CS 441 Discrete Mathematics for CS Lecture 2

Propositional logic

Milos Hauskrecht

milos@cs.pitt.edu 5329 Sennott Square

CS 441 Discrete mathematics for CS

M. Hauskrecht

Course administration

Homework 1

- Is out and posted on the course web page
- Due on Thursday, September 10, 2009.

Recitations this week:

- Friday, September 4, 2009
- 5313 Sennott Square Bldg.

Course web page:

http://www.cs.pitt.edu/~milos/courses/cs441/

CS 441 Discrete mathematics for CS

Propositional logic: review

- **Propositional logic:** a formal language for representing knowledge and for making logical inferences
- A proposition is a statement that is either true or false.
- A compound proposition can be created from other propositions using logical connectives
- The truth of a compound proposition is defined by truth values of elementary propositions and the meaning of connectives.
- The truth table for a compound proposition: table with entries (rows) for all possible combinations of truth values of elementary propositions.

CS 441 Discrete mathematics for CS

M. Hauskrecht

Compound propositions

- More complex propositional statements can be build from the elementary statements using **logical connectives**.
- Logical connectives:
 - Negation
 - Conjunction
 - Disjunction
 - Exclusive or
 - Implication
 - Biconditional

CS 441 Discrete mathematics for CS

Compound propositions

- Let p: 2 is a prime T q: 6 is a prime F
- Determine the truth value of the following statements:
 - ¬p: **F**
 - $p \wedge q : \mathbf{F}$
 - $p \wedge \neg q$: **T**
 - $p \vee q : T$
 - $p \oplus q$: T
 - $p \rightarrow q$: **F**
 - $q \rightarrow p$: T

CS 441 Discrete mathematics for CS

M. Hauskrecht

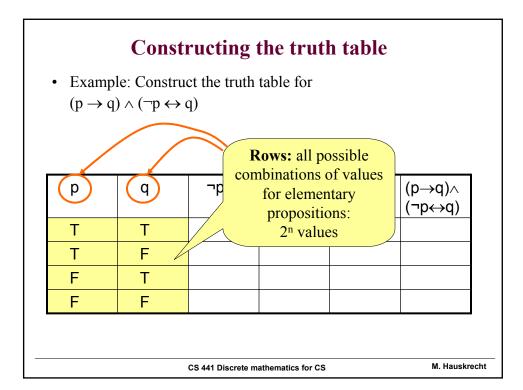
Constructing the truth table

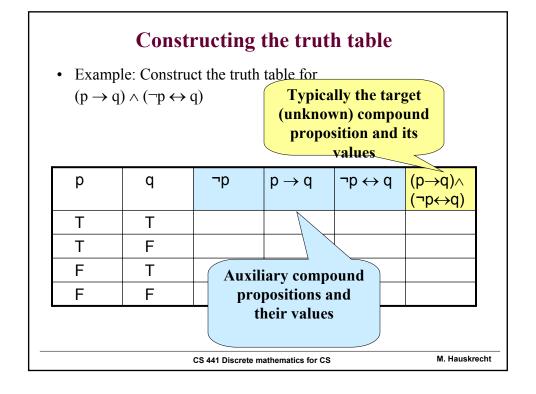
• Example: Construct the truth table for

$$(p \to q) \land (\neg p \leftrightarrow q)$$

р	q	¬р	$p \rightarrow q$	¬p ↔ q	(p→q)∧ (p↔q)
Т	Т				
Т	F				
F	Т				
F	F				

CS 441 Discrete mathematics for CS





Constructing the truth table

 Examples: Construct a truth table for (p → q) ∧ (¬p ↔ q)

р	q	¬р	$p \rightarrow q$	¬p ↔ q	(p→q)∧ (p↔q)
Т	Т	F	Т	F	F
Т	F	F	F	Т	F
F	Т	Т	Т	Т	Т
F	F	Т	Т	F	F

CS 441 Discrete mathematics for CS

M. Hauskrecht

Translation

Logic helps us to define the meaning of statements:

- Mathematical or English statements.

Question: How to translate an English sentence to the logic?

Assume a sentence:

- If you are older than 13 or you are with your parents then you can attend a PG-13 movie.
- The whole sentence is a proposition. It is **True.**
- But this is not the best. We want to parse the sentence to elementary statements that are combined with connectives.

CS 441 Discrete mathematics for CS

Translation

If you are older than 13 or you are with your parents then you can attend a PG-13 movie.

