
UNIT 14 BOOLEAN LOGIC

Structure

- 14.0 Objectives
- 14.1 Introduction
- 14.2 Boolean Basics
- 14.3 Venn Diagrams
- 14.4 Boolean Operators
 - 14.4.1 AND Operator
 - 14.4.2 OR Operator
 - 14.4.3 NOT Operator
 - 14.4.4 Using Parenthesis to Express Correct Logic
- 14.5 Basic Laws
- 14.6 Simplification Laws
- 14.7 Laws Involving Negation
- 14.8 Proximity Operators
- 14.9 Internet Searching
- 14.10 Answers to Self Check Exercises
- 14.11 Summary
- 14.12 Keywords
- 14.13 References and Further Reading

14.0 OBJECTIVES

Boolean logic is used for database searching and is supported by most Web search engines and library databases. Boolean search techniques may be used to perform accurate searches and should eliminate many irrelevant search results.

When you perform a Boolean search, you search the computer database for the key words that best describe your topic. The power of Boolean searching is based on combinations of keywords with connecting terms called operators.

After reading this unit, you will be able to:

- construct more logical queries using the logical operators; and
- formulate more powerful and fast queries on internet searching.

14.1 INTRODUCTION

The term “Boolean” refers to a system of logical thought developed by the English mathematician and computer pioneer, George Boole (1815-64). This concept of logical thought has been developed to a stream of study called Boolean algebra and played a very vital role in computation. This logical analysis concept is very important

in searching methods also in-material of the search domain. However this unit is discussing the topic only in relation to DBMS and to semi-structured documents in Internet.

14.2 BOOLEAN BASICS

In computer operation with binary values, Boolean logic can be used to describe electro magnetically charged memory locations or circuit states that are either charged (1 or true) or not charged (0 or false). The Boolean logic may be explained using logical gates. The basic logic gates comprise AND, OR and NOT. Each gate assesses various information consisting of high or low voltages in accordance with predetermined rules, and produces a single high or low voltage logical conclusion. The voltage itself represents the binary yes-no, true-false, one-zero concept.

When the selecting criteria become complex and include more conditions, the Boolean algebra is used to evaluate the query in DBMS. The clauses or conditions are connected with these Boolean operators (AND, OR, NOT). Each individual clause is evaluated as True or false, then the operators are applied to evaluate the whole condition.

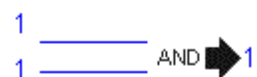
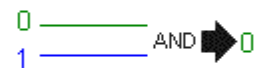
14.3 VENN DIAGRAMS

Boolean concepts are often explained with Venn diagrams, which are generally used to explain the set operations. The Venn diagram shows the search area. In Venn diagram a circle with a word shows the subset of the search area that contains the word written inside it. The overlap of the circles represents the common area of the subsets. For further explanation of our boolean operators the Venn diagram are used.

14.4 BOOLEAN OPERATORS

14.4.1 AND Operator

AND gates will only yield a TRUE result (that is, a 1) if all input is TRUE. Therefore, the top two gates produce a FALSE (binary 0) result.



Truth Table for P AND Q		
P	Q	P AND Q
False	False	False
False	True	False
True	False	False
True	True	True

Truth Table for AND		
AND	False	True
False	False	False
True	False	True

AND - All terms are present

AND means only the data that contain all conditions. A condition consisting of two clauses connected by an AND can be true only if both the clauses are true.

Example:

1. See the Venn diagrams for Library AND Computer Fig 3.1(a)
2. See fig 3.1(b)

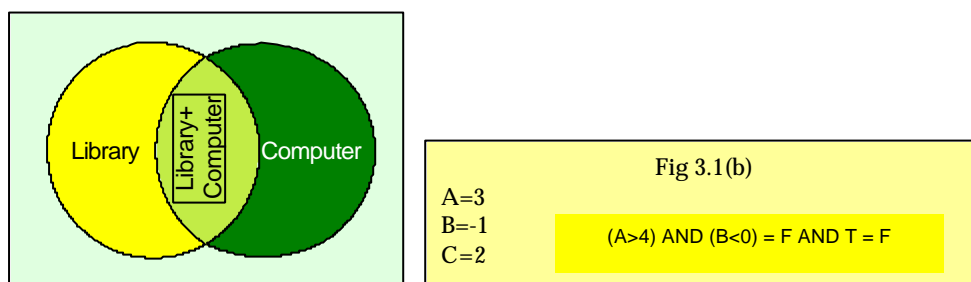
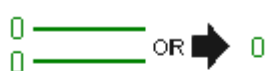
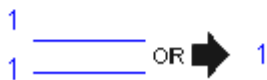
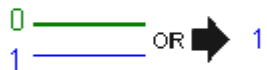


Fig. 3.1 (a) : Librari and Computer

14.4.2 OR Operator



An OR gate will send a TRUE result if any of its inputs is TRUE. Only, if every input is FALSE will it produce a 0 result.



