Propositional Logic: Equivalence

Alice Gao

Lecture 5

Based on work by J. Buss, L. Kari, A. Lubiw, B. Bonakdarpour, D. Maftuleac, C. Roberts, R. Trefler, and P. Van Beek

Outline

Learning goals

By the end of this lecture, you should be able to

(Logical equivalence)

- Prove that two formulas are logically equivalent using logical identities.
- Translate a condition in a block of code into a propositional formula. Simplify an if statement. Determine whether a piece of code is live or dead.

(Circuit design)

- Write down a truth table given a problem description.
- Convert a truth table to a propositional formula.
- Convert a propositional formula to a circuit diagram using AND, OR, NOT, and XOR gates.

Learning goals

By the end of this lecture, you should be able to

(Adequate set of connectives)

- Prove that a connective is definable in terms of a set of connectives.
- Prove that a set of connectives is adequate.
- Prove that a set of connectives is not adequate.

Definition of logical equivalence

Two formulas α and β are logically equivalent, denoted $\alpha \equiv \beta$:

- $\alpha^t = \beta^t$, for every valuation t.
- α and β have the same final column in their truth tables.
- $(\alpha \leftrightarrow \beta)$ is a tautology.

Why do we care about logical equivalence?

- Do these two formulas have the same meaning?
- Do these two circuits behave the same way?
- Do these two pieces of code fragments behave the same way?

Proving logical equivalence

Two approaches:

- Truth tables
- Logical identities

Logical Identities

Commutativity:

$(\alpha \land \beta) \equiv (\beta \land \alpha)$
$(\alpha \lor \beta) \equiv (\beta \lor \alpha)$
$(\alpha \leftrightarrow \beta) \equiv (\beta \leftrightarrow \alpha)$

Associativity

$(\alpha \land (\beta \land \gamma)) \equiv ((\alpha \land \beta) \land \gamma)$
$(\alpha \lor (\beta \lor \gamma)) \equiv ((\alpha \lor \beta) \lor \gamma)$

Distributivity

$(\alpha \lor (\beta \land \gamma)) \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma))$
$(\alpha \land (\beta \lor \gamma)) \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma))$

Idempotence

$(\alpha \lor \alpha) \equiv \alpha$
$(\alpha \land \alpha) \equiv \alpha$

Double Negation

$(\neg(\neg\alpha)) \equiv \alpha$

De Morgan's Laws

$(\neg(\alpha \land \beta)) \equiv ((\neg\alpha) \lor (\neg\beta))$
$(\neg(\alpha \lor \beta)) \equiv ((\neg\alpha) \land (\neg\beta))$

Logical Identities (continued)

Simplification I (Absorbtion):
$(\alpha \wedge \mathsf{T}) \equiv \alpha$
$(\alpha \vee \mathsf{T}) \equiv \mathsf{T}$
$(\alpha \wedge \mathsf{F}) \equiv \mathsf{F}$
$(\alpha \vee \mathsf{F}) \equiv \alpha$

Simplification II
$(\alpha \vee (\alpha \wedge \beta)) \equiv \alpha$
$(\alpha \wedge (\alpha \vee \beta)) \equiv \alpha$

Implication
$(\alpha \rightarrow \beta) \equiv ((\neg\alpha) \vee \beta)$

Contrapositive
$(\alpha \rightarrow \beta) \equiv ((\neg\beta) \rightarrow (\neg\alpha))$

Equivalence
$(\alpha \leftrightarrow \beta) \equiv ((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha))$

Excluded Middle
$(\alpha \vee (\neg\alpha)) \equiv \mathsf{T}$

Contradiction
$(\alpha \wedge (\neg\alpha)) \equiv \mathsf{F}$

CQ 22, 23, and 24 Logical identities and the implication

A logical equivalence proof

Theorem: $(((\neg p) \wedge q) \vee p) \equiv (p \vee q)$.

Proof.

$$
\begin{aligned}
&(((\neg p) \wedge q) \vee p) && && (1)\\
&\equiv (((\neg p) \vee p) \wedge (q \vee p)) && \text{Distributivity} && (2)\\
&\equiv (\mathrm{T} \wedge (q \vee p)) && \text{Excluded Middle} && (3)\\
&\equiv (q \vee p) && \text{Simplification I} && (4)\\
&\equiv (p \vee q) && \text{Commutativity} && (5)
\end{aligned}
$$

$\square$

Proving logical equivalence

"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.

Three possible translations:

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

Prove that all three translations are logically equivalent.

Strategies for proving logical equivalence

- Try getting rid of $\rightarrow$ and $\leftrightarrow$.
- Try moving negations inward using De Morgan's law. $(\neg(p \vee q)) \equiv ((\neg p) \wedge (\neg q))$.
- Work from the more complex side first.
- Switch to different strategies/sides when you get stuck.
- In the end, write the proof in clean "one-side-to-the-other" form and double-check steps.

Proving non-equivalence

"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.
2 possible translations:

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

Theorem: The two translations are NOT logically equivalent.
Which valuation t can we use to prove this theorem?

(A) $t(s) = \mathrm{F}$, $t(c) = \mathrm{T}$, $t(a) = \mathrm{F}$
(B) $t(s) = \mathrm{F}$, $t(c) = \mathrm{F}$, $t(a) = \mathrm{F}$
(C) $t(s) = \mathrm{T}$, $t(c) = \mathrm{F}$, $t(a) = \mathrm{T}$
(D) Two of (a), (b), and (c).
(E) All of (a), (b), and (c).

