CS245 Logic and Computation

Alice Gao

August 1, 2018

Contents

1 Propositional Logic

1.1 Translations

Exercise 1. *Translate the following three sentences into propositional logic.*

- **Nadhi will eat a fruit if it is an apple.**
- **Nadhi will eat a fruit only if it is an apple.**
- **Nadhi will eat a fruit if and only if it is an apple.**

Solution: n: Nadhi will eat a fruit.
a: The fruit is an apple.

- **Nadhi will eat a fruit if it is an apple.**

  Translation: $(a \rightarrow n)$

  If the fruit is an apple, we know that Nadhi will eat it.

  If the fruit is not an apple, Nadhi may or may not eat it.

  The set of apples is a subset of the set of fruits that Nadhi eats.

- **Nadhi will eat a fruit only if it is an apple.**

  Translation: $(n \rightarrow a)$

  If Nadhi eats a fruit, then we know that it is an apple.

  If Nadhi does not eat a fruit, the fruit may or may not be an apple.

  The set of fruits that Nadhi eats is a subset of the set of apples.

- **Nadhi will eat a fruit if and only if it is an apple.**

  Translation: $(n \leftrightarrow a)$

  If Nadhi eats a fruit, then it is an apple.

  If Nadhi does not eat a fruit, then it is not an apple.

  The set of fruits that Nadhi eats and the set of apples coincide.

Exercise 2. *Translate the following sentence into multiple propositional formulas. Show that they are logically equivalent using a truth table.*

Soo-Jin will eat an apple or an orange but not both.

Solution: a: Soo-Jin will eat an apple. o: Soo-Jin will eat an orange.

This sentence translates into an exclusive OR. There are many ways of writing down a formula for an exclusive OR.

- $((a \vee o) \wedge (\neg(a \wedge o)))$

  a or o is true, but not both.

- $((a \vee o) \wedge ((\neg a) \vee (\neg o)))$

  a or o is true, and a is false or o is false.

- $((a \wedge (\neg o)) \vee ((\neg a) \wedge o))$

  a is true and o is false, or a is false and o is true.

- $(\neg(a \leftrightarrow o))$

  It is not the case that a and o have the same truth value.

- $((\neg a) \leftrightarrow o) \equiv (a \leftrightarrow (\neg o))$

  negated a and o have the same truth value.

Exercise 3. *Translate the following sentence into at least three syntactically different propositional formulas. Show that they are logically equivalent using a truth table.*

If it is sunny tomorrow, then I will play golf, provided that I am relaxed.

Solution:

- s: It is sunny tomorrow.
- g: I will play golf.
- r: I am relaxed.

I can think of three ways of translating this sentence into a propositional formula.

- Interpretation 1: If it is sunny tomorrow, then, if I am relaxed, then I will play golf.

  Translation: $(s \to (r \to g))$.

  Sunny tomorrow is the premise for the first.

- Interpretation 2: If it is sunny tomorrow and I am relaxed, then I will play golf.

  Translation: $((s \land r) \to g)$.

  Sunny tomorrow and being relaxed together are premises for playing golf.

- Interpretation 3: If I am relaxed, then, if it is sunny tomorrow, I will play golf.

  Translation: $(r \to (s \to g))$.

  Being relaxed is the premise for the rest.

All three interpretations are logically equivalent.

Exercise 4. *Translate the following sentence into a propositional formula.*

If I ace CS 245, I will get a job at Google; otherwise I will apply for the Geek Squad.

Solution: Define the propositional variables:

- a: I ace CS 245.
- g: I will get a job at Google.
- s: I will apply for the Geek Squad.

First, let's break down this sentence into two parts by the semicolon.

The first part translates into an implication because of the key word "if". It becomes $(a \to g)$.

In the second part, "otherwise" means that "if I don't ace CS 245". After rephrasing, the second part becomes "If I don't ace CS 245, then I will apply for the Geek Squad." This is another implication $((\neg a) \to s)$.

Now the tricky part is: what connective should we use to connect the two parts together? Two natural options are $\wedge$ and $\vee$. The $\vee$ option seems possible because the sentence could be rephrase as "If I ace CS 245, ...; or otherwise ...."
The correct connective to use is $\wedge$ for the following reasons.

Let's consider the scenario in which I ace CS 245, I don't get a job at Google and I apply for the Geek Squad. In this case, is the sentence true or false? Intuitively, the sentence should be false, because the first implication is violated when I ace CS 245 but do not get a job at Google. Now let's look at the truth values of the two possible propositional formulas:

- If we use $\wedge$ as the connective, the resulting formula $((a \to g) \wedge ((\neg a) \to s))$ is false in this scenario. The truth value of the formula is the same as the truth value of the sentence in this scenario.
- If we use $\vee$ as the connective, the resulting formula $((a \to g) \wedge ((\neg a) \to s))$ is true in this scenario. This truth value of the formula is different from the truth value of the sentence in this scenario. Therefore, $\vee$ is not the correct connective to use because the resulting formula has a different meaning from the formula.

Exercise 5. *Translate the following sentence into two propositional formulas and explain why the two formulas are not logically equivalent.*

Sidney will carry an umbrella unless it is sunny.

Solution: Define the propositional variables.
u: Sidney will carry an umbrella.
s: It is sunny.

- Interpretation 1:

  Intuitively, many people understand "unless" as an "exclusive OR", which means that exactly one of the two parts of the sentence is true at a time.

  With this interpretation, "unless" is equivalent to an "if and only if not". The sentence is true under the following two scenarios:

  - It is not sunny and Sidney carries an umbrella.
  - It is sunny and Sidney does not carry an umbrella.

  Note that this interpretation does not allow Sidney to carry an umbrella when it is sunny. So the sentence is false when u and s are both true.

  In propositional logic, this is equivalent to

$$
\begin{align}
&((\neg u) \leftrightarrow s) \tag{1}\\
&\equiv (((\neg u) \wedge s) \vee (u \wedge (\neg s))) \tag{2}\\
&\equiv ((u \vee s) \wedge (\neg(u \wedge s))) \tag{3}\\
&\equiv ((u \vee s) \wedge ((\neg u) \vee (\neg s))). \tag{4}
\end{align}
$$

  All the formulas above are equivalent. They look different but their meanings are the same.

- Interpretation 2:

  Alternatively, you may think of "unless" as meaning "if not". Then the sentence becomes: if it is not sunny, then Sidney will carry an umbrella. In propositional logic, this becomes:

$$
\begin{align}
&((\neg s) \rightarrow u) \tag{5}\\
&\equiv ((\neg(\neg s)) \vee u) \tag{6}\\
&\equiv (s \vee u). \tag{7}
\end{align}
$$

  Under this interpretation, this sentence is true under three scenarios:

  - It is not sunny and Sidney carries an umbrella.

  - It is sunny and Sidney does not carry an umbrella.
  - It is sunny and Sidney carries an umbrella.

Notice that this interpretation allows Sidney to carry an umbrella when it is sunny. So the sentence is true when u and s are both true.

1.1.1 Translations: Additional Exercises

Exercise 6. *Consider the following sentences. If the sentence is a proposition, translate it into a propositional formula. Otherwise, explain why it is not a proposition.*

(A) Does Lidia drink coffee and tea?

(B) Lidia drinks coffee.

(C) Lidia does not drink coffee.

(D) Lidia drinks coffee and tea.

(E) Lidia drinks neither coffee nor tea.

1.2 Structural Induction

Theorem 1. *Every well-formed formula has an equal number of opening and closing brackets.*

Solution:

Proof by Structural Induction. Let $P(\varphi)$ denote that the well-formed formula φ has an equal number of opening and closing brackets. Let $op(\varphi)$ and $cl(\varphi)$ denote the number of opening and closing brackets of φ respectively.

Base case: φ is a propositional variable q. Prove that $P(q)$ holds.

q has zero opening and zero closing bracket. Thus, $P(\varphi)$ holds.

Induction step:

Case 1: φ is $(\neg a)$, where a is well-formed.
Induction hypothesis: Assume that $P(a)$ holds.
We need to prove that $P((\neg a))$ holds.

$$\begin{aligned} op((\neg a)) &= 1 + op(a) && (8)\\ &= 1 + cl(a) \text{ By induction hypothesis} && (9)\\ &= cl((\neg a)) && (10)\end{aligned}$$

Thus, $P((\neg a))$ holds.

Case 2: φ is $(a * b)$ where a and b are well-formed and $*$ is a binary connective.
Induction hypothesis: Assume that $P(a)$ and $P(b)$ hold.
We need to prove that $P((a * b))$ holds.

$$\begin{aligned} op((a * b)) &= 1 + op(a) + op(b) && (11)\\ &= 1 + cl(a) + cl(b) \text{ By induction hypothesis} && (12)\\ &= cl(a * b) && (13)\end{aligned}$$

Thus, $P((a * b))$ holds.

By the principle of structural induction, $P(\varphi)$ holds for every well-formed formula φ. QED

□

Theorem 2. *Every proper prefix of a well-formed formula has more opening than closing brackets.*

Solution:

Proof by Structural Induction. Let $P(\varphi)$ denote that every proper prefix of the well-formed formula φ has more opening than closing brackets.
Let $op(\varphi)$ and $cl(\varphi)$ denote the number of opening and closing brackets of φ respectively.

Base case: φ is a propositional variable q. Prove that $P(q)$ holds.

Induction step:

Case 1: φ is $(\neg a)$, where a is well-formed.
Induction hypothesis: Assume that $P(a)$ holds.
We need to prove that $P((\neg a))$ holds.
Let m denote any proper prefix of a. There are four possible proper prefixes of $(\neg a)$: (, $(\neg$, $(\neg m$, and $(\neg a$. We will prove the four cases separately.

$$op(() = 1 \tag{14}$$
$$cl(() = 0 \tag{15}$$
$$op(() > cl(() \tag{16}$$

$$op((\neg) = 1 \tag{17}$$
$$cl((\neg) = 0 \tag{18}$$
$$op((\neg) > cl(() \tag{19}$$

$$op((\neg m) \tag{20}$$
$$= 1 + op(m) \tag{21}$$
$$> 1 + cl(m) \text{ By the induction hypothesis on } m \tag{22}$$
$$> cl(m) \tag{23}$$
$$= cl((\neg m) \tag{24}$$

$$op((\neg a) \tag{25}$$
$$= 1 + op(a) \tag{26}$$
$$= 1 + cl(a) \text{ By Theorem 1 and } a \text{ is a well-formed formula} \tag{27}$$
$$> cl(a) \tag{28}$$
$$= cl((\neg a) \tag{29}$$

Case 2: φ is $(a * b)$ where a and b are well-formed and $*$ is a binary connective. Let m and n denote any proper prefix of a and b respectively.

Induction hypothesis: Assume that $P(a)$ and $P(b)$ hold. In other words, $P(m)$ and $P(n)$ are true.

We need to prove that $P((a * b))$ holds.

There are six possible proper prefixes of $(a * b)$: (, $(m$, $(a$, $(a*$, $(a * n$, and $(a * b$.

$$op(() = 1 \tag{30}$$
$$cl(() = 0 \tag{31}$$
$$op(() > cl(() \tag{32}$$

$$op((m) \tag{33}$$
$$= 1 + op(m) \tag{34}$$
$$> 1 + cl(m) \text{ By the induction hypothesis on } m \tag{35}$$
$$> cl(m) \tag{36}$$
$$= cl((m) \tag{37}$$

$$op((a) \tag{38}$$
$$= 1 + op(a) \tag{39}$$
$$= 1 + cl(a) \text{ By Theorem 1 and } a \text{ is a well-formed formula} \tag{40}$$
$$> cl(a) \tag{41}$$
$$= cl((a) \tag{42}$$

$$op((a*) \tag{43}$$
$$= 1 + op(a) \tag{44}$$
$$= 1 + cl(a) \text{ By Theorem 1 and } a \text{ is a well-formed formula} \tag{45}$$
$$> cl(a) \tag{46}$$
$$= cl((a*) \tag{47}$$

$$
\begin{aligned}
& op((a * n) && (48)\\
& = 1 + op(a) + op(n) && (49)\\
& = 1 + cl(a) + op(n) \text{ By Theorem 1 and } a \text{ is a well-formed formula} && (50)\\
& > 1 + cl(a) + cl(n) \text{ By the induction hypothesis on } n && (51)\\
& > cl(a) + cl(n) && (52)\\
& = cl((a * n) && (53)
\end{aligned}
$$

$$
\begin{aligned}
& op((a * b) && (54)\\
& = 1 + op(a) + op(b) && (55)\\
& = 1 + cl(a) + cl(b) \text{ By Theorem 1 and } a \text{ is a well-formed formula} && (56)\\
& > cl(a) + cl(b) && (57)\\
& = cl((a * b) && (58)
\end{aligned}
$$

By the principle of structural induction, $P(\varphi)$ holds for every well-formed formula φ. QED

□

1.3 The Semantics of an Implication

Exercise 7. *Do you really understand an implication? We will find out.*

- *Think of an implication as a promise that someone made to you. In what case can you prove that the promise has been broken (i.e. the implication is false)?*
- *When the premise is true, what is the relationship between the truth value of the conclusion and the truth value of the implication?*
- *When the premise is false, the implication is vacuously true. Could you come up with an intuitive explanation for this?*
- *If the conclusion is true, is the implication true or false?*
- *The implication* $(a \to b)$ *is logically equivalent to* $((\neg a) \vee b)$*. Does this equivalent formula make sense to you? Explain.*

1.4 Tautology, Contradiction, and Satisfiable but Not a Tautology

Exercise 8. *Determine whether each of the following formulas is a tautology, satisfiable but not a tautology, or a contradiction.*

- p

  Solution: Answer: Satisfiable but not a tautology.

  Reason: True when p is true and false when p is false.

- $((r \wedge s) \rightarrow r)$

  Solution: Answer: Tautology.

  Reason: When r is true, the conclusion of the implication is true, so the implication is true. When r is false, the premise of the implication is false, so the implication is vacuously true.

- $((\neg(p \leftrightarrow q)) \leftrightarrow (q \vee p))$

  Solution: Answer: Satisfiable but not a tautology

  Reason: It's tempting to say "these two formulas don't mean the same thing so the biconditional is false". However, go back to truth values. When p is true and q is false, both sides of the biconditional are true and the biconditional itself is true. When p and q are both true, the left side is false but the right is true, and so the biconditional is false.

- $((((p \vee q) \wedge (p \vee (\neg q))) \wedge ((\neg p) \vee q)) \wedge ((\neg p) \vee (\neg q)))$

  Solution: Answer: Contradiction

  Reason: The first half can be simplfiied to $(p \vee (q \wedge (\neg q)))$, which is $(p \vee F)$ or p. The second half can be simplfiied to $(\neg p)$. Thus, the entire formula is $(p \wedge (\neg p))$, which is a contradiction.

1.5 Logical Equivalence

Exercise 9. *”If it is sunny, I will play golf, provided that I am relaxed.”*
s: it is sunny. g: I will play golf. r: I am relaxed.

There are three possible translations:

1. $(r \to (s \to g))$
2. $((s \wedge r) \to g)$
3. $(s \to (r \to g))$

Prove that all three translations are logically equivalent.

Solution: Part 1: $(r \to (s \to g)) \equiv ((s \wedge r) \to g)$.

Proof.

$$\begin{align}
&(r \to (s \to g)) && \tag{59}\\
&\equiv (r \to ((\neg s) \vee g)) && \text{Implication} \tag{60}\\
&\equiv ((\neg r) \vee ((\neg s) \vee g)) && \text{Implication} \tag{61}\\
&\equiv (((\neg r) \vee (\neg s)) \vee g) && \text{Associativity} \tag{62}\\
&\equiv (((\neg (r \wedge s)) \vee g) && \text{De Morgan} \tag{63}\\
&\equiv ((r \wedge s) \to g) && \text{Implication} \tag{64}\\
&\equiv ((s \wedge r) \to g) && \text{Commutativity} \tag{65}
\end{align}$$

□

Part 2: $(r \to (s \to g)) \equiv (s \to (r \to g))$.

Proof.

$$\begin{align}
&(r \to (s \to g)) && \tag{66}\\
&\equiv (r \to ((\neg s) \vee g)) && \text{Implication} \tag{67}\\
&\equiv ((\neg r) \vee ((\neg s) \vee g)) && \text{Implication} \tag{68}\\
&\equiv (((\neg r) \vee (\neg s)) \vee g) && \text{Associativity} \tag{69}\\
&\equiv (((\neg s) \vee (\neg r)) \vee g) && \text{Commutativity} \tag{70}\\
&\equiv ((\neg s) \vee ((\neg r) \vee g)) && \text{Associativity} \tag{71}\\
&\equiv ((\neg s) \vee (r \to g)) && \text{Implication} \tag{72}\\
&\equiv (s \to (r \to g)) && \text{Implication} \tag{73}
\end{align}$$

□

Exercise 10. *"If it snows then I will not go to class but I will do my assignment."*
s: it snows. c: I will go to class. a: I will do my assignment.

There are two possible translations:

1. $((s \to (\neg c)) \land a)$
2. $(s \to ((\neg c) \land a))$

Prove that the two translations are NOT logically equivalent.

Solution:

Proof. We need to find a valuation t under which the two formulas have different values.
Consider the truth valuation t where $t(s) = \texttt{F}$, $t(c) = \texttt{T}$, and $t(a) = \texttt{F}$.
The two formulas have different values under t, as shown below.

- $((s \to (\neg c)) \land a)^t = \texttt{F}$
- $(s \to ((\neg c) \land a))^t = \texttt{T}$

□

1.6 Analyzing Conditional Code

Consider the following code fragment:

```
if (input > 0 || !output) {
  if (!(output && queuelength < 100)) {
    P1
  } else if (output && !(queuelength < 100)) {
    P2
  } else {
        P3
  }
} else {
  P4
}
```

Define the propositional variables:

- i: input > 0
- u: output
- q: queuelength < 100

The code fragment becomes the following. We'll call this code fragment #1.

```
if ( i || !u ) {
  if ( !(u && q) ) {
    P1
  } else if ( u && !q ) {
    P2
  } else { P3 }
} else { P4 }
```

Code fragment #2:

```
if (( i && u) && q) {
  P3
} else if (!i && u) {
  P4
} else {
  P1
}
```

Prove that these two pieces of code fragments are equivalent:
Solution:

Prove that the condition leading to P_2 is logically equivalent to F.
The condition leading to P_2:

$$
\begin{aligned}
&(((i \lor (\neg u)) \land (\neg(\neg(u \land q)))) \land (u \land (\neg q))) && && (74)\\
&\equiv (((i \lor (\neg u)) \land (u \land q)) \land (u \land (\neg q))) && \text{Double Negation} && (75)\\
&\equiv ((i \lor (\neg u)) \land ((u \land u) \land (q \land (\neg q)))) && \text{Associativity, Commutativity} && (76)\\
&\equiv ((i \lor (\neg u)) \land (u \land (q \land (\neg q)))) && \text{Idempotence} && (77)\\
&\equiv ((i \lor (\neg u)) \land (u \land \mathtt{F})) && \text{Contradiction} && (78)\\
&\equiv ((i \lor (\neg u)) \land \mathtt{F}) && \text{Simplification 1} && (79)\\
&\equiv \mathtt{F} && \text{Simplification 1} && (80)\\
& && && (81)
\end{aligned}
$$

Prove that the condition leading to P_3 is true if and only if all three variables are true.
The condition leading to P_3:

$$
\begin{aligned}
&(((i \lor (\neg u)) \land (u \land q)) \land (\neg(u \land (\neg q)))) && && (82)\\
&\equiv (((i \lor (\neg u)) \land (u \land q)) \land ((\neg u) \lor (\neg(\neg q)))) && \text{De Morgan} && (83)\\
&\equiv (((i \lor (\neg u)) \land (u \land q)) \land ((\neg u) \lor q)) && \text{Double Negation} && (84)\\
&\equiv ((i \lor (\neg u)) \land (u \land (q \land ((\neg u) \lor q)))) && \text{Associativity} && (85)\\
&\equiv ((i \lor (\neg u)) \land (u \land q)) && \text{Simplification 2} && (86)\\
&\equiv ((i \lor (\neg u)) \land u) \land q) && \text{Associativity} && (87)\\
&\equiv (((i \land u) \lor ((\neg u) \land u)) \land q) && \text{Distributivity} && (88)\\
&\equiv (((i \land u) \lor \mathtt{F}) \land q) && \text{Contradiction} && (89)\\
&\equiv ((i \land u) \land q) && \text{Simplification 1} && (90)
\end{aligned}
$$

Prove that the condition leading to P_4 is true if and only if i is false and u is true.
The condition leading to P_4:

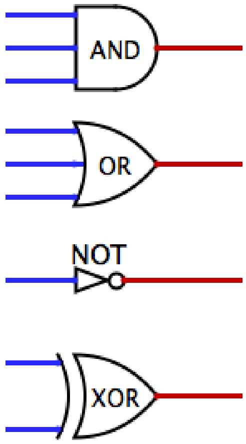
$$
\begin{aligned}
&(\neg(i \lor (\neg u))) && && (91)\\
&((\neg i) \land (\neg(\neg u))) && \text{De Morgan} && (92)\\
&\equiv ((\neg i) \land u) && \text{Double Negation} && (93)
\end{aligned}
$$

The condition leading to P_1:

$$
\begin{aligned}
&((i \lor (\neg u)) \land (\neg(u \land q))) && && (94)\\
&\equiv ((i \lor (\neg u)) \land ((\neg u) \lor (\neg q))) && \text{De Morgan} && (95)\\
&\equiv ((\neg u) \lor (i \land (\neg q))) && \text{Distributivity} && (96)
\end{aligned}
$$

1.7 Circuit Design

Basic gates:



Problem: Your instructors, Alice, Carmen, and Collin, are choosing questions to be put on the midterm. For each problem, each instructor votes either yes or not. A question is chosen if it receives two or more yes votes. Design a circuit, which outputs yes whenever a question is chosen.

1. Draw the truth table based on the problem description.

x	y	z	output
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	F

2. Convert the truth table into a propositional formula.

3. Then, convert the formula to a circuit.

Solution:

Solution 1:

1. Convert the truth table into a propositional formula.

   Convert each row of the truth table to a conjunction.

