Discrete Mathematics and Its Applications

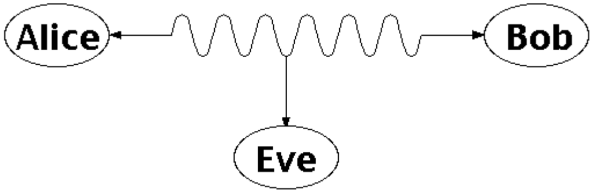
Learning to do proofs from watching the slides is like trying to learn to play football from watching it on TV!

Please, do the exercises!
Use lecture notes as a study guide!

Discrete mathematics is the kind of mathematics one needs to know to communicate with a computer as a designer, programmer, or user.

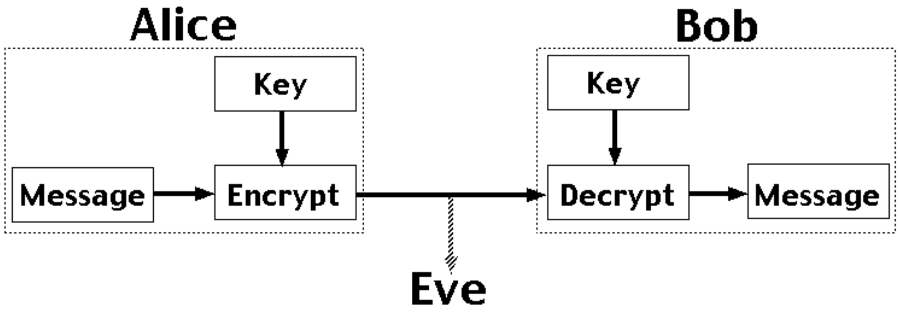
Discrete Mathematics is the branch of Mathematics in which we deal with questions involving finite or countably infinite sets. In particular, this means that the numbers involved are either integers, or numbers closely related to them, such as fractions or ‘modular’ numbers

Use Case1: RSA and Public-key Cryptography



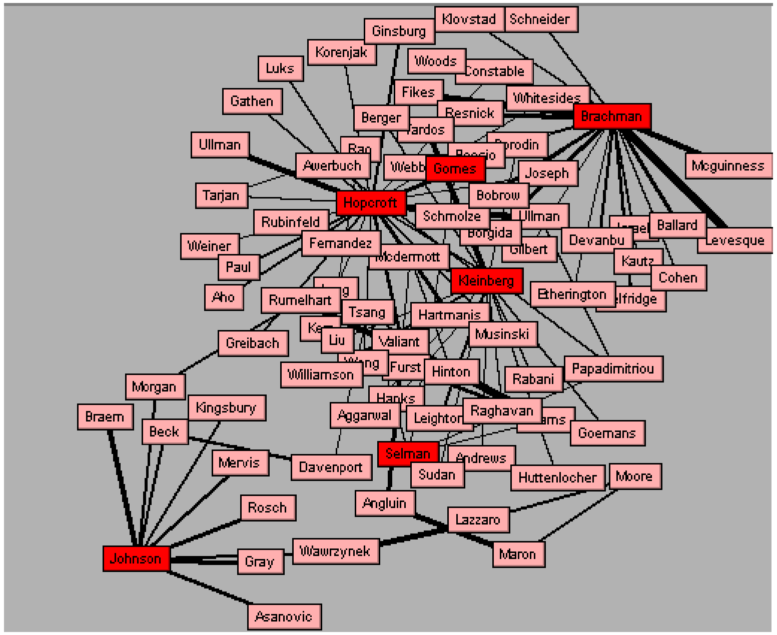
Alice and Bob have never met, but they would like to exchange a message. Eve would like to eavesdrop.

