

Discrete Structures, CSCI-150.

Information

Mon, Tue, Wed, Thr, 11:40 am – 12:50 pm. North C111.

Instructor: Alexey Nikolaev.

Website: <http://a-nikolaev.github.io/ds/>

Grading policy

No late homeworks accepted.

Homeworks twice a week. (Due Thursday and Monday).

Final grade:

HWs: 25%

Exams: 75%

There are two midterms and the final. When computing the final grade, only two best exams out of three are counted, and the worst is dropped.

Information

Propositions

Operators

Fighting Complexity

Equivalence

Course content

Propositional Logic. Operators. Truth tables. Logical equivalence. Rules of inference. Satisfiability. Predicates and quantifiers. Proofs.

Counting. Sum and product rules. Pigeonhole principle. Permutations, $n!$ Binomial coefficients, n choose k . Selection with replacement.

Induction. Hanoi towers. Summation of series. Recurrence. Fibonacci numbers. Catalan numbers. Solving linear recurrence.

Number theory. Divisibility and primes. Modulo-arithmetics. GCD and Euclid's algorithm. Cryptography. RSA.

Sets. Operations, empty set, singleton set, powerset. Natural, rational, real numbers. Diagonalization. Relations and Functions. Counting and Bijection. Partial orders.

Graphs. Bridges of Königsberg. Eulerian and Hamiltonian cycles. Trees, spanning trees. Huffman coding.

Probability. Bernoulli Trials. Random variables. Expected value.

Information

Propositions

Operators

Fighting Complexity

Equivalence

Literature

Information

Propositions

Operators

Fighting Complexity

Equivalence

Primary books:

- *Rosen*
“Discrete Mathematics and its Applications” edition 6 or 7.
(you can find a used or new 6th edition for less than \$40).
- *Lehman and Leighton*
Lecture notes “Mathematics for Computer Science” (2004).
(free, but this is not a complete textbook)

Our first object

Information

Propositions

Operators

Fighting Complexity

Equivalence

Something that is either

true or *false*

Propositions

Information

Propositions

Operators

Fighting Complexity

Equivalence

Def. A **proposition** is a declarative sentence that is either **true** or **false**, but not both.

A good test for a proposition is to ask *“Is it true that ...?”* If that makes sense, it is a proposition.

- One plus two equals three.
- Washington, D.C., is the capital of the US.
- The Moon is a satellite of the Earth.
- Albany is the capital of Canada.
- The Sun is a planet.

Propositions

Information

Propositions

Operators

Fighting Complexity

Equivalence

Def. A **proposition** is a declarative sentence that is either **true** or **false**, but not both.

A good test for a proposition is to ask “*Is it true that ... ?*” If that makes sense, it is a proposition.

- One plus two equals three. *true* ✓
- Washington, D.C., is the capital of the US. *true* ✓
- The Moon is a satellite of the Earth. *true* ✓
- Albany is the capital of Canada. *false* ✓
- The Sun is a planet. *false* ✓ all are propositions

Propositions

Information

Propositions

Operators

Fighting Complexity

Equivalence

Def. A **proposition** is a declarative sentence that is either **true** or **false**, but not both.

A good test for a proposition is to ask “*Is it true that ... ?*” If that makes sense, it is a proposition.

- Three plus four.
- Consider these sentences.
- Does anyone have any questions?
- The largest planet in the Solar System.
- n in a prime number.

Propositions

Information

Propositions

Operators

Fighting Complexity

Equivalence

Def. A **proposition** is a declarative sentence that is either **true** or **false**, but not both.

A good test for a proposition is to ask “*Is it true that ... ?*” If that makes sense, it is a proposition.

- Three plus four. ✗ neither one is a proposition
- Consider these sentences. ✗
- Does anyone have any questions? ✗
- The largest planet in the Solar System. ✗
- n in a prime number. ✗

Propositions

Information

Propositions

Operators

Fighting Complexity

Equivalence

Instead of writing sentences, we will abbreviate them by using *propositional variables*.

It is standard practice to use the lower-case letters: p, q, r, \dots

Then, if

$$p = \text{"It is raining"},$$
$$q = \text{"I have an umbrella"},$$

we can construct *compound propositions* using logical operators:

$$p \text{ and } q = \text{"It is raining, and I have an umbrella"}.$$
$$\text{not } q = \text{"I don't have an umbrella"}.$$

Logical Operators

Information

Propositions

Operators

Fighting Complexity

Equivalence

And (called Conjunction)

p and q

$p \wedge q$ is true when both p and q are true, otherwise false.

