Lecture 4

Equivalence of Formulas
Recall, $\alpha$ and $\beta$ are said to be equivalent ($\alpha \equiv \beta$) iff for any truth valuation $t$, $\alpha^t = \beta^t$.

This can be illustrated by showing via:

1. Truth tables (Their final rows are the same).
2. Valuation Trees (Their leaves are the same).
3. Equivalence laws/Algebra of formulas.

It is important to know how to do all of these and follow the specific instructions for the questions! The equivalence laws/algebra of formulas are in a PDF under Resources on my page. You must know how to manipulate formulas using these!

Do not memorize the steps – learn! *practice, practice, practice*

Code Simplification and Equivalence Examples
Consider the following code:

```java
if ( (input > 0) OR NOT output ) {
    if ( NOT (output AND (queue_length < 100) ) ) {
        P_1
    } else if ( output AND NOT (queue_length < 100) ) {
        P_2
    } else { P_3 }
} else { P_4 }
```

When does each piece of code get executed? Let’s abstract the statement:

Let $i$: input > 0,
$u$: output is true (we will assume it is a boolean),
$q$: queue_length < 100.

```java
if ( i || !u ) {
    if ( !u && q ){
        P_1
    }
    else if ( u && !q){
        P_2
    }
    else { P_3 }
} else { P_4 }
```

By constructing a truth table (Exercise), we can see that $P_2$ is never executed, $P_2$ is *unsatisfiable*. This means that $P_2$ is *dead code*.
We can show that $P_2$ is unsatisfiable by showing it is equivalent to $F$:

$$((i \lor \neg u) \land \neg \neg (u \land q)) \land (u \land \neg q)$$

$\equiv ((i \lor \neg u) \land (u \land q)) \land (u \land \neg q)$  \hspace{1cm} \text{Double Negation}

$\equiv (i \lor \neg u) \land ((u \land (q \land \neg q))$  \hspace{1cm} \text{Associativity}

$\equiv (i \lor \neg u) \land (u \land \neg q)$  \hspace{1cm} \text{Distributivity}

$\equiv (i \lor \neg u) \land (u \land F)$  \hspace{1cm} \text{Contradiction}

$\equiv (i \lor \neg u) \land F$  \hspace{1cm} \text{Simplification I}

$\equiv F$  \hspace{1cm} \text{Simplification I}

From the same truth table, we can also see that $P_3$ is executed, or is satisfiable.

We can show this by finding a truth valuation that satisfies $P_3$.

For example: $t(i) = T, t(u) = T, t(q) = T$ is a truth valuation that satisfies $P_3$.

We can also show that two pieces of code are equivalent. Consider the following fragments:

**Fragment 1:**

```java
if ( i || ~u ) {
    if ( ~((u && q) )
        P1
    }
    else if ( u && ~q )
        P2
    }
else {
    P3
}
else {
    P4
}
```

**Fragment 2:**

```java
if ( i && u && q )
    P3
else if ( ~i && u )
    P4
else {
    P1
}
```

To prove that the two fragments are equivalent, show that each block of code $P_1, P_2, P_3,$ and $P_4$ is executed under equivalent conditions. Since $P_2$ is not mentioned in Fragment 2, its formula is $F$.

<table>
<thead>
<tr>
<th>Block</th>
<th>Fragment 1</th>
<th>Fragment 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$</td>
<td>$(i \lor \neg u) \land \neg((u \land q)$</td>
<td>$\neg (i \land u \land q) \land \neg(\neg i \land u)$</td>
</tr>
<tr>
<td>$P_2$</td>
<td>$(i \lor \neg u) \land \neg((u \land q)$</td>
<td>$F$</td>
</tr>
<tr>
<td></td>
<td>$\land (u \land \neg q)$</td>
<td></td>
</tr>
<tr>
<td>$P_3$</td>
<td>$(i \lor \neg u) \land \neg((u \land q)$</td>
<td>$i \land u \land q$</td>
</tr>
<tr>
<td></td>
<td>$\land \neg(u \land \neg q)$</td>
<td></td>
</tr>
<tr>
<td>$P_4$</td>
<td>$\neg(i \lor \neg u)$</td>
<td>$\neg (i \land u \land q) \land (\neg i \land u)$</td>
</tr>
</tbody>
</table>

Exercise: Show they are equivalent using truth tables, valuation trees and equivalence laws/algebra of formulas.
Semantic Entailment

The notion of satisfiability extends to sets of formulas.

Let $\Sigma$ denote a set of formulas and $t$ a valuation. The valuation of the set of formulas is defined:

$$\Sigma^t = \begin{cases} T & \text{if for each } \beta \in \Sigma, \beta^t = T \\ F & \text{otherwise} \end{cases}$$

$\Sigma^t = T$ means $t$ satisfies $\Sigma$. $\Sigma$ is satisfiable iff there is some valuation $t$ such that $\Sigma^t = T$.

This means that for a given $t$, every formula in the set is satisfied.

Example:

The set $\{(p \rightarrow q) \lor r), p \lor (q \lor s)\}$ is satisfiable.

One valuation that satisfies it is $t$ defined as $t(p) = t(q) = t(r) = t(s) = T$.

Exercise: Find a different valuation that satisfies the set.

Definition of Semantic Entailment:

Let $\Sigma$ be a set of formulas, and let $\alpha$ be a formula.

$\Sigma \models \alpha$, read as $\alpha$ is a logical consequence of $\Sigma$ or $\Sigma$ (semantically) entails $\alpha$, if and only if for any truth valuation $t$, if $\Sigma^t = T$ then also $\alpha^t = T$.

This means that if $\Sigma \models \alpha$, then there is no $t$ such that $\Sigma^t = T$ and $\alpha^t = F$.

We write $\Sigma \not\models \alpha$ to say that $\alpha$ is NOT a logical consequence of $\Sigma$.

If it is the case that $\emptyset \models \alpha$ (where $\emptyset$ is an empty set of formulas), then $\alpha$ is a tautology.

It is always the case that $\{\alpha, \neg\alpha\} \models \beta$, regardless what $\alpha$ and $\beta$ are.

Equivalence can be expressed using the notion of entailment:

Lemma: $\alpha \equiv \beta$ if and only if both $\{\alpha\} \models \beta$ and $\{\beta\} \models \alpha$. 