csci54 – discrete math & functional programming
propositional logic

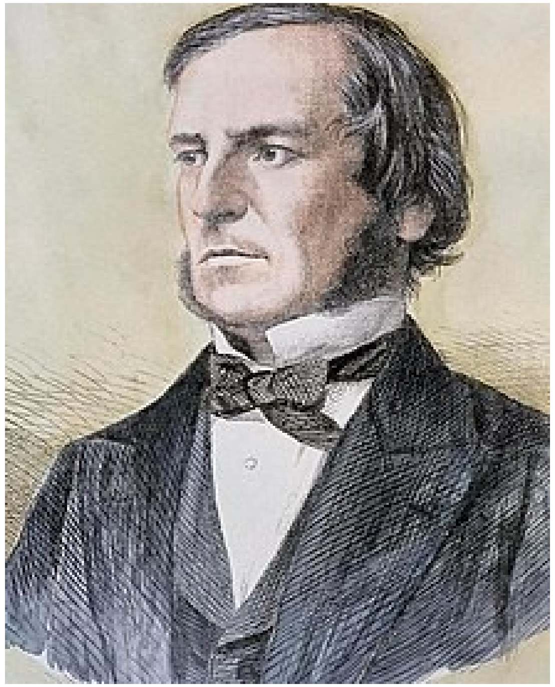
Simplify each of the following Haskell expressions:

(a) `a && not a`

(b) `a || (not a && b)`

(c)

```
(not a || b) && (not b || c) &&
  (not c || not a) && (not c || not b)
```



George
Boole
1815-1864

On "True" and "False"

- logic is the study of valid reasoning

- The starting point:

  A proposition is a statement that is either True or False.

- What are examples of propositions that are True? False? Unknown?

On propositional logic

- the study of propositions: how to formulate, evaluate, manipulate

- atomic proposition: a proposition that is conceptually indivisible

- compound proposition: a proposition that is build up out of conceptually simpler propositions
  - How?

Creating compound propositions

- We can create more complex propositional statements using <u>logical connectives</u>
  - negation (not, ¬, ~)
  - conjunction (and, ∧)
  - disjunction (or, ∨)
  - implication (implies, ⇒, →)

Precedence rules:

- negation binds most tightly
- then conjunction
- then disjunction
- then implication

implication is right-associative

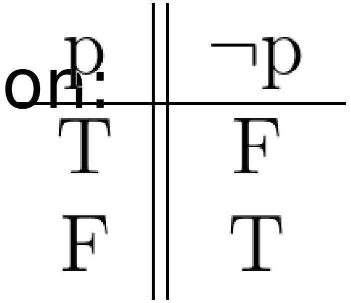
- In particular, a well-formed pro
  defined as:

$$\phi ::= T|F|(\neg\phi)|(\phi \wedge \phi)|(\phi \vee \phi)|(\phi \Rightarrow \phi)$$

Evaluating compound propositional statements

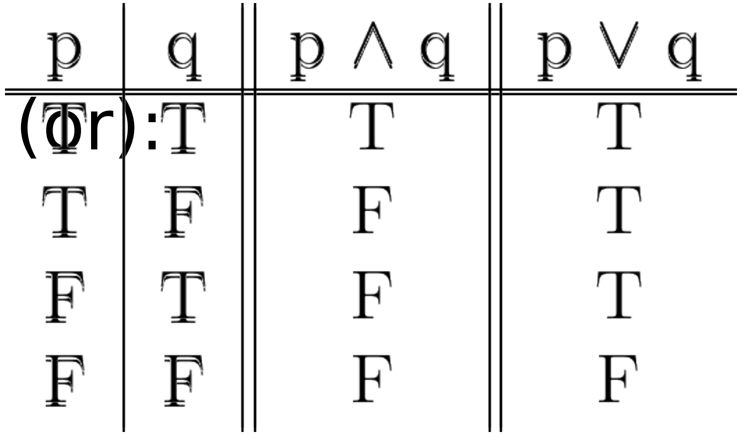
- Convenient to use a truth table to display the relationships between truth values of different propositions