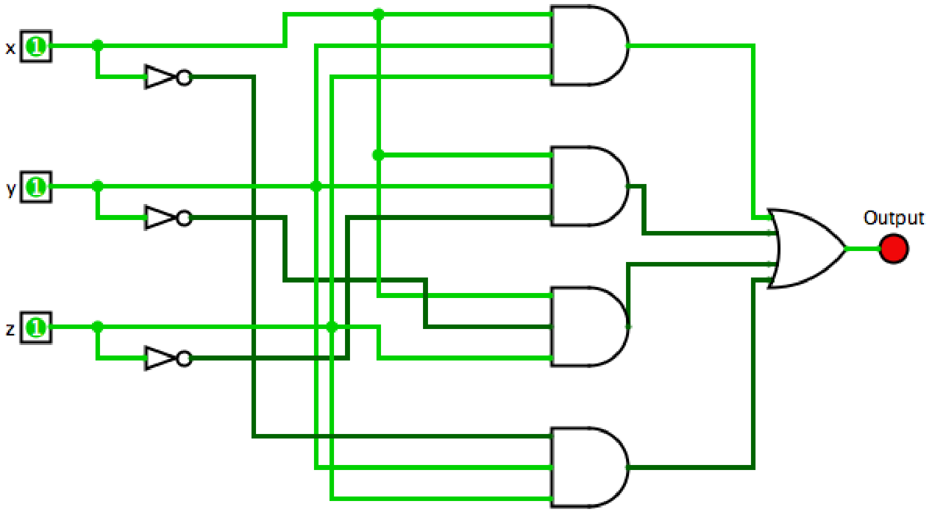
   If a variable is true in that row, write it down. Otherwise, if the variable is false, write down its negation. Then connect all variables or their negations together using AND.

   - $((x \wedge y) \wedge z)$
   - $((x \wedge y) \wedge (\neg z))$
   - $((x \wedge (\neg y)) \wedge z)$
   - $(((\neg x) \wedge y) \wedge z)$

   Connect all formulas into a disjunction.

   $(((((x \wedge y) \wedge z) \vee ((x \wedge y) \wedge (\neg z))) \vee ((x \wedge (\neg y)) \wedge z)) \vee (((\neg x) \wedge y) \wedge z))$

2. Draw the circuit.



   Making a circuit clear and readable can be challenging. Here are some advice on drawing circuits:

   - Determine where to put the inputs and the outputs first.
   - Determine where to put the major gates (the OR at the end, and one AND for each scenario).
   - Try to draw wires horizontally or vertically, not at an angle.
   - Indicate clearly whether two crossing wires are connected or not.

Solution 2:

1. Convert the truth table into a propositional formula.

   Converts rows 1-3 to a propositional formula.

   $(x \land (y \lor z))$

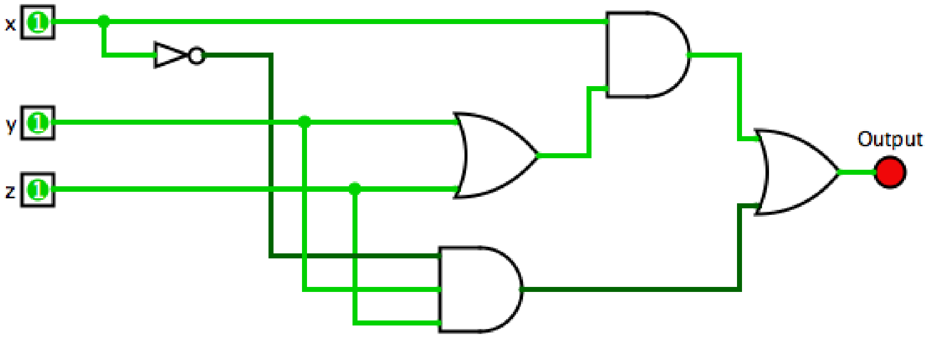
   Convert row 5 to a propositional formula.

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

   Connect all formulas into a disjunction.

   $((x \land (y \lor z)) \lor (((\neg x) \land y) \land z))$

2. Draw the circuit.



Solution 3:

1. Convert the truth table into a propositional formula.

   Convert rows 1 and 5 into a propositional formula.

   $(y \wedge z)$

   Convert rows 2 and 3 into a propositional formula.

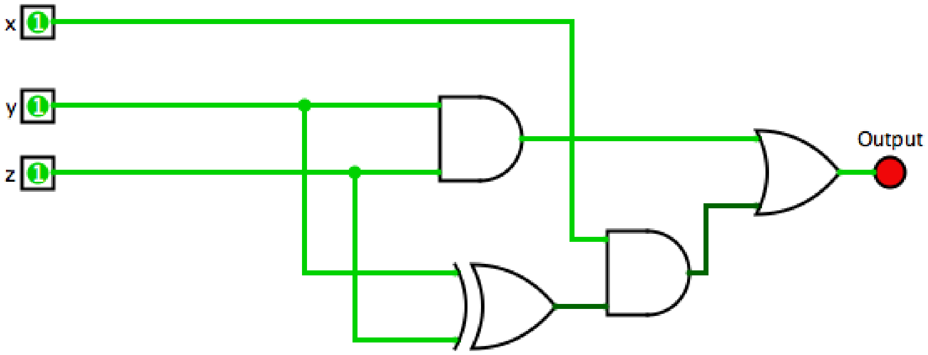
   $(x \wedge (y \oplus z))$

   For convenience, we will use the symbol $\oplus$ to represent an exclusive OR. However, you are only allowed to use this symbol in circuit design problems. You are not allowed to use this symbol for other problems because it is not a basic connective based on the definition of well-formed formulas.

   Connect all formulas into a disjunction.

   $((y \wedge z) \vee (x \wedge (y \oplus z)))$

2. Draw the circuit.



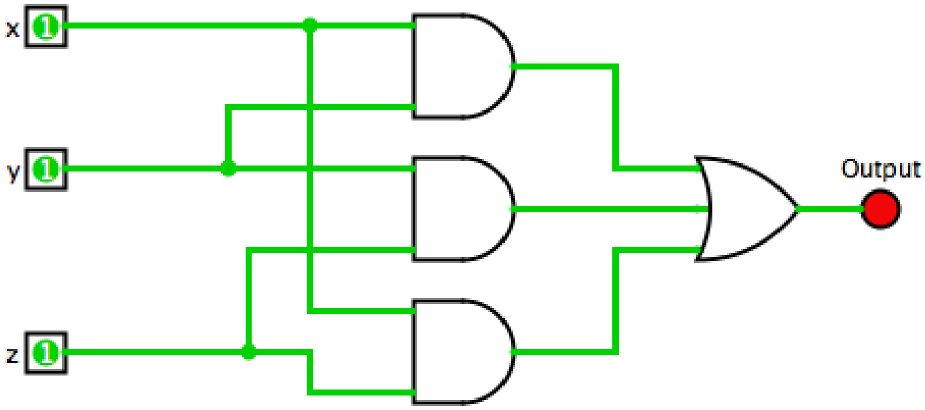
Solution 4 (contributed by Triman Kandola)

1. Convert the truth table into a propositional formula.

   $(((x \wedge y) \vee (x \wedge z)) \vee (y \wedge z))$

   This formula intuitively makes sense. If two people are already voting yes, then I don't care about what the third vote is.

2. Draw the circuit.



1.8 Semantic Entailment

Exercise 11. *Let $\Sigma = \{(p \to q), (q \to r)\}$. Is Σ satisfiable? Why or why not?*

Solution: Σ is satisfied by the truth valuation t where $t(p) = \mathtt{T}$, $t(q) = \mathtt{T}$ and $t(r) = \mathtt{T}$. Note that $(p \to q)^t = \mathtt{T}$ and $(q \to r)^t = \mathtt{T}$. Thus, Σ is satisfiable.

Exercise 12. *Let $\Sigma = \emptyset$. Is Σ satisfiable? Why or why not?*

Solution: Σ is satisfiable. In fact, any truth valuation satisfies Σ.
A truth valuation t satisfies Σ if and only if, for any formula α, if α is in Σ, then $\alpha^t = \mathtt{T}$. Since $\Sigma = \emptyset$, no formula is in Σ. The premise of the implication is false for any α, so the implication is true for every α. Therefore, any truth valuation satisfies $\Sigma = \emptyset$.

Exercise 13. *Let $\Sigma = \{p, (\neg p)\}$. Is Σ satisfiable? Why or why not?*

Solution: Σ is not satisfiable. To show this, we need to show that, under every truth valuation, at least one formula in Σ is false.

Consider an arbitrary truth valuation t. Under t, p is either true or false.

- If $p^t = \mathtt{T}$, then $(\neg p)^t = \mathtt{F}$. t does not satisfy Σ.
- If $p^t = \mathtt{F}$, then t does not satisfy Σ.

In both cases, t does not satisfy Σ. Therefore, no truth valuation can satisfy Σ. Σ is not satisfiable.

Exercise 14. *Prove that* $\{(\neg(p \land q)), (p \to q)\} \models (\neg p)$.

Solution:

Proof. Consider a truth valuation t such that $(\neg(p \land q))^t = \texttt{T}$ and $(p \to q)^t = \texttt{T}$.

Since $(p \to q)^t = \texttt{T}$, it is not the case that $p^t = \texttt{T}$ and $q^t = \texttt{F}$.
Since $(\neg(p \land q))^t = \texttt{T}$, it is not the case that $p^t = \texttt{T}$ and $q^t = \texttt{T}$.

Thus, the two premises are true under two scenarios:

- $p^t = \texttt{F}$ and $q^t = \texttt{T}$: In this case, $(\neg p)^t = \texttt{T}$.

- $p^t = \texttt{F}$ and $q^t = \texttt{F}$: In this case, $(\neg p)^t = \texttt{T}$.

In both scenarios, the conclusion is true. Thus, the entailment holds. □

Exercise 15. *Prove that* $\{(\neg(p \land q)), (p \to q)\} 6\models (p \leftrightarrow q)$.

Solution:

Proof. Consider the truth valuation t where $p^t = \texttt{F}$ and $q^t = \texttt{T}$.

By definitions of the connectives, $(\neg(p \land q))^t = \texttt{T}$, $(p \to q)^t = \texttt{T}$ and $(p \leftrightarrow q)^t = \texttt{F}$. Thus, the entailment does not hold. □

Exercise 16. *Prove that* $\emptyset \models ((p \land q) \to p))$.

Solution:

Proof. Since there is no premise, we need to prove that the conclusion $((p \land q) \to p))$ is a tautology.
Consider any truth valuation t. Under t, p must be either true or false.

- $p^t = \texttt{T}$: The conclusion of the implication $((p \land q) \to p))$is true. Therefore, the implication is true.

- $p^t = \texttt{F}$: The premise of the implication $((p \land q) \to p))$ is true. Therefore, the implication is true.

Thus, the conclusion is true under any truth valuation and is a tautology. The entailment holds. □

Exercise 17. *Prove that* $\{r, (p \to (r \to q))\} \vDash (p \to (q \land r))$.

Solution:

Proof. Consider a truth valuation t where $r^t = \mathtt{T}$ and $(p \to (r \to q))^t = \mathtt{T}$. We need to show that $(p \to (q \land r))^t = \mathtt{T}$.

Consider two cases: $p^t = \mathtt{F}$ and $p^t = \mathtt{T}$.

If $p^t = \mathtt{F}$, then $(p \to (q \land r))^t = \mathtt{T}$.

Otherwise, suppose that $p^t = \mathtt{T}$. We need to show that $(q \land r)^t = \mathtt{T}$.
By the definition of implication, $(r \to q)^t = \mathtt{T}$ since $(p \to (r \to q))^t = \mathtt{T}$. Since $r^t = \mathtt{T}$ and $(r \to q)^t = \mathtt{T}$, then $q^t = \mathtt{T}$ by the definition of implication. By the definition of $\land$, $(q \land r)^t = \mathtt{T}$ since q and r are both true under t. Therefore, $(p \to (q \land r))^t = \mathtt{T}$.

In both cases, the conclusion is true under t. The entailment holds. □

Exercise 18. *Prove that* $\{(\neg p), (q \to p)\} \not\vDash ((\neg p) \land q)$.

Solution:

Remark 1. *We need to come up with a truth valuation under which both premises are true and the conclusion is false.*
$(\neg p)$ *has to be true. So* p *has to be false under this truth valuation.*
$(q \to p)$ *has to be true and* p *is false. Thus,* q *must be false under this truth valuation.*
Therefore, this truth valuation must make p *false and* q *false.*

Proof. Consider the truth valuation t where $p^t = \mathtt{F}$ and $q^t - \mathtt{F}$.

Under this truth valuation, $(\neg p)^t = \mathtt{T}$ and $(q \to p)^t = \mathtt{T}$. Both premises are true.

Under this truth valuation, $((\neg p) \land q)^t = \mathtt{F}$. The conclusion is false.

Therefore, the entailment does not hold. □

Exercise 19. *Prove that* $\{p, (\neg p)\} \vDash r$.

Solution:

Proof. Consider any truth valuation t under which both premises are true. If such a truth valuation exists, we have to show that r must be true under this truth valuation.

However, such a truth valuation does not exist. There are two possible cases. p is true or p is false. If p is false, then this truth valuation does not satisfy the first premise. If p is true under this truth valuation, then $(\neg p)$ must be false. This truth valuation does not satisfy the second premise.

Since no truth valuation satisfies both premises, the entailment holds. □

1.9 Natural Deduction

1.9.1 Strategies for writing a natural deduction proof

General strategies:

- Write down all of the premises
- Leave plenty of space. Then write down the conclusion.
- Look at the conclusion carefully. What is the structure of the conclusion (what is the last connective applied in the formula? Can you apply an introduction rule to produce the conclusion?
- Look at each premise carefully. What is the structure of the premise (what is the last connective applied in the formula)? Can you apply an elimination rule to simplify it?
- Working backwards from the conclusion is often more effective than working forward from the premises. It keeps your eyes on the prize.
- If no rule is applicable, consider using a combination of ¬i and ¬¬e.

Working with subproofs

- To apply an introduction rule to produce the conclusion, **lay down the structure of the subproof before you proceed to fill in the subproof**. That is, draw the box for the subproof, write down the assumption on the first line, copy the conclusion to the last line of the subproof.
- Every subproof must be created to apply a particular rule. If you don't know what rule you are trying to apply, don't create a subproof.
- When filling in a subproof, you can use all the formulas that come before as long as the formula is not in a previous subproof that has already closed.
- Outside of a subproof, you have to use the subproof as a whole. You cannot use any individual formula in the subproof.

1.9.2 And elimination and introduction

Exercise 20. *Show that* $\{(p \land q), (r \land s)\} \vdash (q \land s)$*.*

Solution:

1.	$(p \land q)$	premise
2.	$(r \land s)$	premise
3.	q	$\land$e: 1
4.	s	$\land$e: 2
5.	$(q \land s)$	$\land$i: 3, 4

Exercise 21. *Show that* $((p \land q) \land r) \vdash (p \land (q \land r))$*.*

Solution:

1.	$((p \land q) \land r)$	premise
2.	$(p \land q)$	$\land$e: 1
3.	p	$\land$e: 2
4.	q	$\land$e: 2
5.	r	$\land$e: 1
6.	$(q \land r)$	$\land$i; 4, 5
7.	$(p \land (q \land r))$	$\land$i: 3, 6

1.9.3 Implication introduction and elimination

Exercise 22. *Show that* $\{(p \to q), (q \to r)\} \vdash (p \to r)$.

Solution:

1.	$(p \to q)$	premise
2.	$(q \to r)$	premise
3.	p	assumption
4.	q	→e: 1, 3
5.	r	→e: 2, 4
6.	$(p \to r)$	→i: 3–5

Exercise 23. *Show that* $\{(p \to (q \to r)), (p \to q)\} \vdash (p \to r)$.

Solution:

1.	$(p \to (q \to r))$	premise
2.	$(p \to q)$	premise
3.	p	assumption
4.	$(q \to r)$	→e: 1, 3
5.	q	→e: 2, 3
6.	r	→e: 4, 5
7.	$(p \to r)$	→i: 3-6

Exercise 24. *Show that* $\{(p \to (q \to r))\} \vdash ((p \land q) \to r)$.

Solution:

1.	$(p \to (q \to r))$	premise
2.	$(p \land q)$	assumption
3.	p	$\land$e: 2
4.	$(q \to r)$	$\to$e: 1, 3
5.	q	$\land$e: 2
6.	r	$\to$e: 4, 5
7.	$((p \land q) \to r)$	$\to$i: 2–6

Exercise 25. *Show that* $\{((p \land q) \to r)\} \vdash (p \to (q \to r))$.

Solution:

1.	$((p \land q) \to r)$	premise
2.	p	assumption
3.	q	assumption
4.	$(p \land q)$	$\land$i: 2, 3
5.	r	$\to$e: 1, 4
6.	$(q \to r)$	$\to$i: 3–5
7.	$(p \to (q \to r))$	$\to$i: 2–6

1.9.4 Or elimination and introduction

Exercise 26. *Show that* $\{(p \wedge (q \vee r))\} \vdash ((p \wedge q) \vee (p \wedge r))$.

Solution:

1.	$(p \wedge (q \vee r))$	premise
2.	p	$\wedge$e: 1
3.	$(q \vee r)$	$\wedge$e: 1
4.	q	assumption
5.	$(p \wedge q)$	$\wedge$i: 2, 4
6.	$((p \wedge q) \vee (p \wedge r))$	$\vee$i: 5
7.	r	assumption
8.	$(p \wedge r)$	$\wedge$i: 2, 7
9.	$((p \wedge q) \vee (p \wedge r))$	$\vee$i: 8
10.	$((p \wedge q) \vee (p \wedge r))$	$\vee$e: 3, 4–6, 7–9

Exercise 27. *Show that* $\{((p \wedge q) \vee (p \wedge r))\} \vdash (p \wedge (q \vee r))$.

Solution:

1.	$((p \wedge q) \vee (p \wedge r))$	premise
2.	$(p \wedge q)$	assumption
3.	p	$\wedge$e: 2
4.	q	$\wedge$e: 2
5.	$(q \vee r)$	$\vee$i: 4
6.	$(p \wedge (q \vee r))$	$\wedge$i: 3, 5
7.	$(p \wedge r)$	assumption
8.	p	$\wedge$e: 7
9.	r	$\wedge$e: 7
10.	$(q \vee r)$	$\vee$i: 9
11.	$(p \wedge (q \vee r))$	$\wedge$i: 8, 10
12.	$(p \wedge (q \vee r))$	$\vee$e: 1, 2-6, 7-11

Exercise 28. *Show that* $\{(p \lor q)\} \vdash ((p \to q) \lor (q \to p))$.

Solution:

1.	$(p \lor q)$	premise
2.	p	assumption
3.	q	assumption
4.	p	Reflexivity: 2
5.	$(q \to p)$	$\to$i: 3–4
6.	$((p \to q) \lor (q \to p))$	$\lor$i: 5
7.	q	assumption
8.	p	assumption
9.	q	Reflexivity: 7
10.	$(p \to q)$	$\to$i: 8–9
11.	$((p \to q) \lor (q \to p))$	$\lor$i: 10
12.	$((p \to q) \lor (q \to p))$	$\lor$e: 1, 2–6, 7–11

Exercise 29. *Show that* $\{(p \to q)\} \vdash ((r \lor p) \to (r \lor q))$.

Solution:

1.	$(p \to q)$	premise
2.	$(r \lor p)$	assumption
3.	r	assumption
4.	$(r \lor q)$	$\lor$i: 3
5.	p	assumption
6.	q	$\to$e: 1, 5
7.	$(r \lor q)$	$\lor$i: 6
8.	$(r \lor q)$	$\lor$e: 2, 3–4, 5–7
9.	$((r \lor p) \to (r \lor q))$	$\to$e: 2–8

1.9.5 Negation introduction and double negation elimination

Exercise 30. *Show that* $\{(p \to (\neg p))\} \vdash (\neg p)$.

Solution:

1.	$(p \to (\neg p))$	premise
2.	p	assumption
3.	$(\neg p)$	$\to$e: 1, 2
4.	$\bot$	$\bot$i: 2, 3
5.	$(\neg p)$	$\neg$i, 2–4

Exercise 31. *Show that* $\{(p \to (q \to r)), p, (\neg r)\} \vdash (\neg q)$.

Solution:

1.	$(p \to (q \to r))$	premise
2.	p	premise
3.	$(\neg r)$	premise
4.	$(q \to r)$	$\to$e: 1, 2
5.	q	assumption
6.	r	$\to$e: 4, 5
7.	$\bot$	$\bot$i: 3, 6
8.	$(\neg q)$	$\neg$i: 5–7

Exercise 32. *Show that* $\{((\neg p) \to (\neg q))\} \vdash (q \to p)$.

Solution:

1.	$((\neg p) \to (\neg q))$	premise
2.	q	assumption
3.	$(\neg p)$	assumption
4.	$(\neg q)$	$\to$e: 1, 3
5.	$\bot$	$\bot$i: 2, 4
6.	$(\neg(\neg p))$	$\neg$i: 3-5
7.	p	$\neg\neg$e: 6
8.	$(q \to p)$	$\to$i: 2-7

Exercise 33. *Show that* $\{((p \wedge (\neg q)) \to r), (\neg r), p\} \vdash q$.

Solution:

1.	$((p \wedge (\neg q)) \to r)$	premise
2.	$(\neg r)$	premise
3.	p	premise
4.	$(\neg q)$	assumption
5.	$(p \wedge (\neg q))$	$\wedge$i: 3, 4
6.	r	$\to$e: 1, 5
7.	$\bot$	$\bot$i: 2, 6
8.	$(\neg(\neg q))$	$\neg$i: 4–7
9.	q	$\neg\neg$e: 8

1.9.6 Negation elimination

Exercise 34. *Show that* $\{(p \vee q), (\neg p)\} \vdash q$.

Solution:

1.	$(p \vee q)$	premise
2.	$(\neg p)$	premise
3.	p	assumption
4.	$\bot$	$\bot$i: 2, 3
5.	q	$\bot$e: 4
6.	q	assumption
7.	q	$\vee$e: 1, 3-5, 6

Exercise 35. *Show that* $\emptyset \vdash ((\neg p) \to (p \to (p \to q)))$.

Solution:

1.	$(\neg p)$	assumption
2.	p	assumption
3.	$\bot$	$\bot$i: 1, 2
4.	$(p \to q)$	$\bot$e: 3
5.	$(p \to (p \to q))$	$\to$i: 2-4
6.	$((\neg p) \to (p \to (p \to q)))$	$\to$i: 1-5

1.9.7 Putting them together!

Exercise 36. *Show that* $\{(p \to q), (\neg q)\} \vdash (\neg p)$.

Solution:

Exercise 37. *(Law of excluded middle)* $\emptyset \vdash (a \vee (\neg a))$.

Solution: Solution 1:

1.	$(\neg(a \vee (\neg a)))$	assumption
2.	a	assumption
3.	$(a \vee (\neg a))$	$\vee$i: 2
4.	$\perp$	$\neg$e: 1, 3
5.	$(\neg a)$	$\neg$i: 2–4
6.	$(a \vee (\neg a))$	$\vee$i: 5
7.	$\perp$	$\neg$e: 1, 6
8.	$(\neg(\neg(a \vee (\neg a))))$	$\neg$i: 1–7
9.	$(a \vee (\neg a))$	$\neg\neg$e: 8

Solution 2:

1.	$(\neg(a \vee (\neg a)))$	assumption
2.	$(\neg a)$	assumption
3.	$(a \vee (\neg a))$	$\vee$i: 2
4.	$\perp$	$\neg$e: 1, 3
5.	$(\neg(\neg a))$	$\neg$i: 2–4
6.	a	$\neg\neg$e: 5
7.	$(a \vee (\neg a))$	$\vee$i: 6
8.	$\perp$	$\neg$e: 1, 7
9.	$(\neg(\neg(a \vee (\neg a))))$	$\neg$i: 1–8
10.	$(a \vee (\neg a))$	$\neg\neg$e: 9

Exercise 38. *(De Morgan's Law) Show that* $\{(\neg(\alpha \vee \beta))\} \vdash ((\neg\alpha) \wedge (\neg\beta))$.

