# Warm Up Problem

Let $t$ be a truth valuation, $p$ and $q$ propositional variables with $t(p) = \mathtt{T}$, and let $\alpha$ be a well-formed formula. Which of the following are written using the correct notation? What does the equality-type character mean on each line?

1. $\alpha = \mathtt{T}$
2. $\alpha^t = \mathtt{T}$
3. $\alpha^t \equiv \mathtt{T}$
4. $(p \wedge q)^t = (\mathtt{T} \wedge q)$
5. $(p \wedge q)^t \equiv (\mathtt{T} \wedge q^t)$
6. $(p \wedge q) = (q \wedge p)$
7. $(p \wedge q) \equiv (q \wedge p)$

# *Propositional Logic: Semantics*

Carmen Bruni
With thanks to A. Gao for these slides!

Based on slides by Jonathan Buss, Lila Kari, Anna Lubiw and Steve Wolfman with thanks to B. Bonakdarpour, A. Gao, D. Maftuleac, C. Roberts, R. Trefler, and P. Van Beek

Lecture 5

# Outline for today

- How can we prove that two formulas have the same meaning? (Logical equivalence)
- Analyzing dead code.
- Circuit design.

# Learning goals

Logical equivalence of formulas:

- Prove that the logical equivalence of formulas using truth tables and/or logical identities.
- Describe strategies to prove logical equivalence using logical identities.
- Translate a condition in a block of code into a propositional logic formula.
- Simplify code using truth tables and logical identities.
- Determine whether a piece of code is live or dead using truth tables and logical identities.

Dead Code:

- Determine whether or not a specific line of code is reachable.
- Give examples of parameters that reach a specific line of code.

# Definition of logical equivalence

Two formulas $\alpha$ and $\beta$ are logically equivalent if and only if they have the same value under any valuation.

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

# Why do we care about logical equivalence?

- Will I lose marks if I provide a solution that is syntactically different but logically equivalent to the provided solution?
- Do these two circuits behave the same way?
- Do these two pieces of code fragments behave the same way?

You already know one way of proving logical equivalent. What is it?

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

# Logical Identities

Commutativity

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

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, cont'd

Simplification I (Absorption)

$$(\alpha \wedge \mathtt{T}) \equiv \alpha$$
$$(\alpha \vee \mathtt{T}) \equiv \mathtt{T}$$
$$(\alpha \wedge \mathtt{F}) \equiv \mathtt{F}$$
$$(\alpha \vee \mathtt{F}) \equiv \alpha$$

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

$$(\alpha \vee (\neg \alpha)) \equiv \mathtt{T}$$

Contradiction

$$(\alpha \wedge (\neg \alpha)) \equiv \mathtt{F}$$

# A logical equivalence proof

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

# A logical equivalence proof

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

Proof.

| | |
|---|---|
| $(((\neg p) \wedge q) \vee p)$ | |
| $\equiv (((\neg p) \vee p) \wedge (q \vee p))$ | Distributivity |
| $\equiv (\mathtt{T} \wedge (q \vee p))$ | Excluded Middle |
| $\equiv (q \vee p)$ | Simplification I |
| $\equiv (p \vee q)$ | Commutativity |

□

# A practice problem

"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.

A few translations:

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

Theorem: All three translations are logically equivalent.
Proof: Done in class.

# How do you prove non-equivalence?

"If it snows then I won't 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. $((s \rightarrow (\neg c)) \wedge a)$
2. $(s \rightarrow ((\neg c) \wedge a))$

Theorem: $((s \rightarrow (\neg c)) \wedge a)$ and $(s \rightarrow ((\neg c) \wedge a))$ are not logically equivalent.

# How do you prove non-equivalence?

"If it snows then I won't 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: $((s \to (\neg c)) \wedge a)$ and $(s \to ((\neg c) \wedge a))$ are not logically equivalent.

