

Конъюнкция (от лат. conjunctio союз, связь) — логическая операция, по своему применению максимально приближённая к союзу "и". Синонимы: логическое "И", логическое умножение, иногда просто "И".

Дизъюнкция — (лат. disjunctio - разобщение) логическая операция, по своему применению максимально приближённая к союзу «или» в смысле «или то, или это, или оба сразу». Синонимы: логическое «ИЛИ», включающее «ИЛИ», логическое сложение, иногда просто «ИЛИ».

Исключающее ИЛИ (логическое сложение, строгая дизъюнкция) — булева функция и логическая операция. Результат выполнения операции является истинным только при условии, если является истинным в точности один из аргументов.

Отрицание в логике — унарная операция над суждениями, результатом которой является суждение (в известном смысле) «противоположное» исходному. Обозначается знаком \sim перед или чертой над суждением. Синоним: логическое "НЕ".

Побитовое отрицание (или побитовое НЕ, или дополнение) — это унарная операция, действие которой эквивалентно применению логического отрицания к каждому биту двоичного представления операнда. Другими словами, на той позиции, где в двоичном представлении операнда был 0, в результате будет 1, и, наоборот, где была 1, там будет 0.

Побитовое И — это бинарная операция, действие которой эквивалентно применению логического И к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях операндов. Другими словами, если оба соответствующих бита операндов равны 1, результирующий двоичный разряд равен 1; если же хотя бы один бит из пары равен 0, результирующий двоичный разряд равен 0.

Побитовое ИЛИ — это бинарная операция, действие которой эквивалентно применению логического ИЛИ к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях операндов. Другими словами, если оба соответствующих бита операндов равны 0, двоичный разряд результата равен 0; если же хотя бы один бит из пары равен 1, двоичный разряд результата равен 1.

Побитовое исключающее ИЛИ (или побитовое сложение по модулю два) — это бинарная операция, действие которой эквивалентно применению логического исключающего ИЛИ к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях операндов. Другими словами, если соответствующие биты операндов различны, то двоичный разряд результата равен 1; если же биты совпадают, то двоичный разряд результата равен 0.

Битовые сдвиги относят к битовым операциям. При сдвиге значения битов копируются в соседние по направлению сдвига. Различают несколько видов сдвигов — логический, арифметический и циклический, в зависимости от обработки крайних битов.

Также различают сдвиг влево (в направлении от младшего бита к старшему) и вправо (в направлении от старшего бита к младшему).

Бинарные операторы & являются predetermined для целых типов и bool. Для целых типов оператор & выполняет битовую операцию логического умножения операндов. Для операндов bool оператор & выполняет операцию логического умножения операндов, то есть, если оба оператора — true, результатом будет являться значение true иначе false.

Оператор & вычисляет оба оператора независимо от значения первого из них. Для целочисленных типов |вычисляет результат битовой операции ИЛИ для своих операндов. Для операндов bool| выполняет операцию логического ИЛИ

для своих операндов, то есть результатом будет значение false тогда и только тогда, когда оба операнда имеют значение false.

Оператор ^ выполняет побитовую операцию исключающего OR его операндов.

Для операндов bool оператор ^ выполняет операцию логического исключающего OR операндов, то есть результатом будет являться значение true только в том случае, если ровно один из его операндов имеет значение true.

Логический сдвиг. При логическом сдвиге значение последнего бита по направлению сдвига теряется (копируясь в бит переноса), а первый приобретает нулевое значение. Логические сдвиги влево и вправо используются для быстрого умножения и деления на 2, соответственно.

Оператор сдвига влево (<<) сдвигает первый операнд влево в соответствии с количеством бит, заданным вторым операндом. Тип второго операнда должен быть int или тип, имеющий предопределенное неявное числовое преобразование в int.

Если тип первого операнда — int или uint (32-разрядное число), начало сдвига задается пятью младшими разрядами второго операнда. Фактический сдвиг от 0 до 31 бит.

Если тип первого операнда — long или ulong (64-разрядное число), начало сдвига задается шестью младшими разрядами второго операнда. Фактический сдвиг от 0 до 63 бит.

Старшие разряды, которые находятся не в диапазоне тип первого операнда после смены отбрасываются, а пустые младшие разряды заполняются нулями. Операторы сдвига никогда не вызывают переполнений.

Оператор сдвига вправо (>>) сдвигает первый операнд вправо в соответствии с количеством бит, заданным вторым операндом.

Если тип первого операнда — int или uint (32-разрядное число), начало сдвига задается пятью младшими разрядами второго операнда (второй операнд & 0x1f).

Если тип первого операнда — long или ulong (64-разрядное число), начало сдвига задается пятью младшими разрядами второго операнда (второй операнд & 0x3f).

Если тип первого операнда — int или long, сдвиг вправо является арифметическим сдвигом (пустым старшим разрядом задан знаковый бит). Если тип первого операнда — long или ulong (64-разрядное число), начало сдвига задается шестью младшими разрядами второго операнда (второй операнд & 0x3f).

Если тип первого операнда — uint или ulong, сдвиг вправо является логическим сдвигом

Арифметический сдвиг. Арифметический сдвиг аналогичен логическому, но значение слова считается знаковым числом, представленным в дополнительном коде. Так, при правом сдвиге старший бит сохраняет свое значение. Левый арифметический сдвиг идентичен логическому. В языке C# арифметический сдвиг отсутствует.

Циклический сдвиг. При циклическом сдвиге, значение последнего бита по направлению сдвига копируется в первый бит (и копируется в бит переноса). Также различают циклический сдвиг через бит переноса — при нём первый бит по направлению сдвига получает значение из бита переноса, а значение последнего бита сдвигается в бит переноса. В языке C# циклический сдвиг отсутствует.

Условный оператор AND (&&) выполняет логическое AND своих операндов типа bool, но вычисляет только второй операнд при необходимости.

Условный оператор OR (||) выполняет логическое OR своих операндов типа bool, но вычисляет только второй операнд при необходимости.

Короткозамкнутое вычисление - техника работающая по следующему принципу: Если значение первого операнда в операции AND (&&) ложно, то второй операнд не вычисляется, потому что полное выражение в любом случае будет ложным.

Или если значение первого операнда в операции OR (||) истинно, то второй операнд не вычисляется, потому что полное выражение в любом случае будет истинным.

Для применения теорем Де Моргана к логическому оператору AND или OR и паре операндов, требуется инвертировать оба операнда, заменить (AND на OR) или (OR на AND) и инвертировать все выражение полностью.