This logic sentence is always true:

*F or (not F)*

If *F* is false then *(not F)* is true and the sentence is true. If *F* is true, then the sentence is true because true or’ed with anything is always true. So, it doesn’t matter what the truth value of *F* is, the sentence is true. For example, this is true:

*The moon is made of blue cheese or the moon is not made of blue cheese.*

You can use the Alloy Analyzer to verify that the sentence is always true:

**pred** F {}

**check** { F **or** (**not** F) }  
**check** { (1 = 2) **or** **not** (1 = 2) }

The Alloy Analyzer reports:

#1: No counterexample found.  
#2: No counterexample found.

This sentence is also always true:

*F and (F or G)  
 if and only if  
F*

*“if and only if ”* is the equivalence connector.If a sentence *A if and only if B* is true, then *A* and *B* are equivalent. Thus, *F and (F or G)* is equivalent to *F*. In other words, if you are writing a program and suddenly realize that you have created code with the form *F and (F or G),* you can replace it with simply *F*.

Again, we can verify that the sentence is always true using Alloy:

**pred** F {}  
**pred** G {}

**check** { (F **and** (F **or** G)) **iff** F }

The Alloy Analyzer reports: No counterexample found.

The following sentence is always true:

*if F then G  
 if and only if  
F and G*

So, *if F then G* is equivalent to *F and G*. In Alloy predicates one often wants a precondition to be satisfied. Now we see that predicate can be expressed in either of these two ways (assume that F is the precondition statement):

|  |  |
| --- | --- |
| **pred** Example {  F   G  } | **pred** Example {  F => G  } |

Below is a catalog of statements that are always true, regardless of the truth value of *F*, *G*, and *H*. A sentence that evaluates to true regardless of the truth values of its sub-sentences is said to be “valid.” The catalog comes from the (wonderful) book titled “The Logical Basis for Computer Programing, Volume 1: Deductive Reasoning” by Zohar Manna & Richard Waldinger.

## Basic valid sentences

|  |  |
| --- | --- |
| *F if and only if F* | *F or (not F)* |
| *if F then F* | *if (F and G) then F* |
| *if F then (F or G)* | *F and (F or G)  if and only if F* |
| *F or (F and G)  if and only if F* | *(if F then G else G)  if and only if G* |

## True-false laws

|  |  |
| --- | --- |
| *true* | *not false* |
| *F or true* | *not (F and false)* |
| *if false then F* | *if F then true* |
| *(F or false)  if and only if F* | *(F and true)  if and only if F* |
| *(if true then F)  if and only if F* | *(if true then F else G)  if and only if F* |
| *(if false then F else G)  if and only if G* | *(true if and only if F)  if and only if F* |
| *(false if and only if F)  if and only if (not F)* | |

## Commutativity

|  |  |
| --- | --- |
| *(F and G)  if and only if (G and F*) | *(F or G)  if and only if (G or F)* |
| *(F if and only if G)  if and only if (G if and only if F)* | |

## Associativity

|  |  |
| --- | --- |
| *((F and G) and H)  if and only if (F and (G and H))* | *((F or G) or H)  if and only if (F or (G or H))* |
|  | |

## Transitivity

|  |  |
| --- | --- |
|  |  |

## Contrapositive laws

|  |  |
| --- | --- |
| *(if F then G)  if and only if (if (not G) then (not F)*) | *(if (not F) then G)  if and only if (if (not G) then F)* |
| *(F if and only if G)  if and only if ((not F) if and only if (not G))* | |

## Distributivity

|  |  |
| --- | --- |
| *(F and (G or H))  if and only if ((F and G) or (F and H))* | *(F or (G and H))  if and only if ((F or G) and (F or H))* |
| *(if (F or G) then H)  if and only if* | *(if F then (G or H))  if and only if* |
| *(if (F and G) then H)  if and only if* | *(if F then (G and H))  if and only if* |
| *(if (F and G) then H)  if and only if (if F then (if G then H))* | |

## Laws of negation

|  |  |
| --- | --- |
| *not (not F)  if and only if F* | *not(F and G)  if and only if ((not F) or (not G))* |
| *not(F or G)  if and only if ((not F) and (not G))* | *not (if F then G)  if and only if (F and (not G))* |
| *not (if F then G else H)  if and only if (if F then (not G) else (not H))* | *not(F if and only if G)  if and only if (F if and only if (not G))* |

## Reduction laws

|  |  |
| --- | --- |
| *(if F then G)  if and only if ((not F) or G)* | |
| *(if F then G else H)  if and only if* | *(if F then G else H)  if and only if* |
| *(F if and only if G)  if and only if* | *(F if and only if G)  if and only if* |