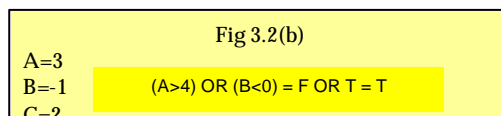
OR - Anyone of the terms are present

A statement that consists of two clauses connected by OR is true as long as at least one of the two conditions is true.

Example:

- 1) See the Venn diagrams for Library OR Computer
- 2) See fig 3.2(b)

Truth Table for OR		
OR	False	True
False	False	True
True	True	True



Truth Table for P OR Q		
P	Q	P OR Q
False	False	False
False	True	True
True	False	True
True	True	True

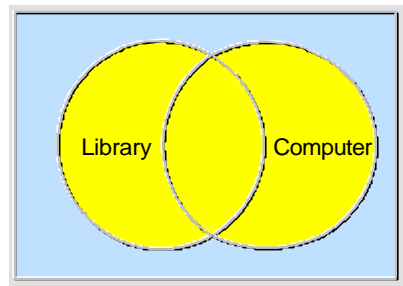


Fig 3.2(a) LIBRARY OR COMPUTER

14.4.3 NOT Operator

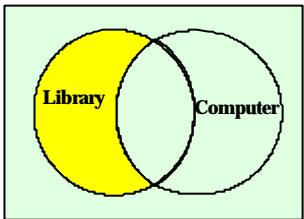


NOT gates, on the other hand, do just one thing. A NOT gate accepts only a single input, either TRUE or FALSE, which it promptly reverses the value. NOT gates can be combined with AND or OR to give NAND (not and) or NOR (not or) gates. These new gates process input in the usual manner and then reverse the result.

By adjusting the gates, binary adders can be used not only for addition, but for subtraction, multiplication and division as well, enabling them to control the computer's data flow and process instructions

Truth table for NOT	
NOT	
False	True
True	False

Truth table for NOT P	
P	NOT P
False	True
True	False



Library NOT Computer

14.4.4 Using Parentheses to Express Correct Search Logic

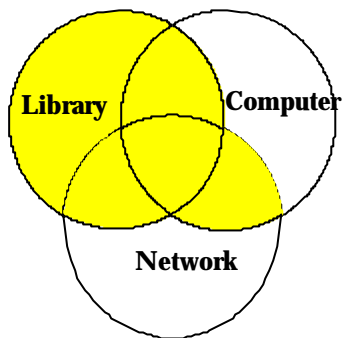
The basic logical operators/gates are explained both by using Venn diagram and logical gates. A complex condition arises when the overall condition contains both AND operators and OR operators. In this situation the resulting truth value depends on the order in which the clauses are evaluated. If we combine these operators in a query, the query is evaluated from left to right in the order of precedence of Boolean operators, i.e., NOT, AND, OR.

That means, an AND operation will be performed before an OR operation if both operations are included in a query. We can use parentheses to specify the desired order for overriding the priority of search operators. As in the case of algebra, expressions in the innermost parentheses are always evaluated first.

Example:

- 1) Library OR (Computer AND Network)

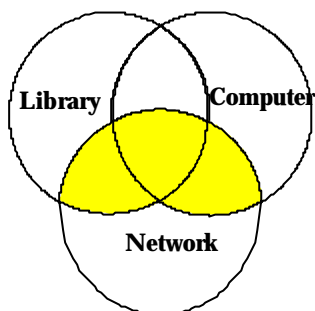
The statement in parentheses - (Computer AND Network) - is processed first. Then Library OR the result is added to the query.



