prime implicant chart*.

2. Выделение ядерных импликант

Если какой-то из мinterмов покрывается единственной импликантой (= если в столбце ровно одна галочка), то эта импликанта будет ядерной, и мы обязаны взять её в разбиение:

	0	1	2	3	4	5	6	7
m_1	✓	✗	✗	✓	✗	✗	✓	✓
m_2	✗	✓	✗	✗	✗	✓	✗	✗
m_3	✗	✗	✓	✓	✗	✗	✗	✓
m_4	✓	✗	✗	✗	✗	✓	✗	✓
m_5	✗	✗	✓	✗	✗	✗	✓	✗

3. Исключение ядерных импликант

Следующим шагом мы исключаем из таблицы найденные ядерные импликанты и все мinterмы, покрытые ими. При этом оставшимся импликантам задаются «псевдонимы»:

	2	3	6
$m_1 = \varphi_1$	✗	✓	✓
$m_3 = \varphi_2$	✓	✓	✗
$m_5 = \varphi_3$	✓	✗	✓

4. Построение минимальной ДНФ

Минимальная ДНФ включает в себя все ядерные импликанты и некоторое число таких импликант из получившейся таблицы, что их дизъюнкция покрывают все оставшиеся мinterмы. Запишем, что означает «покрытие»:

$$\begin{aligned}(\varphi_2 \vee \varphi_3) & \text{ минтерм 2} \\ \wedge (\varphi_1 \vee \varphi_2) & \text{ минтерм 3} \\ \wedge (\varphi_1 \vee \varphi_3) & \text{ минтерм 6}\end{aligned}$$

Переведём это выражение в ДНФ:

$$(\varphi_1 \wedge \varphi_2) \vee (\varphi_1 \wedge \varphi_3) \vee (\varphi_2 \wedge \varphi_3)$$

Минимальные формы получаются добавлением любого из получившихся конъюнктов к выделенным ранее ядерным импликантам:

$$\begin{aligned}f &= m_2 \vee m_4 \vee (\varphi_1 \wedge \varphi_2) = \\ &= m_2 \vee m_4 \vee (\varphi_1 \wedge \varphi_3) = \\ &= m_2 \vee m_4 \vee (\varphi_2 \wedge \varphi_3)\end{aligned}$$

Алгоритм Куайна — Мак-Класки

Определение

Непосредственным приложением метода Петрика является **алгоритм Куайна — Мак-Класки** — табличный метод минимизации, гарантирующий нахождение минимальной ДНФ. Он проходит в два этапа:

1. построение СДНФ при помощи алгоритма Куайна;
2. построение минимальной ДНФ при помощи метода Петрика.

В отличие от карт Карно, алгоритм Куайна — Мак-Класки весьма сложно проделывать вручную, однако он достаточно удобен для программирования.