https://uofi.app.box.com/s/k4ptqnsai0hjwsech7003fyll42zjpxf

Small Step Semantics (30 pts)

1. Show the derivation of the final results in the following programs, considering the initial valuation of program variables shown in the corresponding program state, using the **small-step** operational semantic rules introduced in the class. [$30 = 15 \times 2$]

(i) `(y := 0; if x > 0 then (x := 2*x -1 ; y:= -x)`
`else (x := 2*x + 1 ; y:= -x) fi , {x → 1, y → 1} )`

(y := 0; if x > 0 then (x := 2*x - 1; y := -x) else (x := 2*x + 1; y := -x) fi, {x -> 1, y -> 1})
--> (if x > 0 then (x := 2*x - 1; y := -x) else (x := 2*x + 1; y := -x) fi, {x -> 1, y -> 0})
--> (if 1 > 0 then (x := 2*x - 1; y := -x) else (x := 2*x + 1; y := -x) fi, {x -> 1, y -> 0})
--> (if True then (x := 2*x - 1; y := -x) else (x := 2*x + 1; y := -x) fi, {x -> 1, y -> 0})
--> (x := 2*x - 1; y := -x, {x -> 1, y -> 0})
--> (x := 2*1 - 1; y := -x, {x -> 1, y -> 0})
--> (x := 2 - 1; y := -x, {x -> 1, y -> 0})
--> (x := 1; y := -x, {x -> 1, y -> 0})
--> (y := -x, {x -> 1, y -> 0})
--> (y := -1, {x -> 1, y -> 0})
--> **{x -> 1, y -> -1}**

Small Step Semantics (30 pts)

1. Show the derivation of the final results in the following programs, considering the initial valuation of program variables shown in the corresponding program state, using the **small-step** operational semantic rules introduced in the class. [30 = 15 × 2]

(ii) `(while y < 5 do y:= y + 2; x:= x + y*y od, {x → 1, y → 0})`

(while y < 5 do y := y + 2; x := x + y*y od, {x -> 1, y -> 0})
--> (if y < 5 then (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od), else skip fi, {x -> 1, y -> 0})
--> (if 0 < 5 then (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od), else skip fi, {x -> 1, y -> 0})
--> (if True then (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od), else skip fi, {x -> 1, y -> 0})
--> (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 1, y -> 0})
--> (y := 0 + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 1, y -> 0})
--> (y := 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 1, y -> 0})
--> (x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 1, y -> 2})
--> (x := 1 + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 1, y -> 2})
--> (x := 1 + 2*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 1, y -> 2})
--> (x := 1 + 2*2; while y < 5 do y := y + 2; x := x + y*y od, {x -> 1, y -> 2})
--> (x := 1 + 4; while y < 5 do y := y + 2; x := x + y*y od, {x -> 1, y -> 2})
--> (x := 5; while y < 5 do y := y + 2; x := x + y*y od, {x -> 1, y -> 2})
--> (while y < 5 do y := y + 2; x := x + y*y od, {x -> 5, y -> 2})
--> (if y < 5 then (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od), else skip fi, {x -> 5, y -> 2})
--> (if 2 < 5 then (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od), else skip fi, {x -> 5, y -> 2})
--> (if True then (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od), else skip fi, {x -> 5, y -> 2})
--> (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 5, y -> 2})
--> (y := 2 + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 5, y -> 2})
--> (y := 4; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 5, y -> 2})
--> (x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 5, y -> 4})
--> (x := 5 + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 5, y -> 4})
--> (x := 5 + 4*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 5, y -> 4})
--> (x := 5 + 4*4; while y < 5 do y := y + 2; x := x + y*y od, {x -> 5, y -> 4})
--> (x := 5 + 16; while y < 5 do y := y + 2; x := x + y*y od, {x -> 5, y -> 4})
--> (x := 21; while y < 5 do y := y + 2; x := x + y*y od, {x -> 5, y -> 4})
--> (while y < 5 do y := y + 2; x := x + y*y od, {x -> 21, y -> 4})