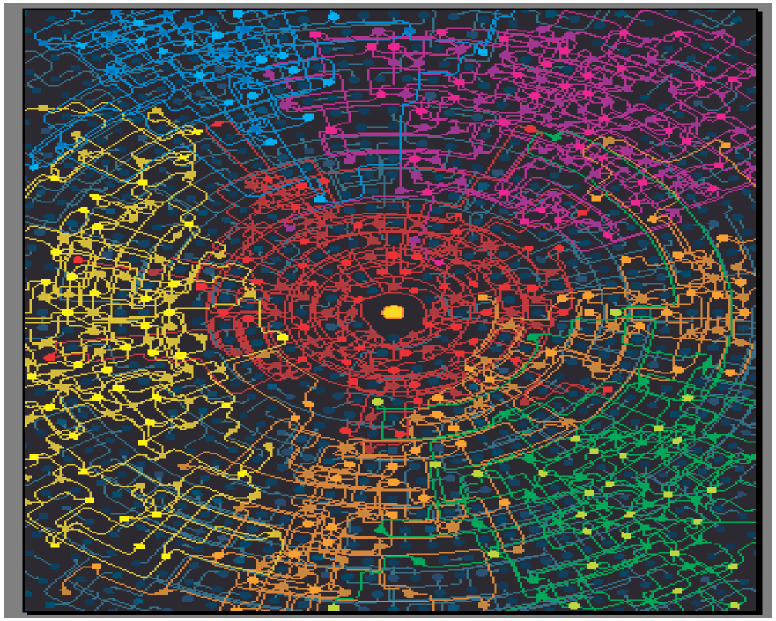
E.g. between you and the Bank of XX.



They could come up with a good encryption algorithm and exchange the encryption key – but how to do it without Eve getting it? (If Eve gets it, all security is lost.)

Use Case2: New Science of Networks





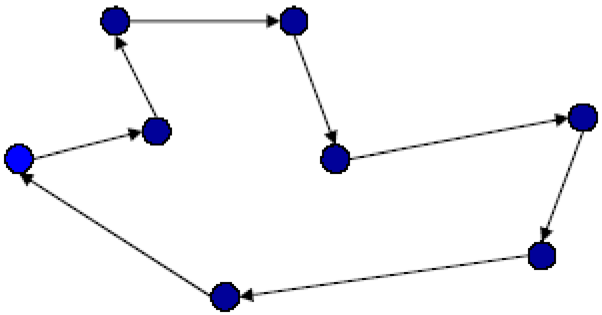
Network of computer scientists
ReferralWeb System
(Kautz and Selman)

Cybercommunities
(Automatically discovered)
Kleinberg et al

Networked or linked structures are ubiquitous

Use Case 3: Traveling Salesman Problem (TSP)

Find a closed tour of minimum length visiting all the cities.



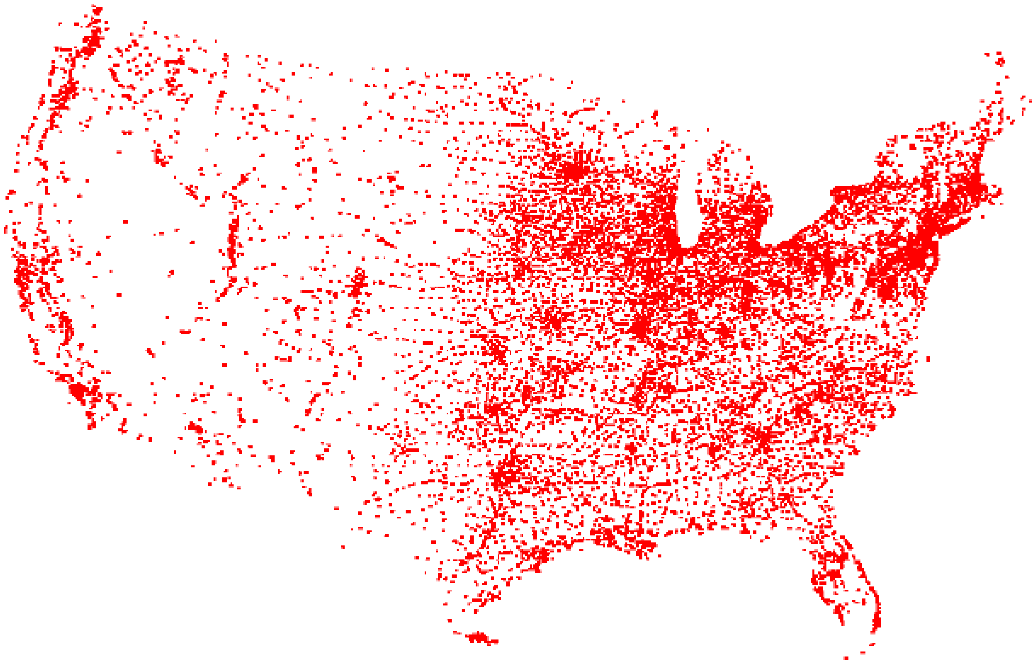
TSP → lots of applications:

Transportation related: scheduling deliveries

Many others: e.g., Scheduling of a machine to drill holes in a circuit board ;
Genome sequencing; etc

Use Case 3: Traveling Salesman Problem (TSP)

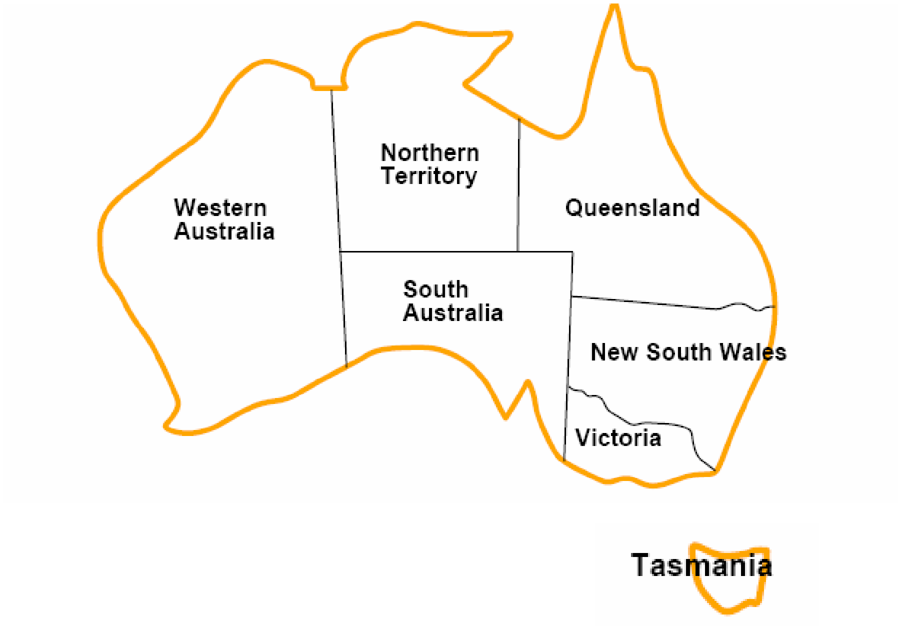
13,509 cities in the US

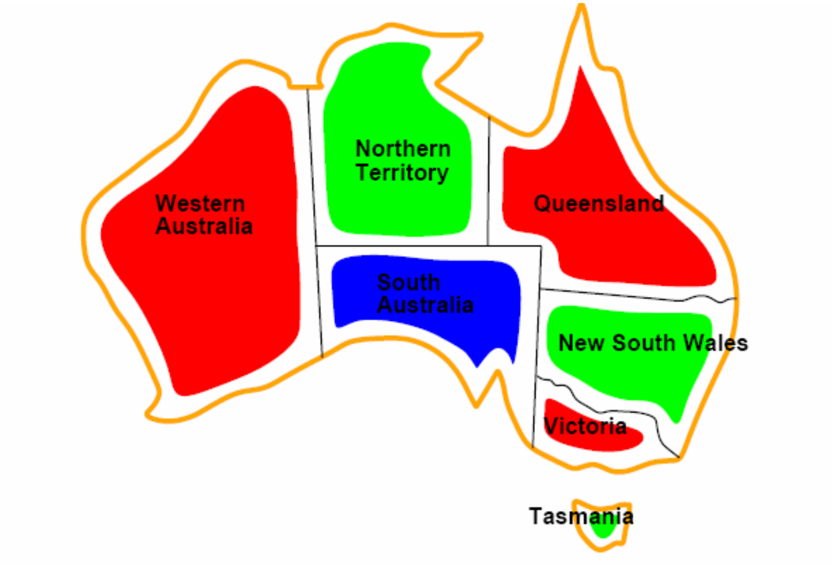


13508!= 1.4759774188460148199751342753208e+49936

The optimal tour!

Use Case 4 : Coloring a Map





How to color this map so that no two adjacent regions have the same color?

Logic

- Crucial for mathematical reasoning
- Important for program design

- (Propositional)Logic is a system based on propositions

- A proposition is a (declarative) statement that is either true or false <u>(not both)</u>

- We say that the truth value of a proposition is either true (T) or false (F)
- Corresponds to 1 and 0 in digital circuits

Example :1

“$y > 5$”

Is this a statement?	yes
Is this a proposition?	no

Its truth value depends on the value of y, but this value is not specified.

We call this type of statement a propositional function or open sentence.

Argument

- An argument is a sequence of propositions. The final proposition is called the conclusion of the argument while the other propositions are called the premises or hypotheses of the argument.