Solution:

1.	$(\neg(\alpha \vee \beta))$	premise
2.	α	assumption
3.	$(\alpha \vee \beta)$	$\vee$i: 2
4.	$\perp$	$\perp$i: 1, 3
5.	$(\neg\alpha)$	$\neg$i: 2–4
6.	β	assumption
7.	$(\alpha \vee \beta)$	$\vee$i: 6
8.	$\perp$	$\perp$i: 1, 7
9.	$(\neg\beta)$	$\neg$i: 6–8
10.	$((\neg\alpha) \wedge (\neg\beta))$	$\wedge$i: 5, 9

Exercise 39. *(De Morgan's Law) Show that* $\{((\neg\alpha) \wedge (\neg\beta))\} \vdash (\neg(\alpha \vee \beta))$.

Solution:

1.	$((\neg\alpha) \wedge (\neg\beta))$	premise
2.	$(\alpha \vee \beta)$	assumption
3.	α	assumption
4.	$(\neg\alpha)$	$\wedge$e: 1
5.	$\perp$	$\perp$i: 3, 4
6.	β	assumption
7.	$(\neg\beta)$	$\wedge$e: 1
8.	$\perp$	$\perp$i: 7, 8
9.	$\perp$	$\vee$e: 2, 3–6, 7–10
10.	$(\neg(\alpha \vee \beta))$	$\neg$i: 2–11

Exercise 40. *(De Morgan's Law) Show that* $\{((\neg\alpha) \vee (\neg\beta))\} \vdash (\neg(\alpha \wedge \beta))$.

Solution:

1.	$((\neg\alpha) \vee (\neg\beta))$	premise
2.	$(\neg\alpha)$	assumption
3.	$(\alpha \wedge \beta)$	assumption
4.	α	$\wedge$e: 3
5.	$\perp$	$\perp$i: 2, 4
6.	$(\neg(\alpha \wedge \beta))$	$\neg$i: 3-5
7.	$(\neg\beta)$	assumption
8.	$(\alpha \wedge \beta)$	assumption
9.	β	$\wedge$e: 8
10.	$\perp$	$\perp$i: 7, 9
11.	$(\neg(\alpha \wedge \beta))$	$\neg$i: 8-10
12.	$(\neg(\alpha \wedge \beta))$	$\vee$e: 1, 2-6, 7-11

Exercise 41. *(De Morgan's Law) Show that* $\{(\alpha \vee \beta)\} \vdash (\neg((\neg\alpha) \wedge (\neg\beta)))$.

Solution:

1.	$(\alpha \vee \beta)$	premise
2.	α	assumption
3.	$((\neg\alpha) \wedge (\neg\beta))$	assumption
4.	$(\neg\alpha)$	$\wedge$e: 3
5.	$\perp$	$\perp$i: 2, 4
6.	$(\neg((\neg\alpha) \wedge (\neg\beta)))$	$\neg$i: 3-5
7.	β	assumption
8.	$((\neg\alpha) \wedge (\neg\beta))$	assumption
9.	$(\neg\beta)$	$\wedge$e: 8
10.	$\perp$	$\perp$i: 7, 9
11.	$(\neg((\neg\alpha) \wedge (\neg\beta)))$	$\neg$i: 8-10
12.	$(\neg((\neg\alpha) \wedge (\neg\beta)))$	$\vee$e: 1, 2-6, 7-11

Exercise 42. *(De Morgan's Law) Show that* $\{(\neg(\alpha \wedge \beta))\} \vdash ((\neg\alpha) \vee (\neg\beta))$.

Solution:

1.	$(\neg(\alpha \wedge \beta))$	premise
2.	$(\neg((\neg\alpha) \vee (\neg\beta)))$	assumption
3.	$(\neg\alpha)$	assumption
4.	$((\neg\alpha) \vee (\neg\beta))$	$\vee$i: 3
5.	$\bot$	$\bot$i: 2, 4
6.	α	PBC: 3-5
7.	$(\neg\beta)$	assumption
8.	$((\neg\alpha) \vee (\neg\beta))$	$\vee$i: 7
9.	$\bot$	$\bot$i: 2, 8
10.	β	PBC: 7-9
11.	$(\alpha \wedge \beta)$	$\wedge$i: 6, 10
12.	$\bot$	$\bot$i: 1, 11
13.	$((\neg\alpha) \vee (\neg\beta))$	PBC: 2-12

Exercise 43. *Show that* $\{(\neg(p \to q))\} \vdash (q \to p)$.

Solution:

1.	$(\neg(p \to q))$	premise
2.	q	assumption
3.	$(\neg p)$	assumption
4.	p	assumption
5.	q	reflexive: 2
6.	$(p \to q)$	$\to$i: 4-5
7.	$\bot$	$\bot$i: 1, 6
8.	$(\neg(\neg p))$	$\neg$i: 3-7
9.	p	$\neg\neg$e: 8
10.	$(q \to p)$	$\to$i: 2-9

1.9.8 Putting them together: Additional exercises

Exercise 44. $\{(\neg(p \to q))\} \vdash p$.

Exercise 45. $\{((p \to q) \to p)\} \vdash p$.

Exercise 46. $\{((p \to q) \to q)\} \vdash ((\neg q) \to p)$.

Exercise 47. $\vdash ((p \to q) \vee (q \to r))$

Exercise 48. $\{(p \to (q \vee r))\} \vdash ((p \to q) \vee (p \to r))$.

1.9.9 Other problems

Exercise 49. *E4 Exercise 4: Prove that for any set of propositional formulas* Σ *and any propositional variables* p *and* q*, if* $\Sigma \vdash p$*, then* $\Sigma \vdash ((\neg p) \to q)$*.*

Solution:

Proof. Let Σ be a set of propositional formulas and let p and q be propositional variables. Assume that $\Sigma \vdash p$. This means that the following proof exists.

1.	Σ	premises
2.	...	...
3.	p	

Using the above proof, we will construct a natural deduction proof for $\Sigma \vdash ((\neg p) \to q)$.

1.	Σ	premises
2.	...	...
3.	p	
4.	$(\neg p)$	assumption
5.	$\perp$	$\perp$i: 3, 4
6.	q	$\perp$e: 5
7.	$((\neg p) \to q)$	$\to$i: 4-6

Therefore, $\Sigma \vdash ((\neg p) \to q)$ holds. □

1.10 Soundness and Completeness of Natural Deduction

1.10.1 The soundness of inference rules

Exercise 50. *The following inference rule is called Disjunctive syllogism.*

$$\frac{(\neg\alpha) \quad (\alpha \vee \beta)}{\beta} \textit{ Disjunctive syllogism}$$

where α and β are well-formed propositional formulas.

Prove that this inference rule is sound. That is, prove that the following semantic entailment holds.

$$\{(\neg\alpha), (\alpha \vee \beta)\} \vDash \beta$$

You must use **the definition of semantic entailment** *to write your proof. Do not use any other technique such as truth table, valuation tree, logical identities, natural deduction, soundness, or completeness.*

Solution:

Remark 2. *To prove that an entailment holds, we need to consider all truth valuations under which all of the premises are true. For each such truth valuation, we need to show that the conclusion is true.*

The proof typically looks like the following.

- *Consider a truth valuation t under which all of the premises are true.*
- *If premise 1 is true under t, then α must be ... under t and β must be ... under t. If premise 2 is true under t, then ...*
- *There are ... cases that we need to consider.*
- *Case 1: this case is impossible because .../... the conclusion is true under t.*
- *Case 2: ...*
- *The conclusion is true in every case. Therefore, the entailment holds.*

Proof. Consider a truth valuation t under which $(\neg\alpha)^t = \mathtt{T}$ and $(\alpha \vee \beta)^t = \mathtt{T}$. We need to show that $\beta^t = \mathtt{T}$.
Since $(\alpha \vee \beta)^t = \mathtt{T}$, at least one of α and β is true under t.
Since $(\neg\alpha)^t = \mathtt{T}$, α is false under t. Therefore, β must be true under t.
Therefore, the entailment holds. □

Exercise 51. *Consider the following inference rule:*

$$\frac{(\alpha \rightarrow \beta)}{(\beta \rightarrow \alpha)} \textit{ Flip the implication}$$

where α and β are well-formed propositional formulas.

Prove that this inference rule is NOT sound. That is, prove the following statement:

$$\{(\alpha \rightarrow \beta)\}\ 6\models (\beta \rightarrow \alpha)$$

You must use **the definition of semantic entailment** *to write your proof. Do not use any other technique such as truth table, valuation tree, logical identities, natural deduction, soundness, or completeness.*

Solution:

Remark 3. *To prove that an entailment does not hold, we need to find a concrete counterexample, which shows that, there is a truth valuation t under which all of the premises are true and the conclusion is false.*

A concrete counterexample consist of the following:

- *Choose concrete formulas for α and β. In the following proof, we let α be p and β be q where p and q are propositional variables.*
- *Choose a truth valuation t such that all the premises are true and the conclusion is false.*

Choosing a concrete formula for each symbol is important. In the proof below, if we do not assign concrete formulas to α and β, then we cannot make claims about their truth values under t. We want to find a truth valuation under which β is true and α is false. This is not possible if β is $(r \wedge (\neg r))$ and α is $(r \vee (\neg r))$.

The difficult part is coming up with a counterexample that works. After that, writing up the proof is straightforward.

Proof. To prove that the entailment does not hold, we need to find one counterexample.

Let p and q be two propositional variables. Let α be p and let β be q. Consider a truth valuation t under which $p^t = \mathtt{F}$ and $q^t = \mathtt{T}$.

Under t, the premise is true. $(\alpha \rightarrow \beta)^t = (p \rightarrow q)^t = \mathtt{T}$.
Under t, the conclusion is false. $(\beta \rightarrow \alpha)^t = (q \rightarrow p)^t = \mathtt{F}$.

We found a truth valuation under which the premise is true and the conclusion is false. Thus, the entailment does not hold. □

1.10.2 Soundness and Completeness of Natural Deduction

Exercise 52. *Prove or disprove this statement: If* $\{a, b\} \vdash c$*, then* $\emptyset \vDash ((a \wedge b) \to c)$*.* a*,* b*, and* c *are well-formed propositional formulas.*

Solution:

Remark 4. *The statement is an implication, and the premise and the conclusion of the implication differ in two ways. The premise is about the existence of a natural deduction proof, whereas the conclusion is about an entailment. Moreover, the premise has* a *and* b *on the left hand side, whereas the conclusion has everything on the right hand side. Thus, there are two ways for us to transform the premise into the conclusion.*

Approach 1:

A visual representation of approach 1:

$$\{a, b\} \vdash c \quad \to \quad \{a, b\} \vDash c \quad \to \quad \emptyset \vDash ((a \wedge b) \to c)$$

First, we transform $\{a, b\} \vdash c$ *(the existence of a proof) to* $\{a, b\} \vDash c$ *(an entailment) by using the soundness of natural deduction.*

Then, we move a *and* b *from the left hand side to the right hand side by proving that* $\{a, b\} \vDash c$ *are* $\emptyset \vDash ((a \wedge b) \to c)$ *equivalent by the definition of entailment.*

Approach 2:

A visual representation of approach 2:

$$\{a, b\} \vdash c \quad \to \quad \emptyset \vdash ((a \wedge b) \to c) \quad \to \quad \emptyset \vDash ((a \wedge b) \to c)$$

First, we move a *and* b *from the left hand side to the right hand side by proving that* $\{a, b\} \vdash c$ *and* $\emptyset \vdash ((a \wedge b) \to c)$ *are equivalent.*

Then, we transform $\emptyset \vdash ((a \wedge b) \to c)$ *(the existence of a proof) to* $\emptyset \vDash ((a \wedge b) \to c)$ *(an entailment) by the soundness of natural deduction.*

See the two proofs on the following page.

Proof 1. We will prove the statement.

Assume $\{a, b\} \vdash c$ holds.

By the soundness of natural deduction, the entailment $\{a, b\} \vDash c$ holds.

Consider a truth valuation t under which $a^t = \texttt{T}$ and $b^t = \texttt{T}$. We know that $c^t = \texttt{T}$ by $\{a, b\} \vDash c$. Therefore, by the definition of an implication, we know that $((a \wedge b) \to c)$ is a tautology.

Consider a truth valuation t. There is no formula in $\emptyset$. Thus, t satisfies $\emptyset$. t also satisfies $((a \wedge b) \to c)$ since $((a \wedge b) \to c)$ is a tautology. Therefore, the entailment $\emptyset \vDash ((a \wedge b) \to c)$ holds.

□

Proof 2. We will prove the statement.

Assume $\{a, b\} \vdash c$ holds. Thus, there is a natural deduction proof which starts with a and b as the premises and ends with c.

1.	a	premise
2.	b	premise
3.	...	...
4.	c	...

We construct a natural deduction proof for $\emptyset \vdash ((a \wedge b) \to c)$ as follows.

1.	$(a \wedge b)$	assumption
2.	a	$\wedge$e: 1
3.	b	$\wedge$e: 1
4.	...	...
5.	c	...
6.	$((a \wedge b) \to c)$	$\to$i: 1-5

This proof shows that $\emptyset \vdash ((a \wedge b) \to c)$ holds.

By the soundness of natural deduction, the entailment $\emptyset \vDash ((a \wedge b) \to c)$ holds.

□

Exercise 53. *Prove or disprove this statement: If $\{\alpha\} \models \beta$, then $\emptyset \vdash (\beta \to \alpha)$. α and β are well-formed propositional formulas.*

Solution:

Remark 5. *To show that the implication is false, we need to choose concrete formulas for α and β such that the premise is true and the conclusion is false.*

By inspecting the premise and the conclusion, we see that the concrete formulas need to make sure that α entails β, but β does not entail α.

Choosing α to be p and β to be $(p \vee q)$ satisfy both requirements.

Proof. We will disprove the statement.

Let p and q be two propositional variables. Let α be p and let β be $(p \vee q)$.

First, we prove that $\{\alpha\} \models \beta$ holds. Consider a truth valuation t under which α is true. This means that $p^t = \mathtt{T}$. Under t, β is true because $(p \vee q)^t = \mathtt{T}$. Therefore, the entailment $\{\alpha\} \models \beta$ holds.

Now, we prove that $\emptyset \nvdash (\beta \to \alpha)$ holds. To show that such a proof does not exist, it suffices to show that the corresponding entailment $\emptyset \models (\beta \to \alpha)$ does not hold. Then by the contrapositive of the soundness of natural deduction, we have that $\emptyset \nvdash (\beta \to \alpha)$ holds.

To prove that $\emptyset$ 6$\models$ $(\beta \to \alpha)$ (or $\emptyset$ 6$\models$ $((p \vee q) \to p)$), we consider a truth valuation t such that $p^t = \mathtt{F}$ and $q^t = \mathtt{T}$. Under t, $\beta^t = (p \vee q)^t = \mathtt{T}$ and $\alpha^t = p^t = \mathtt{F}$. Therefore, $\emptyset$ 6$\models$ $(\beta \to \alpha)$ holds.

□

2 Predicate Logic

2.1 Translations

Exercise 54. *Let the domain be the set of animals. Let $B(x)$ mean that x is a bear. Let $H(x)$ mean that x likes honey.*

Translate "every bear likes honey" into predicate logic.

Solution: People often come up with the following two translations. See the formulas and the corresponding explanations below.

- $(\forall x\ (B(x) \wedge H(x)))$

  This formula says that every animal x is a bear and likes honey.

  This formula is an incorrect translation. The original sentence does not require every animal to be a bear. The sentence simply ignores any animal that is not a bear and focuses on animals that are bears.

- $(\forall x\ (B(x) \to H(x)))$

  This formula says that for every animal x, if x is a bear, then x likes honey.

  This is a correct translation. If an animal is a bear, then it must like honey as required by the original sentence. If an animal is not a bear, then the premise of the implication is false, which means that the implication is vacuously true. (In other words, we don't care about animals that are not bears.)

To differentiate between two predicate formulas, it is often a useful exercise to come up with a domain for which one formula is true and the other formula is false.

Consider a domain, which contains a bear A who likes honey and a rabbit B.

- For this domain, the first formula is false. When x is rabbit B, x is not a bear.
- For this domain, the second formula is true. When x is bear A, it likes honey, so the implication is true. When x is rabbit B, it is not a bear, so the implication is vacuously true. Since the implication is true for every element of the domain, the formula is true.

In general, consider a domain D and a predicate $P(x)$.
The following sentence

> "All <things in D for which P is true> have the property Q."

translates into the formula

$$(\forall x\ (P(x) \to Q(x))).$$

Exercise 55. *Let the domain be the set of animals. Let $B(x)$ mean that x is a bear. Let $H(x)$ mean that x likes honey.*

Translate "some bear likes honey" into predicate logic.

Solution: People often come up with the following two translations. See the formulas and the corresponding explanations below.

- $(\exists x\ (B(x) \wedge H(x)))$

  This formula says that there is an animal x, which is a bear and likes honey.

  This formula is the correct translation. The original sentence requires that there is a bear in the domain. Furthermore, it requires that there is a bear in the domain that likes honey. This formula guarantees both.

- $(\exists x\ (B(x) \rightarrow H(x)))$

  This formula says that there is an animal x, which is either not a bear, or is a bear and likes honey.

  This sentence is an incorrect translation, although many people think that it makes intuitive sense. The problem with this formula comes from the fact that the implication is vacuously true when the premise is false. This formula does not guarantee that there has to be a bear in the domain. As soon as we find an animal that is not a bear in the domain, the premise of the implication is false and the implication is vacuously true. This does not correspond to the original sentence, which requires that there is a bear in the domain.

To differentiate these two formulas, let's consider a domain, which contains a rabbit B. For this domain, the original sentence should be false because there is no bear.

- For this domain, the first formula is false. We cannot find a bear in the domain, which is required by the formula.
- For this domain, the second formula is true. When x is rabbit B, B is not a bear, so the premise of the implication is false. Thus, the implication is vacuously true. Since we have found an animal which makes the implication true, the formula is true.

In general, consider a domain D and a predicate $P(x)$.
The following sentence

> "Some <thing in D for which P is true> have the property Q."

translates into the formula

$$(\exists x\ (P(x) \wedge Q(x))).$$

Based on the two exercises above, could you summarize the general patterns of translations? Which binary connectives usually go with the universal and the existential quantifiers?

As a general rule of thumb, the universal quantifier is often used in conjunction with the implication ($\rightarrow$), and the existential quantifier is often used in conjunction with the conjunction ($\wedge$). We've seen examples of both above.

The universal quantifier

- $\forall$ and $\rightarrow$: This universal quantifier pairs well with the implication. This combination is used to make a statement about a subset of the domain. Therefore, we use the premise of the implication to restrict our attention to this subset. We don't have to worry about any element that is not in this subset because the implication is vacuously true for any such element.

- $\forall$ and $\wedge$: This combination is not impossible. However, it is a very strong statement. This combination is claiming that every element of the domain must satisfy the properties connected by the $\wedge$. If this is what you meant to express, then go ahead and use this combination.

The existential quantifier

- $\exists$ and $\wedge$: The existential quantifier pairs well with the conjunction. This combination can be used to express the fact that there exists an element of domain which has the two properties connected by the conjunction.

- $\exists$ and $\rightarrow$: This combination does not make sense logically. The main reason is that it is too easy to make such a formula true. As soon as we find an element of the domain, which makes the premise of the implication false, the implication is vacuously true and the formula is true as well.

Exercise 56. *Translate the following sentences into predicate formulas.*

Let the domain contain the set of all students and courses. Define the following predicates:
$C(x)$: x is a course.
$S(x)$: x is a student.
$T(x, y)$: student x has taken course y.

1. Every student has taken some course.

   Solution: $(\forall x\ (S(x) \to (\exists y\ (C(y) \land T(x, y)))))$

2. A student has taken a course.

   Solution: $(\exists x\ (S(x) \land (\exists y\ (C(y) \land T(x, y)))))$

3. No student has taken every course.

   Solution: $(\neg(\exists x\ (S(x) \land (\forall y\ (C(y) \to T(x, y))))))$

4. Some student has not taken any course.

   Solution: $(\exists x\ (S(x) \land (\forall y\ (C(y) \to (\neg T(x, y))))))$

5. Every student has taken every course.

   Solution: $(\forall x\ (S(x) \to (\forall y\ (C(y) \to T(x, y)))))$

Exercise 57. *Translating "at least", "at most", and "exactly". Translate the following sentences into predicate formulas.*

- There is at least one bear.

  Solution:

  $$(\exists x\ B(x))$$

- There are at least two bears.

  Solution:

  $$(\exists x\ (\exists y\ ((B(x) \wedge B(y)) \wedge (x \neq y))))$$

  The formula says: there are two bears x and y, and x and y must be different. Note that, if we don't have $(x \neq y)$, the formula only guarantees that there exists one bear because x and y could refer to the same animal in the domain.

- There is at most one bear.

  Solution:

  $$(\neg(\exists x\ (\exists y\ ((B(x) \wedge B(y)) \wedge (x \neq y)))))$$

  The negation of "at most one" is "at least two". Therefore, the sentence is equivalent to "It is not the case that there exist two different bears".

  Using the generalized De Morgan's laws, we can show that the above formula is logically equivalent to the formula below.

  $$(\forall x\ (\forall y\ ((B(x) \wedge B(y)) \to (x = y))))$$

  This formula says that: If we can find two bears x and y, then x and y must refer to the same bear. To understand this formula, imagine that I made the claim that there is at most one bear. Then your goal is to disprove my claim. You find two bears in the domain and show them to me. For my claim to be true, I have to be able to prove that the two bears you found are actually the same bear. I have to be able to do this no matter which two bears you show to me.

  Yet another translation is that: ((there is no bear) or (there is exactly one bear)). We can use any translation of "there is exactly one bear" on the next page.

  $$((\forall x\ (\neg B(x))) \vee (\exists y\ (B(y) \wedge (\forall z\ (B(z) \to (y = z))))))$$

- There is exactly one bear.

  Solution: One translation is: there is at least one bear and there is at most one bear.

  $$((\exists z\ B(z)) \wedge ((\neg(\exists x\ (\exists y\ ((B(x) \wedge B(y)) \wedge (x \neq y)))))))$$

  Another translation: there is at least one bear and if there is another bear, then the two bears must be the same.

  $$(\exists x\ (B(x) \wedge (\forall y\ (B(y) \rightarrow (x = y)))))$$

2.2 Semantics of Predicate Formulas

Consider this language of predicate logic:

- Constant symbols: a, b, c
- Variable symbols: x, y, z
- Function symbols: $f^{(1)}, g^{(2)}$
- Predicate symbols: $P^{(1)}, Q^{(2)}$

2.2.1 Evaluating Formulas with No Variables

Exercise 58. *Give an interpretation I such that $Q(f(c), a)^I = \mathtt{T}$ where $dom(I) = \{1, 2, 3\}$.*

Solution:

Remark 6. *I don't like to work with weird functions. So let's fix the function f to something simple first. Let f^I be $f^I(x) = x, \forall x \in dom(I)$. Given this, we simplify the formula below.*

$$f(c)^I = f^I(c^I) = c^I \tag{97}$$

$$Q(f(c), a)^I = Q(c, a)^I \tag{98}$$

The function g does not appear in the formula. Nevertheless, we need to give it an interpretation. Let g^I be $g^I(x) = 1, \forall x \in dom(I)$.