Parse:

- If (you are older than 13 or you are with your parents) then (you can attend a PG-13 movie)
 - A= you are older than 13
 - B= you are with your parents
 - C=you can attend a PG-13 movie
- Translation: $A \vee B \rightarrow C$
- Why do we want to do this?
- Inference: Assume I know that $A \vee B \rightarrow C$ is true and A is true. Then we can conclude that C is true as well.
- $A \lor B \to C$ and A are both true then C is true

CS 441 Discrete mathematics for CS

M. Hauskrecht

Translation

- General rule for translation.
- Look for patterns corresponding to logical connectives in the sentence and use them to define elementary propositions.
- Example:

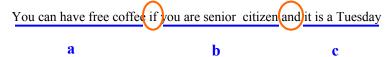
You can have free coffee if you are senior citizen and it is a Tuesday

Step 1 find logical connectives

CS 441 Discrete mathematics for CS

Translation

- · General rule for translation.
- Look for patterns corresponding to logical connectives in the sentence and use them to define elementary propositions.
- Example:



Step 2 break the sentence into elementary propositions

CS 441 Discrete mathematics for CS

M. Hauskrecht

Translation

- · General rule for translation.
- Look for patterns corresponding to logical connectives in the sentence and use them to define elementary propositions.
- Example:

You can have free coffee if you are senior citizen and it is a Tuesday

b

c

Step 3 rewrite the sentence in propositional logic

 $b \wedge c \rightarrow a$

CS 441 Discrete mathematics for CS

Translation

- Assume two elementary statements:
 - p: you drive over 65 mph; q: you get a speeding ticket
- Translate each of these sentences to logic
 - you do not drive over 65 mph. (¬p)
 - you drive over 65 mph, but you don't get a speeding ticket. (p ∧ ¬q)
 - you will get a speeding ticket if you drive over 65 mph. $(p \rightarrow q)$
 - if you do not drive over 65 mph then you will not get a speeding ticket. $(\neg p \rightarrow \neg q)$
 - driving over 65 mph is sufficient for getting a speeding ticket. $(p \rightarrow q)$
 - you get a speeding ticket, but you do not drive over 65 mph. (q \rightarrow \forall p)

CS 441 Discrete mathematics for CS

M. Hauskrecht

Computer representation of True and False

We need to encode two values True and False:

- Computers represents data and programs using 0s and 1s
- Logical truth values True and False
- A bit is sufficient to represent two possible values:
 - 0 (False) or 1(True)
- A variable that takes on values 0 or 1 is called a **Boolean** variable.
- <u>Definition</u>: A bit string is a sequence of zero or more bits. The **length** of this string is the number of bits in the string.

CS 441 Discrete mathematics for CS M. Hauskrecht

Bitwise operations

• T and F replaced with 1 and 0

р	q	p ∨ q	p∧q
1	1	1	1
1	0	1	0
0	1	1	0
0	0	0	0

р	¬р
1	0
0	1

CS 441 Discrete mathematics for CS

M. Hauskrecht

Bitwise operations

• Examples:

$$\vee 0110 \ 1010 \qquad \wedge \ 0110 \ 1010$$

$$\oplus$$
 0110 1010

Bitwise operations

• Examples:

 $\begin{array}{c} 1011\ 0011 \\ \vee\ \underline{0110\ 1010} \\ 1111\ 1011 \end{array}$

1011 0011 ^ 0110 1010 1011 0011

 \oplus 0110 1010

CS 441 Discrete mathematics for CS

M. Hauskrecht

Bitwise operations

• Examples:

 $\begin{array}{r}
1011\ 0011 \\
 \times \ \underline{0110\ 1010} \\
1111\ 1011
\end{array}$

 $\begin{array}{c} 1011\ 0011 \\ \land\ \underline{0110\ 1010} \\ 0010\ 0010 \end{array}$

 $\begin{array}{c} 1011\ 0011 \\ \oplus \ \ \underline{0110\ 1010} \\ 1101\ 1001 \end{array}$

CS 441 Discrete mathematics for CS

Tautology and Contradiction

• Some propositions are interesting since their values in the truth table are always the same

Definitions:

- A compound proposition that is always true for all possible truth values of the propositions is called a **tautology**.
- A compound proposition that is always false is called a **contradiction**.
- A proposition that is neither a tautology nor contradiction is called a **contingency**.

Example: $p \lor \neg p$ is a **tautology.**

р	٦p	p ∨ ¬p
Т	F	Т
F	Т	Т

CS 441 Discrete mathematics for CS

M. Hauskrecht

Tautology and Contradiction

• Some propositions are interesting since their values in the truth table are always the same

Definitions:

- A compound proposition that is always true for all possible truth values of the propositions is called a **tautology**.
- A compound proposition that is always false is called a contradiction.
- A proposition that is neither a tautology nor contradiction is called a **contingency**.

Example: $p \land \neg p$ is a contradiction.

р	٦p	p ∧ ¬p
T	F	F
F	Т	F

CS 441 Discrete mathematics for CS

- We have seen that some of the propositions are equivalent. Their truth values in the truth table are the same.
- Example: $p \rightarrow q$ is equivalent to $\neg q \rightarrow \neg p$ (contrapositive)

р	q	$p \rightarrow q$	$\neg q \rightarrow \neg p$
Т	Т	Т	Т
Т	F	F	F
F	Т	Т	Т
F	F	Т	Т

 Equivalent statements are important for logical reasoning since they can be substituted and can help us to make a logical argument.

