

PROBLEM 1 *English to logic*

Rewrite each of the following English sentences as an expression over propositions. Include both a mapping from symbols to propositions and the final expression (see the example). If there are ambiguities, explain where they arise, and give two non-equivalent interpretations.

1. (example) If I forget my keys I can't get into the house unless my roommate is home.

$K$ : I remember my keys

$H$ : I can enter my house

$R$ : My roommate is home

$$(\neg K \wedge \neg R) \rightarrow \neg H$$

2. I prefer oranges to apples, although apples are less messy to eat

$P$ : I prefer apples to oranges

$M$ : apples are less messy than oranges

$$P \wedge M$$

3. If you can prove  $P \neq NP$  (or  $P = NP$ , though I hope you don't), you'll become famous and I'll give you an A in this class

$E$ : You can prove  $P = NP$

$N$ : You can prove  $P \neq NP$

$F$ : You'll be famous

$A$ : I'll give you an A

$H$ : I hope  $E$

$$((E \vee N) \rightarrow (F \wedge A)) \wedge \neg H$$

*note: the "or" is a separate claim, implicitly anded with others*

4. Python programmers must be lazy because Python programs are so much shorter than the equivalent Java or C++ programs

$L$ : Python programmers are lazy

$J$ : Python programs are shorter than Java programs

$C$ : Python programs are shorter than C++ programs

$$(J \wedge C) \rightarrow L$$

*note: the "or" linguistically means "and" logically in this case*

PROBLEM 2 *If Statements*

Write an expression for when the following function returns the given return values. Use the variables  $a$  and  $b$  as your propositions.

```
def f(a,b):
    if a:
        return "left"
    elif b:
        return "right"
    else:
        return "up"
```

```
public static String f(boolean a, boolean b){
    if(a)
        return "left";
    else if(b)
        return "right";
    else
        return "up";
}
```

Returns "right" when  $b \wedge \neg a$

Returns "up" when  $\neg b \wedge \neg a$

PROBLEM 3 *Truth Tables*

Fill in the following truth tables

A	B	C	$(A \vee C) \leftrightarrow (B \wedge C)$		
0	0	0	0	1	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	1	1

A	B	C	$((A \oplus B) \vee (A \oplus C)) \vee (B \oplus C)$				
0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	1
0	1	0	1	1	0	1	1
0	1	1	1	1	1	1	0
1	0	0	1	1	1	1	0
1	0	1	1	1	0	1	1
1	1	0	0	1	1	1	1
1	1	1	0	0	0	0	0

In each of the blanks below, put 1<sup>st</sup> if the first truth table above is the given idea; 2<sup>nd</sup> if the second truth table is; leave it blank if neither is.

\_\_\_ at least one of  $A$ ,  $B$ , and  $C$  is 1

\_\_\_ at least one of  $A$ ,  $B$ , and  $C$  is 0

\_\_\_  $A$ ,  $B$ , and  $C$  are all the same

2<sup>nd</sup>  $A$ ,  $B$ , and  $C$  are not all the same

\_\_\_ either  $A$  and  $C$  are both false or  $B$  and  $C$  are both true, but not both

1<sup>st</sup> either  $A$  and  $C$  are both false or  $B$  and  $C$  are both true, or both