Library OR Computer AND Network

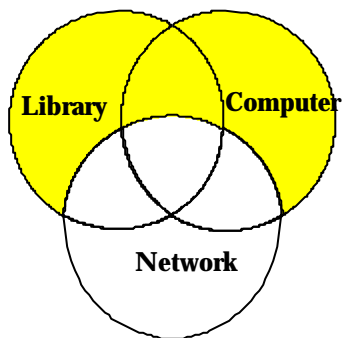
2) (Library OR Computer) AND Network

The statement in parentheses - 'Library OR Computer' - is processed first. Then AND Network is added to the query.



(Library OR Computer) AND Network

3) (Library OR computer) NOT Network. The statement in parentheses - (Library OR Computer) - is processed first.



(Library OR Computer) NOT Network

4) See FIG 3.3 (b)

Fig 3.3 (b)	
A=3	(A>4) AND (B<0) OR (C>1)
B=-1	= F AND T OR T
C=2	= F OR T
	= T
	(A>4) AND ((B<0) OR (C>1))
	= F AND (T OR T)
	= F AND T
	= F

14.5 BASIC LAWS

- Commutative Laws
 - $(p \text{ AND } q) = (q \text{ AND } p)$
 - $(p \text{ OR } q) = (q \text{ OR } p)$
 - $(p = q) = (q = p)$
- Associative Laws (not for combinations of AND & OR)
 - $p \text{ AND } (q \text{ AND } r) = (p \text{ AND } q) \text{ AND } r = p \text{ AND } q \text{ AND } r$
 - $p \text{ OR } (q \text{ OR } r) = (p \text{ OR } q) \text{ OR } r = p \text{ OR } q \text{ OR } r$
- Distributive Laws
 - $p \text{ OR } (q \text{ AND } r) = (p \text{ OR } q) \text{ AND } (p \text{ OR } r)$
 - $p \text{ AND } (q \text{ OR } r) = (p \text{ AND } q) \text{ OR } (p \text{ AND } r)$

14.6 SIMPLIFICATION LAWS

- OR -simplification
 - $p \text{ OR TRUE} = \text{TRUE}$
 - $p \text{ OR FALSE} = p$
 - $p \text{ OR } p = p$
 - $p \text{ OR } (p \text{ AND } q) = p$
- AND-simplification
 - $p \text{ AND TRUE} = p$
 - $p \text{ AND FALSE} = \text{FALSE}$
 - $p \text{ AND } p = p$
 - $p \text{ AND } (p \text{ OR } q) = p$

14.7 LAWS INVOLVING NEGATION

- Law of Negation: $\sim(\sim p) = p$
- Law of the Excluded Middle: $p \text{ OR } \sim p = \text{TRUE}$
- Law of Contradiction: $p \text{ AND } \sim p = \text{FALSE}$
- De Morgan's Laws
 - $\sim(p \text{ AND } q) = \sim p \text{ OR } \sim q$
 - $\sim(p \text{ OR } q) = \sim p \text{ AND } \sim q$

14.8 PROXIMITY OPERATORS

Many library databases and Internet search engines also support variations on the basic Boolean operators, known as **proximity operators**. Proximity, or positional operators are not really part of Boolean logic, but they serve a similar function in formulating search statements. These include ADJACENT, WITHIN, FOLLOWED BY and NEAR. ADJACENT and WITH require that the words appear next to each other, NEAR requires that the search terms appear in close proximity and FOLLOWED BY requires that one term follow another.

- **Within.** The Within proximity operator, w/n, works with two query expressions--one preceding and the other following the operator--and determines if the two expressions appear within a number or n words of one another in the database, in the order that they appear in the query. For example, the query **truth w/3 beauty** will retrieve works containing truth within three words of beauty excluding stopwords. So, while five words intercede between the phrase truth and beauty in the phrase: **truth of much and most rare beauty** it would still appear in a search because the stopwords of and do not count.

The Within operator is unidirectional, from left to right. It will match only those where the order of appearance of terms in query also matches.

The number that you designate with the within operator may immediately follow w/ or may be separated from the w/ with a blank. If w/ is not followed by a number, it will be treated as a query term, not as an operator.

- **Adjacency.** The adjacency operator, ADJ, is equivalent to the Within proximity operator with a defined range of one (w/1). Like within, ADJ locates a term or terms within a given range of another term or terms. Stopwords that appear between the two expressions used with the ADJ operator, moreover, do not count; for example, the query **city adj lights** would retrieve documents containing the string **city lights** as well as documents containing the string **city of lights**. Also similar to the Within operator, adjacency works with multi- as well as single-word expressions. Parentheses may be used to set off multiple-word expressions.