- An argument is valid whenever the truth of all its premises implies the truth of its conclusion.

Example 2

- “Please do not fall asleep.”

Is this a statement? no

It’s a request.

Is this a proposition? no

Only statements can be propositions.

Example 3

- “$x < y$ if and only if $y > x$.”

Is this a statement? yes

Is this a proposition? yes

… because its truth value does not depend on specific values of x and y.

What is the truth value of the proposition? true

Logical Operators (Connectives)

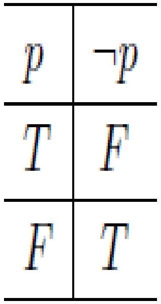
We will examine the following logical operators:

- Negation (NOT, $\neg$)
- Conjunction (AND, $\wedge$)
- Disjunction (OR, $\vee$)
- Exclusive-or (XOR, $\oplus$)
- Implication (if – then, $\rightarrow$)
- Biconditional (if and only if, $\leftrightarrow$)

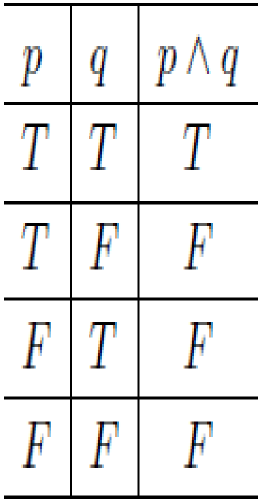
- Truth tables can be used to show how these operators can combine propositions to compound propositions.

Convention

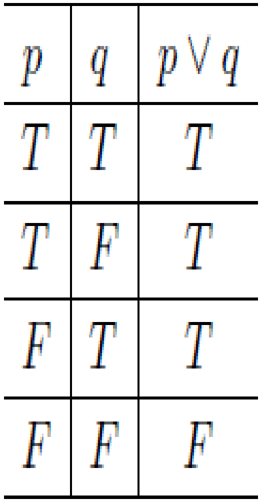
- Unary Operator, Symbol: ¬ Negation (NOT)
- Binary Operator, Symbol: ∧ Conjunction (AND)
- Binary Operator, Symbol: ∨ Disjunction (OR)
- Binary Operator, Symbol: →Implication (if - then)
- Binary Operator, Symbol: ↔ Biconditional (if and only if)



