

Lab 2

Knowledge Representation (Propositional logic)

Example 1

```
# Atomic propositions (simple, indivisible)
```

```
p = True # "It is raining"
```

```
q = False # "I have an umbrella"
```

```
r = True # "The ground is wet"
```

```
print("p: 'It is raining' = {p}")
```

```
print("q: 'I have an umbrella' = {q}")
```

```
print("r: 'The ground is wet' = {r}")
```

```
# Propositions with Boolean value
```

```
p = True # "It is raining"
```

```
q = False # "I have an umbrella"
```

```
# Logical operators
```

```
conjunction = p and q #  $p \wedge q$ 
```

```
disjunction = p or q #  $p \vee q$ 
```

```
negation = not p #  $\neg p$ 
```

```
implication = not p or q #  $p \rightarrow q$ 
```

```
print(p and q) # or print(disjunction)
```

```
print(p or q) # or print(disjunction)
```

```
print(not p) # or print(negation)
```

```
print(not p or q) # or print(implication)
```

Example 1

More structured approach with classes

```
class Proposition:
    def __init__(self, name, value):
        self.name = name
        self.value = value

    def __and__(self, other):
        return self.value and other.value

    def __or__(self, other):
        return self.value or other.value

    def __invert__(self):
        return not self.value

    def implies(self, other):
        return not self.value or other.value

# Define and give the value of the propositions
rain = Proposition("rain", True)
umbrella = Proposition("umbrella", False)

print(~rain) # NOT
print(rain & umbrella) # AND
print(rain | umbrella) # OR
print(rain.implies(umbrella)) # IF-THEN
```