--> (if y < 5 then (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od) else skip fi, {x -> 21, y -> 4})
--> (if 4 < 5 then (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od) else skip fi, {x -> 21, y -> 4})
--> (if True then (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od) else skip fi, {x -> 21, y -> 4})
--> (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 21, y -> 4})
--> (y := 4 + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 21, y -> 4})
--> (y := 6; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 21, y -> 4})
--> (x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 21, y -> 6})
--> (x := 21 + y*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 21, y -> 6})
--> (x := 21 + 6*y; while y < 5 do y := y + 2; x := x + y*y od, {x -> 21, y -> 6})
--> (x := 21 + 6*6; while y < 5 do y := y + 2; x := x + y*y od, {x -> 21, y -> 6})
--> (x := 21 + 36; while y < 5 do y := y + 2; x := x + y*y od, {x -> 21, y -> 6})
--> (x := 57; while y < 5 do y := y + 2; x := x + y*y od, {x -> 21, y -> 6})
--> (while y < 5 do y := y + 2; x := x + y*y od, {x -> 57, y -> 6})
--> (if y < 5 then (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od) else skip fi, {x -> 57, y -> 6})
--> (if 6 < 5 then (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od) else skip fi, {x -> 57, y -> 6})
--> (if False then (y := y + 2; x := x + y*y; while y < 5 do y := y + 2; x := x + y*y od) else skip fi, {x -> 57, y -> 6})
--> (skip, {x -> 57, y -> 6})
--> **{x -> 57, y -> 6}**

Semantic Argument

$$\frac{v \vDash \neg p}{v \nvDash p}$$

$$\frac{v \nvDash \neg p}{v \vDash p}$$

$$\frac{v \vDash p \wedge q}{v \vDash \ p\ and\ v \vDash q}$$

$$\frac{v \nvDash p \wedge q}{v \nvDash \ p\ or\ v \nvDash q}$$

$$\frac{v \vDash p \vee q}{v \vDash \ p\ or\ v \vDash q}$$

$$\frac{v \nvDash p \vee q}{v \nvDash \ p\ and\ v \nvDash q}$$

Semantic Argument

$$\frac{v \vDash p \Rightarrow q}{v \nvDash p\ or\ v \vDash q}$$

$$\frac{v \nvDash p \Rightarrow q}{v \vDash p \text{ and } v \nvDash q}$$

$$\frac{v \vDash p \Leftrightarrow q}{v \vDash \ p \wedge q\ or\ v \nvDash p \vee q}$$

$$\frac{v \nvDash p \Leftrightarrow q}{v \vDash p \wedge \neg q\ or\ v \vDash \neg p \wedge q}$$

$$\frac{v \vDash p\ and\ v \nvDash \text{p}}{v \vDash false}$$

Logic (20 pts)

In this section, T and F refer to the constant symbols for True and False, respectively. A, B, C, etc. are arbitrary formulas in propositional logic.

2. Either prove the following with semantic argument or show a truth value assignment which evaluates to false: [$20 = 4 \times 5$]

1. $(\neg A) \iff (A \implies F)$ (4)

Suppose that this statement is invalid. Then:

1. v ⊭ (¬ A) <==> (A ==> False) (Assumption)
2. v ⊨ (¬ A) ∧ ¬ (A ==> False) or v ⊨ ¬ (¬ A) ∧ (A ==> False)
   (By 1. and semantics of <==>)
3. v ⊨ (¬ A) ∧ ¬ (A ==> False) (By 1. and semantics of <==>)
   a. v ⊨ (¬ A) and v ⊨ ¬ (A ==> False) (By 3. and semantics of ∧)
   b. v ⊨ (¬ A) (By 3a.)
   c. v ⊭ A (By 3b. and semantics of ¬)
   d. v ⊨ ¬ (A ==> False) (By 3a.)
   e. v ⊭ (A ==> False) By 3d. and semantics of ¬)
   f. v ⊨ A and v ⊭ False (By 3e. and semantics of ==>)
   g. v ⊨ A and v ⊨ ¬ A (By 3c. and 3f.)
   h. v ⊨ False (By 3g. and semantics of "and")
4. v ⊨ ¬ (¬ A) ∧ (A ==> False) (By 1. and semantics of <==>)
   a. v ⊨ ¬ (¬ A) and v ⊨ (A ==> False) (By 4 and semantics of ∧)
   b. v ⊨ ¬ (¬ A) (By 4a.)
   c. v ⊭ (¬ A) (By 4b. and semantics of ¬)
   d. v ⊨ A (By 4c. and semantics of ¬)
   e. v ⊨ (A ==> False) (By 4a.)
   f. v ⊭ A or v ⊨ False (By 4e. and semantics of ==>)
   g. v ⊨ A and v ⊭ A (By 4d. and 4f.)
   h. v ⊨ A and v ⊨ ¬ A (By 4g. and semantics of ¬)
   i. v ⊨ False (By 3h. and semantics of "and")
5. v ⊨ False or v ⊨ False (By 2., 3h., and 4i.)
6. v ⊨ False (By 5. and definition of "or" [It actually isn't defined in the slides.])