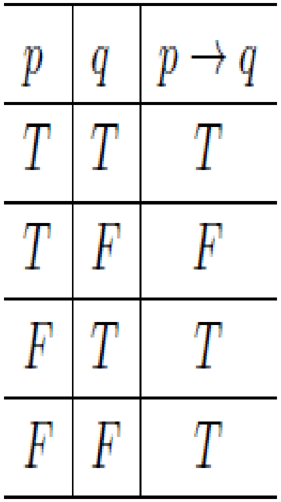
p	$\neg p$
T	F
F	T



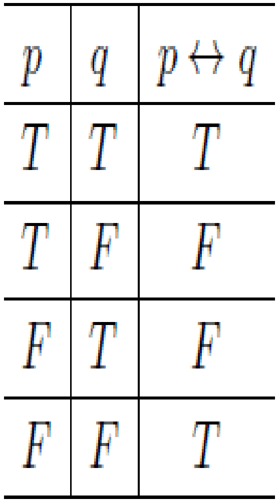
p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F



p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F



p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T



p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Laws

Law	Name
$P \lor \neg P \equiv T$	Excluded middle law
$P \land \neg P \equiv F$	Contradiction law
$P \lor F \equiv P$ $P \land T \equiv P$	Identity laws
$P \lor T \equiv T$ $P \land F \equiv F$	Domination laws
$P \lor P \equiv P$ $P \land P \equiv P$	Idempotent laws
$\neg(\neg P) \equiv P$	Double-negation law

LAWS

Law	Name
$P \vee Q \equiv Q \vee P$ $P \wedge Q \equiv Q \wedge P$	Commutative laws
$(P \vee Q) \vee R \equiv P \vee (Q \vee R)$ $(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$	Associative laws
$(P \vee Q) \wedge (P \vee R) \equiv P \vee (Q \wedge R)$ $(P \wedge Q) \vee (P \wedge R) \equiv P \wedge (Q \vee R)$	Distributive laws
$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$ $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$	De Morgan’s laws
$P \vee (P \wedge Q) \equiv P$ $P \wedge (P \vee Q) \equiv P$	Absorption laws

We can prove all laws by truth tables

P	Q	¬	(P ∧ Q)	⇔	¬P	∨	¬Q
T	T	F	T	T	F	F	F
T	F	T	F	T	F	T	T
F	T	T	F	T	T	T	F
F	F	T	F	T	T	T	T

NOTE:

Please try by yourself for others

In general, 2^n rows are required if a compound proposition involves n propositional variables to get the combination of all truth values.

Exclusive Or (XOR)

- Binary Operator, Symbol: $\oplus$

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

Have a fun!!!

- Example:
  - Let p be the statement “Subrata learns discrete mathematics.” and q the statement “Subrata will find a good job.” Express the statement $p \longrightarrow q$ as a statement in English.

Solution: Any of the following -

“If Subrata learns discrete mathematics, then he will find a good job.

“Subrata will find a good job when he learns discrete mathematics.”

“For Subrata to get a good job, it is sufficient for him to learn discrete mathematics.”

“Subrata will find a good job unless he does not learn discrete mathematics.”

Cont..

- How can this English sentence be translated into a logical expression?

"You cannot ride the roller coaster if you are under 5 feet tall unless you are older than 20 years old."

Solution: Let *q*, *r*, and *s* represent "You can ride the roller coaster," "You are under 5 feet tall," and "You are older than 20 years old." The sentence can be translated into:

$$(r \wedge \neg s) \rightarrow \neg q.$$

Example: 4

Is $p \to q \equiv \neg q \to \neg p$?

Ans: We construct a truth table as follows.

p	q	$p \to q$	$\neg q \to \neg p$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

Since in each row of the truth table, the truth values of the two formulas match, they are equivalent.