Analyzing Conditional Code

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

Define the propositional variable below.

i: input > 0

u: output

q: queuelength < 100

Your code fragment

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

Your friend's code fragment

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

Two Equivalent Code Fragments

Prove that the two code fragments are equivalent.

Listing 1: Your code

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

Listing 2: Your friend's code

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

P_2 is Dead Code

- Write down the condition leading to P_2 in your code fragment.
- Prove that P_2 is dead code. That is, the conditions leading to P_2 is logically equivalent to F.

When is P_3 executed?

- Write down the condition leading to P_3 in your code fragment.
- Prove that P_3 is executed if and only if i, u and q are all true.

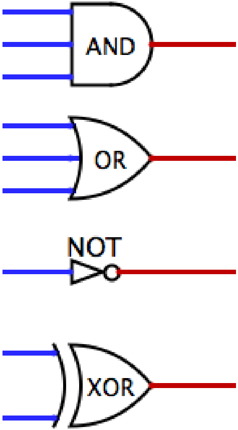
When is P_4 executed?

- Write down the condition leading to P_4 in your code fragment.
- Prove that P_4 is executed if and only if i is false and u is true.

Digital Circuits

- An electronic computer is made up of a number of circuits.
- The basic elements of circuits are called logic gates.
- A gate is an electronic device that operates on a collection of binary inputs and produces a binary output.

Logical Gates



A circuit design 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.

Design the circuit

1. Draw the truth table based on the problem description.
2. Convert the truth table to a propositional formula.
3. Convert the formula to a circuit.

Step 1: Draw the truth table

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

Adequate Sets of Connectives - Questions

- Why did we learn these five connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$?
- Using these connectives, can we express every propositional logic formula that we ever want to write?
- Are there any connectives in this set that are not necessary?
- Are there other connectives that we could define and use? Is there another set of connectives that we should have studied instead?

Adequate Sets of Connectives - Answers

- Why did we learn these five connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$?
- Using these connectives, can we express every propositional logic formula that we ever want to write?
  Yes. Can you prove this?
- Are there any connectives in this set that are not necessary?
  Yes. Recall that $(x \rightarrow y) \equiv ((\neg x) \vee y)$. We don't need $\rightarrow$ at all. (We say that $\rightarrow$ is definable in terms of $\neg$ and $\vee$.)
- Are there other connectives that we could define and use?
  Yes. For example, we could define NOR and NAND connectives.
- Is there another set of connectives that we should have studied instead?
  That depends on what we want to use them for.

A few adequate and non-adequate sets

If a set of connectives is sufficient to express every possible propositional formula, we call it an adequate set of connectives. That is, any other connective not in this set is definable in terms of the ones in this set.

Theorem 1. $\{\wedge, \vee, \neg\}$ is an adequate set of connectives.

Theorem 2. $\{\wedge, \neg\}$, $\{\vee, \neg\}$, and $\{\rightarrow, \neg\}$ are adequate sets.

Theorem 3. The set $\{\wedge, \vee\}$ is *not* an adequate set of connectives.

An adequate set to start with

Theorem 1. $\{\wedge, \vee, \neg\}$ is an adequate set of connectives.

Hint: We need to show that every other connective can be definable in terms of the connectives in this set. For every connective, we can write down its truth table. Can you convert every truth table to a propositional formula using only $\wedge$, $\vee$ and $\neg$ as connectives?

Other sets are also adequate

Theorem 2. $\{\wedge, \neg\}$, $\{\vee, \neg\}$, and $\{\rightarrow, \neg\}$ are adequate sets.

By Theorem 1, the set $\{\wedge, \vee, \neg\}$ is adequate.

To prove that $\{\wedge, \neg\}$ is adequate, we need to show that $\vee$ is definable in terms of $\wedge$ and $\neg$.

To prove that $\{\vee, \neg\}$ is adequate, we need to show that $\wedge$ is definable in terms of $\vee$ and $\neg$.

To prove that $\{\rightarrow, \neg\}$ is adequate, we need to show that each of $\vee$ and $\wedge$ is definable in terms of $\rightarrow$ and $\neg$.

A non-adequate set

Theorem 3. The set $\{\wedge, \vee\}$ is *not* an adequate set of connectives.

Consider any formula which uses only $\wedge$ and $\vee$ as connectives. Consider a valuation t under which every propositional variable is true. What is the truth value of the formula under t?

a. Always true
b. Always false
c. Sometimes true and sometimes false
d. Not enough information to tell

A non-adequate set

Theorem 3. The set $\{\wedge, \vee\}$ is *not* an adequate set of connectives.

Lemma: For any formula which uses only $\wedge$ and $\vee$ as connectives, under a valuation which makes every variable true, the formula is true.
Prove this lemma using structural induction.

This lemma means that it is impossible to negate a formula using only $\wedge$ and $\vee$. Why?

Revisiting the learning goals

By the end of this lecture, you should be able to

(Logical equivalence)

- Prove that two formulas are logically equivalent using logical identities.
- Translate a condition in a block of code into a propositional formula. Simplify an if statement. Determine whether a piece of code is live or dead.

(Circuit design)

- Write down a truth table given a problem description.
- Convert a truth table to a propositional formula.
- Convert a propositional formula to a circuit diagram using AND, OR, NOT, and XOR gates.

Revisiting the learning goals

By the end of this lecture, you should be able to

(Adequate set of connectives)

- Prove that a connective is definable in terms of a set of connectives.
- Prove that a set of connectives is adequate.
- Prove that a set of connectives is not adequate.