I like to deal with the predicates last. So let's assign meanings to the constant symbols. Let $c^I = 1$ and $a^I = 2$. Then, we have that $Q(c, a)^I$ is true if and only if $\langle 1, 2\rangle \in Q^I$.
The constant symbol b does not appear in the formula, but we still need to give it an interpretation. Let $b^I = 1$.

Finally, let's define Q^I. Above the above analysis, at a minimum, we need $\langle 1, 2\rangle \in Q^I$. We could include other tuples in Q^I if we like, but they don't affect the truth value of this formula. Thus, let $Q^I = \{\langle 1, 2\rangle\}$.
P does not appear in the formula, but we still need to give it an interpretation. Let P^I be the empty set.

Solution Text: The interpretation I is given below.

- $dom(I) = \{1, 2, 3\}$.
- $a^I = 2, b^I = 1, c^I = 1$.
- $f^I(x) = x, \forall x \in dom(I)$, $g^I(x) = 1, \forall x \in dom(I)$.
- $Q^I = \{\langle 1, 2\rangle\}$, $P^I = \emptyset$.

Therefore, $Q(f(c), a)^I = \mathtt{T}$ since all of the following hold:

$$f(c)^I = f^I(1) = 1 \tag{99}$$

$$a^I = 2 \tag{100}$$

$$\langle 1, 2\rangle \in Q^I. \tag{101}$$

Exercise 59. *Give an interpretation I such that $Q(f(c), a)^I = \mathtt{F}$.*

Solution:

Remark 7. *All we need to do is make one small adjustment to the interpretation in exercise 58.*

To make the formula false, we need to make sure the tuple $\langle 1, 2\rangle$ is not in Q^I. Let Q^I be the empty set.

Solution Text: The modified interpretation I is given below.

- $dom(I) = \{1, 2, 3\}$.
- $a^I = 2, b^I = 1, c^I = 1$.
- $f^I(x) = x, \forall x \in dom(I)$, $g^I(x) = 1, \forall x \in dom(I)$.
- $Q^I = \emptyset$, $P^I = \emptyset$.

Therefore, $Q(f(c), a)^I = \mathtt{F}$ since all of the following hold:

$$f(c)^I = f^I(1) = 1 \tag{102}$$

$$a^I = 2 \tag{103}$$

$$\langle 1, 2\rangle \notin Q^I. \tag{104}$$

2.2.2 Evaluating Formulas with Free Variables Only

Exercise 60. *Give an interpretation I and an environment E such that $Q(f(x), a)^{(I,E)} = \mathtt{T}$.*

Solution:

Remark 8. *Let's start with the interpretation in the solution to exercise 58. Consider an arbitrary environment E. We simplify the formula below.*

$$f(x)^{(I,E)} = f^I(E(x)) = E(x) \tag{105}$$

$$a^I = 2 \tag{106}$$

Thus, the formula is true if and only if $\langle E(x), 2\rangle \in Q^I$.

The only tuple in Q^I is $\langle 1, 2\rangle$. Thus, it is sufficient to let $E(x) = 1$. Even though y and z do not appear in the formula, we still need to include mappings for them in the definition of the environment. Let $E(y) = 1$ and $E(z) = 1$.

Solution Text: The interpretation I and the environment E are given below.

- $dom(I) = \{1, 2, 3\}$.
- $a^I = 2, b^I = 1, c^I = 1$.
- $f^I(x) = x, \forall x \in dom(I)$, $g^I(x) = 1, \forall x \in dom(I)$.
- $Q^I = \{\langle 1, 2\rangle\}$, $P^I = \emptyset$.

The environment E is $E(x) = 1$, $E(y) = 1$, and $E(z) = 1$.
Given I and E, we can show that $Q(f(x), a)^{(I,E)} = \mathtt{T}$ because

$$E(x) = 1 \tag{107}$$

$$f(x)^{(I,E)} = f^I(1) = 1 \tag{108}$$

$$a^I = 2 \tag{109}$$

$$\langle 1, 2\rangle \in Q^I. \tag{110}$$

Exercise 61. *Give an interpretation I such that $Q(f(x), a)^{(I,E)} = \texttt{F}$.*

Solution:

Remark 9. *Let's start with the interpretation in the solution to exercise 60, and modify Q^I to be the empty set.*
Under I and E, the formula is false, using similar reasoning as exercise 59.

Solution Text: The interpretation I and the environment E are given below.

- $dom(I) = \{1, 2, 3\}$.
- $a^I = 2, b^I = 1, c^I = 1$.
- $f^I(x) = x, \forall x \in dom(I)$, $g^I(x) = 1, \forall x \in dom(I)$.
- $Q^I = \{\langle 1, 2\rangle\}$, $P^I = \emptyset$.

The environment E is $E(x) = 1$, $E(y) = 1$, and $E(z) = 1$.

Given I and E, we can show that $Q(f(x), a)^{(I,E)} = \texttt{F}$ because

$$E(x) = 1 \tag{111}$$

$$f(x)^{(I,E)} = f^I(1) = 1 \tag{112}$$

$$a^I = 2 \tag{113}$$

$$\langle 1, 2\rangle \notin Q^I. \tag{114}$$

2.2.3 Evaluating Formulas with Free and Bound Variables

Exercise 62. *Give an interpretation I and an environment E such that $(\exists x\ Q(x,y))^{(I,E)} =$ $\mathtt{T}$. Assume that the domain is $dom(I) = \{1,2,3\}$.*

Solution:

Remark 10. *Here is more explanation to help you understand how I came up with the I and E above.*
y is a free variable in α. The value of y is given by the environment. Let's arbitrarily define $E(y) = 2$. Even though x and z do not appear in the formula, we need to define their mappings as part of E. Let $E(x) = 1$ and $E(z) = 1$.

To make α true, there must be at least one tuple in Q^I and the second value in the tuple (the value of y in the tuple) must be 2 because $E(y) = 2$. Let Q^I be $\{\langle 1,2\rangle\}$.

We need to define the rest of I even though the symbols do not appear in the formula.

Solution Text: The interpretation I is shown below.

- $dom(I) = \{1,2,3\}$.
- $a^I = 2, b^I = 1, c^I = 1$.
- $f^I(x) = x, \forall x \in dom(I)$, $g^I(x) = 1, \forall x \in dom(I)$.
- $Q^I = \{\langle 1,2\rangle\}$, $P^I = \emptyset$.

The environment E is $E(x) = 1, E(y) = 2, E(z) = 1$.
Given the I and E above, we know that $Q(x,y)^{(I,E[x\mapsto 1])} = \mathtt{T}$ because all of the following hold:

$$E[x \mapsto 1](x) = 1 \tag{115}$$

$$E[x \mapsto 1](y) = 2 \tag{116}$$

$$\langle 1,2\rangle \in Q^I \tag{117}$$

Hence, by the satisfaction rules for $\exists$, $(\exists x\ Q(x,y))^{(I,E)} = \mathtt{T}$.

Exercise 63. *Give an interpretation I and an environment E such that $(\forall x\ Q(x,y))^{(I,E)} = \mathtt{T}$. Assume that the domain is $dom(I) = \{1,2,3\}$.*

Solution:

Remark 11. *Let's start with the I and E in exercise 62.*

We will modify Q^I. To make the formula true, we must be able to replace x by any value in the domain. Furthermore, for each tuple in Q^I, the second value in the tuple must be 2 because the environment maps y to 2. Thus, let $Q^I = \{\langle 1,2\rangle, \langle 2,2\rangle, \langle 3,2\rangle\}$.

Solution Text: The interpretation I is shown below.

- $dom(I) = \{1,2,3\}$.
- $a^I = 2, b^I = 1, c^I = 1$.
- $f^I(x) = x, \forall x \in dom(I)$, $g^I(x) = 1, \forall x \in dom(I)$.
- $Q^I = \{\langle 1,2\rangle, \langle 2,2\rangle, \langle 3,2\rangle\}$, $P^I = \emptyset$.

The environment E is $E(x) = 1, E(y) = 2, E(z) = 1$.

We will prove that $(\forall x\ Q(x,y))^{(I,E)} = \mathtt{T}$. Consider all possible values of x. By the definition of Q^I, the following statements hold.

- $[x \mapsto 1]$: $Q(x,y)^{(I,E[x\mapsto 1])} = \mathtt{T}$ because all of the following hold.

$$\begin{aligned}&E[x \mapsto 1](x) = 1\\&E[x \mapsto 1](y) = 2\\&\langle 1,2\rangle \in Q^I.\end{aligned}$$

- $[x \mapsto 2]$: $Q(x,y)^{(I,E[x\mapsto 2])} = \mathtt{T}$ because all of the following hold.

$$\begin{aligned}&E[x \mapsto 2](x) = 2\\&E[x \mapsto 2](y) = 2\\&\langle 2,2\rangle \in Q^I.\end{aligned}$$

- $[x \mapsto 3]$: $Q(x,y)^{(I,E[x\mapsto 3])} = \mathtt{T}$ because all of the following hold.

$$\begin{aligned}&E[x \mapsto 3](x) = 3\\&E[x \mapsto 3](y) = 2\\&\langle 3,2\rangle \in Q^I.\end{aligned}$$

Therefore, by the satisfaction rules for $\forall$, $(\forall x\ Q(x,y))^{(I,E)} = \mathtt{T}$.

2.2.4 Evaluating Formulas with Bound Variables Only

Exercise 64. *Give an interpretation I and an environment E such that $(\exists x(\forall y\, Q(x,y)))^{(I,E)} = \mathtt{T}$. Start with the domain $dom(I) = \{1,2,3\}$.*

Solution:

Remark 12. *To make the formula true, there must be at least 3 tuples in Q^I because y (the second value of each tuple) could take any of the 3 possible values in the domain.*

The first element of all three tuples must be the same because there must be one value for x that makes $Q(x,y)$ true.

Note that, when choosing the value of x, we do not know the value of y yet. Our choice of value for x cannot depend on the value of y.

One definition of Q^I that satisfies all these requirements is $Q^I = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle\}$.

Solution Text: The interpretation I is given below.

- $dom(I) = \{1,2,3\}$.
- $a^I = 2, b^I = 1, c^I = 1$.
- $f^I(x) = x, \forall x \in dom(I)$, $g^I(x) = 1, \forall x \in dom(I)$.
- $Q^I = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle\}$, $P^I = \emptyset$.

Let E be an arbitrary environment.

We will prove that $(\exists x(\forall y\; Q(x,y)))^{(I,E)} = \mathtt{T}$.

By the satisfaction rules of $\exists$, we need to show that $(\forall y\; Q(x,y))^{(I,E[x\mapsto d_x])} = \mathtt{T}$ for some $d_x indom(I)$.
Consider $d_x = 1$. We now need to show that $Q(x,y)^{(I,E[x\mapsto d_x][y\mapsto d_y)} = \mathtt{T}$ for every $d_y \in dom(I)$. Consider all possible values of y.

- $[y \mapsto 1]$: $Q(x,y)^{(I,E[x\mapsto 1][y\mapsto 1])} = \mathtt{T}$ because all of the following hold.

$$E[x \mapsto 1][y \mapsto 1](x) = 1 \tag{118}$$
$$E[x \mapsto 1][y \mapsto 1](y) = 1 \tag{119}$$
$$\langle E[x \mapsto 1][y \mapsto 1](x), E[x \mapsto 1][y \mapsto 1](y)\rangle = \langle 1,1\rangle \in Q^I. \tag{120}$$

- $[y \mapsto 2]$: $Q(x,y)^{(I,E[x\mapsto 1][y\mapsto 2])} = \mathtt{T}$ because all of the following hold.

$$E[x \mapsto 1][y \mapsto 2](x) = 1 \tag{121}$$
$$E[x \mapsto 1][y \mapsto 2](y) = 2 \tag{122}$$
$$\langle E[x \mapsto 1][y \mapsto 2](x), E[x \mapsto 1][y \mapsto 2](y)\rangle = \langle 1,2\rangle \in Q^I. \tag{123}$$

- $[y \mapsto 3]$: $Q(x,y)^{(I,E[x\mapsto 1][y\mapsto 3])} = \mathtt{T}$ because all of the following hold.

$$E[x \mapsto 1][y \mapsto 3](x) = 1 \tag{124}$$

$$E[x \mapsto 1][y \mapsto 3](y) = 3 \tag{125}$$

$$\langle E[x \mapsto 1][y \mapsto 3](x), E[x \mapsto 1][y \mapsto 3](y)\rangle = \langle 1, 3\rangle \in Q^I. \tag{126}$$

By the satisfaction rules of $\forall$, $(\exists x(\forall y\ Q(x,y)))^{(I,E[x\mapsto 1])} = \mathtt{T}$ holds. By the definition of $\exists$, $(\exists x(\forall y\ Q(x,y)))^{(I,E)} = \mathtt{T}$ holds.

Exercise 65. *Give an interpretation I and an environment E such that* $(\exists x(\forall y\, Q(x,y)))^{(I,E)} = \mathtt{F}$*. Start with the domain* $dom(I) = \{1,2,3\}$*.*

Solution:

Remark 13. *The formula has no free variables. The bound variables get their meanings through the quantifiers. Thus, there is no need to define an environment. We only need to define an interpretation to evaluate the formula.*

There are many ways to make the formula false. An easy solution is to let Q^I be the empty set. Then, $Q^I(x,y)$ is always false and the formula must be false as well.

If there are tuples in Q^I, we need to make sure that Q^I does not have three tuples such that the first value of all three tuples are the same and the second value in all three tuples are all different.

Solution Text:
The interpretation I is shown below.

- $dom(I) = \{1,2,3\}$.
- $a^I = 2, b^I = 1, c^I = 1$.
- $f^I(x) = x, \forall x \in dom(I)$, $g^I(x) = 1, \forall x \in dom(I)$.
- $Q^I = \{\langle 1,1\rangle, \langle 2,2\rangle, \langle 3,3\rangle\}$, $P^I = \emptyset$.

Let E be an arbitrary environment.

We will prove that $(\exists x(\forall y\ Q(x,y)))^{(I,E)} = \mathtt{F}$.

By the satisfaction rules for $\exists$, we need to show that $(\forall y\ Q(x,y))^{(I,E[x\mapsto d_x])} = \mathtt{F}$ holds for every $d_x \in dom(I)$.

Consider all possible values of x.

- $[x \mapsto 1]$:

  By the rules of satisfaction for $\forall$, to prove that $(\forall y\ Q(x,y))^{(I,E[x\mapsto 1])} = \mathtt{F}$, we need to prove that $Q(x,y)^{(I,E[x\mapsto 1][y\mapsto d_y])} = \mathtt{F}$ for some $d_y \in dom(I)$.

  $Q(x,y)^{(I,E[x\mapsto 1][y\mapsto 2])} = \mathtt{F}$ holds since all of the following statements hold.

$$E[x \mapsto 1][y \mapsto 2](x) = 1 \tag{127}$$
$$E[x \mapsto 1][y \mapsto 2](y) = 2 \tag{128}$$
$$\langle E[x \mapsto 1][y \mapsto 2](x), E[x \mapsto 1][y \mapsto 2](y)\rangle = \langle 1,2\rangle \notin Q^I \tag{129}$$

  Therefore, $Q(x,y)^{(I,E[x\mapsto 1][y\mapsto 2])} = \mathtt{F}$ holds, which means that $(\forall y\ Q(x,y))^{(I,E[x\mapsto 1])} = \mathtt{F}$ holds.

- $[x \mapsto 2]$:
  $Q(x,y)^{(I,E[x\mapsto 2][y\mapsto 1])} = \mathtt{F}$ holds because all of the following statements hold.

  $$E[x \mapsto 2][y \mapsto 1](x) = 2 \tag{130}$$
  $$E[x \mapsto 2][y \mapsto 1](y) = 1 \tag{131}$$
  $$\langle E[x \mapsto 2][y \mapsto 1](x), E[x \mapsto 2][y \mapsto 1](y)\rangle = \langle 2, 1\rangle \notin Q^I \tag{132}$$

  Therefore, $(\forall y\ Q(x,y))^{(I,E[x\mapsto 2])} = \mathtt{F}$ holds.

- $[x \mapsto 3]$:
  $Q(x,y)^{(I,E[x\mapsto 3][y\mapsto 1])} = \mathtt{F}$ holds because all of the following statements hold.

  $$E[x \mapsto 3][y \mapsto 1](x) = 3 \tag{133}$$
  $$E[x \mapsto 3][y \mapsto 1](y) = 1 \tag{134}$$
  $$\langle E[x \mapsto 3][y \mapsto 1](x), E[x \mapsto 3][y \mapsto 1](y)\rangle = \langle 3, 1\rangle \notin Q^I \tag{135}$$

  Therefore, $(\forall y\ Q(x,y))^{(I,E[x\mapsto 3])} = \mathtt{F}$ holds.

By the satisfaction rules of $\exists$, we have proven that $(\exists x(\forall y\ Q(x,y)))^{(I,E)} = \mathtt{F}$.

2.3 Semantic Entailment

Collected Wisdom:

- **Semantic entailment and natural deduction are two ways of proving the same argument.** Did you notice that Q1a of assignment 6 is the same as Q2d of assignment 7 (in Spring 2018)? We asked you prove the same argument, once with semantic entailment and once with natural deduction. **If you have trouble proving a statement using semantic entailment, you may want to try natural deduction first, and then convert it to a semantic entailment argument.**

Exercise 66. *Show that* $\{(\forall x\ P(x))\} \vDash (\exists x\ P(x))$.

Solution:

Proof. Consider an interpretation I such that $(\forall x\, P(x))^I = \mathtt{T}$. We will prove that $(\exists x\, P(x))^I = \mathtt{T}$.

Consider an arbitrary environment E. Let $d_1 \in dom(I)$ be a domain element.

By the satisfaction rules for $\forall$, $P(x)^{(I,E[x \mapsto d_1])} = \mathtt{T}$. Therefore, $E[x \mapsto d_1](x) = d_1 \in P^I$.

By the satisfaction rules for $\exists$, $(\exists x\ P(x))^I = \mathtt{T}$. □

Exercise 67. *Show that* $\{(\exists x\ P(x))\} \nvDash (\forall x\ P(x))$.

Solution:

Proof. To prove that the entailment does hold, we need to find an interpretation I such that $(\exists x\ P(x))^I = \mathtt{T}$ and $(\forall x\ P(x))^I = \mathtt{F}$.

Consider the interpretation I below.

- $dom(I) = \{1, 2\}$.
- $P^I = \{1\}$.

Let E be an arbitrary environment.

$P(x)^{(I,E[x \mapsto 1])} = \mathtt{T}$ holds since $E[x \mapsto 1](x) = 1 \in P^I$. By the satisfaction rules for $\exists$, $(\exists x\ P(x))^{(I,E)} = \mathtt{T}$.

$P(x)^{(I,E[x \mapsto 2])} = \mathtt{F}$ holds since $E[x \mapsto 2](x) = 2 \notin P^I$. By the satisfaction rules for $\forall$, $(\forall x\ P(x))^{(I,E)} = \mathtt{F}$ holds. □

Exercise 68. *Show that* $\{(\forall x\ (\alpha \rightarrow \beta))\} \vDash ((\forall x\ \alpha) \rightarrow (\forall x\ \beta))$*, where* x *is a variable symbol and* α *and* β *are well-formed predicate formulas.*

Solution:

Proof. Consider an interpretation I and an environment E such that $(\forall x\ (\alpha \rightarrow \beta))^{(I,E)} = \mathtt{T}$. We will prove that $((\forall x\ \alpha) \rightarrow (\forall x\ \beta))^{(I,E)} = \mathtt{T}$.

To show that $((\forall x\ \alpha) \rightarrow (\forall x\ \beta))^{(I,E)} = \mathtt{T}$, we assume that $(\forall x\ \alpha)^{(I,E)} = \mathtt{T}$.

By the satisfaction rule for $\forall$, we have that

$$\alpha^{(I,E[x \mapsto d])} = \mathtt{T} \text{ for every } d \in \mathit{dom}(I).$$

By our assumption, $(\forall x\ (\alpha \rightarrow \beta))^{(I,E)} = \mathtt{T}$. By the satisfaction rule for $\forall$, we have that

$$(\alpha \rightarrow \beta)^{(I,E[x \mapsto d])} = \mathtt{T} \text{ for every } d \in \mathit{dom}(I).$$

By the satisfaction rule for an implication, we have that

$$\beta^{(I,E[x \mapsto d])} = \mathtt{T} \text{ for every } d \in \mathit{dom}(I).$$

By the satisfaction rule for $\forall$, we have that

$$(\forall x\ \beta)^{(I,E)} = \mathtt{T}.$$

Thus, the entailment holds. □

Exercise 69. *Show that* $\{((\forall x\ \alpha) \to (\forall x\ \beta))\} \not\models (\forall x\ (\alpha \to \beta))$*, where* x *is a variable symbol and* α *and* β *are well-formed predicate formulas.*

Solution:

Remark 14. *The most important step for the proof below is to come up with the concrete example such that the premises are all true and the conclusion is false.*

I first chose concrete formulas for α *and* β*. This step is important. Without doing so, I may not be able to make claims about whether* α *and* β *are true or false under a particular interpretation.*

Next, I construct an interpretation to satisfy the two requirements. I start by picking a domain containing two elements. It is small enough to be manageable and large enough to give me a few possibilities to experiment with.

Then, I try to find definitions for P^I *and* Q^I *to satisfy the two requirements.*

First, I want to make the conclusion $(\forall x\ (P(x) \to Q(x)))$ *false. To do this, it is sufficient to make* P *to be true and* Q *to be false for one value of* x *(so that the implication* $(P(x) \to Q(x))$ *is false). I used* $x = 2$ *for this case and made sure that* $2 \in P^I$ *and* $2 \notin Q^I$*.*

Next, I want to make the premise true. Since $2 \notin Q^I$*, then* $(\forall x\ Q(x))$ *is false. So the conclusion of the premise is false. To make the premise true, I have to make the premise of the premise false. This means that, I need to make sure at least one domain element is not in* P^I*. Therefore, I defined* P^I *such that* $1 \notin P^I$*.*

Proof. Let α be $P(x)$ and let β be $Q(x)$, where P and Q are unary predicates. Consider the following interpretation:

- $dom(I) = \{1, 2\}$
- $P^I = \{2\}$ and $Q^I = \{1\}$

We need to show that $((\forall x\ P(x)) \to (\forall x\ Q(x)))^{(I,E)} = \mathtt{T}$ and $(\forall x\ (P(x) \to Q(x)))^{(I,E)} = \mathtt{F}$. Let E be an arbitrary environment.