Example: 5

Show that

$$\neg(p \leftrightarrow q) \equiv \neg p \leftrightarrow q$$

$$\begin{aligned}
\neg(p \leftrightarrow q) &\equiv \neg\big((p \wedge q) \vee (\neg p \wedge \neg q)\big) && \text{Biconditional} \\
&\equiv \neg(p \wedge q) \wedge \neg(\neg p \wedge \neg q) && \text{De Morgan} \\
&\equiv (\neg p \vee \neg q) \wedge (p \vee q) && \text{De Morgan, Double negation} \\
&\equiv (\neg p \wedge p) \vee (\neg p \wedge q) \vee (\neg q \wedge p) \vee (\neg q \wedge q) && \text{Distributivity} \\
&\equiv (\neg p \wedge q) \vee (\neg q \wedge p) && \text{Constants} \\
&\equiv (\neg p \wedge q) \vee (\neg\neg p \wedge \neg q) && \text{Double negation} \\
&\equiv \neg p \leftrightarrow q && \text{Biconditional}
\end{aligned}$$

Example: 6

Show that

$$p \to q \equiv p \leftrightarrow p \wedge q$$

$$\begin{aligned}
p \leftrightarrow p \wedge q &\equiv \big(\neg p \vee (p \wedge q)\big) \wedge \big(p \vee \neg(p \wedge q)\big) && \text{Biconditional} \\
&\equiv \big(\neg p \vee (p \wedge q)\big) \wedge \big(p \vee (\neg p \vee \neg q)\big) && \text{De Morgan} \\
&\equiv (\neg p \vee p) \wedge (\neg p \vee q) \wedge \big(p \vee (\neg p \vee \neg q)\big) && \text{Distributivity} \\
&\equiv \neg p \vee q && \text{Constants} \\
&\equiv p \to q && \text{Implication}
\end{aligned}$$

Example: 7

- To take discrete mathematics, you must have taken calculus or a course in computer science.
  - P: take discrete mathematics
  - Q: take calculus
  - R: take a course in computer science
- $P \rightarrow \mathbf{Q} \vee R$

Example: 8

- School is closed if more than 50 C temperature or humidity is above 99.
  - P: School is closed
  - Q: 50 C temperature
  - R: humidity is above 99
  - $Q \wedge R \rightarrow P$

Equivalent Statements

P	Q	¬(P∧Q)	(¬P)∨(¬Q)	¬(P∧Q)↔(¬P)∨(¬Q)
T	T	F	F	T
T	F	T	T	T
F	T	T	T	T
F	F	T	T	T

- The statements ¬(P∧Q) and (¬P) ∨ (¬Q) are logically equivalent, since they have the same truth table, or put it in another way, ¬(P∧Q) ↔(¬P) ∨ (¬Q) is always true.

Tautologies and Contradictions

- A tautology is a statement that is always true.
- Examples:
  - $R \vee (\neg R)$
  - $\neg(P \wedge Q) \leftrightarrow (\neg P) \vee (\neg\ Q)$
- A contradiction is a statement that is always false.
- Examples:
  - $R \wedge (\neg R)$
  - $\neg(\neg(P \wedge Q) \leftrightarrow (\neg P) \vee (\neg Q))$
- The negation of any tautology is a contradiction, and the negation of any contradiction is a tautology.

Normal Forms

Normal forms are standard forms, sometimes called canonical or accepted forms.

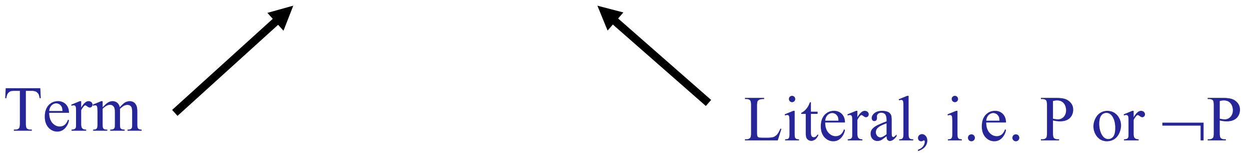
A logical expression is said to be in *disjunctive normal form (DNF)* if it is written as a disjunction, in which all *terms* are conjunctions of *literals*.