Which valuation $t$ can we use to prove this theorem?

a. $s^t = \mathtt{F}$, $(\neg c)^t = \mathtt{F}$, $a^t = \mathtt{F}$
b. $s^t = \mathtt{F}$, $(\neg c)^t = \mathtt{T}$, $a^t = \mathtt{F}$
c. $s^t = \mathtt{T}$, $(\neg c)^t = \mathtt{T}$, $a^t = \mathtt{T}$
d. Two of these.
e. All of these.

# Collected Wisdom

- Try getting rid of $\rightarrow$ and $\leftrightarrow$.
- Try moving negations inward. $\neg(p \vee q) \equiv (\neg p) \wedge (\neg q)$.
- Work from the more complex side first, BUT
- 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.

```
if ( (input > 0) or (not output) ) {
    if ( not (output and (queuelength < 100) ) ) {
        P1
    } else if ( output and (not (queuelength < 100)) ) {
        P2
    } else { P3 }
} else { P4 }
```

When does each piece of code get executed?

Let $i$: `input > 0`,
$u$: `output`,
$q$: `queuelength < 100`.

# A Code Example, cont'd

| $i$ | $u$ | $q$ | $(i \vee (\neg u))$ | $(\neg(u \wedge q))$ | $(u \wedge (\neg q))$ | |
|---|---|---|---|---|---|---|
| T | T | T | T | | | |
| T | T | F | T | | | |
| T | F | T | T | | | |
| T | F | F | T | | | |
| F | T | T | F | | | $P_4$ |
| F | T | F | F | | | $P_4$ |
| F | F | T | T | | | |
| F | F | F | T | | | |

# A Code Example, cont'd

| $i$ | $u$ | $q$ | $(i \vee (\neg u))$ | $(\neg(u \wedge q))$ | $(u \wedge (\neg q))$ | |
|---|---|---|---|---|---|---|
| T | T | T | T | F | F | $P_3$ |
| T | T | F | T | T | | $P_1$ |
| T | F | T | T | T | | $P_1$ |
| T | F | F | T | T | | $P_1$ |
| F | T | T | F | | | $P_4$ |
| F | T | F | F | | | $P_4$ |
| F | F | T | T | T | | $P_1$ |
| F | F | F | T | T | | $P_1$ |

# Finding Live Code

Prove that $P_3$ is live code. That is, the conditions leading to $P_3$ is satisfiable.

Theorem:
$$\Big((i \vee (\neg u)) \wedge \big((\neg(\neg(u \wedge q))) \wedge (\neg(u \wedge (\neg q)))\big)\Big) \equiv ((i \wedge u) \wedge q)$$