- Truth table for negation:



p	¬p
T	F
F	T

- For conjunction (and) and disjunction (or):

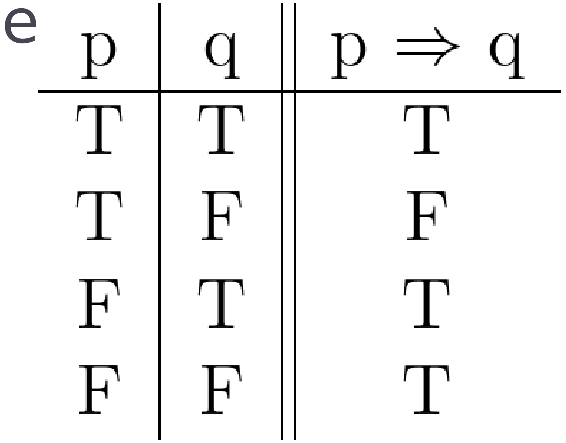


p	q	p ∧ q	p ∨ q
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

$$\phi ::= T|F|(\neg\phi)|(\phi \wedge \phi)|(\phi \vee \phi)|(\phi \Rightarrow \phi)$$

Implication

- What does it mean to say "p implies q"?
  - p q is true if q is true or p is false



p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

- What is the truth value of each of the following statements?
  - 1 + 1 = 2 implies that 2 + 3 = 5
  - 1 + 1 = 2 implies that 2 + 3 = 6
  - 1 + 1 = 3 implies that 2 + 3 = 5
  - 1 + 1 = 3 implies that 2 + 3 = 6

A little more on implications

- p q
  - “if p, then q”
  - “p implies q”
  - “p only if q”
  - “q whenever p”
  - “q, if p"
  - “q is necessary for p”
  - “p is sufficient for q”

- Bidirectional implication p q
  - "p if and only if q", "p iff q"
  - True only when p and q have same truth value: either both true or both false.

Example

- "Since Sandra is wearing a soccer jersey, she must be a soccer player."

- This compound proposition is composed of 2 atomic propositions:
  - (1) = Sandra is wearing a soccer jersey
  - (2) = Sandra is a soccer player

- The compound proposition can written as:
  - (1) ⇔ (2)

inspired by:
https://philosophy.lander.edu/logic/diagram_quiz.html

Passwords

- "A password is valid only if it is at least 8 characters long, is not one that you have used previously, and contains at least 2 of the following: a number, a lowercase character, an uppercase character."

- This is a compound proposition that is composed of how many atomic propositions?

- What are the 6 atomic propositions?
- How can you write the compound proposition in terms of the atomic propositions?

categorizing well-formed formulas (wff)

- A formula in propositional logic is one of:
  - tautology (valid): if it evalutes to T in all cases
  - satisfiable: evaluates to T in some cases
  - contingency (falsifiable): evaluates to F in some cases
  - contradiction (unsatisfiable): evaluates to F in all cases

- Consider the following formula:

$$(p \lor q) \Rightarrow (\neg p \land \neg q)$$

- Which of the following describes the formula: tautology, satisfiable, contingency, contradiction? Why?

a collection of tautologies

$(p \Rightarrow q) \wedge p \Rightarrow q$	Modus Ponens
$(p \Rightarrow q) \wedge \neg q \Rightarrow \neg p$	Modus Tollens
$p \vee \neg p$	Law of the Excluded Middle
$p \Leftrightarrow \neg\neg p$	Double Negation
$p \Leftrightarrow p$	
$p \Rightarrow p \vee q$	
$p \wedge q \Rightarrow p$	

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

logical equivalence

- Two propositions are logically equivalent (written) if they have exactly identical truth tables (i.e. their truth values are the same under every truth assignment)