Similarly, a logical expression is said to be in *conjunctive normal form (CNF)* if it is written as a conjunction of disjunctions of literals.

DNF and CNF

- Disjunctive Normal Form (DNF)

  $(.. \wedge .. \wedge ..) \vee (.. \wedge .. \wedge ..) \vee \ldots \vee (.. \wedge ..)$



Term

Literal, i.e. P or ¬P

Examples: $(P \wedge Q) \vee (P \wedge \neg Q)$

$P \vee (Q \wedge R)$

- Conjunctive Normal Form (CNF)

  $(.. \vee .. \vee ..) \wedge (.. \vee .. \vee ..) \wedge \ldots \wedge (.. \vee ..)$

Examples: $(P \vee Q) \wedge (P \vee \neg Q)$

$P \wedge (Q \vee R)$

Converting Expressions to DNF or CNF

The following procedure converts an expression to DNF or CNF:

1. Remove all ⇒ and ⇔.
2. Move ¬ inside. (Use De Morgan's law.)
3. Use distributive laws to get proper form.

Simplify as you go. (e.g. double-neg., idemp., comm., assoc.)

Example: CNF-satisfaction

A conjunctive normal form (CNF) is a Boolean expression consisting of one or more disjunctive formulas connected by an AND symbol ($\wedge$). A disjunctive formula is a collection of one or more (positive and negative) literals connected by an OR symbol ($\vee$).

Example:

$$(a) \wedge (\neg a \vee \neg b \vee c \vee d) \wedge (\neg c \vee \neg d) \wedge (\neg d)$$

Problem (CNF-satisfaction): Give an algorithm that receives as input a CNF *form* and returns Boolean assignments for each literal in *form* such that *form* is true

Method to construct DNF

- Construct a truth table for the proposition.
- Use the rows of the truth table where the proposition is True to construct minterms
  - If the variable is true, use the propositional variable in the minterm
  - If a variable is false, use the negation of the variable in the minterm
- Connect the minterms with $\vee$’s.

How to find the DNF of $(p \vee q)\rightarrow\neg r$

p	q	r	$(p \vee q)$	$\neg r$	$(p \vee q)\rightarrow\neg r$
T	T	T	T	F	F
T	*T*	*F*	*T*	*T*	*T*
T	F	T	T	F	F
T	*F*	*F*	*T*	*T*	*T*
F	T	T	T	F	F
F	*T*	*F*	*T*	*T*	*T*
F	*F*	*T*	*F*	*F*	*T*
F	*F*	*F*	*F*	*T*	*T*

There are five sets of input that make the statement true. Therefore there are five minterms.

Cont.

p	q	r	$(p \vee q)$	$\neg r$	$(p \vee q) \rightarrow \neg r$
T	T	T	T	F	F
T	*T*	*F*	*T*	*T*	*T*
T	F	T	T	F	F
T	*F*	*F*	*T*	*T*	*T*
F	T	T	T	F	F
F	*T*	*F*	*T*	*T*	*T*
F	*F*	*T*	*F*	*F*	*T*
F	*F*	*F*	*F*	*T*	*T*

From the truth table we can set up the DNF

$$(p \vee q) \rightarrow \neg r \Leftrightarrow (p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge \neg r) \vee$$
$$(\neg p \wedge q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r)$$

Exercises

- Show that $P \to Q \equiv \neg P \lor Q$: by truth table
- Show that $(P \to Q) \land (P \to R) \equiv P \to (Q \land R)$
- Write a short note on **Tseitin's encoding**