CS 441 Discrete mathematics for CS

M. Hauskrecht

Logical equivalence

<u>Definition</u>: The propositions p and q are called <u>logically</u> equivalent if $p \leftrightarrow q$ is a tautology (alternately, if they have the same truth table). The notation $p \le q$ denotes p and q are logically equivalent.

Example of important equivalences

- DeMorgan's Laws:
- 1) $\neg (p \lor q) \iff \neg p \land \neg q$
- 2) $\neg (p \land q) \iff \neg p \lor \neg q$

Example: Negate "The summer in Mexico is cold and sunny" with DeMorgan's Laws

Solution: ?

CS 441 Discrete mathematics for CS

<u>Definition</u>: The propositions p and q are called <u>logically</u> equivalent if p ↔ q is a tautology (alternately, if they have the same truth table). The notation p <=> q denotes p and q are logically equivalent.

Example of important equivalences

- DeMorgan's Laws:
- 1) $\neg (p \lor q) \iff \neg p \land \neg q$
- 2) $\neg (p \land q) \iff \neg p \lor \neg q$

Example: Negate "The summer in Mexico is cold and sunny" with DeMorgan's Laws

Solution: "The summer in Mexico is not cold or not sunny."

CS 441 Discrete mathematics for CS

M. Hauskrecht

Equivalence

Example of important equivalences

- DeMorgan's Laws:
- 1) $\neg (p \lor q) \iff \neg p \land \neg q$
- 2) $\neg (p \land q) \iff \neg p \lor \neg q$

To convince us that two propositions are logically equivalent use the truth table

р	q	¬р	¬q	¬(p ∨ q)	p ∧ q¬
Т	Т	F	F		
Т	F	F	Т		
F	Т	Т	F		
F	F	Т	Т		

CS 441 Discrete mathematics for CS

Example of important equivalences

- DeMorgan's Laws:
- 1) $\neg (p \lor q) \iff \neg p \land \neg q$
- 2) $\neg (p \land q) \iff \neg p \lor \neg q$

To convince us that two propositions are logically equivalent use the truth table

р	q	¬р	¬q	¬(p ∨ q)	pr ∧ qr
Т	Т	F	F	F	
Т	F	F	Т	F	
F	Т	Т	F	F	
F	F	Т	Т	Т	

CS 441 Discrete mathematics for CS

M. Hauskrecht

Equivalence

Example of important equivalences

- DeMorgan's Laws:
- 1) $\neg (p \lor q) \iff \neg p \land \neg q$
- 2) $\neg (p \land q) \iff \neg p \lor \neg q$

To convince us that two propositions are logically equivalent use the truth table

р	q	¬р	¬q	¬(p ∨ q)	¬p ∧ ¬q
T	Т	F	F	F	F
Т	F	F	Т	F	F
F	Т	Т	F	F	F
F	F	Т	Т	Т	Т

CS 441 Discrete mathematics for CS

Example of important equivalences

- DeMorgan's Laws:
- 1) $\neg (p \lor q) \iff \neg p \land \neg q$
- 2) $\neg (p \land q) \iff \neg p \lor \neg q$

To convince us that two propositions are logically equivalent use the truth table

р	q	٦p	¬q	¬(p ∨ q)	¬p ∧ ¬q
Т	Т	F	F	F	F
Т	F	F	Т	F	F
F	Т	Т	F	F	F
F	F	Т	Т	T	T

CS 441 Discrete mathematics for CS

M. Hauskrecht

Important logical equivalences

- Identity
 - $p \wedge T \iff p$
 - $p \lor F \iff p$
- Domination
 - $p \lor T \iff T$
 - $p \wedge F \iff F$
- Idempotent
 - $p \lor p \iff p$
 - $\ p \wedge p \ <=> p$

CS 441 Discrete mathematics for CS

Important logical equivalences

- Double negation
 - $\neg (\neg p) \iff p$
- Commutative
 - $p \lor q \iff q \lor p$
 - $p \wedge q \iff q \wedge p$
- Associative
 - $-\ (p\vee q)\vee r\ <=>\ p\vee (q\vee r)$
 - $-(p \wedge q) \wedge r \iff p \wedge (q \wedge r)$

CS 441 Discrete mathematics for CS

M. Hauskrecht

Important logical equivalences

- Distributive
 - $p \lor (q \land r) \iff (p \lor q) \land (p \lor r)$
 - $p \wedge (q \vee r) \iff (p \wedge q) \vee (p \wedge r)$
- De Morgan
 - $\neg (p \lor q) <=> \neg p \land \neg q$
 - $\quad (p \land q) \iff \neg p \lor \neg q$
- Other useful equivalences
 - $p \lor \neg p <=> T$
 - $-p \land \neg p \iff F$
 - $p \rightarrow q \iff (\neg p \lor q)$

CS 441 Discrete mathematics for CS