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.**

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.

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.

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.

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.

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.*

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

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* $((\neg a) \lor b)$. *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
- $((r \land s) \to r)$
- $((\neg(p \leftrightarrow q)) \leftrightarrow (q \lor p))$
- $((((p \lor q) \land (p \lor (\neg q))) \land ((\neg p) \lor q)) \land ((\neg p) \lor (\neg q)))$

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. $(r \rightarrow (s \rightarrow g))$
2. $((s \wedge r) \rightarrow g)$
3. $(s \rightarrow (r \rightarrow g))$

Prove that all three translations are logically equivalent.

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. $((s \to (\neg c)) \wedge a)$
2. $(s \to ((\neg c) \wedge a))$

Prove that the two translations are NOT logically equivalent.

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 }
```

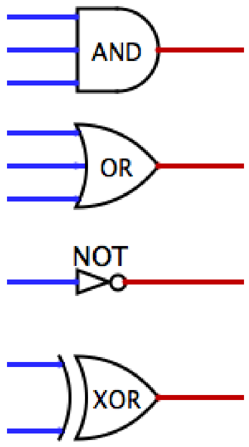
Code fragment #2:

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

Prove that these two pieces of code fragments are equivalent:

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.

1.8 Semantic Entailment

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

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

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

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

Exercise 15. *Prove that* $\{(\neg(p \wedge q)), (p \rightarrow q)\} \not\models (p \leftrightarrow q)$.

Exercise 16. *Prove that* $\emptyset \models ((p \wedge q) \rightarrow p))$.

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

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

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

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 $\neg$i and $\neg\neg$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* $\{(p \wedge q), (r \wedge s)\} \vdash (q \wedge s)$.

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

1.9.3 Implication introduction and elimination

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

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

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

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

1.9.4 Or elimination and introduction

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

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

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

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

1.9.5 Negation introduction and double negation elimination

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

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

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

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

1.9.6 Negation elimination

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

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

1.9.7 Putting them together!

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

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

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

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

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

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

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

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

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 $\Sigma \vdash ((\neg p) \to q)$.*

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.*

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

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.

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

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.*

Exercise 51. *Consider the following inference rule:*

$$\frac{(\alpha \to \beta)}{(\beta \to \alpha)} \text{ *Flip the implication*}$$

where α *and* β *are well-formed propositional formulas.*

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

$$\{(\alpha \to \beta)\} \not\models (\beta \to \alpha)$$

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.*

1.10.2 Soundness and Completeness of Natural Deduction

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

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

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.

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.

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.
2. A student has taken a course.
3. No student has taken every course.
4. Some student has not taken any course.
5. Every student has taken every course.

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

- There is at least one bear.

- There are at least two bears.

- There is at most one bear.

- There is exactly one bear.

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 $dom(I) = \{1, 2, 3\}$.*

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

2.2.2 Evaluating Formulas with Free Variables Only

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

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

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 $dom(I) = \{1, 2, 3\}$.*

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

2.2.4 Evaluating Formulas with Bound Variables Only

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

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* $dom(I) = \{1,2,3\}$.

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* $\{(\forall x\ P(x))\} \vDash (\exists x\ P(x))$.

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

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.*

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.*

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

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

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

Remark 1. *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* $(P(x) \vee Q(y))$. *It turns out that, having this concrete predicate formula allows us to prove the entailment.*

Remark 2. *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* $(P(x) \vee Q(y))$ *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* $(P(x) \vee Q(y))$ *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* $(P(x) \vee Q(y))$ *must be true.*

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. $\{(\exists x\,(P(x) \to Q(x))), (\forall y\ P(y))\} \vDash (\exists x\ Q(x))$

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

2.4.2 Exists-introduction

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

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

2.4.3 Forall-introduction

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

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

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

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. $\{(\forall x\,((\neg P(x)) \wedge Q(x)))\} \vdash (\forall x\,(P(x) \rightarrow Q(x)))$.

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

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. $\{(\forall x\,(P(x) \rightarrow Q(x)))\} \vdash ((\forall x\,(\neg Q(x))) \rightarrow (\forall x\,(\neg P(x))))$.

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

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

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

2.4.5 Exists-elimination

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

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

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

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

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

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

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

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

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

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:*

$$\{(\forall x\ \alpha)\} \vDash \alpha[t/x] \tag{1}$$

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 $(\forall x\ \alpha)^{(I,E)} = \mathtt{T}$. We need to show that $\alpha[t/x]^{(I,E)} = \mathtt{T}$.

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

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

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

Thus, the entailment holds and the inference rule is sound. $\square$

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

$$\{\alpha[t/x]\} \vDash (\exists x\ \alpha) \tag{2}$$

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. □

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

$$\frac{\alpha[t/x]}{(\forall x\ \alpha)}\ \forall i* \tag{3}$$

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

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

$$\frac{(\exists x\ \alpha)}{\alpha[t/x]}\ \exists e* \tag{4}$$

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

2.5.2 Additional Exercises

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

$$\frac{(\forall x(\alpha \to \beta)) \quad \alpha[t/x]}{\beta[t/x]} \,\forall e1 \tag{5}$$

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.*

$$\frac{(\forall x(\alpha \to \beta)) \quad (\neg\beta[t/x])}{(\neg\alpha[t/x])} \,\forall e2 \tag{6}$$

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.*

$$\frac{(\forall x(\alpha \to \beta)) \quad \beta[t/x]}{\alpha[t/x]} \,\forall e3 \tag{7}$$

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.*

$$\frac{(\forall x(\alpha \to \beta)) \quad (\neg\alpha[t/x])}{(\neg\beta[t/x])} \,\forall e4 \tag{8}$$

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. □

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?

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?

Exercise 132. *Consider the Hoare triple* $(\!|\, (x > 0) \,|\!)\ C_3\ (\!|\, ((y * y) < x) \,|\!)$.

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?

Exercise 133. *Consider the Hoare triple* $(\!|\, (x > 0) \,|\!)\ C_4\ (\!|\, ((y * y) < x) \,|\!)$.

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?

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

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

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!) ⦈
```

