Review of Propositional Logic (Continue)

- **Hierarchy** of propositional operators: when multiple operators appear in the same proposition, we calculate in the following order: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.
  - If we want to calculate an operator of lower hierarchy first, we need to use parenthesis.

1. Remove redundant parenthesis in the following statements:
   a. $\big(p \rightarrow ((\neg q) \wedge r)\big)$ $\qquad p \rightarrow \neg q \wedge r$
   b. $\big((p \vee q) \wedge r\big) \rightarrow \big(\neg(s \vee p)\big)$ $\qquad (p \vee q) \wedge r \rightarrow \neg(s \vee p)$

- **Associativity** for propositional operators:
  - $\wedge$ and $\vee$ are associative, which means $(p \wedge q) \wedge r$ and $p \wedge (q \wedge r)$ are *logically equivalent*. Formally we consider them **left associative**, in other words we define: $p \vee q \vee r$ means $(p \vee q) \vee r$.
  - Implication is **right associative**: $p \rightarrow q \rightarrow r$ means $\big(p \rightarrow (q \rightarrow r)\big)$.
  - Biconditional is also **right associative**: $p \leftrightarrow q \leftrightarrow r$ means $\big(p \leftrightarrow (q \leftrightarrow r)\big)$.

2. What is the truth value of $T \leftrightarrow F \leftrightarrow F$?
   $T \leftrightarrow F \leftrightarrow F$ means $T \leftrightarrow (F \leftrightarrow F)$ which is $True$.

- **Truth Table** is used to capture the truth values of a compound proposition. Here we show the truth values of some propositions.

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

p	$\neg p$
F	T
T	F

3. Create the truth table for proposition $\neg p \vee q$:

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

   From this example, we can see that $\neg p \vee q$ and $p \rightarrow q$ have the same truth value despite of the truth values of p and q.

Semantic Equality and Syntactic Equality

4. True or False:
   a. $2 + 2 = 4$ $\qquad True$
   b. $2 + 2 \equiv 4$ $\qquad False$

- Here, the difference between the two expressions is the semantic equality and syntactic equality.
- "Syntactic" means something about grammar. Two expressions need to be textually identical to be **syntactically equal**.
  - For example: $p \to q \to r \equiv p \to (q \to r)$, with or without parenthesis, these two expressions are evaluated in the same procedure.

- "Semantic" means something about meaning. Both $2 + 2$ and 4 has numerical value 4, so they are **semantically equal or logically equivalent**.
  - In logic, we usually use " $\Leftrightarrow$ " instead of " $=$ " to represent semantical equality. For example, we have $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$, and we can say these two propositions are semantically equal or logically equivalent.
  - " $\leftrightarrow$ " is an operation, when we say "$p \leftrightarrow q$" we focus on the biconditional operation itself and the result can be either true or false. " $\Leftrightarrow$ " shows the result, when we say "$p \Leftrightarrow q$" we focus on that p and q are semantically equal. In other words, $p \Leftrightarrow q$ means $p \leftrightarrow q \Leftrightarrow T$.

- It is easy to see that syntactic equality *logically implies* semantic equality: It is always true that if $A \equiv B$, then $A = B$; which is denoted as $(A \equiv B) \Rightarrow (A = B)$.
  - The difference between " $\to$ " and " $\Rightarrow$ " is like the difference between " $\leftrightarrow$ " and " $\Leftrightarrow$ ". If $p \to q \Leftrightarrow T$, then we can say p **logically implies** q, which is denoted as $p \Rightarrow q$.
  - As an aside, technically speaking, "If p, then q" we use in real life means $p \Rightarrow q$ instead of $p \to q$; because it is hidden in the sentence that we are saying something true: "(It is true that) if p, then q."

States in Propositional Logic

- A **state,** in general, means a collection of variables with their current values (or a collection of **bindings**):
  - For example, $\{c_1 = 2,\ str_1 = "asdf"\}$ is a state, $c_1 = 2$ is a binding, $str_1 = "asdf"$ is another binding.
- We usually use letter σ to represent a state.
- Since each variable is bound to exactly one value, a state can also be considered as a function. For example, $\sigma = \{x = 5, p = T\}$, then $\sigma(x) = 5$ and $\sigma(p) = T$.

- In propositional logic, a state σ is a collection of proposition variables and their truth values.
  - For example, $\sigma_1 = \{p = T,\ q = F\}$ is state in propositional logic.

- A state is **ill-formed** if one of the follows is true:
  - There is more than one binding for the same variable. For example, $\{p = T,\ p = F\}$ is ill-formed.
  - There are bindings involved with more than a single variable. For example, $\{p \vee \neg p = T\},\ \{p \wedge q = F\}$ are ill-formed.

- A proposition p is **satisfied by** a state σ if it evaluates to $True$ in that state, which is denoted as $\sigma \vDash p$. On the other hand, if σ doesn't satisfy p, we have $\sigma \nvDash p$. For now, we can consider that $\sigma \nvDash p$ means that p evaluates to $False$ in σ (this definition will be updated in the future).
  - Note that, if σ is ill-formed then we cannot ask whether σ satisfies p.

