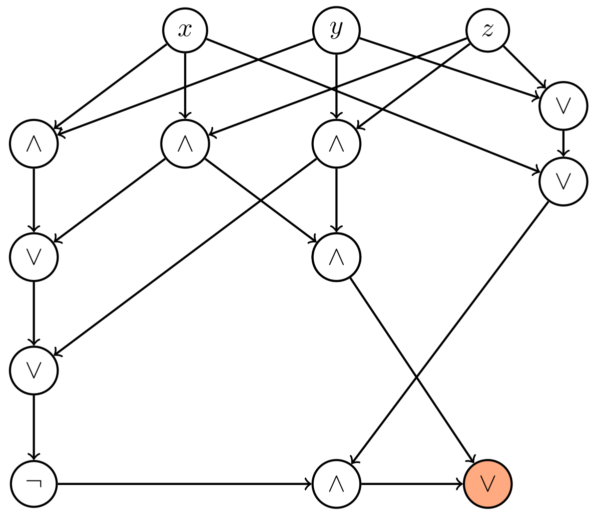
Домашнее задание 18

Ткачев Андрей, группа 166

15 февраля 2017 г.

Задача 1. Приведем в качестве доказательства схему вычисляющую $XOR(x, y, z)$.

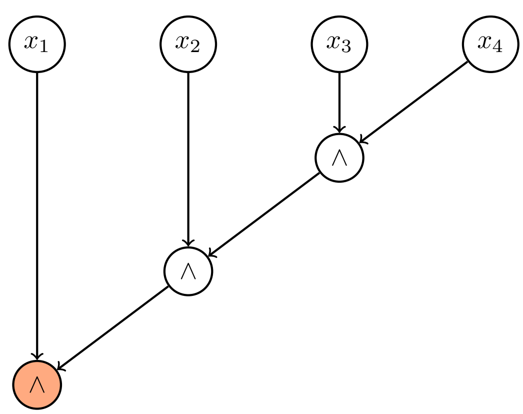


$x \oplus y \oplus z$ принимает значение 0 либо когда $x = y = z = 0$, либо когда какие-то 2 переменные равны 1, а третья — 0; в остальных случаях (только один истинный аргумент или все аргументы истинны) значение — 1. Собственно, приведенная схема ведет себя так же.

Задача 2. Последние 9 наборов значений переменных x_1, x_2, x_3, x_4 в стандартном порядке имеют вид

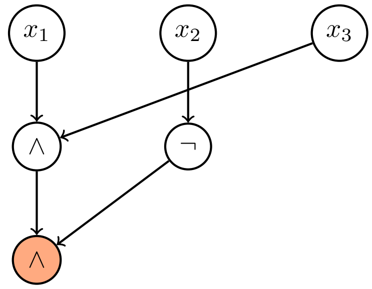
- 0111
- 1000
- 1001 $\vdots$
- 1111

Эти наборы отличаются от оставшихся тем, что в них старший бит равен 1 или последние три бита равны 1. В виде схемы последнее предложение записывается так:

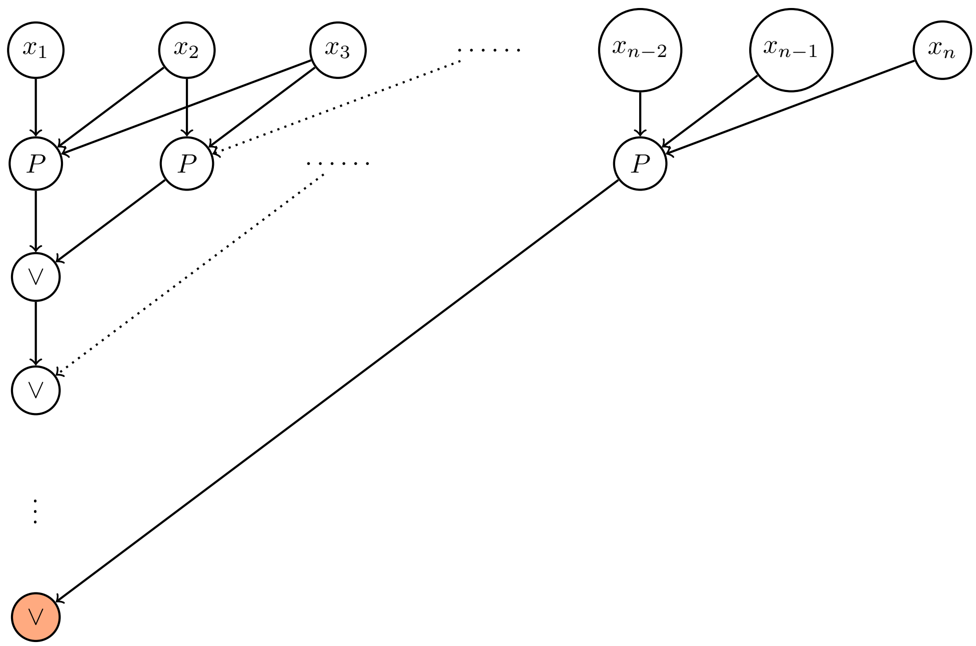


Длина данной схемы равна 7, что явно меньше 15, и она вычисляет требуемую функцию.

Задача 3. Обозначим за элемент P с тремя входами и одним выходом следующую схему:



Нетрудно видеть, что данная схема вычисляет функцию, которая истинна только на наборе аргументов $1, 0, 1$ (порядок важен). Тогда составим схему для входа размера n, определяющую, входит ли подстрока 101 в $x_1 x_2 \dots x_n$..



Корректность работы схемы можно доказать по индукции по размеру входа (База: $n = 3$, результатом является просто результат работы схемы P, которая корректна. Предположение: верно для $n = k$. Шаг: строка из $k + 1$ аргументов содержит подстроку 101 если ее содержит подстрока из первых k аргументов или последние три аргумета, т.е. результат работы схемы на этом входе — дизъюнкиция выхода схемы на k аргументов и схемы P на последних трех аргументах, что и требовалось).

Оценим размер полученной схемы. Зная, что $|P| = 3$, получаем, что размер схемы

$$n + \underbrace{(n-2)|P|}_{\text{n - 2 раза используем схему } P} + \underbrace{(n-3)}_{\text{n - 3 раза используем дизъюнкцию}} =$$

$$= n + 3n - 6 + n - 3 = 5n - 9 = O(n)$$

Задача 4. Умножение двоичного числа на 3 равносильно сумме этого числа и его же, умноженного на 2.

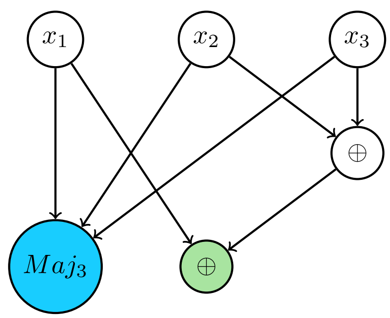
Умножение двоичного числа на 2 есть приписывание к нему справа нулевого бита.

Если двоичное число в своей записи имеет n бит, то результатом умножения на три можест стать $n + 2$ битное число. Таким образом, мы строим схему на n входов $x_n, \dots, x_1$ (пусть x_n - страший бит) и $n + 2$ выходами $y_{n+2}, \dots, y_1$.