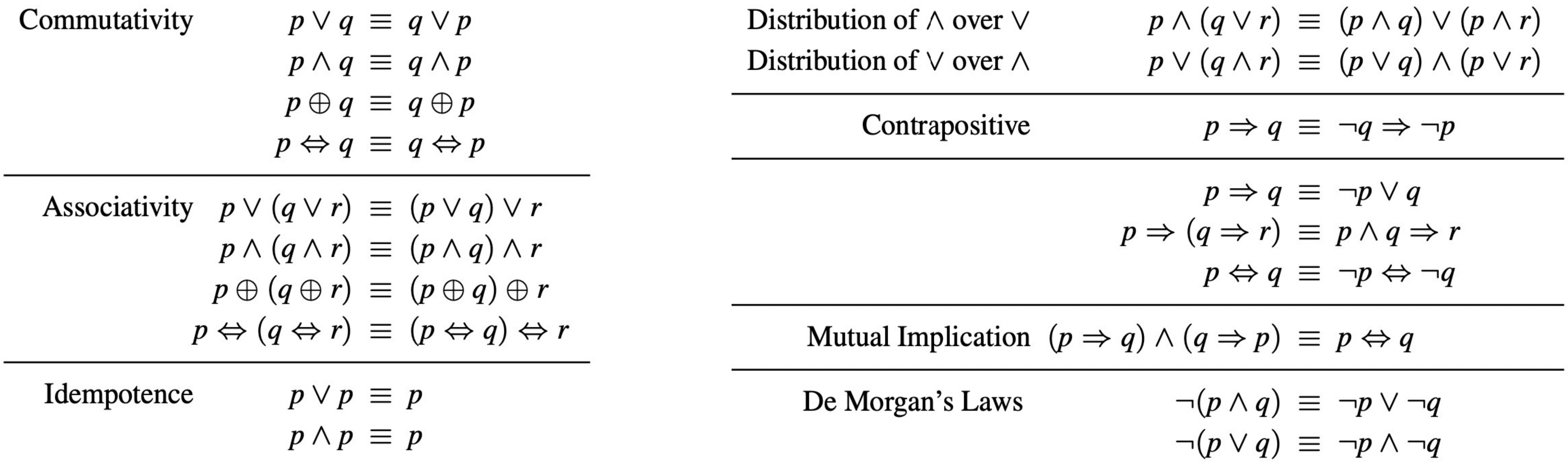
Simplify each of the following Haskell expressions:

(a)
```
a && not a
```

(b)
```
a || (not a && b)
```

(c)
```
(not a || b) && (not b || c) &&
  (not c || not a) && (not c || not b)
```

some logically equivalent propositions



Commutativity	$p \vee q \equiv q \vee p$
	$p \wedge q \equiv q \wedge p$
	$p \oplus q \equiv q \oplus p$
	$p \Leftrightarrow q \equiv q \Leftrightarrow p$
Associativity	$p \vee (q \vee r) \equiv (p \vee q) \vee r$
	$p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$
	$p \oplus (q \oplus r) \equiv (p \oplus q) \oplus r$
	$p \Leftrightarrow (q \Leftrightarrow r) \equiv (p \Leftrightarrow q) \Leftrightarrow r$
Idempotence	$p \vee p \equiv p$
	$p \wedge p \equiv p$

Distribution of $\wedge$ over $\vee$	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
Distribution of $\vee$ over $\wedge$	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
Contrapositive	$p \Rightarrow q \equiv \neg q \Rightarrow \neg p$
	$p \Rightarrow q \equiv \neg p \vee q$
	$p \Rightarrow (q \Rightarrow r) \equiv p \wedge q \Rightarrow r$
	$p \Leftrightarrow q \equiv \neg p \Leftrightarrow \neg q$
Mutual Implication	$(p \Rightarrow q) \wedge (q \Rightarrow p) \equiv p \Leftrightarrow q$
De Morgan's Laws	$\neg(p \wedge q) \equiv \neg p \vee \neg q$
	$\neg(p \vee q) \equiv \neg p \wedge \neg q$

$$\boxed{(\neg a \vee b) \wedge (\neg b \vee c) \wedge (\neg c \vee \neg a) \wedge (\neg c \vee \neg b)}$$