First, we will show that $((\forall x\ P(x)) \to (\forall x\ Q(x)))^{(I,E)} = \mathtt{T}$.

$P(x)^{(I,E[x\mapsto 1])} = \mathtt{F}$ because $E[x \mapsto 1](x) = 1 \notin P^I$. By the satisfaction rule for $\forall$, $(\forall x\ P(x))^{(I,E)} = \mathtt{F}$.

By the satisfaction rule for an implication, $((\forall x\ P(x)) \to (\forall x\ Q(x)))^{(I,E)} = \mathtt{T}$ because $(\forall x\ P(x))^{(I,E)} = \mathtt{F}$.

Next, we will show that $(\forall x\ (P(x) \to Q(x)))^{(I,E)} = \mathtt{F}$.

$(P(x) \to Q(x))^{(I,E[x\mapsto 2])} = \mathtt{F}$ because $E[x \mapsto 2](x) = 2 \in P^I$ and $E[x \mapsto 2](x) = 2 \notin Q^I$.

By the satisfaction rule for $\forall$, $(\forall x\ (P(x) \to Q(x)))^{(I,E)} = \mathtt{F}$.

In summary, the entailment does not hold.

□

Exercise 70. *Show that* $\{(\exists y\ (\forall x\ Q(x,y)))\} \vDash (\forall x\ (\exists y\ Q(x,y)))$.

Solution:

Proof. Consider an interpretation I such that $(\exists y\ (\forall x\ Q(x,y)))^{(I,E)} = \texttt{T}$. We will prove that $(\forall x\ (\exists y\ Q(x,y)))^{(I,E)} = \texttt{T}$. Let E be an arbitrary environment.

By the satisfaction rules for $\exists$, we have

$$(\forall x\ Q(x,y))^{(I,E[y\mapsto d_y])} = \texttt{T} \text{ for some } d_y \in \mathit{dom}(I).$$

By the satisfaction rules for $\forall$, we have

$$Q(x,y)^{(I,E[y\mapsto d_y][x\mapsto d])} = \texttt{T}, \text{ for some } d_y \in \mathit{dom}(I) \text{ and for every } d \in \mathit{dom}(I).$$

Note that in the environment $E[y \mapsto d_y][x \mapsto d]$, the value of d_y was chosen first and does not depend on the value of d. Thus, the environment $E[y \mapsto d_y][x \mapsto d]$ is equivalent to the environment $E[x \mapsto d][y \mapsto d_y]$. It does not matter whether we chose the value for x or the value for y first. Therefore, we rewrite the formula above as follows.

$$Q(x,y)^{(I,E[x\mapsto d][y\mapsto d_y])} = \texttt{T}, \text{ for every } d \in \mathit{dom}(I) \text{ and for some } d_y \in \mathit{dom}(I).$$

By the satisfaction rule for $\exists$, we have

$$(\exists y\ Q(x,y))^{(I,E[x\mapsto d])} = \texttt{T} \text{ for every } d \in \mathit{dom}(I).$$

By the satisfaction rule for $\forall$, we have that

$$(\forall x\ (\exists y\ Q(x,y)))^{(I,E)} = \texttt{T}.$$

□

Exercise 71. *Show that* $\{(\forall x\ (\exists y\ Q(x,y)))\} \not\models (\exists y\ (\forall x\ Q(x,y)))$.

Solution:

Remark 15. *If I attempt to prove the entailment, what would happen?*

By the satisfaction rules for $\forall$*, we have*

$$(\exists y\ Q(x,y))^{(I,E[x\mapsto d_x])} = \mathtt{T} \textit{ for every } d_x \in dom(I).$$

By the satisfaction rules for $\exists$*, we have*

$$Q(x,y)^{(I,E[x\mapsto d_x][y\mapsto d_y])} = \mathtt{T}$$

for every $d \in dom(I)$ *and for some* $d_y \in dom(I)$ *where the value of* d_y *may depend on the value of* d_x*.*
Note that the value of d_y *may depend on the value of* d_x*. In other words, for every value of* d_x*, we may choose a different value of* d_y *to satisfy the formula. Therefore, we CANNOT switch the two overrides in the environment. The following formula is FALSE.*

$$\textit{A false formula: } Q(x,y)^{(I,E[y\mapsto d_y][x\mapsto d_x])} = \mathtt{T}$$

for some $d_y \in dom(I)$ *and for every* $d_x \in dom(I)$*.*

Proof. To prove that the entailment does not hold, we need to find an interpretation I such that $(\forall x\ (\exists y\ Q(x,y)))^{(I,E)} = \mathtt{T}$ and $(\exists y\ (\forall x\ Q(x,y)))^{(I,E)} = \mathtt{F}$.

Consider the interpretation I below.

- $dom(I) = \{1, 2\}$.
- $Q^I = \{\langle 1,1\rangle, \langle 2,2\rangle\}$.

First, we will show that $(\forall x\ (\exists y\ Q(x,y)))^{(I,E)} = \mathtt{T}$. Let E be an arbitrary environment. Consider all possible values of x.

- $[x \mapsto 1]$: $Q(x,y)^{(I,E[x\mapsto 1][y\mapsto 1])} = \mathtt{T}$ because

  $$\langle E[x\mapsto 1][y\mapsto 1](x), E[x\mapsto 1][y\mapsto 1](y)\rangle = \langle 1,1\rangle \in Q^I.$$

  By the satisfaction rule for $\exists$, $(\exists y\ Q(x,y))^{(I,E[x\mapsto 1])} = \mathtt{T}$.

- $[x \mapsto 2]$: $Q(x,y)^{(I,E[x\mapsto 2][y\mapsto 2])} = \mathtt{T}$ because

  $$\langle E[x\mapsto 2][y\mapsto 2](x), E[x\mapsto 2][y\mapsto 2](y)\rangle = \langle 2,2\rangle \in Q^I.$$

  By the satisfaction rule for $\exists$, $(\exists y\ Q(x,y))^{(I,E[x\mapsto 2])} = \mathtt{T}$.

Thus, by the satisfaction rule for $\forall$, $(\forall x\ (\exists y\ Q(x,y)))^{(I,E)} = \mathtt{T}$.

Next, we will show that $(\exists y\ (\forall x\ Q(x,y)))^{(I,E)} = \mathtt{F}$. Let E be an arbitrary environment. Consider all possible values of y.

- $[y \mapsto 1]$: $Q(x,y)^{(I,E[x\mapsto 2][y\mapsto 1])} = \mathtt{F}$ because

  $$\langle E[x \mapsto 2][y \mapsto 1](x), E[x \mapsto 2][y \mapsto 1](y)\rangle = \langle 2, 1\rangle \notin Q^I.$$

  By the satisfaction rule for $\exists$, $(\forall x\ Q(x,y))^{(I,E[y\mapsto 1])} = \mathtt{F}$.

- $[y \mapsto 2]$: $Q(x,y)^{(I,E[x\mapsto 1][y\mapsto 2])} = \mathtt{F}$ because

  $$\langle E[x \mapsto 1][y \mapsto 2](x), E[x \mapsto 1][y \mapsto 2](y)\rangle = \langle 1, 2\rangle \notin Q^I.$$

  By the satisfaction rule for $\forall$, $(\forall x\ Q(x,y))^{(I,E[y\mapsto 2])} = \mathtt{F}$.

Thus, by the satisfaction rule for $\forall$, $(\exists y\ (\forall x\ Q(x,y)))^{(I,E)} = \mathtt{F}$.

Hence, the entailment does not hold. □

Exercise 72. *Show that* $\{(\forall x\ (\exists y\ (P(x) \lor Q(y))))\} \vDash (\exists y\ (\forall x\ (P(x) \lor Q(y))))$.

Remark 16. *Wait a second! In exercise 71, didn't we just show that this entailment does NOT hold? Not quite. In exercise 71, we dealt with a generic predicate formula $Q(x, y)$ without knowing any additional information about the predicate. In this question, we are working with a much more concrete predicate formula $(P(x) \lor Q(y))$. It turns out that, having this concrete predicate formula allows us to prove the entailment.*

Remark 17. *Let's write out a proof sketch first.*

To prove that the conclusion is true, we need to find one value $d_y \in dom(I)$ for y such that $(P(x) \lor Q(y))$ is true for every possible value for x. The value of y only influences the $Q(y)$ part of the formula. Does there exist a value for y such that $Q(y)$ is true?

Let's suppose that we know that there is some $d_y \in dom(I)$ for y such that $Q(y)$ is true. Would this help us prove the conclusion? For sure. If $Q(y)$ is true for $y = d_y$, then $(P(x) \lor Q(y))$ must be true for $y = d_y$ regardless of the value of x. We just found a value for y which will make the conclusion true.

We know how to prove the conclusion for the case when $Q(y)$ for at least one value of y. What if $Q(y)$ is always false? Let's look at the premise. If $Q(y)$ is always false, for the premise to be true, $P(x)$ must be true for every possible value of x. If $P(x)$ is true for every possible value of x, then to prove that the conclusion is true, we could choose any value for y. For any value of y, $P(x)$ is true for any value of x, so $(P(x) \lor Q(y))$ must be true.

Proof. Consider an interpretation and environment (I, E) such that

$$(\forall x\ (\exists y\ (P(x) \lor Q(y))))^{(I,E)} = \mathtt{T}.$$

Consider two cases:
Case 1: Assume that $((\exists y\ Q(y))^{(I,E)} = \mathtt{T}$.
By the satisfaction rule for $\exists$, we have that

$$Q(y)^{(I,E[y\mapsto d_y])} = \mathtt{T}$$

for some $d_y \in dom(I)$.
By the definition of $\lor$, we have that

$$(P(x) \lor Q(y))^{(I,E[y\mapsto d_y][x\mapsto d_x])} = \mathtt{T}$$

for some $d_y \in dom(I)$ and for every $d_x \in dom(I)$.
By the satisfaction rule for $\forall$, we have that

$$(\forall x\, (P(x) \lor Q(y)))^{(I,E[y\mapsto d_y])} = \mathtt{T}$$

for some $d_y \in dom(I)$.

By the satisfaction rule for ∀, we have that

$$(\exists y\,(\forall x\,(P(x) \vee Q(y))))^{(I,E)} = \mathtt{T}.$$

Case 2: Assume that $((\exists y\ Q(y))^{(I,E)} = \mathtt{F}$.
By the satisfaction rule for ∀, we have that

$$(\exists y\ (P(x) \vee Q(y)))^{(I,E[x \mapsto d_x])} = \mathtt{T}.$$

for every $d_x \in dom(I)$.

□

2.3.1 Semantic Entailment - Additional Exercises

Exercise 73. $\{((\forall x\ P(x)) \vee (\forall x\ Q(x)))\} \vDash (\forall x\,(P(x) \vee Q(x)))$.

Exercise 74. $\{(\exists x\,(P(x) \rightarrow Q(x))), (\forall y\ P(y))\} \vDash (\exists x\ Q(x))$

Exercise 75. $\{((\exists x\ P(x)) \vee (\exists x\ Q(x)))\} \vDash (\exists x\,(P(x) \vee Q(x)))$.

2.4 Natural Deduction

- $\forall$e (analogous to $\wedge$e)
- $\forall$i (analogous to $\wedge$i)
  - We know nothing about the fresh variable u except that u is a domain element. (If u is special, our conclusion may not be valid.)
  - The fresh variable u cannot escape the subproof box. For example, we cannot conclude $\alpha[u/x]$ outside of the box.
  - When you choose the fresh variable u, make sure that it has not appears anywhere outside of the subproof box in the proof.
- $\exists$e (analogous to $\vee$e)
  - Proof by cases.
  - The conclusion may have nothing to do with the starting formula.
- $\exists$i (analogous to $\vee$i)

2.4.1 Forall-elimination

Exercise 76. *Show that* $\{P(t), (\forall x\ (P(x) \to (\neg Q(x))))\} \vdash (\neg Q(t))$.

Solution:

1.	$P(t)$	premise
2.	$(\forall x\ (P(x) \to (\neg Q(x))))$	premise
3.	$(P(t) \to (\neg Q(t)))$	$\forall$e: 2
4.	$(\neg Q(t))$	$\to$e: 1, 3

2.4.2 Exists-introduction

Exercise 77. *Show that* $\{(\neg P(y))\} \vdash (\exists x\ (P(x) \to Q(y)))$.

Solution:

1.	$(\neg P(y))$	premise
2.	$P(y)$	assumption
3.	$\perp$	$\perp$i: 1, 2
4.	$Q(y)$	$\perp$e: 3
5.	$(P(y) \to Q(y))$	$\to$i: 2-4
6.	$(\exists x\ (P(x) \to Q(y)))$	$\exists$i: 5

Exercise 78. *Show that* $\{(\forall x\ P(x))\} \vdash (\exists y\ P(y))$.

Solution:

1.	$(\forall x\ P(x))$	premise
2.	$P(u)$	$\forall$e: 1
3.	$(\exists y\ P(y))$	$\exists$i: 2

2.4.3 Forall-introduction

Exercise 79. *Show that* $\{(\forall x\ P(x))\} \vdash (\forall y\ P(y))$.

Solution:

1.	$(\forall x\ P(x))$	premise
2.	u fresh	assumption
3.	$P(u)$	∀e: 1
4.	$(\forall y\ P(y))$	∀i: 2-3

Exercise 80. *Show that* $\{(\forall x\ (P(x) \to Q(x))), (\forall x\ P(x))\} \vdash (\forall x\ Q(x))$.

Solution:

1.	$(\forall x\ (P(x) \to Q(x)))$	premise
2.	$(\forall x\ P(x))$	premise
3.	u fresh	assumption
4.	$P(u)$	∀e: 2
5.	$(P(u) \to Q(u))$	∀e: 1
6.	$Q(u)$	→e: 4, 5
7.	$(\forall x\ Q(x))$	∀i: 3-6

Exercise 81. *Show that* $\{(\forall x\ (P(x) \to Q(x)))\} \vdash ((\forall x\ P(x)) \to (\forall y\ Q(y)))$.

Solution:

1.	$(\forall x\ (P(x) \to Q(x)))$	premise
2.	$(\forall x\ P(x))$	assumption
3.	u fresh	assumption
4.	$P(u)$	∀e: 2
5.	$(P(u) \to Q(u))$	∀e: 1
6.	$Q(u)$	→e: 4, 5
7.	$((\forall y\ Q(y))$	∀i: 3-6
8.	$((\forall x\ P(x)) \to (\forall y\ Q(y)))$	→i: 2-7

2.4.4 Forall-introduction - Additional Exercises

Exercise 82. $\{(\forall x\,(\forall y\ P(x,y)))\} \vdash (\forall y\,(\forall x\ P(x,y)))$.

Exercise 83. $\{(\forall x\,((\neg P(x)) \wedge Q(x)))\} \vdash (\forall x\,(P(x) \to Q(x)))$.

Exercise 84. $\{(\forall x\,(P(x) \wedge Q(x)))\} \vdash (\forall x\,(P(x) \to Q(x)))$.

Exercise 85. $\{(\forall x\,(P(x) \wedge Q(x)))\} \vdash ((\forall x\ P(x)) \wedge (\forall x\ Q(x)))$.

Exercise 86. $\{((\forall x\ P(x)) \vee (\forall x\ Q(x)))\} \vdash (\forall x\,(P(x) \vee Q(x)))$.

Exercise 87. $\{(\forall x\,(P(x) \to Q(x)))\} \vdash ((\forall x\,(\neg Q(x))) \to (\forall x\,(\neg P(x))))$.

Exercise 88. $\{(\forall x\,(\forall y\,(R(x,y) \to R(y,x))))\} \vdash (\forall x\,(\forall y\,(R(y,x) \to R(x,y))))$.

Exercise 89. $\{(\forall x\,(\forall y\,(\forall z\,((R(x,y) \wedge R(y,z)) \to R(x,z))))), (\forall x\,(\neg R(x,x)))\}$
$\vdash (\forall x\,(\forall y\,(\forall z\,(\neg((R(x,y) \wedge R(y,z)) \wedge R(z,x))))))$.

Exercise 90. $\{(\forall x\,(\forall y\,(\forall z\,((R(x,y) \wedge R(x,z)) \to R(y,z))))), (\forall x\ R(x,x))\}$
$\vdash (\forall x\,(\forall y\,(\forall z\,((R(x,y) \wedge R(y,z)) \to R(x,z)))))$.

2.4.5 Exists-elimination

Exercise 91. *Show that* $\{(\exists x\ P(x))\} \vdash (\exists y\ P(y))$.

Solution:

1.	$(\exists x\ P(x))$	premise
2.	$P(u)$, u fresh	assumption
3.	$(\exists y\ P(u))$	∃i: 2
4.	$(\exists y\ P(y))$	∃e: 1, 2-3

Exercise 92. *Show that* $\{(\forall x\ (P(x) \to Q(x))), (\exists x\ P(x))\} \vdash (\exists x\ Q(x))$.

Solution:

1.	$(\forall x\ (P(x) \to Q(x)))$	premise
2.	$(\exists x\ P(x))$	premise
3.	$P(u)$, u fresh	assumption
4.	$(P(u) \to Q(u))$	∀e: 1
5.	$Q(u)$	→e: 3, 4
6.	$(\exists x\ Q(x))$	∃i: 5
7.	$(\exists x\ Q(x))$	∃e: 2, 3-6

Exercise 93. *Show that* $\{(\forall x\ (Q(x) \to R(x))), (\exists x\ (P(x) \wedge Q(x)))\} \vdash (\exists x\ (P(x) \wedge R(x)))$.

Solution:

1.	$(\forall x\ (Q(x) \to R(x)))$	premise
2.	$(\exists x\ (P(x) \wedge Q(x)))$	premise
3.	$(P(u) \wedge Q(u))$, u fresh	assumption
4.	$P(u)$	∧e: 3
5.	$Q(u)$	∧e: 3
6.	$(Q(u) \to R(u))$	∀e: 1
7.	$R(u)$	→e: 5, 6
8.	$(P(u) \wedge R(u))$	∧i: 4, 7
9.	$(\exists x\ (P(x) \wedge R(x)))$	∃i: 8
10.	$(\exists x\ (P(x) \wedge R(x)))$	∃e: 2, 3-9

2.4.6 Exists-Elimination - Additional Exercises

Exercise 94. $\{(\exists x\, (P(x) \to Q(x))), (\forall y\ P(y))\} \vdash (\exists x\ Q(x))$

Exercise 95. $\{(\exists x\, (\exists y\ P(x, y)))\} \vdash (\exists y\, (\exists x\ P(x, y)))$.

Exercise 96. $\{(\exists x\, ((\neg P(x)) \wedge (\neg Q(x))))\} \vdash (\exists x\, (\neg(P(x) \wedge Q(x))))$.

Exercise 97. $\{(\exists x\, ((\neg P(x)) \vee Q(x)))\} \vdash (\exists x\, (\neg(P(x) \wedge (\neg Q(x)))))$.

Exercise 98. $\{(\exists x\, (P(x) \wedge Q(x)))\} \vdash ((\exists x\ P(x)) \wedge (\exists x\ Q(x)))$.

Exercise 99. $\{((\exists x\ P(x)) \vee (\exists x\ Q(x)))\} \vdash (\exists x\, (P(x) \vee Q(x)))$.

2.4.7 Putting them together

Exercise 100. *Show that* $\{(\exists x\ P(x)), (\forall x\ (\forall y\ (P(x) \to Q(y))))\} \vdash (\forall y\ Q(y))$.

Solution: Solution 1:

1.	$(\exists x\ P(x))$	premise
2.	$(\forall x\ (\forall y\ (P(x) \to Q(y))))$	premise
3.	y_0 fresh	assumption
4.	$P(x_0)$, x_0 fresh	assumption
5.	$(\forall y\ (P(x_0) \to Q(y)))$	∀e: 2
6.	$(P(x_0) \to Q(y_0))$	∀e: 5
7.	$Q(y_0)$	→e: 4, 6
8.	$Q(y_0)$	∃e: 1, 4-7
9.	$(\forall y\ Q(y))$	∀i: 3-8

Solution 2:

1.	$(\exists x\ P(x))$	premise
2.	$(\forall x\ (\forall y\ (P(x) \to Q(y))))$	premise
3.	$P(x_0)$, x_0 fresh	assumption
4.	y_0 fresh	assumption
5.	$(\forall y\ (P(x_0) \to Q(y)))$	∀e: 2
6.	$(P(x_0) \to Q(y_0))$	∀e: 5
7.	$Q(y_0)$	→e: 3, 6
8.	$(\forall y\ Q(y))$	∀i: 4-7
9.	$(\forall y\ Q(y))$	∃e: 3-8

Exercise 101. *Show that* $\{(\exists y\ (\forall x\ P(x,y)))\} \vdash (\forall x\ (\exists y\ P(x,y)))$.

Solution:

1.	$(\exists y\ (\forall x\ P(x,y)))$	premise
2.	$(\forall x\ P(x,y_0))$, y_0 fresh	assumption
3.	x_0 fresh	assumption
4.	$P(x_0,y_0)$	∀e: 2
5.	$(\exists y\ P(x_0,y))$	∃i: 4
6.	$(\forall x\ (\exists y\ P(x,y)))$	∀i: 3-5
7.	$(\forall x\ (\exists y\ P(x,y)))$	∃e: 1, 2-6

Exercise 102. *Show that* $\{(\neg(\exists x\ P(x)))\} \vdash (\forall x\ (\neg P(x)))$. *(De Morgan)*

Solution:

1.	$(\neg(\exists x\ P(x))$	premise
2.	u fresh	assumption
3.	$P(u)$	assumption
4.	$(\exists x\ P(x))$	$\exists$i: 3
5.	$\perp$	$\perp$i: 1, 4
6.	$(\neg P(u))$	$\neg$i: 3-5
7.	$(\forall x\ (\neg P(x)))$	$\forall$i: 2-6

Exercise 103. *Show that* $\{(\forall x\ (\neg P(x)))\} \vdash (\neg(\exists x\ P(x)))$. *(De Morgan)*

Solution:

1.	$(\forall x\ (\neg P(x)))$	premise
2.	$(\exists x\ P(x))$	assumption
3.	$P(u)$, u fresh	assumption
4.	$(\neg P(u))$	$\forall$e: 2
5.	$\perp$	$\perp$i: 3, 4
6.	$\perp$	$\exists$e: 2, 3-5
7.	$(\neg(\exists x\ P(x)))$	$\neg$i: 2-6

Exercise 104. *Show that* $\{(\exists x\ (\neg P(x)))\} \vdash (\neg(\forall x\ P(x)))$. *(De Morgan)*

Solution:

1.	$(\exists x\ (\neg P(x)))$	premise
2.	$(\forall x\ P(x))$	assumption
3.	$(\neg P(u))$, u fresh	assumption
4.	$P(u)$	$\forall$e: 2
5.	$\perp$	$\perp$i: 3, 4
6.	$\perp$	$\exists$e: 1, 3-5
7.	$(\neg(\forall x\ P(x)))$	$\neg$i: 2-6

Exercise 105. *Show that* $\{(\neg(\forall x\ P(x)))\} \vdash (\exists x\ (\neg P(x)))$. *(De Morgan)*

Solution:

1.	$(\neg(\forall x\ P(x)))$	premise
2.	$(\neg(\exists x\ (\neg P(x))))$	assumption
3.	u fresh	assumption
4.	$(\neg P(u))$	assumption
5.	$(\exists x\ (\neg P(x)))$	$\exists$i: 4
6.	$\perp$	2, 5
7.	$P(u)$	PBC: 4-6
8.	$(\forall x\ P(x))$	$\forall$i: 3-7
9.	$\perp$	$\perp$i: 1, 8
10.	$(\exists x\ (\neg P(x)))$	PBC: 2-9

2.4.8 Putting them together - Additional Exercises

Exercise 106. $\{(\forall x\,(P(x) \to (\neg Q(x))))\} \vdash (\neg(\exists x\,(P(x) \land Q(x))))$.