Суммирование реализуем, как обычное сложение столбиком

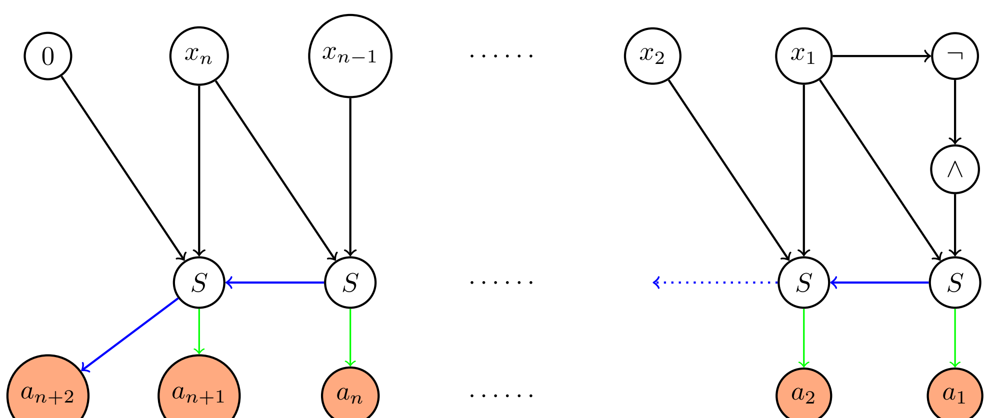
$$
\begin{array}{r}
x_n\, x_{n-1} \ldots x_1 0 \\
x_n\ \ldots x_2 x_1 \\
\hline
(x_n \oplus r_n)\ \ \ldots\ \ (x_1 \oplus x_2 \oplus r_1)(x_1 \oplus x_2) x_1
\end{array}
$$

Для красоты введем вспомогательную схему S с тремя входами — аргументы сложения и значение, которое переносится с прошлых разрядов, и двумя выходами — результат сложения по модулю и число, которое необходимо перенести в следущий разряд (1, если единиц среди аргументов больше одной).



Зеленый выход — результат сложения, который будет записан в разряд, синий — число для переноа в следующий разряд.

Позволим себе использовать элемент «0» (т.к. мы его можем получить из $x \wedge \bar{x}$). Будем считать так же, что если на изображение в S входит только две «стрелки» причем одна из них от нуля, то это означает, что недостающие аргументы — нули (во избежание нагромаждения стрелок).



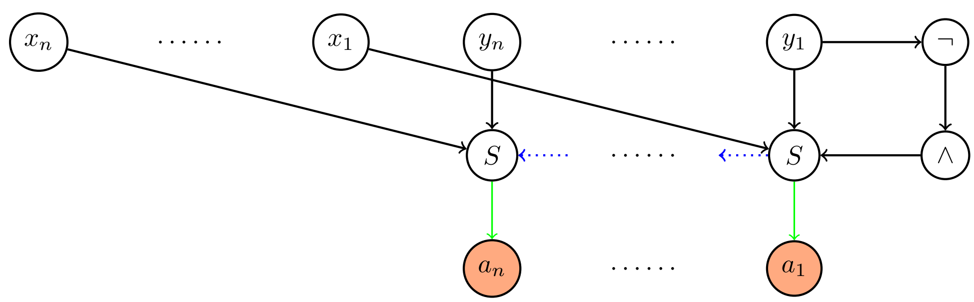
Оценим размер схемы S. В ней используется два xor, каждый из которых требует 5 базисных функций (два отрицания, две конъюнкции и дизъюнкция) и функция Maj_3, которая записывается за 4 базовых функции (две конъюнкции и две дизъюнкции). Итого: $|S| = 14$.

Теперь оценим нашу схему. В ней используется ровно $n+1$ подсхема S, две элементарных функции для получения нуля и n аргументнов. Итого: $(n+1)|S| + n + 2 = 14n + 16 = O(n)$.

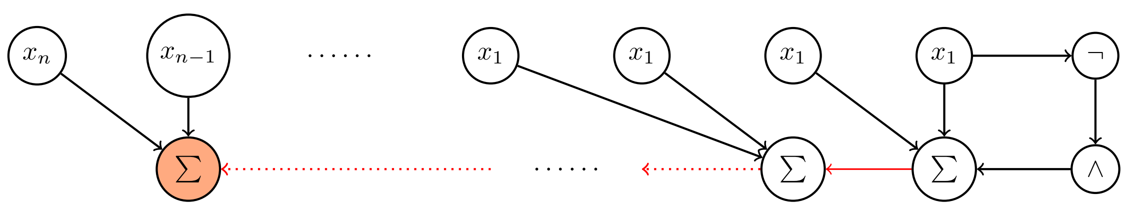
Задача 5. Воспользуемся признаком делимости на 3 в четверичной системе счисления: число делится на 3 в четверичной системе счисления, когда его сумма цифр делится на 3 (следует из того, что в этой системе i-ый разряд на самом деле является числом $4^i \cdot a = (4^i - 1)a + a = (4-1)(\ldots)a + a = 3(...)a + a$, т.е. любое число можно записать как сумму цифр и некую сумму чисел, кратных трем). Из четверичной системы мы можем легко перейти в двоичную — достаточно каждую цифру представить как двочиное двухбитное число. Тогда получим признак делимости на 3 для двоичных чисел: двоичное число делится на три, если сумма чисел, образованных битами 0 и 1, 2 и $3 \ldots, n-1$ и n делится на три.