Or (called Disjunction)

p or q

$p \vee q$ is true when p or q or both are true, otherwise false.

Negation

not p

$\neg p$ is true when p is false, otherwise false.

Truth tables

Information

Propositions

Operators

Fighting Complexity

Equivalence

And (Conjunction)

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

Or (Disjunction)

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

Negation

p	$\neg p$
T	F
F	T

Think of the truth tables as our ultimate definition of the logical connectives (operators).

More Operators. Implication

Information

Propositions

Operators

Fighting Complexity

Equivalence

Implication

if p then q

$p \rightarrow q$ is true if whenever p is true, so is q , otherwise false.

Truth table:

*An implication is true when
the if-part is false or the then-part is true.*

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

So, $p \rightarrow q$ is equivalent to $(\neg p) \vee q$.

“I need an umbrella, if it’s raining”.

“If the Earth is flat, my brother is a physicist”.

More Operators. Implication

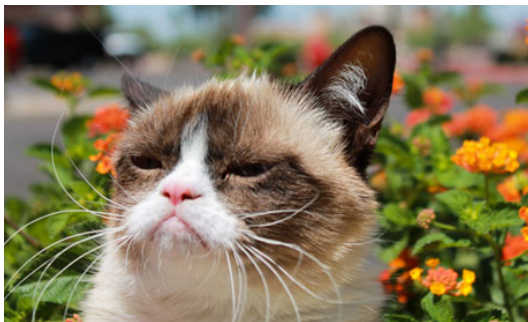
Information

Propositions

Operators

Fighting Complexity

Equivalence



1. *"If he is hungry, he is grumpy".*
2. *"He is hungry".*

Is he grumpy?

More Operators. Implication

Information

Propositions

Operators

Fighting Complexity

Equivalence

“If he is hungry, he is grumpy”.

$$h \rightarrow g = T$$

If an implication is true, we can make conclusions about g , if we know h .

We know that *“he is hungry”*,

$$h = T,$$

it's only possible what we is grumpy

$$g = T.$$

h	g	$h \rightarrow g$
T	T	T
F	T	T
T	F	F
F	F	T

More Operators. Implication

Information

Propositions

Operators

Fighting Complexity

Equivalence



1. *"If he is hungry, he is grumpy".*
2. *"He is not hungry".*

Is he happy?

More Operators. Implication

Information

Propositions

Operators

Fighting Complexity

Equivalence

“If he is hungry, he is grumpy”.

$$h \rightarrow g = T$$

If an implication is true, we can make conclusions about g , if we know h .

If we know that *“he is not hungry”*,

$$h = F,$$

then

g can be T or F .

h	g	$h \rightarrow g$
T	T	T
F	T	T
T	F	F
F	F	T

More Operators. Implication

Information

Propositions

Operators

Fighting Complexity

Equivalence

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



Big Al told these guys that dogs can't look up.

Their thoughts:

$p \rightarrow q$ = "If dogs can look up, Big Al is a liar".

p = "Dogs can look up"

q = "Big Al is a liar"

$(\neg p) \vee q$ = "Dogs can't look up, or Big Al is a liar".

$p \rightarrow q$ is equivalent to $(\neg p) \vee q$

More Operators. Biconditional

Biconditional

p *if and only if* q

$p \leftrightarrow q$

is true when p and q have the same truth values, otherwise false.

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

Often, “if and only if” is abbreviated to *iff*:

p *iff* q

“You can take the flight if and only if you buy a ticket.”

Theorems are often formulated as implications or biconditionals.

Information

Propositions

Operators

Fighting Complexity

Equivalence

Combined truth tables for connectives \neg , \wedge , \vee , \rightarrow , and \leftrightarrow

Information

Propositions

Operators

Fighting Complexity

Equivalence

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	F	T	T	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
F	F	T	F	F	T	T

Complex compound propositions?

Information

Propositions

Operators

Fighting Complexity

Equivalence

Let's take a complex compound proposition:

$$q \vee ((\neg q) \wedge r)$$

q or ((not q) and r)

q	r	\dots
T	T	\dots
F	T	\dots
T	F	\dots
F	F	\dots

Complex compound propositions?

Information

Propositions

Operators

Fighting Complexity

Equivalence

Let's take a complex compound proposition:

$$q \vee ((\neg q) \wedge r)$$

q or ((not q) and r)

q	r	$\neg q$	\dots
T	T	F	\dots
F	T	T	\dots
T	F	F	\dots
F	F	T	\dots

Complex compound propositions?