Exercise 107. $\{(\forall x\,(P(x) \lor Q(x)))\} \vdash ((\forall x\ P(x)) \lor (\exists x\ Q(x)))$.

Exercise 108. $\{(\forall x\,(P(x) \to (Q(x) \lor R(x)))), (\neg(\exists x\,(P(x) \land R(x))))\} \vdash (\forall x\,(P(x) \to Q(x)))$.

Exercise 109. $\{(\exists x\,(P(x) \land Q(x))), (\forall x\,(P(x) \to R(x)))\} \vdash (\exists x\,(R(x) \land Q(x)))$.

Exercise 110. $\{(\exists x\,(\exists y\,(S(x,y) \lor S(y,x))))\} \vdash (\exists x\,(\exists y\ S(x,y)))$.

Exercise 111. $\{(\forall x\,(\exists y\ R(x,y)))\} \vdash (\neg(\forall x\ R(x,x)))$.
This is false. Can you prove it?

Exercise 112. $\{(\forall x\,(\exists y\ R(x,y)))\} \vdash (\forall x\,(\exists y\,(\exists z\,(R(x,y) \land R(x,z)))))$.

Exercise 113. $\{(\forall x\,(P(x) \lor Q(x))), (\exists x\,(\neg Q(x))), (\forall x\,(R(x) \to (\neg P(x))))\} \vdash_{ND} (\exists x\,(\neg R(x)))$.

Exercise 114. $\emptyset \vdash (\exists y\,(R(y) \to (\forall x\ R(x))))$.

Exercise 115. $\{(\forall x\,(\exists y\,(P(x) \lor Q(y))))\} \vdash (\exists y\,(\forall x\,(P(x) \lor Q(y))))$.

Exercise 116. $\{(\forall x\,((\exists y\ P(y)) \to Q(x)))\} \vdash (\forall x\,(\exists y\,(P(y) \to Q(x))))$.

Exercise 117. $\{(\forall x\,(P(x,x) \lor (\forall y\ Q(x,y))))\} \vdash (\forall x\,((\exists y\ P(x,y)) \lor Q(x,x)))$.

Exercise 118. $\vdash ((\forall x\,(\exists y\ R(x,y))) \lor (\neg(\forall x\ R(x,x))))$.

2.5 Soundness and Completeness of Natural Deduction

2.5.1 Proving that an inference rule is sound or not sound

Lemma 1. *Let t be a predicate term. Let I be an interpretation with domain $dom(I)$. Let E be an environment. Then we have that*

$$t^{(I,E)} \in dom(I).$$

Lemma 2. *Let α be a well-formed predicate formula. Let t be a predicate term. Let I and E be an interpretation and environment. Let x be a variable. Then we have that*

$$\alpha[t/x]^{(I,E)} = \alpha^{(I,E[x \mapsto t^{(I,E)}])}.$$

Exercise 119. *Prove that the $\forall e$ inference rule is sound. That is, prove that the entailment holds:*

$$\{(\forall x\ \alpha)\} \vDash \alpha[t/x] \tag{136}$$

where α be a Predicate formula, x is a variable, and t is a Predicate term.

The proof sketch below is like an outline or a master plan. I will lay down the plan first. Then I will fill in the missing details.

Proof Sketch. Consider an interpretation and environment (I, E) such that $(\forall x\ \alpha)^{(I,E)} = \texttt{T}$. We need to show that $\alpha[t/x]^{(I,E)} = \texttt{T}$.

$(\forall x\ \alpha)^{(I,E)} = \texttt{T}$ holds because ...

$\alpha^{(I,E[x \mapsto t^{(I,E)}])} = \texttt{T}$ holds because ...

$\alpha[t/x]^{(I,E)} = \texttt{T}$ holds because ...

Thus, the entailment holds and the inference rule is sound. □

Solution: Let (I, E) be an interpretation and environment such that $(\forall x\ \alpha)^{(I,E)} = \texttt{T}$.

By the satisfaction rule for $\forall$, we have that $\alpha^{(I,E[x \mapsto d])} = \texttt{T}$, for every $d \in dom(I)$.

By Lemma 1, $t^{(I,E)}$ is some domain element. Thus, we have that $\alpha^{(I,E[x \mapsto t^{(I,E)}])} = \texttt{T}$.

By Lemma 2, we have that $\alpha[t/x]^{(I,E)} = \alpha^{(I,E[x \mapsto t^{(I,E)}])}$. Thus, we have that $\alpha[t/x]^{(I,E)} = \texttt{T}$.

Exercise 120. *Prove that the $\exists i$ inference rule is sound. That is, prove that the entailment holds:*

$$\{\alpha[t/x]\} \vDash (\exists x\ \alpha) \tag{137}$$

where α is a predicate formula, t is a predicate term, and x is a variable.

Proof Sketch. Consider an interpretation and environment (I, E) such that $\alpha[t/x]^{(I,E)} = \mathtt{T}$. We need to show that $(\exists x\ \alpha)^{(I,E)} = \mathtt{T}$.

$\alpha[t/x]^{(I,E)} = \mathtt{T}$ holds because ...

$\alpha^{(I,E[x \mapsto t^{(I,E)}])} = \mathtt{T}$ holds because ...

$(\exists x\ \alpha)^{(I,E)} = \mathtt{T}$ holds because ...

Thus, the entailment holds and the inference rule is sound. □

Solution: Let (I, E) be an interpretation and environment such that $\alpha[t/x]^{(I,E)} = \mathtt{T}$.

By Lemma 2, we have that $\alpha[t/x]^{(I,E)} = \alpha^{(I,E[x \mapsto t^{(I,E)}])}$. Thus, we have that $\alpha^{(I,E[x \mapsto t^{(I,E)}])} = \mathtt{T}$.

By Lemma 1, $t^{(I,E)}$ is some domain element. Thus, by the satisfaction rule for $\exists$, we have that $(\exists x\ \alpha)^{(I,E)} = \mathtt{T}$.

Exercise 121. *Prove that the following inference rule is NOT sound.*

$$\frac{\alpha[t/x]}{(\forall x\ \alpha)}\ \forall i* \tag{138}$$

where α is a predicate formula, t is a predicate term, and x is a variable.

Proof Sketch. Define the symbols in the language of Predicate logic that we consider.

Choose α to be a concrete Predicate formula. Choose t to be a concrete Predicate term.

Define an interpretation and an environment (I, E).

Show that $\alpha[t/x]^{(I,E)} = \mathtt{T}$.

Show that $(\forall x\ \alpha)^{(I,E)} = \mathtt{F}$. □

Solution:

Remark 18. *How did I come up with the interpretation and the environment below? Given the definition of $dom(I)$, E, and α and t, we can simplify the premise and the conclusion.*

The premise becomes $\alpha[t/x] = P(x)[y/x] = P(y)$.

The conclusion becomes $(\forall x\ P(x))$.

To make the premise true, we need to define $E(y)$ and P^I such that $P(y)$ is true. If we choose $E(y) = 1$, then we need $1 \in P^I$. If we choose $E(y) = 2$, then we need $2 \in P^I$. In the solution below, I chose $E(y) = 1$ and $1 \in P^I$.

To make the conclusion false, we need to make sure that $P(x)$ is false for one element of the domain. By the above definition, we already know that $P(x)$ is true when x is 1. The only other element of the domain is 2. So we need to make sure that $P(x)$ is false when x is 2, which means that $2 \notin P^I$.

Solution Text:
We need to provide an interpretation I and an environment E such that $\alpha[t/x]^{(I,E)} = \mathtt{T}$ and $(\forall x\ \alpha)^{(I,E)} = \mathtt{F}$.

Consider the language of predicate logic where $P^{(1)}$ is a unary predicate and x and y are variables.

Let α be $P(x)$ and let t be y. Let the interpretation I be defined below.

- $dom(I) = \{1, 2\}$
- $P^I = \{1\}$

Let the environment E be defined by $E(x) = 1$ and $E(y) = 1$.

First, we show that $\alpha[t/x]^{(I,E)} = \mathtt{T}$. By Lemma 2, $\alpha[t/x]^{(I,E)} = \alpha^{(I,E[x\mapsto t^{(I,E)}])}$. By the definition of the term t, $t^{(I,E)} = y^{(I,E)} = E(y) = 1$. Thus, $\alpha[t/x]^{(I,E)} = \alpha^{(I,E[x\mapsto 1])} = P(x)^{(I,E[x\mapsto 1])} = \mathtt{T}$ because $E[x \mapsto 1](x) = 1 \in P^I$.

Next, we show that $(\forall x\ \alpha)^{(I,E)} = \mathtt{F}$. By the satisfaction rule for $\forall$, we need to show that $\alpha^{(I,E[x\mapsto d])} = \mathtt{F}$ for at least one $d \in \mathit{dom}(I)$. We have that $\alpha^{(I,E[x\mapsto 2])} = P(x)^{(I,E[x\mapsto 2])} = \mathtt{F}$ because $E[x \mapsto 2](x) = 2 \notin P^I$.

Exercise 122. *Prove that the following inference rule is NOT sound.*

$$\frac{(\exists x\ \alpha)}{\alpha[t/x]}\ \exists e* \tag{139}$$

where α is a predicate formula, t is a predicate term, and x is a variable.

Proof Sketch. Define the symbols in the language of Predicate logic that we consider.

Choose α to be a concrete Predicate formula. Choose t to be a concrete Predicate term.

Define an interpretation and an environment (I, E).

Show that $(\exists x\ \alpha)^{(I,E)} = \mathtt{T}$.

Show that $\alpha[t/x]^{(I,E)} = \mathtt{F}$. □

Solution: We need to provide an interpretation I and an environment E such that $(\exists x\ \alpha)^{(I,E)} = \mathtt{T}$ and $\alpha[t/x]^{(I,E)} = \mathtt{F}$.

Consider the language of predicate logic where $P^{(1)}$ is a unary predicate and x and y are variables.

Let α be $P(x)$ and let t be y. Let the interpretation I be defined below.

- $dom(I) = \{1, 2\}$
- $P^I = \{1\}$

Let the environment E be defined by $E(x) = 1$ and $E(y) = 2$.

First, we show that $(\exists x\ \alpha)^{(I,E)} = \mathtt{T}$.
By the definition of P^I, we have that

$$\alpha^{(I,E[x\mapsto 1])} = \mathtt{T}$$

because $E[x \mapsto 1](x) = 1 \in P^I$.
By the satisfaction rule for $\exists$, we have that

$$(\exists x\ \alpha)^{(I,E)} = \mathtt{T}.$$

Second, we show that $\alpha[t/x]^{(I,E)} = \mathtt{F}$.
By Lemma 2, we have that

$$\alpha[t/x]^{(I,E)} = \alpha^{(I,E[x\mapsto t^{(I,E)}])}.$$

We need to prove that

$$\alpha^{(I,E[x\mapsto t^{(I,E)}])} = \mathtt{F}.$$

By Lemma 1, $t^{(I,E)}$ is a domain element. By the definitions of t, I, and E, we have that

$$t^{(I,E)} = y^{(I,E)} = E(y) = 2.$$

By the definition of P^I, we have that

$$\alpha^{(I, E[x \mapsto t^{(I,E)}])} = \alpha^{(I, E[x \mapsto 2])} = \mathtt{F}$$

because $E[x \mapsto 2](x) = 2 \notin P^I$.
Therefore, we have that $\alpha^{(I, E[x \mapsto t^{(I,E)}])} = \mathtt{F}$.
The entailment does not hold and the inference rule is not sound.

2.5.2 Additional Exercises

Exercise 123. *Prove that the following inference rule is sound.*

$$\frac{(\forall x(\alpha \to \beta)) \quad \alpha[t/x]}{\beta[t/x]}\ \forall e1 \tag{140}$$

where α and β are predicate formulas, t is a predicate term, and x is a variable.

Exercise 124. *Prove that the following inference rule is sound.*

$$\frac{(\forall x(\alpha \to \beta)) \quad (\neg\beta[t/x])}{(\neg\alpha[t/x])}\ \forall e2 \tag{141}$$

where α and β are predicate formulas, t is a predicate term, and x is a variable.

Exercise 125. *Prove that the following inference rule is NOT sound.*

$$\frac{(\forall x(\alpha \to \beta)) \quad \beta[t/x]}{\alpha[t/x]}\ \forall e3 \tag{142}$$

where α and β are predicate formulas, t is a predicate term, and x is a variable.

Exercise 126. *Prove that the following inference rule is NOT sound.*

$$\frac{(\forall x(\alpha \to \beta)) \quad (\neg\alpha[t/x])}{(\neg\beta[t/x])}\ \forall e4 \tag{143}$$

where α and β are predicate formulas, t is a predicate term, and x is a variable.

2.5.3 Proofs using the soundness and completeness theorems

Exercise 127. *Let Σ be a set of Predicate formulas and let α be a Predicate formula. If $\Sigma \cup \{(\neg\alpha)\}$ is unsatisfiable, then $\Sigma \vdash \alpha$.*

Proof Sketch. Assume that $\Sigma \cup \{(\neg\alpha)\}$ is unsatisfiable. This means that, for any interpretation and environment (I, E), at least one formula in $\Sigma \cup \{(\neg\alpha)\}$ is false.

Prove that $\Sigma \vDash \alpha$. Consider an interpretation and environment (I, E). Assume that every formula in Σ is true under (I, E). Prove that α is true under (I, E).

We have $\Sigma \vdash \alpha$ by the completeness of Natural Deduction. □

Solution:

Remark 19. *What does it mean for a set of formula Σ to be unsatisfiable?*

- *It means that "for every (I, E), at least one formula in Σ is false."*

  Example 1: The set $\{P(x), (\neg P(x))\}$ is unsatisfiable. Under any (I, E), if $P(x)$ is true, then $(\neg P(x))$ must be false. If $(\neg P(x))$ is true, then $P(x)$ must be false.

- *It DOES NOT mean that " for every (I, E), at least one formula in Σ is a contradiction."*

  Example 2: The set $\{(P(x) \wedge (\neg P(x)))\}$ is unsatisfiable. Under any (I, E), $(P(x) \wedge (\neg P(x)))$ is always false. Note that this is only one type of unsatisfiable set.

- *It DOES NOT mean that "for one pair (I, E), at least one formula in Σ is false."*

When proving the entailment $\Sigma \vDash \alpha$, why do we only consider the cases when every formula in Σ is true?

By the definition of entailment, we only need to verify that α is true in the case when every formula in Σ is true under an (I, E). Thus, we do not need to consider the case when a formula in Σ is false under an (I, E).

Solution Text: Assume that $\Sigma \cup \{(\neg\alpha)\}$ is unsatisfiable. This means that, for any interpretation and environment (I, E), at least one formula in $\Sigma \cup \{(\neg\alpha)\}$ is false.

We need to prove that $\Sigma \vDash \alpha$. Consider an interpretation and environment (I, E). Assume that every formula in Σ is true under (I, E). We need to prove that α is true under (I, E).

Under the (I, E) we are considering, every formula in Σ is true and at least one formula in $\Sigma \cup \{(\neg\alpha)\}$ is false. Therefore, it must be that $(\neg\alpha)^{(I,E)} = \mathtt{F}$. By the definition of $\neg$, $\alpha^{(I,E)} = \mathtt{T}$. Therefore, the entailment $\Sigma \vDash \alpha$ holds.

$\Sigma \vdash \alpha$ holds by the completeness of Natural Deduction.

Exercise 128. *Let Σ be a set of Predicate formulas and let α be a Predicate formula. If $\Sigma \vdash \alpha$, then $\Sigma \cup \{(\neg\alpha)\}$ is unsatisfiable.*

Exercise 129. *Show that there is no natural deduction proof for $\{(\exists x\ P(x))\} \vdash P(t)$, where P is a unary predicate, t is a term and x is a variable.*

3 Program Verification

3.1 Partial and Total Correctness

Exercise 130. *Consider the Hoare triple* $(\!|\, (x > 0) \,|\!)\ C_1\ (\!|\, ((y * y) < x) \,|\!)$.

If we run C_1 *starting with the state* $(x = 5), (y = 5)$, C_1 *terminates in the state* $(x = 5), (y = 0)$.

Is the Hoare triple satisfied under partial correctness?

Solution: The answer is ‘not enough information to tell.”

The definition of partial correctness has an implication in it: If a starting state satisfies the precondition and the program terminates when run with this starting state, then the terminating state satisfies the postcondition.

For the given example, the starting state $(x = 5)$ satisfies the precondition $x > 0$, the program terminates, and the terminating state $((x = 5), (y = 0))$ satisfies the postcondition since $y * y = 0 < 5 = 5$. Therefore, this example satisfies the implication in the definition of partial correctness.

However, to verify partial correctness, we need to consider all possible starting states which satisfy the precondition, e.g. $(x = 1)$, $(x = 2)$, etc. We do not know the terminating states for the other possible starting states. Therefore, we do not have enough information to determine the answer.

Exercise 131. *Consider the Hoare triple* $(\!|\, (x > 0) \,|\!)\ C_2\ (\!|\, ((y * y) < x) \,|\!)$.

If we run C_2 *starting with the state* $(x = 5), (y = 5)$, C_2 *terminates in the state* $(x = 5), (y = 3)$.

Is the Hoare triple satisfied under partial correctness?

Solution: The answer is NO.

The definition of partial correctness has an implication in it: If a starting state satisfies the precondition and the program terminates when run with this starting state, then the terminating state satisfies the postcondition.

For the given example, the starting state $x = 5$ satisfies the precondition $x > 0$, and the terminating state $(x = 5, y = 3)$ does NOT satisfy the postcondition $y * y = 0 < 5 = 5$. Therefore, this example does not satisfy this implication in the definition of partial correctness.

To verify that partial correctness is not satisfied, it is sufficient to find one counterexample as shown above. Therefore, the triple is NOT satisfied under partial correctness.

Exercise 132. *Consider the Hoare triple* $\llparenthesis\, (x > 0) \,\rrparenthesis \;\; C_3 \;\; \llparenthesis\, ((y * y) < x) \,\rrparenthesis$.

If we run C_3 *starting with the state* $(x = -3), (y = 5)$, C_3 *terminates in the state* $(x = -3), (y = 0)$.

Is the Hoare triple satisfied under partial correctness?

Solution: The answer is “not enough information to tell”.

For the given example, the starting state $x = -3$ does not satisfy the precondition.

To verify partial correctness, we only need to consider starting states that satisfy the precondition. Therefore, the example is irrelevant for us.

Since we do not know how the program behaves for starting states that satisfy the precondition, we do not have enough information to determine the answer.

Exercise 133. *Consider the Hoare triple* $\llparenthesis\, (x > 0) \,\rrparenthesis \;\; C_4 \;\; \llparenthesis\, ((y * y) < x) \,\rrparenthesis$.

If we run C_4 *starting with the state* $(x = 2), (y = 5)$, C_4 *does not terminate.*

Is the Hoare triple satisfied under partial correctness?

Solution: The answer is “not enough information to tell”.

The definition of partial correctness has an implication in it: If a starting state satisfies the precondition and the program terminates when run with this starting state, then the terminating state satisfies the postcondition.

For the given example, the starting state $x = 2$ satisfies the precondition but the program does not terminate. Note that, program termination is a premise of the implication in the definition of partial correctness. Therefore, this example satisfies the implication in the definition of partial correctness.

However, to verify partial correctness, we need to consider all possible starting states that satisfy the precondition. Since we do not know how the program behaves for other possible starting states, we do not have enough information to determine the answer.

Exercise 134. *Is the following Hoare triple satisfied under partial and/or total correctness?*

```
⦇ (x = 1) ⦈
while (1) {
   x = 0
};
⦇ (y = 1) ⦈
```

Solution: The triple is satisfied under partial correctness, and it is not satisfied under total correctness.

The program does not terminate for any starting state. Therefore, partial correctness is automatically satisfied. (If the program does not terminate for a starting state, then the premise of the implication is false and the implication is vacuously true.)

The program does not terminate for any starting state. Therefore, total correctness is NOT satisfied.

The key difference between partial and total correctness is that partial correctness does not require program termination, whereas total correctness does.

Exercise 135. *Is the following Hoare triple satisfied under partial and/or total correctness?*

```
⦇ true ⦈
y = 1;
z = 0;
while (z != x) {
   z = z + 1;
   y = y * z;
}
⦇ (y = x!) ⦈
```

Solution: The triple is satisfied under partial correctness, and it is NOT satisfied under total correctness.

The precondition is true. This means that there is no required precondition. In other words, any starting state satisfies the precondition.

If the starting state has $x \geq 0$, we can verify that the program terminates and computes $y = x!$ correctly. (We are not able to prove this yet because we haven't learned the techniques to construct the proof. However, we could verify this on a case-by-case basis.) Therefore, for this case, partial and total correctness are both satisfied.

If the starting state has $x < 0$, the while loop runs forever and does not terminate. Therefore, for this case, only partial correctness is satisfied. Total correctness is not satisfied because the program does not terminate.

In summary, partial correctness is satisfied because it is satisfied in both cases. Total correctness is NOT satisfied because the program does not terminate for some starting states where $x < 0$.

3.2 Assignment Statements

Complete the following annotations.