Реализуем схему P, которая будет однократно выполнять суммирование, описанное выше, чтобы после многократного ее применения получиль простое двухбитное число, для которого понятно, как определить кратность.

Для начала при помощи схемы суммирования бит S из одной из прошлых задач построим схему сумматора $\sum$ с $2n$ входами и n выходами (т.е. в результате указанного суммирования мы не можем получить число больше исходного(так если мы выполним суммировние для $2^{n+1} - 1$ мы получим число не больше, чем $\frac{3n}{2}$, двоичная запись которого много меньше n бит).

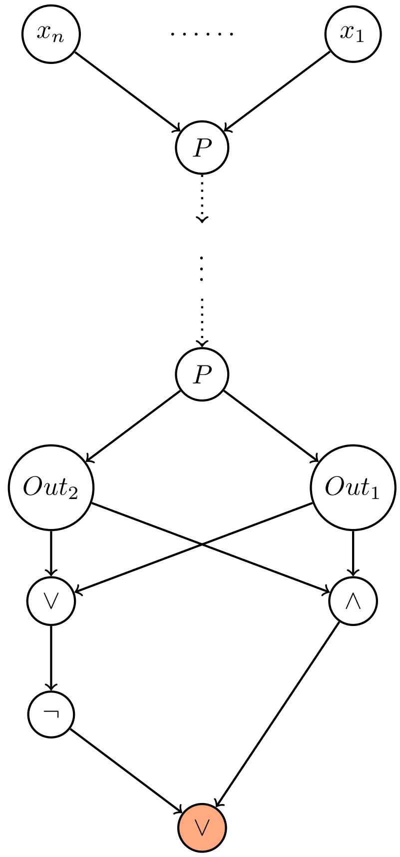


Теперь построим P. Позволим себе некоторую вольность: мы не будем рисовать, как мы передаем сумматору все $2n$ аргументов, довольствуемся лишь условным обозначением — стрелки от $x_i x_{i-1}$ будут означать, что эти переменные идут как первые два младших бита в одном аргументе сложения, в то время как остальные биты этого слагаемого мы примем за нули (которые мы можем синтезировать из базисных функций), стрелка от другого сумматора — есть второе число для сложения. Стрелка же от конъюнкции $x \wedge \bar{x}$ в данном случае означает, что в качестве второго числа для сложения мы принимаем n нулей.



Стоит сказать, что наше n не обязательно четно. В этом случае схема отличается лишь тем, что в один из сумматоров вместо второй переменной передается 0, как лидирующий бит. Позволим себе не загромождать схему деталями, не меняющими ход решения (ведь ноль мы все равно умеем получать, и более того уже получили). Теперь поймем, что нам нужно знать, является ли выход схемы P числом кратным трем. Для этого пирменим ее же к выходу схемы P на входе из нашего числа. Поймем, также что применение схемы P к числу с двумя младшими значащими битами мы получим это же число, т.е. не страшно, если мы применим схему P ко входу достаточно много раз. Поймем, что наверняка достаточно применить схему P всего лишь n раз (в самом деле, после применения число значащих бит уменьшается больше чем в два раза $n \to log_2(\frac{3n}{2}) \to log_2(\frac{3log_2(\frac{3n}{2})}{2})\ldots)$, таким образом после n-ого применения мы наверняка получим число с двумя младшими значащими битами. Для них определит делимость на три элементарно — они должны одновременно равняться 0 либо 1. И ответ делится ли это маленькое число на три или нет и является ответом на глобальный вопрос задачи.