The ADJ operator is unidirectional, from left to right. Given the query **city adj lights**, the adjacency operator would retrieve only those documents in which **lights** occurs after **city**. Occurrences of **city** after **lights** would not be considered hits. Use the NEAR operator to search for occurrences of bidirectional adjacency.

- **NEAR.** The NEAR proximity operator, near/n, acts on two query expressions--one preceding and the other following it--and determines if the two expressions appear within n words of one another in the database. The difference between NEAR and Within is that **NEAR is bidirectional**: the queries on either side of the near/n operator may appear in any order in the database to match the near query. For example, the query **books near/5 wisdom** will produce works containing both terms within five words of one another--excluding stopwords--without distinguishing which comes first. The NEAR operator works on multi- as well as single-word expressions. Parentheses may be used to set off multiple-word expressions.

The number designated with the NEAR operator may immediately follow near/ or may be separated from near with a blank. Note that the number is optional, but if you omit n, you must omit the slash as well, and the program will use a default value

of one. Therefore, the operator will function as bidirectional adjacency; for example, the query **purple near heart** would return the same results as **purple near/l heart**.

The tolerance of NEAR varies by search engine with a range 9 to 15 words being typical.

14.9 INTERNET SEARCHING

The search strategy to be followed in internet is also varies as, it is a semi-structured collection of documents. It may be worth to mention some of them.

- In Internet multiple consecutive words are treated as a phrase
- The words in the exception list is used for searching, such as a, an, the etc.
- Punctuation marks such as the period (.), colon (:), semicolon (;), and comma (,) are ignored during a search.
- To search for a word or phrase containing quotation marks, enclose the entire phrase in quotation marks and then double the quotation marks around the word or words you want to surround with quotes. For example, “World Wide Web or “Web” searches for World Wide Web or “Web”.
- Another useful search parameter that may be used with some search engines and databases is truncation. Truncation allows the searcher to insert a truncation symbol, usually an * or ? For example, the search term teen* will locate the terms teens, teenagers, and teenaged. Some search engines automatically truncate your search terms to find plural, -ing, or -ed endings. First Search requires a + as a wildcard at the end of a word to find simple plural forms: -s or -es, but does not support truncation in the middle of a word.

Self Check Exercise

M=Today is Monday

R=It is Raining

S=It is Snowing

Prepare the truth table for

- a) (M AND R) OR S
- b) M AND (R OR S)

14.10 ANSWERS TO SELF CHECK EXERCISES

Assumption: If it is raining then it is not snowing and viz.

M	R	S	M AND R	(M AND R) OR S	R OR S	M AND (R OR S)
T	T	T	T	T	Not Possible	Not Possible
T	T	F	T	T	T	T
T	F	T	F	T	T	T

T	F	F	F	F	F	F
F	T	T	F	T	Not Possible	Not Possible
F	T	F	F	F	T	F
F	F	T	F	T	T	F
F	F	F	F	F	F	F

14.11 SUMMARY

Boolean search commands have been used by professionals for searching through traditional databases for years. Despite this, they are also supported by all the major search services. In fact, many professionals might benefit by abandoning Boolean commands when using web search engines. But since there is a comfort level in using what is already familiar.

Electronic records, like print records, are organized into separate fields. A typical web page is composed of the following major fields: title, domain, host, URL, and link. When information is entered into a record's field, some search engines allow you to retrieve it by using the correct field label in combination with your search terms. Where available, field searching on the Web is a very powerful tool. It allows you to specify exactly where you want the search engine to look in the Web document. Many search engines gives you advanced/power search facility also.

14.12 KEYWORDS

- Venn Diagram** : A graphic image consisting of circles that represents the logical relationship of two sets of information.
- Truth Table** : A table that describes a logic function by listing all possible input values and indicating for each combination the output value.
- Wild Card** : A pattern matching character

14.13 REFERENCES AND FURTHER READING

Baase, Sara, Computer Algorithms: Introduction to Design and Analysis 3rd ed: Addison -Wesley, Massachusetts, 2000.

Rood, Harold, J, Logic and Structured Design for Computer Programmers, P WS Kent Publ House, Boston, 1992

Spanior, Otto, Computer Arithmetoic Logic and Design, John Wiley & Sons, Chichester,1981