$\llparenthesis \qquad\qquad \rrparenthesis$
`x = 2;`
$\llparenthesis\, (x = 2)\, \rrparenthesis$

Solution:

$\llparenthesis\, (2 = 2)\, \rrparenthesis$
`x = 2;`
$\llparenthesis\, (x = 2)\, \rrparenthesis$ assignment

$\llparenthesis \qquad\qquad \rrparenthesis$
`x = 2;`
$\llparenthesis\, (x = y)\, \rrparenthesis$

Solution:

$\llparenthesis\, (2 = y)\, \rrparenthesis$
`x = 2;`
$\llparenthesis\, (x = y)\, \rrparenthesis$ assignment

$\llparenthesis \qquad\qquad \rrparenthesis$
`x = 2;`
$\llparenthesis\, (x = 0)\, \rrparenthesis$

Solution:

$\llparenthesis\, (2 = 0)\, \rrparenthesis$
`x = 2;`
$\llparenthesis\, (x = 0)\, \rrparenthesis$ assignment

$\llparenthesis \qquad\qquad\qquad\qquad \rrparenthesis$

x = x + 1;

$\llparenthesis\, (x = (n + 1))\, \rrparenthesis$

Solution:

$\llparenthesis\, ((x + 1) = (n + 1))\, \rrparenthesis$

x = x + 1;

$\llparenthesis\, (x = (n + 1))\, \rrparenthesis$ assignment

$\llparenthesis \qquad\qquad\qquad\qquad \rrparenthesis$

x = y;

$\llparenthesis\, ((2 * x) = (x + y))\, \rrparenthesis$

Solution:

$\llparenthesis\, ((2 * y) = (y + y))\, \rrparenthesis$

x = y;

$\llparenthesis\, ((2 * x) = (x + y))\, \rrparenthesis$ assignment

Exercise 136. *Show that the following triple is satisfied under partial correctness.*

```
$\llparenthesis\ (y = 6)\ \rrparenthesis$
x = y + 1;
$\llparenthesis\ (x = 7)\ \rrparenthesis$
```

Solution:

```
$\llparenthesis\ (y = 6)\ \rrparenthesis$
$\llparenthesis\ ((y + 1) = 7)\ \rrparenthesis$        implied (A)
x = y + 1;
$\llparenthesis\ (x = 7)\ \rrparenthesis$              assignment
```

Proof of implied (A):
Assume that $y = 6$. Adding 1 to both sides, we get $y + 1 = 6 + 1 = 7$.

Exercise 137. *Show that the following triple is satisfied under partial correctness.*

```
$\llparenthesis\ ((x = x_0) \wedge (y = y_0))\ \rrparenthesis$
t = x;
x = y;
y = t;
$\llparenthesis\ ((x = y_0) \wedge (y = x_0))\ \rrparenthesis$
```

Solution:

```
$\llparenthesis\ ((x = x_0) \wedge (y = y_0))\ \rrparenthesis$
$\llparenthesis\ ((y = y_0) \wedge (x = x_0))\ \rrparenthesis$      implied (A)
t = x;
$\llparenthesis\ ((y = y_0) \wedge (t = x_0))\ \rrparenthesis$       assignment
x = y;
$\llparenthesis\ ((x = y_0) \wedge (t = x_0))\ \rrparenthesis$       assignment
y = t;
$\llparenthesis\ ((x = y_0) \wedge (y = x_0))\ \rrparenthesis$       assignment
```

Proof of implied (A):
Assume that $((x = x_0) \wedge (y = y_0))$ is true. By the definition of $\wedge$, $x = x_0$ and $y = y_0$ are both true. By the definition of $\wedge$, $((y = y_0) \wedge (x = x_0))$ is true.

3.3 Conditional Statements

Exercise 138. *Show that the following triple is satisfied under partial correctness.*

```
$⦇ true ⦈$
if (x > y) {
  max = x;
} else {
  max = y;
}
$⦇ (((x > y) \wedge (max = x)) \vee ((x \leq y) \wedge (max = y))) ⦈$
```

Solution:

```
$⦇ true ⦈$
if (x > y) {
  $⦇ (x > y) ⦈$                                                    if-then-else
  $⦇ (((x > y) \wedge (x = x)) \vee ((x \leq y) \wedge (x = y))) ⦈$           implied (A)
  max = x;
  $⦇ (((x > y) \wedge (max = x)) \vee ((x \leq y) \wedge (max = y))) ⦈$       assignment
} else {
  $⦇ (\neg(x > y)) ⦈$                                               if-then-else
  $⦇ (((x > y) \wedge (y = x)) \vee ((x \leq y) \wedge (y = y))) ⦈$           implied (B)
  max = y;
  $⦇ (((x > y) \wedge (max = x)) \vee ((x \leq y) \wedge (max = y))) ⦈$       assignment
}
$⦇ (((x > y) \wedge (max = x)) \vee ((x \leq y) \wedge (max = y))) ⦈$         if-then-else
```

Proof of implied (A)
Assume that $(x > y)$. By the definition of $=$, $x = x$ is true. By the definition of $\wedge$, $((x > y) \wedge (x = x))$ is true. By the definition of $\vee$, $(((x > y) \wedge (x = x)) \vee ((x \leq y) \wedge (x = y)))$ is true.

Proof of implied (B)
Assume that $(\neg(x > y))$ is true. By the definition of $>$, $(x \leq y)$ is true. By the definition of $=$, $y = y$ is true. By the definition of $\wedge$, $((x \leq y) \wedge (y = y))$ is true. By the definition of $\vee$, $(((x > y) \wedge (y = x)) \vee ((x \leq y) \wedge (y = y)))$ is true.

Exercise 139. *Show that the following triple is satisfied under partial correctness.*

```
$\llparenthesis\, (x = 3) \,\rrparenthesis$
if (x > 0) {
  x = 1;
} else {
  x = 0;
}
$\llparenthesis\, (x \geq 0) \,\rrparenthesis$
```

Solution:

```
$\llparenthesis\, (x = 3) \,\rrparenthesis$
if (x > 0) {
  $\llparenthesis\, ((x = 3) \wedge (x > 0)) \,\rrparenthesis$          if-then-else
  $\llparenthesis\, (1 \geq 0) \,\rrparenthesis$                        implied (A)
  x = 1;
  $\llparenthesis\, (x \geq 0) \,\rrparenthesis$                        assignment
} else {
  $\llparenthesis\, ((x = 3) \wedge (\neg(x > 0))) \,\rrparenthesis$     if-then-else
  $\llparenthesis\, (0 \geq 0) \,\rrparenthesis$                        implied (B)
  x = 0;
  $\llparenthesis\, (x \geq 0) \,\rrparenthesis$                        assignment
}
$\llparenthesis\, (x \geq 0) \,\rrparenthesis$                          if-then-else
```

Proof of implied (A) $(((x = 3) \wedge (x > 0)) \to (1 \geq 0))$
Assume that $((x = 3) \wedge (x > 0))$ is true. $1 \geq 0$ is true by the definition of $\geq$.

Proof of implied (B) $(((x = 3) \wedge (\neg(x > 0))) \to (0 \geq 0))$
The premise $((x = 3) \wedge (\neg(x > 0)))$ is false. $(x = 3)$ means that x is positive. $(\neg(x > 0))$ means that x is not positive. These two formulas contradict each other and cannot be true at the same time. Therefore, $((x = 3) \wedge (\neg(x > 0)))$ is false by the property of $\wedge$. The implication is vacuously true.

Exercise 140. *Show that the following triple is satisfied under partial correctness.*

```
$\llparenthesis\ true\ \rrparenthesis$
if (max < x) {
  max = x;
}
$\llparenthesis\ (max \geq x)\ \rrparenthesis$
```

Solution:

```
$\llparenthesis\ true\ \rrparenthesis$
if (max < x) {
  $\llparenthesis\ (max < x)\ \rrparenthesis$      if-then
  $\llparenthesis\ (x \geq x)\ \rrparenthesis$        implied (A)
  max = x;
  $\llparenthesis\ (max \geq x)\ \rrparenthesis$      assignment
}
$\llparenthesis\ (max \geq x)\ \rrparenthesis$        if-then
                        implied (B)  $((\neg(max < x)) \rightarrow (max \geq x))$
```

Proof of implied (A)
Assume that $(max < x)$ is true. $(x \geq x)$ is true by the definition of $\geq$.

Proof of implied (B)
Assume that $(\neg(max < x))$ is true. By the definition of $\neg$ and $<$, $(max \geq x)$ is true.

Exercise 141. *Show that the following triple is satisfied under partial correctness.*

```
⦇ $true$ ⦈
if (x % 2 == 1) {
  x = x + 1;
}
⦇ $(\exists u\,(x = (2 * u)))$ ⦈
```

Solution:

```
⦇ $true$ ⦈
if (x % 2 == 1) {
  ⦇ $((x\%2) = 1)$ ⦈                if-then
  ⦇ $(\exists u\ (x + 1 = (2 * u)))$ ⦈    implied (A)
  x = x + 1;
  ⦇ $(\exists u\ (x = (2 * u)))$ ⦈        assignment
}
⦇ $(\exists u\ (x = (2 * u)))$ ⦈          if-then
                                  implied (B) $((\neg(x\%2 = 1)) \rightarrow ((\exists u\ (x = (2 * u)))))$
```

Proof of implied (A)
Assume that $((x\%2) = 1)$ is true. This means that x is odd. By the definition of an odd integer, there exists an integer u such that $x = 2 * u - 1$ or $x + 1 = 2 * u$. Therefore, $(\exists u\ (x + 1 = (2 * u)))$ is true.

Proof of implied (B)
Assume that $((\neg(x\%2 = 1))$ is true, which meant that x is even. By the definition of an even integer, there exists an integer u such that $x = 2 * u$. Therefore, $((\exists u\ (x = (2 * u))))$ is true.

Exercise 142. *Show that the following triple is satisfied under partial correctness.*

```
$\llparenthesis\, \mathit{true} \,\rrparenthesis$
if (x < 5) {
  r = 0;
} else {
  if (x > 10) {
    r = 0;
  } else {
    r = 1;
  }
}
$\llparenthesis\, ((((x < 5) \vee (x > 10)) \wedge (r = 0)) \vee (((5 \leq x) \wedge (x \leq 10)) \wedge (r = 1))) \,\rrparenthesis$
```

Solution:

```
$\llparenthesis\, \mathit{true} \,\rrparenthesis$
if (x < 5) {
  $\llparenthesis\, \mathit{true} \wedge (x < 5) \,\rrparenthesis$                                                         if−then−else
  $\llparenthesis\, ((((x < 5) \vee (x > 10)) \wedge (0 = 0)) \vee (((5 \leq x) \wedge (x \leq 10)) \wedge (0 = 1))) \,\rrparenthesis$       implied (A)
  r = 0;
  $\llparenthesis\, ((((x < 5) \vee (x > 10)) \wedge (r = 0)) \vee (((5 \leq x) \wedge (x \leq 10)) \wedge (r = 1))) \,\rrparenthesis$       assignment
} else {
  $\llparenthesis\, \mathit{true} \wedge (x \geq 5) \,\rrparenthesis$                                                         if−then−else
  if (x > 10) {
    $\llparenthesis\, \mathit{true} \wedge (x \geq 5) \wedge (x > 10) \,\rrparenthesis$                                       if−then−else
    $\llparenthesis\, ((((x < 5) \vee (x > 10)) \wedge (0 = 0)) \vee (((5 \leq x) \wedge (x \leq 10)) \wedge (0 = 1))) \,\rrparenthesis$     implied (B)
    r = 0;
    $\llparenthesis\, ((((x < 5) \vee (x > 10)) \wedge (r = 0)) \vee (((5 \leq x) \wedge (x \leq 10)) \wedge (r = 1))) \,\rrparenthesis$     assignment
  } else {
    $\llparenthesis\, \mathit{true} \wedge (x \geq 5) \wedge (x \leq 10) \,\rrparenthesis$   if−then−else
    $\llparenthesis\, ((((x < 5) \vee (x > 10)) \wedge (1 = 0)) \vee (((5 \leq x) \wedge (x \leq 10)) \wedge (1 = 1))) \,\rrparenthesis$     implied (C)
    r = 1;
    $\llparenthesis\, ((((x < 5) \vee (x > 10)) \wedge (r = 0)) \vee (((5 \leq x) \wedge (x \leq 10)) \wedge (r = 1))) \,\rrparenthesis$     assignment
  }
  $\llparenthesis\, ((((x < 5) \vee (x > 10)) \wedge (r = 0)) \vee (((5 \leq x) \wedge (x \leq 10)) \wedge (r = 1))) \,\rrparenthesis$         if−then−else
}
$\llparenthesis\, ((((x < 5) \vee (x > 10)) \wedge (r = 0)) \vee (((5 \leq x) \wedge (x \leq 10)) \wedge (r = 1))) \,\rrparenthesis$           if−then−else
```

3.4 Conditional Statements: Additional Exercises

Exercise 143. *Show that the following triple is satisfied under partial correctness.*

```
⦇ true ⦈
x = a * a;
y = b * b;
z = x + y;
if (b > a) {
   z = z + 2 * a * b;
} else {
   z = z − 2 * a * b;
}
⦇ ((∃u (u ∗ u = z)) ⦈
```

3.5 While Loops

Exercise 144. *Show that the following triple is satisfied under partial correctness.*

```
⦇ (x ≥ 0) ⦈
y = 1;
z = 0;
while (z != x) {
   z = z + 1;
   y = y * z;
}
⦇ (y = x!) ⦈
```

Remark 20. *There is a while loop in the program. To complete the proof, we need to come up with an invariant for the while loop. We produce the following table, which contains the values of all the variables in the program whenever the execution reaches the while test* $z! = x$.

Note: We can choose any non-negative value for x*. For the following table, we chose* $x = 5$.

Note: In the table, I wrote y as a factorial. Doing this is helpful for seeing a relationship between the variables (With this, it is easy to see that $y = z!$ *in every row of the table). Also, the post-condition says that y should be a factorial. If we want to make progress towards that post-condition, then it makes sense that y is equal to some factorial at every iteration of the loop.*

x	z	y
5	0	1 = 0!
5	1	1 = 1!
5	2	2 = 2!
5	3	6 = 3!
5	4	24 = 4!
5	5	120 = 5!

Given the table, we can try to come up with relationship between the variables. For the relationship to be an invariant, it has to be true in every row of the truth table.
For example,

- $(\neg(z = x))$ *is NOT an invariant. It is NOT true in the last row of the table.*
- $(z \leq x)$ *IS an invariant. It is true in every row of the table.*
- $(y = z!)$ *IS an invariant. It is true in every row of the table.*
- $(y = x!)$ *is NOT an invariant. It is only true in the last row of the table and not true in any other row.*

- $((z \leq x) \land (y = z!))$ *IS an invariant.*

Note: We can combine one or more invariants with an $\land$ *to produce new invariants. If* A *and* B *are invariants, then* $(A \land B)$ *is an invariant as well.*

So far, we have found three invariants: $(z \leq x)$, $(y = z!)$, *and* $((z \leq x) \land (y = z!))$. *Which of these invariants will lead to valid proofs? It turns out that both the second and third invariants will both lead to valid proofs.*

How do I choose an invariant to complete my proof? The only sure way of answering this question is to try completing the proof with the invariant. The proof is valid if and only if we can prove all of the implied conditions using the invariant.

However, there are two strategies to speed up this process of selecting ani nvariant that works.

- **A useful invariant is often similar to the post-condition.** *In our example, both invariants that work (*$(y = z!)$ *and* $((z \leq x) \land (y = z!))$*) have the component* $(y = z!)$, *which is similar to the post-condition* $(y = x!)$.

  This makes intuitive sense. An invariant describes the progress we are making towards the post-condition at every iteration of the loop. Therefore, it is only natural that the invariant looks similar to the post-condition.

- **The last implied condition (implied C) is often the most difficult to satisfy.** *Thus, to test whether an invariant works, it may be more efficient to try proving implied (C) first.*

See the completed solution below with the invariant $(y = z!)$.

Solution:

```
$\llparenthesis (x \geq 0) \rrparenthesis$
$\llparenthesis (1 = 0!) \rrparenthesis$                       implied (A)
y = 1;
$\llparenthesis (y = 0!) \rrparenthesis$                       assignment
z = 0;
$\llparenthesis (y = z!) \rrparenthesis$                       assignment
while (z != x) {
  $\llparenthesis ((y = z!) \land (\neg(z = x))) \rrparenthesis$     partial-while
  $\llparenthesis ((y * (z + 1)) = (z + 1)!) \rrparenthesis$        implied (B)
  z = z + 1;
  $\llparenthesis ((y * z) = z!) \rrparenthesis$                     assignment
  y = y * z;
  $\llparenthesis (y = z!) \rrparenthesis$                           assignment
}
$\llparenthesis ((y = z!) \land (\neg(\neg(z = x)))) \rrparenthesis$  partial-while
$\llparenthesis (y = x!) \rrparenthesis$                              implied (C)
```

Proof of implied (A):
Assume that $(x \geq 0)$ is true. $(1 = 0!)$ is true by the definition of factorial.

Proof of implied (B):
Assume that $(y = z!)$ and $(\neg(z = x))$ are true.
Multiplying $(z + 1)$ on both sides of $(y = z!)$, we get that $y * (z + 1) = (z + 1)!$ is true.

Proof of implied (C):
Assume that $(y = z!)$ and $(\neg(\neg(z = x)))$ are true. By the definition of $\neg$, $(\neg(\neg(z = x)))$ is equivalent to $(z = x)$. Since $(y = z!)$ and $(z = x)$ are both true, we know that $(y = x!)$ must be true.

Exercise 145. *Show that the following triple is satisfied under partial correctness.*

```
$\llparenthesis\, (x \geq 0) \,\rrparenthesis$
y = 1;
z = 0;
while (z < x) {
  z = z + 1;
  y = y * z;
}
$\llparenthesis\, (y = x!) \,\rrparenthesis$
```

Solution:

```
$\llparenthesis\, (x \geq 0) \,\rrparenthesis$
$\llparenthesis\, ((1 = 0!) \wedge (0 \leq x)) \,\rrparenthesis$            implied (A)
y = 1;
$\llparenthesis\, ((y = 0!) \wedge (0 \leq x)) \,\rrparenthesis$            assignment
z = 0;
$\llparenthesis\, ((y = z!) \wedge (z \leq x)) \,\rrparenthesis$            assignment
while (z < x) {
  $\llparenthesis\, (((y = z!) \wedge (z \leq x)) \wedge (z < x)) \,\rrparenthesis$  partial-while
  $\llparenthesis\, (y * (z + 1) = (z + 1)!) \,\rrparenthesis$               implied (B)
  z = z + 1;
  $\llparenthesis\, (y * z = z!) \,\rrparenthesis$                           assignment
  y = y * z;
  $\llparenthesis\, (y = z!) \,\rrparenthesis$                               assignment
}
$\llparenthesis\, (((y = z!) \wedge (z \leq x)) \wedge (\neg(z < x))) \,\rrparenthesis$    partial-while
$\llparenthesis\, (y = x!) \,\rrparenthesis$                                   implied (C)
```

Proof of implied (A):
Assume that $x \geq 0$ is true. Then $(0 \leq x)$ is true by the definitions of $\leq$ and $\geq$. $(1 = 0!)$ is true by the definition of factorial.

Proof of implied (B):
Assume that $(y = z!)$, $(z \leq x)$ and $(z < x)$ are true.
Multiplying $(z + 1)$ on both sides of $(y = z!)$, we get that $y * (z + 1) = (z + 1)!$ is true.

Proof of implied (C):
Assume that $(y = z!)$, $(z \leq x)$ and $(\neg(z < x))$ are true. By the definition of $\neg$, $(\neg(z < x))$ is equivalent to $(z \geq x)$. Since $(z \leq x)$ and $(z \geq x)$, it must be that $(z = x)$. Since $(y = z!)$ and $(z = x)$ are both true, we know that $(y = x!)$ must be true.

3.6 While Loops: Additional Exercises

Exercise 146. *Show that the following triple is satisfied under partial correctness.*

```
$⦇ ((n \geq 0) \wedge (a \geq 0)) ⦈$
s = 1;
i = 0;
while ( i != n ) {
   s = s * a;
   i = i + 1;
}
$⦇ (s = a^n) ⦈$
```

Exercise 147. *Show that the following triple is satisfied under partial correctness.*

```
$⦇ ((n \geq 0) \wedge (a \geq 0)) ⦈$
s = 1;
i = 0;
while ( i < n ) {
   s = s * a;
   i = i + 1;
}
$⦇ (s = a^n) ⦈$
```

3.7 Array Assignments

Exercise 148. *Show that the following triple is satisfied under partial correctness.*

$\llparenthesis\, ((A[x] = x0) \wedge (A[y] = y0))\, \rrparenthesis$

t = A[x];

A[x] = A[y];

A[y] = t;

$\llparenthesis\, ((A[x] = y0) \wedge (A[y] = x0))\, \rrparenthesis$ *array assignment*

Solution:

$\llparenthesis\, ((A[x] = x0) \wedge (A[y] = y0))\, \rrparenthesis$

$\llparenthesis\, ((A\{x \leftarrow A[y]\}\{y \leftarrow A[x]\}[x] = y0) \wedge (A\{x \leftarrow A[y]\}\{y \leftarrow A[x]\}[y] = x0))\, \rrparenthesis$ implied

t = A[x];

$\llparenthesis\, ((A\{x \leftarrow A[y]\}\{y \leftarrow t\}[x] = y0) \wedge (A\{x \leftarrow A[y]\}\{y \leftarrow t\}[y] = x0))\, \rrparenthesis$ assignment

A[x] = A[y];

$\llparenthesis\, ((A\{y \leftarrow t\}[x] = y0) \wedge (A\{y \leftarrow t\}[y] = x0))\, \rrparenthesis$ array assignment

A[y] = t;

$\llparenthesis\, ((A[x] = y0) \wedge (A[y] = x0))\, \rrparenthesis$ array assignment

Proof of implied:
We will prove that $((A[x] = x0) \wedge (A[y] = y0)) \to (((A\{x \leftarrow A[y]\}\{y \leftarrow A[x]\}[x] = y0) \wedge (A\{x \leftarrow A[y]\}\{y \leftarrow A[x]\}[y] = x0))$.

Proof. **Case 1: Assume that $x = y$.** Assume that $A[x] = x0$ and $A[y] = y0$ are true.

Since $x = y$, we can re-write the conclusion as follows.

$$A\{x \leftarrow A[y]\}\{y \leftarrow A[x]\}[x] = A\{y \leftarrow A[y]\}\{y \leftarrow A[y]\}[y] = A[y] = y0, \text{ and}$$
$$A\{x \leftarrow A[y]\}\{y \leftarrow A[x]\}[y] = A\{x \leftarrow A[x]\}\{x \leftarrow A[x]\}[x] = A[x] = x0.$$

Case 2: Assume that $x \neq y$. Assume that $A[x] = x0$ and $A[y] = y0$ are true.

Consider the array $A\{x \leftarrow A[y]\}\{y \leftarrow A[x]\}$. The first assignment $\{x \leftarrow A[y]\}$ changes the xth element of the array to $A[y]$. The second assignment $\{y \leftarrow A[x]\}$ changes the yth element of the array to $A[x]$. Since $x \neq y$, the two assignments are modifying two different elements in the array and do not affect each other.

Therefore, we have that

$$A\{x \leftarrow A[y]\}\{y \leftarrow A[x]\}[y] = A[x] = x0.$$

because the second assignment changes the yth element to $A[x]$.

$$A\{x \leftarrow A[y]\}\{y \leftarrow A[x]\}[x] = A[y] = y0.$$

because the first assignment changes the xth element to $A[y]$.