Таким образом итоговая схема имеет вид:



Теперь оценим длину схемы. Мы использовали n переменных, n схем P и 6 элементарных функций. Размер схемы P составляет $\frac{n}{2}$ сумматоров и 2 элементарные функции. Сумматор состоит из n схем S, которые в свою очередь имеют размер 14, и двух базисных функций. Таким образом итоговы размер схемы: $6 + n + n \cdot (\frac{n}{2} \cdot (14n + 2) + 2) = O(n^3)$.

Задача 6. Как мы помним, в данном базисе каждая функция представляется полиномом жигалкина, а именно суммой вида $a_0 \oplus (a_1 \wedge x_1) \ldots (b_0 \wedge x_1 x_2) \ldots (z \wedge x_1 \wedge \ldots \wedge x_n)$. Поймем, сколько узлов содержит схема, вычисляющая данный полином. Всего операторов сложения не более чем $1 + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{n} = 2^n$ (по оператору на каждое слагаемое). Узлов вида x_i в схеме не более n. Отбросим все слагаемые с 0-ми коэффициентами. Посмотрим теперь внимательно на слагаемые вида $x_i x_j, i \neq j$. Если считать, что все узлы из одной переменной у нас уже есть, то для вычисление узлов $x_i x_j$ нужна лишь одна конъюнкция на каждое такое слагаемое, т.е. $\binom{n}{2}$. Это наблюдение можно индуктивно продолжить: пусть у нас вычислены все узлы вида $x_{l_1} \ldots x_{l_k}$ и узлы вида x_i, тогда чтобы вычислить все узлы вида $x_{l_1} \ldots x_{l_{k+1}}$ необходимо применить конъюнкцию к узлам первого и второго видов, т.е. всего $\binom{n}{k+1}$ конъюнкций. Таким образом для вычисления полинома требуется не более $\binom{n}{2} + \ldots + \binom{n}{n} = 2^n - n - 1$ конъюнкций. Итого узлов в схеме не более чем $2^n + 2^n - n - 1 < 2^n$.

Задача 7. Воспользуемся определение схемы. В базисе $\{f_1, \ldots, f_m\}$ последовательность функций g_i таких, что $g_i = x_j$ или $g_i = f_j(g_{l_1}, \ldots, g_{l_p})$, где $l_1, \ldots, l_p < i$, называется схемой.

Докажем по индукции по номеру функции в последовательности, что все функции в цепочки линейны, если линейны $f_1, \ldots, f_m$.

База. Для $i = 1$ очевидно, что $g_1 = x_j \Rightarrow g_1 -$ линейна.

Предположение. Пусть $\forall i \leq k$ верно, что g_i - линейна.

Шаг. Докажем, что g_{k+1} так же линейная. Если $g_{k+1} = x_j$, то ничего доказывать и не надо. В противном случае $g_{k+1} = f_j(g_{l_1}, \ldots, g_{l_p})$. По условию f_j — линейна. Пусть $a_{z_1}, \ldots, a_{z_c}$ — ненулевые коэффициенты линейной записи f_j. Тогда распишим $f_j(\vec{g})$ в линейном виде:

$$f_j(\vec{g}) = a_0 \oplus g_{l_{z_1}} \ldots \oplus g_{l_{z_c}}$$

Но по предположению индукции $\{g_{l_{z_i}}\}$ — линейный функции (индексы меньше $k+1$). Значит каждую из них можно записать в их линейном виде:

$$f_j(\vec{g}) = a_0 \oplus ((c_{l_{z_1}1} \wedge x_{a1}) \oplus \ldots) \oplus \ldots \oplus ((c_{l_{z_c}1} \wedge x_{z1}) \oplus \ldots)$$

Причем, если в записи повторяются какие-то слагаемы, то в силу правил сложения по модулю, они либо в сумме дают 0, либо эквивалентны одному из этих слагаемых. Т.е. на самом деле

$$f_j()\vec{g}) = a_0 \oplus (y_1 \wedge x_{d_1}) \oplus \ldots \oplus (y_s \wedge x_{d_s})$$

Но тогда $f_j(\vec{g})$ — линейная функция, значит g_{k+1} — линеная функция. А значит по принципу полной мат. индукции любая функция в цепочке — линейна, а значит, какую-бы из них не выбрали выходом схемы, схема будет вычислять линейную функцию.