Proof: In class

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
}
```

# Simplifying Code

To prove that the two fragments are equivalent, show that each block of code $P_1$, $P_2$, $P_3$, and $P_4$ is executed under equivalent conditions.

| Block | Fragment 1 | Fragment 2 |
|---|---|---|
| $P_1$ | $\big(i \vee (\neg u)\big) \wedge \big(\neg(u \wedge q)\big)$ | $\big(\neg(i \wedge u \wedge q)\big) \wedge \big(\neg((\neg i) \wedge u)\big)$ |
| $P_2$ | $\big(i \vee (\neg u)\big) \wedge \big(\neg(\neg(u \wedge q))\big) \wedge \big(u \wedge (\neg q)\big)$ | F |
| $P_3$ | $\big(i \vee (\neg u)\big) \wedge \big(\neg(\neg(u \wedge q))\big) \wedge \big(\neg(u \wedge (\neg q))\big)$ | $(i \wedge u \wedge q)$ |
| $P_4$ | $\big(\neg(i \vee (\neg u))\big)$ | $\big(\neg(i \wedge u \wedge q)\big) \wedge \big((\neg i) \wedge u\big)$ |

# Another logic puzzle

A very special island is inhabited only by knights and knaves. Knights always tell the truth, and knaves always lie.

You meet three inhabitants: Alice, Rex and Bob.

Alice says, "Rex is a knave."

Rex says, "It is false that Bob is a knave."

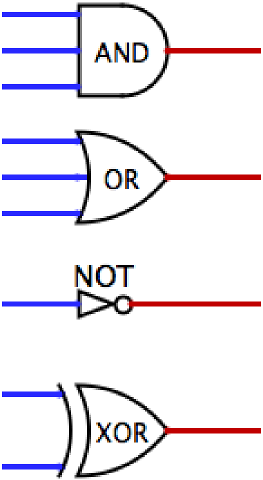
Bob claims, "I am a knight or Alice is a knight."

Can you determine who is a knight and who is a knave?

# 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.

# 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 | |

# Draw the truth table

| 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 |

1. Convert the truth table a propositional formula.

   For convenience, we will use the symbol $\oplus$ to represent an exclusive OR connective. This is a temporary convenience only. You are not allowed to use this connective unless otherwise specified in the problem.

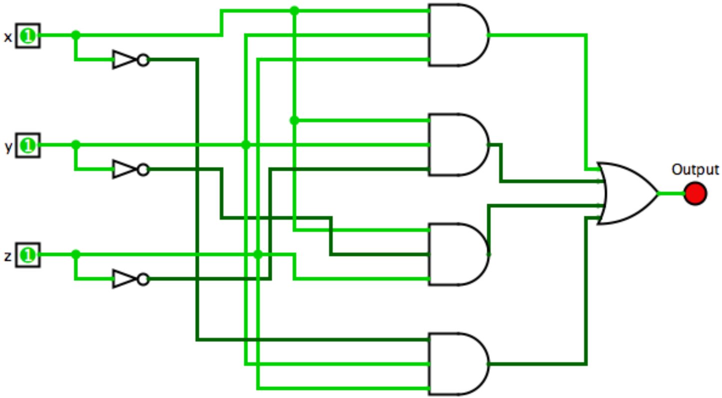
2. Then, convert the formula to a circuit.

# Solution 1

1. Convert each row of the truth table to a conjunction.
   - $((x \wedge y) \wedge z)$
   - $((x \wedge y) \wedge (\neg z))$
   - $((x \wedge (\neg y)) \wedge z)$
   - $(((\neg x) \wedge y) \wedge z)$
2. Connect all formulas to form a disjunction. (Below is not well formed to save confusion)

   $$((x \wedge y) \wedge z) \vee ((x \wedge y) \wedge (\neg z)) \vee ((x \wedge (\neg y)) \wedge z) \vee (((\neg x) \wedge y) \wedge z)$$
3. Draw the circuit.

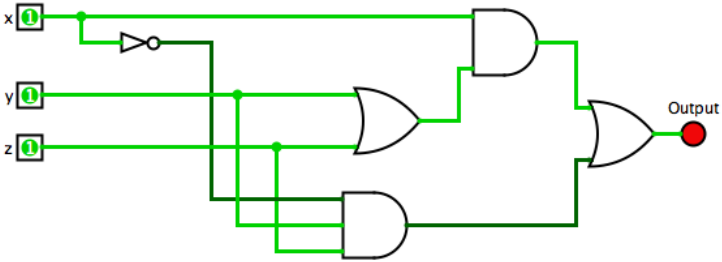
# Circuit 1



# Solution 2

1. Converts rows 1-3 to a propositional formula.
   $(x \wedge (y \vee z))$
2. Convert row 5 to a propositional formula.
   $(((\neg x) \wedge y) \wedge z)$
3. Connect all formulas into a disjunction.
   $((x \wedge (y \vee z)) \vee (((\neg x) \wedge y) \wedge z))$
4. Draw the circuit.

# Circuit 2



# Solution 3

1. Convert rows 1 and 5 into a propositional formula.
   $(y \wedge z)$
2. Convert rows 2 and 3 into a propositional formula.
   $(x \wedge (y \oplus z))$
3. Connect all formulas into a disjunction.
   $((y \wedge z) \vee (x \wedge (y \oplus z)))$
4. Draw the circuit.

# Circuit 3