□

3.8 Putting them together

Exercise 149. *(Reversing an array)*
Consider an array R of n integers, $R[1], R[2], ..., R[n]$.
Consider the following program which reverses the elements inside the array R.
Let r_x denote the element at index x in the array R before the program execution.
Prove that the following triple is satisfied under total correctness.

```
$\llparenthesis\, ((\forall x\,(1 \leq x \leq n \rightarrow R[x] = r_x)))\,\rrparenthesis$
j = 1;
while (2*j <= n) {
   t = R[j];
   R[j] = R[n+1-j];
   R[n+1-j] = t;
   j = j + 1;
}
$\llparenthesis\, ((\forall x\,(1 \leq x \leq n \rightarrow R[x] = r_{n+1-x})))\,\rrparenthesis$
```

Solution: Since there is a while loop in the program, we need to come up with an invariant for the while loop.
Consider the following invariant.

$$\begin{aligned} Inv(j) \equiv & (((\forall x\,(1 \leq x \leq j-1 \rightarrow (R[x] = r_{n+1-x} \wedge R[n+1-x] = r_x))) \\ & \wedge (\forall x\,(j \leq x \leq (n+1)/2 \rightarrow (R[x] = r_x \wedge R[n+1-x] = r_{n+1-x})))) \\ & \wedge (j \leq n/2+1)) \end{aligned}$$

Using the above invariant, we complete the annotations for the program as shown below.

```
$\llparenthesis\, (\forall x\,(1 \leq x \leq n \rightarrow R[x] = r_x))\,\rrparenthesis$
$\llparenthesis\, Inv(1)\,\rrparenthesis$                  implied (A)
j = 1;
$\llparenthesis\, Inv(j)\,\rrparenthesis$                  assignment
while (2*j <= n) {
   $\llparenthesis\, (Inv(j) \wedge (2*j \leq n))\,\rrparenthesis$  partial-while
   $\llparenthesis\, Inv(j+1)[R\{j \leftarrow R[n+1-j]\}\{n+1-j \leftarrow R[j]\}/R]\,\rrparenthesis$  implied (B)
   t = R[j];
   $\llparenthesis\, Inv(j+1)[R\{j \leftarrow R[n+1-j]\}\{n+1-j \leftarrow t\}/R]\,\rrparenthesis$  assignment
   R[j] = R[n+1-j];
   $\llparenthesis\, Inv(j+1)[R\{n+1-j \leftarrow t\}/R]\,\rrparenthesis$  array assignment
   R[n+1-j] = t;
   $\llparenthesis\, Inv(j+1)\,\rrparenthesis$                       array assignment
   j = j + 1;
   $\llparenthesis\, Inv(j)\,\rrparenthesis$  assignment
}
```

$\llparenthesis\, (Inv(j) \land (\neg(2 * j \leq n)))\, \rrparenthesis$ partial−**while**
$\llparenthesis\, ((\forall x\, (1 \leq x \leq n \to R[x] = r_{n+1-x})))\, \rrparenthesis$ implied (C)

It remains to prove the implied (A), (B), and (C).

To prove implied (C), we first prove Lemma 3 below.

Lemma 3. *The two formulas below are logically equivalent.*

$$\begin{aligned}&(\forall x\, ((1 \leq x \leq (n+1)/2) \to (R[x] = r_x \land R[n+1-x] = r_{n+1-x})))) \\ &\equiv (\forall x\, ((1 \leq x \leq n) \to (R[x] = r_x)))\end{aligned}$$

Proof. Starting with the top formula is logically equivalent to the following formula

$$\begin{aligned}&(\forall x\, (1 \leq x \leq (n+1)/2 \to R[x] = r_x))) \\ &\land (\forall x\, (1 \leq x \leq (n+1)/2 \to R[n+1-x] = r_{n+1-x})))\end{aligned} \tag{144}$$

We will transform formula 144 as follows. Let $y = n+1-x$. Then $x = n+1-y$. Plugging $x = n+1-y$ into formula 144, we have

$$\big(\forall x\, \big(1 \leq n+1-y \leq (n+1)/2 \to R[y] = r_y\big)\big)\big)$$

Let's simplify the inequality $1 \leq n+1-y \leq (n+1)/2$.

$$1 \leq n+1-y \to y \leq n$$

$$n+1-y \leq (n+1)/2 \to y \geq (n+1)/2$$

Thus, the inequality becomes:

$$(n+1)/2 \leq y \leq n$$

The formula becomes:

$$\big(\forall x\, \big((n+1)/2 \leq y \leq n \to R[y] = r_y\big)\big)\big)$$

Changing y back into x, we have

$$(\forall x\, ((n+1)/2 \leq x \leq n \to R[x] = r_x)))$$

□

Implied (A):

$$(\forall x\,(1 \leq x \leq n \rightarrow R[x] = r_x)) \rightarrow Inv(1)$$

Proof. Assume that the premise is true. Our goal is to prove that the conclusion is true. Let's simplify the conclusion below.
The conclusion is $Inv(1)$. Take the invariant $Inv(j)$ and plug in $j = 1$, we have

$$\begin{aligned}&(((\forall x\,(1 \leq x \leq 0 \rightarrow (R[x] = r_{n+1-x} \wedge R[n+1-x] = r_x)))\\ &\wedge (\forall x\,(1 \leq x \leq (n+1)/2 \rightarrow (R[x] = r_x \wedge R[n+1-x] = r_{n+1-x}))))\\ &\wedge (1 \leq n/2+1))\end{aligned}$$

$1 \leq x \leq 0$ is always false for any integer x. Thus the first part of the above formula is always true. We can simplify the formula to the following.

$$\begin{aligned}&(\forall x\,(1 \leq x \leq (n+1)/2 \rightarrow (R[x] = r_x \wedge R[n+1-x] = r_{n+1-x}))))\\ &\wedge (1 \leq n/2+1))\end{aligned}$$

We can further simplify $1 \leq n/2+1$ and get $0 \leq n$. The formula becomes the following:

$$\begin{aligned}&(\forall x\,(1 \leq x \leq (n+1)/2 \rightarrow (R[x] = r_x \wedge R[n+1-x] = r_{n+1-x}))))\\ &\wedge (0 \leq n))\end{aligned} \tag{145}$$

Note that $0 \leq n$ is true because n is positive. There is an implicit assumption that the array has at least one element. By Lemma 3, formula 145 is logically equivalent to the premise. Therefore, the implied holds. □

To prove implied (C), we first prove Lemma 4 below.

Lemma 4. *Assume that n is odd (The array R has an odd number of elements). The two formulas below are logically equivalent.*

$$(((\forall x\,(1 \leq x \leq (n-1)/2 \rightarrow (R[x] = r_{n+1-x} \land R[n+1-x] = r_x))) \land (R[(n+1)/2] = r_{(n+1)/2})$$
$$\equiv ((\forall x\,(1 \leq x \leq n \rightarrow R[x] = r_{n+1-x})))$$

Proof. □

Implied (C):

$$(Inv(j) \wedge (\neg(2 * j \leq n))) \rightarrow ((\forall x\,(1 \leq x \leq n \rightarrow R[x] = r_{n+1-x})))$$

Proof. Let's simplify the premise.

$$\begin{gathered}(2j \leq n) \equiv (2j > n) \equiv (2j \geq n+1) \equiv (j \geq (n+1)/2)\\ (j \leq n/2+1) \equiv (j \leq (n+2)/2)\end{gathered}$$

If n is odd, then $j = (n+1)/2$. If n is even, then $j = (n+2)/2$.

Case 1: n is even and $j = (n+2)/2$. Plugging in $j = (n+2)/2$ into the premise, we have

$$\begin{aligned}&(((\forall x\,(1 \leq x \leq n/2 \rightarrow (R[x] = r_{n+1-x} \wedge R[n+1-x] = r_x)))\\ &\wedge (\forall x\,((n+1)/2 \leq x \leq (n+1)/2 \rightarrow (R[x] = r_x \wedge R[n+1-x] = r_{n+1-x}))))\\ &\wedge (j = (n+1)/2).\end{aligned} \tag{146}$$

$(n+1)/2 \leq x \leq (n+1)/2$ is always false. Thus, formula 146 is always true. We can simplify the formula to the following.

$$\begin{aligned}&(((\forall x\,(1 \leq x \leq n/2 \rightarrow (R[x] = r_{n+1-x} \wedge R[n+1-x] = r_x)))\\ &\wedge (j = (n+1)/2).\end{aligned} \tag{147}$$

We can prove that the formula 147 is logically equivalent to the conclusion of implied (C). (The argument is similar to Lemma 3). Therefore, when n is even, if the premise of implied (C) is true, then the conclusion of implied (C) must be true.

Case 2: n is odd and $j = (n+1)/2$. Plugging in $j = (n+1)/2$ into the premise, we have the following. (I've omitted the $j = (n+1)/2$ part of the formula.

$$\begin{aligned}&(((\forall x\,(1 \leq x \leq (n-1)/2 \rightarrow (R[x] = r_{n+1-x} \wedge R[n+1-x] = r_x)))\\ &\wedge (\forall x\,((n+1)/2 \leq x \leq (n+1)/2 \rightarrow (R[x] = r_x \wedge R[n+1-x] = r_{n+1-x}))))\end{aligned} \tag{148}$$

In formula 148, x must be equal to $(n+1)/2$. Thus, we can simplify formula 148 as follows.

$$\begin{aligned}&(\forall x\,((n+1)/2 \leq x \leq (n+1)/2 \rightarrow (R[x] = r_x \wedge R[n+1-x] = r_{n+1-x})))\\ &\equiv \big(R[(n+1)/2] = r_{(n+1)/2} \wedge R[(n+1)/2] = r_{(n+1)/2}\big))\\ &\equiv R[(n+1)/2] = r_{(n+1)/2}\end{aligned}$$

With this simplification, the premise becomes:

$$\begin{aligned}&(((\forall x\,(1 \leq x \leq (n-1)/2 \rightarrow (R[x] = r_{n+1-x} \wedge R[n+1-x] = r_x)))\\ &\wedge (R[(n+1)/2] = r_{(n+1)/2})\end{aligned}$$

By Lemma 4, the above formula is logically equivalent to the conclusion of implied (C). Therefore, when n is odd, if the premise of implied (C) is true, then the conclusion of implied (C) must be true.

□

Implied (B):

$$(Inv(j) \wedge (2 * j \leq n)) \rightarrow Inv(j+1)[R\{j \leftarrow R[n+1-j]\}\{n+1-j \leftarrow R[j]\}/R]$$

Proof. For convenience, let's define $Inv_p(j)$ to be the following formula:

$$(((\forall x\,(1 \leq x \leq j-1 \rightarrow (R[x] = r_{n+1-x} \wedge R[n+1-x] = r_x))) \tag{149}$$
$$\wedge\,(\forall x\,(j \leq x \leq (n+1)/2 \rightarrow (R[x] = r_x \wedge R[n+1-x] = r_{n+1-x})))) \tag{150}$$

The premise of implied (B) becomes

$$Inv_p(j) \wedge (j \leq n/2+1)) \wedge (2j \leq n)$$

For convenience, let R' denote $R\{j \leftarrow R[n+1-j]\}\{n+1-j \leftarrow R[j]\}$. The conclusion of implied (B) becomes

$$\begin{aligned}
&(((\forall x\,(1 \leq x \leq j \rightarrow (R'[x] = r_{n+1-x} \wedge R'[n+1-x] = r_x)))\\
&\wedge\,(\forall x\,(j+1 \leq x \leq (n+1)/2 \rightarrow (R'[x] = r_x \wedge R'[n+1-x] = r_{n+1-x}))))\\
&\wedge\,(j \leq n/2))\\
&\equiv Inv_p(j+1)[R'/R] \wedge (j \leq n/2)
\end{aligned}$$

To prove implied (B), it is sufficient for us to prove the following two implications:

Implication 1: $((j \leq n/2+1)) \wedge (2j \leq n)) \rightarrow (j \leq n/2)$**.**
Proof of implication 1: Assume that $(j \leq n/2+1))$ and $(2j \leq n)$ are true. $(2j \leq n)$ is equivalent to $(j \leq n/2)$. Since $n/2 < n/2+1$, we know that $(j \leq n/2)$, which is the conclusion that we need.

Implication 2: $Inv_p(j) \rightarrow Inv_p(j+1)[R'/R]$**.**
Proof of implication 2: Let's recall that what $Inv_p(j)$ and $Inv_p(j+1)[R'/R]$ are.

$$\begin{aligned}
Inv_p(j) \equiv\ &(((\forall x\,(1 \leq x \leq j-1 \rightarrow (R[x] = r_{n+1-x} \wedge R[n+1-x] = r_x)))\\
&\wedge\,(\forall x\,(j \leq x \leq (n+1)/2 \rightarrow (R[x] = r_x \wedge R[n+1-x] = r_{n+1-x}))))
\end{aligned}$$

$$\begin{aligned}
Inv_p(j+1)[R'/R] \equiv\ &(((\forall x\,(1 \leq x \leq j \rightarrow (R'[x] = r_{n+1-x} \wedge R'[n+1-x] = r_x)))\\
&\wedge\,(\forall x\,(j+1 \leq x \leq (n+1)/2 \rightarrow (R'[x] = r_x \wedge R'[n+1-x] = r_{n+1-x}))))
\end{aligned}$$

To prove implication 2, it is sufficient to prove the following implications.

For any $1 \leq x \leq j-1$,

$$(R[x] = r_{n+1-x} \wedge R[n+1-x] = r_x) \rightarrow (R'[x] = r_{n+1-x} \wedge R'[n+1-x] = r_x).$$

Recall that $R' = R\{j \leftarrow R[n+1-j]\}\{n+1-j \leftarrow R[j]\}$. The two assignments only potentially affect the elements at indices j and $n+1-j$. Thus, for any index x where

$1 \leq x \leq j-1$, the element of R is not affected. Thus, for any $1 \leq x \leq j-1$, $R[x] = R'[x]$. Thus, the implication holds.

For $j+1 \leq x \leq (n+1)/2$,

$$\big(R[x] = r_x \wedge R[n+1-x] = r_{n+1-x}\big) \rightarrow \big(R'[x] = r_x \wedge R'[n+1-x] = r_{n+1-x}\big).$$

Recall that $R' = R\{j \leftarrow R[n+1-j]\}\{n+1-j \leftarrow R[j]\}$. The two assignments only potentially affect the elements at indices j and $n+1-j$. Thus, for any index x where $j+1 \leq x \leq (n+1)/2$, the element of R is not affected. Thus, for any $j+1 \leq x \leq (n+1)/2$, $R[x] = R'[x]$. Thus, the implication holds.

For $x = j$,

$$\big(R[x] = r_x \wedge R[n+1-x] = r_{n+1-x}\big) \rightarrow \big(R'[x] = r_{n+1-x} \wedge R'[n+1-x] = r_x\big).$$

This is equivalent to the following implication:

$$\big(R[j] = r_j \wedge R[n+1-j] = r_{n+1-j}\big) \rightarrow \big(R'[j] = r_{n+1-j} \wedge R'[n+1-j] = r_j\big).$$

Since $j < (n+1)/2$, $j \neq n+1-j$. Thus, we have that

$$R'[j] = R\{j \leftarrow R[n+1-j]\}\{n+1-j \leftarrow R[j]\}[j] = R[n+1-j] = r_{n+1-j}.$$

by the first assignment $\{j \leftarrow R[n+1-j]\}$ and by our assumption that $R[n+1-j] = r_{n+1-j}$.

$$R'[n+1-j] = R\{j \leftarrow R[n+1-j]\}\{n+1-j \leftarrow R[j]\}[n+1-j] = R[j] = r_j.$$

by the second assignment$\{n+1-j \leftarrow R[j]\}$ and by our assumption that $R[j] = r_j$.

□

4 Undecidability

4.1 Prove that a problem is decidable

Collected Wisdom:

- When you describe an algorithm, make sure that it terminates. For example, if a set S is infinite, your algorithm cannot iterate through every element of S. For another example, it is okay to draw the truth table of a given formula because the truth table has finite size.

- An algorithm usually considers several cases. Make sure that you clearly indicate the return value of the algorithm in every case.

Exercise 150. *The propositional-satisfiability problem: Is the propositional formula α satisfiable?*
Prove that the propositional-satisfiability problem is decidable.

Solution: We are given the propositional formula α. We will draw the truth table of α. The formula α must have a finite number of propositional variables in it. Therefore, the truth table will have a finite size and we will need a finite amount of time to draw the truth table.

We will go through every cell in the final column of the truth table, where the truth value of α is indicated. If we can find one row of the truth table in which α is true, then α is satisfiable. Otherwise, if α is false in every row of the truth table, then α is not satisfiable.

Exercise 151. *The propositional-tautology problem: Is the propositional formula α a tautology?*
Prove that the propositional-tautology problem is decidable.

4.2 The Halting Problem is Undecidable

Exercise 152. *The Halting Problem: Given a program P and an input I, does P terminate when run with input I?*
Prove that the Halting Problem is undecidable.

Solution: This proof is adapted from a proof by Luwei Zhang. Thanks, Luwei!

Proof by Contradiction. Assume that the halting problem is decidable. There exists an algorithm H such that H takes a program P and an input I for I and returns yes if P terminates when run with input I and no otherwise.
We will construct a program X which takes a program Y as its input. X works as follows.

- X called $H(Y, Y)$ to predict whether program Y will terminate when run with input Y.
- If $H(Y, Y)$ returns yes, X goes into an infinite loop and does not terminate.
- If $H(Y, Y)$ returns no, X terminates immediately.

Now, consider what happens if we run the program X with itself as input.
Suppose that H predicts that X terminates when run with input X. Then by the construction of X, when $H(X, X)$ returns yes, X goes into an infinite loop and does not terminate. H's prediction was wrong.
Suppose that H predicts that X does not terminate when run with input X. Then by the construction of X, when $H(X, X)$ returns no, X terminates immediately. H's prediction was wrong again.
Therefore, H does not correctly predict whether X terminates when run with input X. This contradicts with our assumption that H can decide the Halting Problem.

□

4.3 Prove that a problem is undecidable

Collected Wisdom:

- Suppose that we are trying to prove that problem X is undecidable. In your reduction, make the inputs to the algorithm for solving problem X relate to P and I. After all, we are trying to construct an algorithm to determine whether P terminates when run with input I.
- To verify whether a reduction leads to a valid proof, consider two different cases: (1) P terminates when run with input I. (2) P does not terminate when run with input I. A reduction works if and only if the constructed algorithm gives the correct answer for both cases.
- A few useful constructions:
  1. Construct a program which runs P with input I.
  2. Construct a program which does nothing and terminates immediately.
  3. Construct a program which has an infinite loop and runs forever.
  4. Construct a program, which ignores its input and does one of 1, 2, and 3.

Exercise 153. *The halting-no-input problem: Given a program P that requires no input, does P halt?*
Prove that the halting-no-input problem is undecidable.

Solution:

Proof by Contradiction. Assume that there is an algorithm B which solves the halting-no-input problem. We will construct an algorithm A to solve the halting problem.

Algorithm A takes two inputs a program P and an input I. It works as follows:

- Constructs a program P', which runs P with input I.
- Runs algorithm B with the program P' as the input and returns the result $B(P')$.

By our construction of algorithm A, P' halts if and only if P halts on input I. Therefore, if algorithm B solves the halting-no-input problem for input P', then algorithm A solves the halting problem for inputs P and I.

By our assumption, algorithm B solves the halting-no-input problem. Thus, algorithm A solves the halting problem.

This contradicts the fact that the halting problem is undecidable.

□

Exercise 154. *The both-halt problem: Given two programs $P1$ and $P2$ that take no input, do both programs halt?*
Prove that the both-halt problem is undecidable.

Solution:

Proof by Contradiction. Assume that there is an algorithm B which solves the both-halt problem. We will construct an algorithm A to solve the halting problem.

Algorithm A takes two inputs a program P and an input I. It works as follows:

- Constructs a program P', which runs P with input I.
- Runs algorithm B with the two programs P' and P' as its inputs and returns the result $B(P', P')$.

By our construction of algorithm A, both programs P' and P' halt if and only if P halts on input I. Therefore, if algorithm B solves the both-halt problem for inputs P' and P', then the algorithm A solves the halting problem for inputs P and I.

By our assumption, algorithm B solves the both-halt problem. Thus, algorithm A solves the halting problem.

This contradicts the fact that the halting problem is undecidable.

□

Remark 21. *Other reductions:*

- *Let $P1$ do nothing. Let $P2$ run P with input I. (This works.)*
- *Let $P1$ contain an infinite loop. Let $P2$ run P with input I. (This does NOT work.)*

Remark 22. *A variant of this problem:*
Consider the both-run-forever problem: Given two programs $P1$ and $P2$, do both programs run forever?
Prove that the both-run-forever problem is undecidable.

Exercise 155. *We say that two problems agree on all input if and only if, for every input x, either they both run forever, or they both halt and return the same value.*
The program-agreement problem: Given two programs, do they agree on all inputs?
Prove that the program-agreement problem is undecidable.

Exercise 156. *The total-correctness problem: Given a Hoare triple, is the triple satisfied under total correctness?*
Prove that the total correctness problem is undecidable.

Exercise 157. *The partial-correctness problem: Given a Hoare triple, is the triple satisfied under partial correctness?*
Prove that the partial-correctness problem is undecidable.

Solution:

Proof by Contradiction. Assume that there is an algorithm B which solves the total-correctness problem. We will construct an algorithm A to solve the halting problem.

Algorithm A takes two inputs a program P and an input I. It works as follows:

- Constructs a program P', which runs P with input I.
- Constructs the Hoare triple $(\!|\, true \,|\!)\ P'\ (\!|\, false \,|\!)$.
- Runs algorithm B with the Hoare triple $(\!|\, true \,|\!)\ P'\ (\!|\, false \,|\!)$ as its input.
- Return the negation of the result $B((\!|\, true \,|\!)\ P'\ (\!|\, false \,|\!))$. (If B returns true, then A returns false, and vice versa.)

If the program P' terminates, then the triple $(\!|\, true \,|\!)\ P'\ (\!|\, false \,|\!)$ is NOT satisfied under partial correctness because the postcondition is false. Therefore, the only way for the triple to be satisfied under partial correctness is when P' does not terminate.

By our construction of algorithm A, P halts on input I if and only if P' halts. P' halts if and only if the triple $(\!|\, true \,|\!)\ P'\ (\!|\, false \,|\!)$ is NOT satisfied under partial correctness. Thus, P halts on input I if and only if the triple $(\!|\, true \,|\!)\ P'\ (\!|\, false \,|\!)$ is NOT satisfied under partial correctness.

Therefore, if algorithm B solves the partial-correctness problem for input $(\!|\, true \,|\!)\ P'\ (\!|\, false \,|\!)$, then the algorithm A solves the halting problem for inputs P and I.

By our assumption, algorithm B solves the partial-correctness problem. Thus, algorithm A solves the halting problem.

This contradicts the fact that the halting problem is undecidable.

□

Exercise 158. *The exists-halting-input problem: Given a program P, does there exist an input I such that P halts with input I?*
Prove that this problem is undecidable.

Exercise 159. *The halt-every-input problem: Given a program P, does P halt for every input?*
Prove that the halt-every-input problem is undecidable.