Information

Propositions

Operators

Fighting Complexity

Equivalence

Let's take a complex compound proposition:

$$q \vee ((\neg q) \wedge r)$$

q or ((not q) and r)

q	r	$\neg q$	$(\neg q) \wedge r$	\dots
T	T	F	F	\dots
F	T	T	T	\dots
T	F	F	F	\dots
F	F	T	F	\dots

Complex compound propositions?

Information

Propositions

Operators

Fighting Complexity

Equivalence

Let's take a complex compound proposition:

$$q \vee ((\neg q) \wedge r)$$

q or ((not q) and r)

q	r	$\neg q$	$(\neg q) \wedge r$	$q \vee ((\neg q) \wedge r)$
T	T	F	F	T
F	T	T	T	T
T	F	F	F	T
F	F	T	F	F

Complex compound propositions?

The number of rows in the truth table of a compound proposition is equal to 2^n , where n is the number of used propositional variables.

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

p	q	r	...
T	T	T	...
F	T	T	...
T	F	T	...
F	F	T	...
T	T	F	...
F	T	F	...
T	F	F	...
F	F	F	...

Each of the three variables can take two possible values, so the system has $2 \cdot 2 \cdot 2 = 8$ possible states.

Information

Propositions

Operators

Fighting Complexity

Equivalence

Equivalence

Information

Propositions

Operators

Fighting Complexity

Equivalence

Two compound propositions are equivalent if they have the same truth values for all possible cases (have the same truth tables).

q	r	$q \vee ((\neg q) \wedge r)$	$q \vee r$
T	T	T	T
F	T	T	T
T	F	T	T
F	F	F	F

Therefore, these two propositions are logically equivalent!

We write it as follows

$$q \vee ((\neg q) \wedge r) \equiv q \vee r$$

Note that the statement of the equivalence of two compound propositions, $a \equiv b$, is not a proposition itself.

Equivalent formulae

Information

Propositions

Operators

Fighting Complexity

Equivalence

$$(a \wedge b) \equiv (b \wedge a) \quad \text{commutativity of } \wedge$$

$$(a \vee b) \equiv (b \vee a) \quad \text{commutativity of } \vee$$

$$((a \wedge b) \wedge c) \equiv (a \wedge (b \wedge c)) \quad \text{associativity of } \wedge$$

$$((a \vee b) \vee c) \equiv (a \vee (b \vee c)) \quad \text{associativity of } \vee$$

$$\neg(\neg a) \equiv a \quad \text{double-negation elimination}$$

$$(a \rightarrow b) \equiv (\neg b \rightarrow \neg a) \quad \text{contraposition}$$

$$(a \rightarrow b) \equiv (\neg a \vee b) \quad \text{implication elimination}$$

$$(a \leftrightarrow b) \equiv (a \rightarrow b) \wedge (b \rightarrow a) \quad \text{biconditional elimination}$$

$$\neg(a \wedge b) \equiv (\neg a \vee \neg b) \quad \text{De Morgan's Law}$$

$$\neg(a \vee b) \equiv (\neg a \wedge \neg b) \quad \text{De Morgan's Law}$$

$$(a \wedge (b \vee c)) \equiv (a \wedge b) \vee (a \wedge c) \quad \text{distributivity of } \wedge \text{ over } \vee$$

$$(a \vee (b \wedge c)) \equiv (a \vee b) \wedge (a \vee c) \quad \text{distributivity of } \vee \text{ over } \wedge$$

Equivalent formulae

Information

Propositions

Operators

Fighting Complexity

Equivalence

If **True** is a compound propositions that is always true, and **False** is a proposition that is always false:

$$a \wedge \mathbf{True} \equiv a \quad \text{identity}$$

$$a \vee \mathbf{False} \equiv a \quad \text{identity}$$

$$a \vee \mathbf{True} \equiv \mathbf{True} \quad \text{domination}$$

$$a \wedge \mathbf{False} \equiv \mathbf{False} \quad \text{domination}$$

$$a \vee \neg a \equiv \mathbf{True} \quad \text{negation}$$

$$a \wedge \neg a \equiv \mathbf{False} \quad \text{negation}$$

Examples of such always true compound propositions:

True : $p \vee \neg p$, $p \leftrightarrow p$, $p \rightarrow (p \vee q)$, etc.

False : ... find a few examples of always false propositions.