3.2 Assignment Statements

Complete the following annotations.

$\llparenthesis \qquad\qquad \rrparenthesis$

```
x = 2;
```

$\llparenthesis\, (x = 2)\, \rrparenthesis$

$\llparenthesis \qquad\qquad \rrparenthesis$

```
x = 2;
```

$\llparenthesis\, (x = y)\, \rrparenthesis$

$\llparenthesis \qquad\qquad \rrparenthesis$

```
x = 2;
```

$\llparenthesis\, (x = 0)\, \rrparenthesis$

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

```
x = x + 1;
```

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

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

```
x = y;
```

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

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

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

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

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

3.3 Conditional Statements

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

```
$(\!|\, true \,|\!)$
$\textbf{if}$ $(x > y)$ {
    $max = x;$
} $\textbf{else}$ {
    $max = y;$
}
$(\!|\, (((x > y) \land (max = x)) \lor ((x \leq y) \land (max = y))) \,|\!)$
```

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

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

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

```
(| true |)
if (max < x) {
    max = x;
}
(| (max ≥ x) |)
```

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

```
⦇ true ⦈
if (x % 2 == 1) {
   x = x + 1;
}
⦇ (∃u (x = (2 ∗ u))) ⦈
```

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

```
(| true |)
if (x < 5) {
   r = 0;
} else {
   if (x > 10) {
      r = 0;
   } else {
      r = 1;
   }
}
```

$\llparenthesis\, ((((x < 5) \vee (x > 10)) \wedge (r = 0)) \vee (((5 \leq x) \wedge (x \leq 10)) \wedge (r = 1)))\, \rrparenthesis$

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 {
    $z = z - 2 * a * b;$
}
$(\!|\ ((\exists u\ (u * u = z))\ |\!)$
```

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 3. *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.*

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

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

So far, we have found three invariants: $(z \leq x)$, $(y = z!)$, *and* $((z \leq x) \wedge (y = z!))$. *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* $((z \leq x) \wedge (y = z!))$*) 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 $(y = z!)$.

Exercise 145. *Show that the following triple is satisfied under partial correctness.*

```
⦇ $(x \geq 0)$ ⦈
y = 1;
z = 0;
while (z < x) {
    z = z + 1;
    y = y * z;
}
⦇ $(y = x!)$ ⦈
```

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.*

$(\!|\ ((A[x] = x0) \wedge (A[y] = y0))\ |\!)$
$t\ =\ A[x];$
$A[x]\ =\ A[y];$
$A[y]\ =\ t;$
$(\!|\ ((A[x] = y0) \wedge (A[y] = x0))\ |\!)$ $\quad array\ \ assignment$

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.

```
$\llparenthesis\, ((\forall x\, (1 \leq x \leq n \to R[x] = r_x)))\, \rrparenthesis$
j = 1;
while (2*j <= n) {
    t = R[j];
    R[j] = R[n+1-j];
    R[n+1-j] = t;
    j = j + 1;
}
$\llparenthesis\, ((\forall x\, (1 \leq x \leq n \to R[x] = r_{n+1-x})))\, \rrparenthesis$
```

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.

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.

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.

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.

Remark 4. *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 5. *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.

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.