5. True or False:
   a. $\{p = F\} \vDash \neg p$ — $True$
   b. $\{p = T, q = T, r = F\} \nvDash p \wedge \neg q$ — $True$
   c. $\emptyset \vDash T \wedge (F \to T)$ — $True$
   d. If $\sigma \vDash \neg p$, then $\sigma \nvDash p$ — $True$

e. Let σ and τ be two states, if $\sigma \vDash p$ and $\sigma \subseteq \tau$, then $\tau \vDash p$. $True$
f. $\{p = T\} \vDash p \vee q$ $True$
g. $\{p = T\} \vDash p \wedge q$ Cannot decide.

- In example 5.g, the state $\{p = T\}$ is not enough to decide whether the proposition $p \wedge q$'s truth value (because it contains more variables), in this case, we say the state $\{p = T\}$ is **not proper** for proposition $p \wedge q$. Otherwise, such as example 5.a, 5.b…, the state contains enough variables to decides the truth value of the proposition, then we say the state is **proper** for the proposition.

- In example 5.c, the proposition $T \wedge (F \rightarrow T)$ is always $True$ in any state, in this case, we can omit the state and denote it as $\vDash T \wedge (F \rightarrow T)$. In general, if a proposition p is $True$ in all states, then $\vDash p$ (read p is **valid**).
  - The opposite to $\vDash p$ is $\nvDash p$ (read p is **not valid**), which means p is not $True$ in all states; or in other words, there exists some states that makes p be $False$.

- We say proposition p is a **tautology** if $\vDash p$ (p is always $True$ or $p \Leftrightarrow T$), and we say p is a **contradiction** if $\vDash \neg p$ (p is always $False$ or $p \Leftrightarrow F$).
- If $\nvDash p$ and $\nvDash \neg p$ (p is neither always $True$ nor always $False$), then p is a **contingency**.

6. Let p be a proposition, is it true that it is either a tautology, or a contradiction, or a contingency? Yes.

7. Is $\neg(p \rightarrow q)$ a tautology, or a contradiction, or a contingency?
   - We can use a truth table for the proposition. If the column for the proposition is all T then it is a tautology, if it is all F then it is a contradiction, or else it is a contingency.

p	q	$p \rightarrow q$	$\neg(p \rightarrow q)$	$p \wedge \neg q$
F	F	T	F	F
F	T	T	F	F
T	F	F	T	T
T	T	T	F	F

<u>How to decide whether two propositions are logically equivalent?</u>

8. Is it true that $\neg(p \rightarrow q) \Leftrightarrow p \wedge \neg q$?
   - We can check whether these two propositions have the same truth value in all states. In other words, we check in a truth table that whether the columns for these two propositions have the same truth value in each row.

   - Since logical equivalency means semantical equivalency, so we can check whether $\vDash \neg(p \rightarrow q) \leftrightarrow p \wedge \neg q$.

- We can always find whether two propositions are logically equivalent or not using the above two ways. However, when we are given some complicated propositions, it is quite time consuming to create the truth table. Fortunately, we have a set of proven logic equivalencies and logic rules that we can use directly to prove the logic equivalency:

  - **Law of contrapositive:** $p \rightarrow q \Leftrightarrow \neg q \rightarrow \neg p$
  - **Definition of implication:** $p \rightarrow q \Leftrightarrow \neg p \vee q$
  - **Negation of implication:** $\neg(p \rightarrow q) \Leftrightarrow p \wedge \neg q$
  - **Definition of biconditional:** $(p \leftrightarrow q) \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$

  - **Commutativity:** $p \vee q \Leftrightarrow q \vee p$

$p \land q \Leftrightarrow q \land p$
$(p \leftrightarrow q) \Leftrightarrow (q \leftrightarrow p)$

- Associativity: $(p \lor q) \lor r \Leftrightarrow p \lor (q \lor r)$
  $(p \land q) \land r \Leftrightarrow p \land (q \land r)$
- Distributivity: $(p \lor q) \land r \Leftrightarrow (p \land r) \lor (q \land r)$
  $(p \land q) \lor r \Leftrightarrow (p \lor r) \land (q \lor r)$
- Identity: $p \land T \Leftrightarrow p$
  $p \lor F \Leftrightarrow p$
- Idempotency: $p \lor p \Leftrightarrow p$
  $p \land p \Leftrightarrow p$
- Domination: $p \lor T \Leftrightarrow T$
  $p \land F \Leftrightarrow F$
- Absurdity: $(F \to p) \Leftrightarrow T$
- Contradiction: $p \land \neg p \Leftrightarrow F$
- Excluded middle: $p \lor \neg p \Leftrightarrow T$
- Double negation: $\neg\neg p \Leftrightarrow p$

- DeMorgan's Laws: $\neg(p \land q) \Leftrightarrow (\neg p \lor \neg q)$
  $\neg(p \lor q) \Leftrightarrow (\neg p \land \neg q)$

- Transitivity: $(p \to q) \land (q \to r) \Rightarrow (p \to r)$
  $(p \leftrightarrow q) \land (q \leftrightarrow r) \Rightarrow (p \leftrightarrow r)$

- Modus Ponens: $(p \to q) \land p \Rightarrow q$
- Modus Tollens: $(p \to q) \land \neg q \Rightarrow \neg p$

- or-introduction (and-elimination):
  $p \Rightarrow p \lor q$ "I had dinner" logically implies "I had dinner or lunch".
  $p \land q \Rightarrow p$ "I had dinner and lunch" logically implies "I had dinner".

- or-elimination: $(p \lor q) \land (p \to r) \land (q \to r) \Rightarrow r$