Thus, by contradiction, (¬ A) <==> (A ==> F) must be a valid statement.

2. $(A \vee B) \iff ((\neg A) \implies B)$ (4)

The statement doesn't hold for values A = True, B = True

3. $(A \wedge B) \iff (\neg((\neg A) \vee (\neg B)))$ (4)

Suppose this statement is invalid. Then:

1. $v \not\models (A \wedge B) <==> (\neg ((\neg A) \vee (\neg B)))$
2. $v \models (A \wedge B) \wedge \neg (\neg ((\neg A) \vee (\neg B)))$ or $v \models \neg (A \wedge B) \wedge (\neg ((\neg A) \vee (\neg B)))$ (By 1 and semantics of $<==>$)
3. $v \models (A \wedge B) \wedge \neg (\neg ((\neg A) \vee (\neg B)))$ (By 1 and semantics of $<==>$)
    a. $v \models (A \wedge B)$ and $v \models \neg (\neg ((\neg A) \vee (\neg B)))$ (By 3 and semantics of $\wedge$)
    b. $v \models A$ and $v \models B$ (By 3a. and semantics of $\wedge$)
    c. $v \models \neg (\neg ((\neg A) \vee (\neg B)))$ (By 3 and semantics of $\wedge$)
    d. $v \not\models (\neg ((\neg A) \vee (\neg B)))$ (By 3c. and semantics of $\neg$)
    e. $v \models ((\neg A) \vee (\neg B))$ (By 3d. and semantics of $\neg$)
    f. $v \models \neg A$ or $v \models \neg B$ (By 3e. and semantics of $\vee$)
    g. $v \models A$ and $v \models \neg A$ (By 3b. and 3f.)
    h. $v \models$ False (By 3g. and semantics of "and")
4. $v \models \neg (A \wedge B) \wedge (\neg ((\neg A) \vee (\neg B)))$ (By 1 and semantics of $<==>$)
    a. $v \models \neg (A \wedge B)$ and $v \models (\neg ((\neg A) \vee (\neg B)))$ (By 4 and semantics of $\wedge$)
    b. $v \models \neg (A \wedge B)$ (By 4a.)
    c. $v \not\models A \wedge B$ (By 4b. and semantics of $\neg$)
    d. $v \not\models A$ or $v \not\models B$ (By 4c. and semantics of $\vee$)
    e. $v \models (\neg ((\neg A) \vee (\neg B)))$ (By 4a.)
    f. $v \not\models (\neg A) \vee (\neg B)$ (By 4f. and semantics of $\neg$)
    g. $v \not\models \neg A$ and $v \not\models \neg B$ (By 4g. and semantics of $\vee$)
    h. $v \models A$ and $v \models B$ (By 4h. and semantics of $\neg$)
    i. ($v \not\models A$ and $v \models A$) or ($v \not\models B$ and $v \models B$) (By 4d. and 4.h)
    j. $v \not\models$ False or $v \not\models$ False (By 4i. and semantics of "and")
5. $v \models$ False or $v \models$ False or $v \models$ False (By 2., 3h., and 4j.)
6. $v \models$ False (By 5. and definition of "or" [It actually isn't defined in the slides.])

Thus, by contradiction, $(A \wedge B) <==> (\neg ((\neg A) \vee (\neg B)))$ must be a valid statement.

4. $(A \iff B) \iff ((A \implies B) \land (B \implies A))$ (4)

Suppose this statement is invalid. Then:

1. v ⊭ (A <==> B) <==> ((A ==> B) ∧ (B ==> A))
2. v ⊨ (A <==> B) ∧ ¬ ((A ==> B) ∧ (B ==> A))
   or v ⊨ ¬ (A <==> B) ∧ ((A ==> B) ∧ (B ==> A)) (By 1. and semantics of <==>)
3. v ⊨ (A <==> B) ∧ ¬ ((A ==> B) ∧ (B ==> A)) (By 2.)
   a. v ⊨ (A <==> B) and v ⊨ ¬ ((A ==> B) ∧ (B ==> A)) (By 3. and semantics of ∧)
   b. v ⊨ (A <==> B) (By 3a.)
   c. v ⊨ A ∧ B or v ⊭ A ∨ B (By 3b. and semantics of <==>)
      i. (v ⊨ A and v ⊨ B) (By 3c. and semantics of ∧)
      ii. (v ⊭ A and v ⊭ B) (By 3c. and semantics of ∨)
   d. (v ⊨ A and v ⊨ B) or (v ⊭ A and v ⊭ B) (By 3ci. and 3cii.)
   e. v ⊨ ¬ ((A ==> B) ∧ (B ==> A)) (By 3a.)
   f. v ⊭ (A ==> B) ∧ (B ==> A) (By 3e. and semantics of ¬)
   g. v ⊭ A ==> B or v ⊭ B ==> A (By 3f. and semantics of ∧)
   h. v ⊭ A ==> B (By 3g.)
      i. (v ⊨ A and v ⊭ B) (By 3h. and semantics of ==>)
   i. v ⊭ B ==> A (By 3h.)
      i. (v ⊨ B and v ⊭ A) (By 3i. and semantics of ==>)
   j. (v ⊨ A and v ⊭ B) or (v ⊨ B and v ⊭ A) (By 3hi. and 3ii.)
   k. ((v ⊨ A and v ⊨ B) or (v ⊭ A and v ⊭ B))
      and ((v ⊨ A and v ⊭ B) or (v ⊨ B and v ⊭ A)) (By 3d. and 3j.)
   l. ((v ⊨ A and v ⊨ B) and (v ⊨ A and v ⊭ B))
      or ((v ⊨ A and v ⊨ B) and (v ⊨ B and v ⊭ A))
      or ((v ⊭ A and v ⊭ B) and (v ⊨ A and v ⊭ B))
      or ((v ⊭ A and v ⊭ B) and (v ⊨ B and v ⊭ A)) (By 3k. and semantics of "and")
   m. (v ⊨ A and v ⊨ B) and (v ⊨ A and v ⊭ B) (By 3l.)
      i. v ⊨ B and v ⊭ B (By 3m.)
      ii. v ⊨ False (By 3mi. and semantics of "and")
   n. (v ⊨ A and v ⊨ B) and (v ⊨ B and v ⊭ A) (By 3l.)
      i. v ⊨ A and v ⊭ A (By 3n.)
      ii. v ⊨ False (By 3ni. and semantics of "and")
   o. (v ⊭ A and v ⊭ B) and (v ⊨ A and v ⊭ B) (By 3l.)
      i. v ⊭ A and v ⊨ A (By 3o.)
      ii. v ⊨ False (By 3oi. and semantics of "and")
   p. (v ⊭ A and v ⊭ B) and (v ⊨ B and v ⊭ A) (By 3l.)
      i. v ⊭ B and v ⊨ B (By 3p.)
      ii. v ⊨ False (By 3pi. and semantics of "and")
   q. v ⊨ False or False or False or False (By 3l., 3mii., 3nii., 3oii., 3pii.)
   r. v ⊨ Fase (By 3q. and definition of "or" [It actually isn't defined in the slides.])

