Artificial Intelligence

Propositional Logic
Playing with it in Python

Model

- Model?
  - Set of Boolean propositions
  - Set of assignments
- Common representations
  - { x, y }
    - Meaning a model where x is True and y is also True
    - Any other proposition not represented is assumed False
  - x/True y/True
    - Meaning a model where x is True, y is also True
    - No other propositions exist
- So:
  - { x, y } is set of assignments for the propositions { x, y, z }
    - Because, in this notation, omitted propositions are assumed False
  - X/True y/True is NOT a set of assignments for { x, y, z }
    - Because is makes not assignment to z

Is an expression/formula satisfiable?

- Yes, if there is a model that makes it True

Model |= Satisfaction

- Left side
  - Model / assignments
- Right side
  - Boolean expression
  - Logical formula
- Satisfaction?
  - If the assignments make the expression True
    - "the assignments satisfy the expression"
- Exercises with M = {x/True, y/False} - signal the satisfaction case(s)
  - M|=(x=>y)
  - M|=(x and y)
  - M|=y
  - M|=(x or y)

Model |= Satisfaction

- (solution) Exercise with M = {x/True, y/False} - signal the satisfaction case(s)
  - M|=(x=>y)
  - M|=(x and y)
  - M|=y
  - M|=(x or y)

Logical consequence

- AKA Logical Implication
- CNF = Conjunctive Normal Form
- Lexp 1, ... , Lexp n |= Rexp 1 , ... , Rexp n
- All the models that satisfy the left-side, must also satisfy the right-side
  - But the right-side might satisfy more models
- Exercise: which are logical consequences?
  - (p=>q),q |= p
  - (p and q) |= p
  - (p or q) |= p
  - (p or q), (not p) |= q
  - (p=>q), p |= q

Logical consequence

- (solution) Exercise: which are logical consequences?
  - (p=>q),q |= p
  - (p and q) |= p
  - (p or q) |= p
  - (p or q), (not p) |= q
  - (p=>q), p |= q

Logical equivalence

- L /// R
- L |= R
- R |= L
- Both must happen
- Exercise - signal the logical equivalence case(s)

$$((\neg u) \lor v) \equiv (\neg(u \land (\neg v)))$$

$$(a \land (b \lor c)) \equiv ((a \land b) \lor (a \land c))$$

$$((\neg x) \land (\neg y)) \equiv (\neg(x \land y))$$

$$((\neg x) \lor (\neg y)) \equiv (\neg(x \lor y))$$

$$((\neg u) \land v) \equiv (\neg(u \lor (\neg v)))$$

Logical equivalence

- (solution) Exercise - signal the logical equivalence case(s)
- @Left, equivalent
- @Right, not equivalent

$$((\neg u) \vee v) \equiv (\neg(u \wedge (\neg v)))$$

$$(a \wedge (b \vee c)) \equiv ((a \wedge b) \vee (a \wedge c))$$

$$((\neg x) \wedge (\neg y)) \equiv (\neg(x \wedge y))$$

$$((\neg x) \vee (\neg y)) \equiv (\neg(x \vee y))$$

$$((\neg u) \wedge v) \equiv (\neg(u \vee (\neg v)))$$

Python challenge

- Check companion file am_logical_helper.py
- Study companion class "LogicalHelper"
- Solve the exercises in these slides using an instance of the class