1. (*2 points*) The associative law for intersection states that $A \cap (B \cap C) = (A \cap B) \cap C)$ for any sets A, B, and C. Verify this law by reducing it to the associative law for propositional logic.

Answer:

$$\begin{aligned} A \cap (B \cap C) &= \{x \mid x \in A \cap (B \cap C)\} && \text{(Definition of } \cap) \\ &= \{x \mid (x \in A) \wedge ((x \in B) \wedge (x \in C))\} && \text{(Definition of } \cap) \\ &= \{x \mid ((x \in A) \wedge (x \in B)) \wedge (x \in C)\} && \text{(Associative law for logic)} \\ &= \{x \mid ((x \in A \cap B)) \wedge (x \in C)\} && \text{(Definition of } \cap) \\ &= (A \cap B) \cap C && \text{(Definition of } \cap) \end{aligned}$$

2. (*3 points*) Let a, b, and c be values of type *int* given as hexadecimal numbers in Java as

`a = 0xABCD1234` `b = 0x5678EF09` `c = 0xFFFF`

Find the values of the following Java expressions, writing the answers as hexadecimal numbers. Do not just give the value, which you could get Java to compute for you; show enough work or explain your reasoning, to show how the answer is computed.

a) `(a << 16) | (b >>> 16)` **b)** `a & (c << 16)` **c)** `(a & (c << 16)) | (b & c)`

Answer:

a) `a << 16` shifts a 16 bits to the left which is four hexadecimal digits, filling in with zeros on the right, giving `0x12340000`. `b >>> 16` shifts a 16 bits to the rightwhich is four hexadecimal digits, filling in with zeros on the left, giving `0x00005678`. When those two numbers are combined with a bitwise or operation, or'ing with zero has no effect, and so the value of `(a << 16) | (b >>> 16)` is `0x12345678`.

b) By similar reasoning, `c << 16` is `0xFFFF0000`. Since and'ing with 1 has no effect and and'ing with 0 results in zero, `a & (c << 16)` is `0xABCD0000`.

c) `b & c` is `0x0000EF09`. (Note that `0xFFFF` is still a 32-bit number, which can be written in full as `0x0000FFFF`.) When the answer from part b) is or'ed with `b & c`, the result is `0xABCDEF09`.

3. (*4 points*) Consider the two 16-bit integers n and m shown below. First, compute the three 16-bit integers $\tilde{}n$, and $n \;\&\; m$, and $n \mid m$. Then, what subset of $\{15, 14, \ldots, 1, 0\}$ does each of the integers n, m, $\tilde{}n$, $n \;\&\; m$, and $n \mid m$ correspond to? (Write out each set in full using the usual set notation.)

$$n = 1001\ 1101\ 1000\ 0101$$
$$m = 0101\ 1001\ 1100\ 0111$$

Answer:

n	$= 1001\ 1101\ 1000\ 0101$	$\{15, 12, 11, 10, 8, 7, 2, 0\}$
m	$= 0101\ 1001\ 1100\ 0111$	$\{14, 12, 11, 8, 7, 6, 2, 1, 0\}$
$\tilde{}n$	$= 0110\ 0010\ 0111\ 1010$	$\{14, 13, 9, 6, 5, 4, 3, 1\}$
$n \mathbin{\&} m$	$= 0001\ 1001\ 1000\ 0101$	$\{12, 11, 8, 7, 2, 0\}$
$n \mid m$	$= 1101\ 1101\ 1100\ 0111$	$\{15, 14, 12, 11, 10, 8, 7, 6, 2, 1, 0\}$

4. (*3 points*) What is computed by the following method? (Hint: Write N in binary!) Explain your answer.

```
int countSomething( int N ) {
    int ct = 0;
    for (int i = 0; i <= 31; i++) {
        if ( (N & 1) == 1 ) {
            ct++;
        }
        N = N >>> 1;
    }
    return ct;
}
```

Answer:

When N is written as a binary number, it is made up of 1's and 0's. This function counts the number of 1's in that binary expansion of N. (If you think of N as representing a subset of $\{31, 30, 29, \ldots, 1, 0\}$, then the function computes the cardinality of that subset.)

The test `if ( (N & 1) == 1 )` tests whether the rightmost bit in N is 1, and if so the value of *ct* is incremented. The assignment `N = N >>> 1` shifts N one bit to the right, so that the next time through the loop, the next bit from the original N is being tested. This is done 32 times, so that every bit from the original N is tested, and *ct* is incremented one for each bit that is a 1.

5. (*2 points*) Describe the set $\{1, 2, 3\} \times \mathbb{N}$. Show that you understand its structure.

Answer:

This set is similar to three copies of $\mathbb{N}$, one for each value in the set $\{1, 2, 3\}$, From the 1 we get elements of $\{1, 2, 3\} \times \mathbb{N}$ of the form $(1, 0), (1, 1), (1, 2), (1, 3), (1, 4), \ldots,$ with one element for each number in $\mathbb{N}$. From the 2, we get $(2, 0), (2, 1), (2, 2), (2, 3), (2, 4), \ldots.$ And similarly for the 3. We could write out the whole set using set notation with ellipses as

$$\begin{aligned}\{1, 2, 3\} \times \mathbb{N} = \{\ & (1, 0), (1, 1), (1, 2), (1, 3), (1, 4), \ldots, \\ & (2, 0), (2, 1), (2, 2), (2, 3), (2, 4), \ldots, \\ & (3, 0), (3, 1), (3, 2), (3, 3), (3, 4), \ldots \\ \}\end{aligned}$$

6. (*5 points*)

a) Consider the function $f : \mathbb{Z} \to \mathbb{Z}$ given by $f(n) = n + 1$. Is f a one-to-one function? Is f an onto function? Justify your answers.

b) Now, consider the function $g : \mathbb{N} \to \mathbb{N}$ given by $g(n) = n + 1$. Is g a one-to-one function? Is g an onto function? Justify your answers.

Answer:

a) f is one-to-one. Suppose $f(n) = f(m)$. This means $n + 1 = m + 1$, which implies $n = m$. [Here, I've shown that for any $n, m \in \mathbb{Z}$, if $f(n) = f(m)$, then $n = m$. This is the definition of one-to-one.] It is onto, since given $m \in \mathbb{Z}$, we can let $n = m - 1$, which is in $\mathbb{Z}$, and then $f(n) = m$. [Here, I've shown that fora any $m \in \mathbb{Z}$, there is an $n \in \mathbb{Z}$ such that $f(n) = m$. This is the definition of onto.]

b) g is one-to-one by an argument identical to the proof that f is one-to-one. However, g is not onto, since there is no $n \in \mathbb{N}$ such that $f(n) = 0$. This follows from the fact that since $n \geq 0$ for all $n \in \mathbb{N}$, then $f(n) = n + 1 > 0$; so it is impossible that $f(n) = 0$. [The proof here shows that it is not the case that $\forall m \in \mathbb{N}, \exists n \in \mathbb{N}, f(n) = m$. The disproof is by giving the counterexample $m = 0$.]