4. v ⊨ ¬ (A <==> B) ∧ ((A ==> B) ∧ (B ==> A)) (By 2.)
    a. v ⊨ ¬ (A <==> B) and v ⊨ ((A ==> B) ∧ (B ==> A)) (By 4. and semantics of ∧)
    b. v ⊨ ¬ (A <==> B) (By 4a.)
    c. v ⊭ A <==> B (By 4b. and semantics of ¬)
    d. v ⊨ A ∧ ¬ B or v ⊨ ¬ A ∧ B (By 4c. and semantics of <==>)
    e. v ⊨ A ∧ ¬ B (By 4d.)
        i. v ⊨ A and v ⊨ ¬ B (By 4di.)
        ii. v ⊨ A and v ⊭ B (By 4dii. and semantics of ¬)
    f. v ⊨ ¬ A ∧ B (By 4d.)
        i. v ⊨ ¬ A and v ⊨ B (By 4fi.)
        ii. v ⊭ A and v ⊨ B (By 4fii. and semantics of ¬)
    g. (v ⊨ A and v ⊭ B) or (v ⊭ A and v ⊨ B) (By 4d., 4eii., 4fii.)
    h. v ⊨ (A ==> B) ∧ (B ==> A) (By 4a.)
    i. v ⊨ A ==> B and v ⊨ B ==> A (By 4h. and semantics of ∧)
    j. v ⊨ A ==> B (By 4i.)
        i. v ⊭ A or v ⊨ B (By 4j. and semantics of ==>)
    k. v ⊨ B ==> A (By 4i.)
        i. v ⊭ B or v ⊨ A (By 4k. and semantics of ==>)
    l. (v ⊭ A or v ⊨ B) and (v ⊭ B or v ⊨ A) (By 4i., 4ji., 4ki.)
    m. (v ⊭ A and v ⊭ B) or (v ⊭ A and v ⊨ A)
    or (v ⊨ B and v ⊭ B) or (v ⊨ B and v ⊨ A) (By 4l. and semantics of "and")
    n. (v ⊭ A and v ⊨ A) (By 4m.)
        i. v ⊨ False (By 4n. and semantics of "and")
    o. (v ⊨ B and v ⊭ B) (By 4m.)
        i. v ⊨ False (By 4o. and semantics of "and")
    p. (v ⊭ A and v ⊭ B) or False or False or (v ⊨ B and v ⊨ A) (By 4m., 4n., 4o.)
    q. (v ⊭ A and v ⊭ B) or (v ⊨ B and v ⊨ A) (By 4p. and semantics of "or")
    r. ((v ⊨ A and v ⊭ B) or (v ⊭ A and v ⊨ B))
    and ((v ⊭ A and v ⊭ B) or (v ⊨ B and v ⊨ A)) (By 4a., 4g., 4q.)
    s. ((v ⊨ A and v ⊭ B) and (v ⊭ A and v ⊭ B))
    or ((v ⊨ A and v ⊭ B) and (v ⊨ B and v ⊨ A))
    or ((v ⊭ A and v ⊨ B) and (v ⊭ A and v ⊭ B))
    or ((v ⊭ A and v ⊨ B) and (v ⊨ B and v ⊨ A)) (By 4r. and semantics of "and")
    t. (v ⊨ A and v ⊭ B) and (v ⊭ A and v ⊭ B) (By 4s.)
        i. v ⊨ A and v ⊭ A (By 4t.)
        ii. v ⊨ False (By 4ti. and semantics of "and")
    u. (v ⊨ A and v ⊭ B) and (v ⊨ B and v ⊨ A) (By 4s.)
        i. v ⊭ B and v ⊨ B (By 4u.)
        ii. v ⊨ False (By 4ui. and semantics of "and")
    v. (v ⊭ A and v ⊨ B) and (v ⊭ A and v ⊭ B) (By 4s.)

        i. $v \models B$ and $v \not\models B$ (By 4v.)
        ii. $v \models$ False (By 4vi. and semantics of "and")
    w. ($v \not\models A$ and $v \models B$) and ($v \models B$ and $v \models A$) (By 4s.)
        i. $v \not\models A$ and $v \models A$ (By 4w.)
        ii. $v \models$ False (By 4wi. and semantics of "and")
5. $v \models$ False or False or False or False (By 4s., 4tii., 4uii., 4vii., 4wii.)
6. $v \models$ False (By 5. and semantics of "or")

Thus, by contradiction, (A <==> B) <==> ((A ==> B) $\wedge$ (B ==> A)) must be a valid statement.

5. $(A \wedge (A \implies B)) \implies B$ (4)

Suppose this statement is invalid. Then:

1. v ⊭ (A ∧ (A ==> B)) ==> B
2. v ⊨ (A ∧ (A ==> B)) and v ⊭ B (By 1. and semantics of ==>)
3. v ⊨ A ∧ (A ==> B) (By 2.)
4. v ⊨ A and v ⊨ (A ==> B) (By 3. and semantics of ∧)
5. v ⊨ A ==> B (By 4.)
6. v ⊨ ¬ A or v ⊨ B (By 5. and semantics of ==>)
7. v ⊭ A (By 6. and semantics of ¬)
8. v ⊨ A and (v ⊭ A or v ⊨ B) and v ⊭ B (By 2., 4., 6., 7.)
9. (v ⊭ A or v ⊨ B) and (v ⊨ A and v ⊭ B) (By 8.)
10. (v ⊭ A and v ⊨ A and v ⊭ B) or (v ⊨ B and v ⊨ A and v ⊭ B)
    (By 9. and semantics of "and")
11. (v ⊭ A and v ⊨ A and v ⊭ B) (By 10.)
12. v ⊨ False and v ⊭ B (By 11. and semantics of "and")
13. v ⊨ False (By 12. and semantics of "and")
14. (v ⊨ B and v ⊨ A and v ⊭ B) (By 10.)
15. (v ⊨ B and v ⊭ B and v ⊨ A) (By 14. and semantics of "and")
16. v ⊨ False and v ⊨ A (By 15. and semantics of "and")
17. v ⊨ False (By 16. and semantics of "and")

Thus, by contradiction, (A ∧ (A ==> B)) ==> B must be a valid statement.