Задача 8. $f(x_1, \ldots, x_n)$ — не линейна $\Rightarrow$ ее представление в полиноме Жигалкина имееет хотябы одно слагаемое более чем одной переменной. Из всех таких нелинейных слагаемых выберем одно с наименьшим числом множителей. Переназовем аргументы так, чтобы первые два множителя были x_1 и x_2. В качестве значения остальных переменных в этом слагаемом выберем 1. Все же остальные переменные примем за 0 (эти константы есть в нашем базисе, значит можно их подставить в функцию). Так как мы выбрали слагаемое с наименьшим числом множителей, то все прочие слагаемые с более чем 1 множителем отличаются хотябы одной переменной, а значит равны 0. В свою очередь все однозначные слагаемые в сумме дают некую константу 0 или 1. При этом сами x_1 и x_2 могли как входить в полином, так и нет, а значит $f(x_1, x_2, 1, \ldots, 1, 0, \ldots, 0) = x_1 \wedge x_2 \oplus [x_1] \oplus [x_2] \oplus [1]$ (аргументы будем писать в таком красивом виде, просто потому что так удобно; $[a]$ означает, что слагаемое a может и не существовать).

Рассмотрим все варианты.

1. $f(x_1, x_2, 1, \ldots, 0 \ldots) = x_1 \wedge x_2$ — готово, мы научились получать конъюнкцию в нашем базисе.

2. $f(x_1, x_2, 1, \ldots, 0, \ldots) = g = x_1 \wedge x_2 \oplus x_1$ (или $x_1 \wedge x_2 \oplus x_2$). Построим таблицу истинности для этого выражения.

x_1	x_2	g
0	0	0
0	1	0
1	0	1
1	1	0

Но посмотрим на выражение $f(x_1, \bar{x}_2, 1, \ldots, 0 \ldots) = x_1 \wedge \bar{x}_2 \oplus x_1$.

x_1	x_2	g
0	0	0
0	1	0
1	0	0
1	1	1

Что эквивалентно конъюнкции. Т.е. мы выразили конъюнкцию в нашем базисе.

3. $f(x_1, x_2, 1, \ldots, 0, \ldots) = g = x_1 \wedge x_2 \oplus x_1 \oplus x_2$. Это выражение, как мы помним из прошлого ДЗ, истинно на все наборах аргументов, кроме $x_1 = x_2 = 0$. Значит, $f(\bar{x}_1, \bar{x}_2, \ldots)$ истинно на всех наборах, кроме $x_1 = x_2 = 1$.

Тогда $\neg f(\bar{x}_1, \bar{x}_2, \ldots)$ эквивалентно конъюнкции.

4. $f(x_1, x_2, 1, \ldots, 0, \ldots) = g = x_1 \wedge x_2 \oplus x_1 \oplus 1$ (или $x_1 \wedge x_2 \oplus x_2 \oplus 1$). Построим таблицу истинности для выражения $x_1 \wedge \bar{x}_2 \oplus x_1 \oplus 1$.

x_1	x_2	g
0	0	1
0	1	1
1	0	0
1	1	1

Т.е. $\neg f(x_1, \bar{x}_2, 1, \ldots, 0, \ldots)$ — ровно конъюнкция x_1 и x_2.

5. $f(x_1, x_2, 1, \ldots, 0, \ldots) = g = x_1 \wedge x_2 \oplus x_1 \oplus x_2 \oplus 1$. Данное выражение истинно только при $x_1 = x_2 = 0$, а значит $f(\neg x_1, \neg x_2, 1 \ldots, 0 \ldots) = x_1 \wedge x_2$.

Таким образом мы выразили конъюнкцию через функции базиса.