CSE 311 Foundations of Computing I
Autumn 2012
Lecture 2
More Propositional Logic
Application: Circuits
Propositional Equivalence

Administrative

• Course web: [http://www.cs.washington.edu/311](http://www.cs.washington.edu/311)
  – Check it often: homework, lecture slides
• Office Hours: 2 × 7 = 14 hours; check the web
• Homework:
  – Paper turn-in (stapled) handed in at the start of class on due date (Wednesday); no online turn in.
  – Individual. OK to discuss with a couple of others but nothing recorded from discussion and write-up done much later
  – Homework 1 available (on web), due October 3

Recall…Connectives

<table>
<thead>
<tr>
<th>p</th>
<th>~p</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ∧ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ∨ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ⊕ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

• Implication
  – p implies q
  – whenever p is true q must be true
  – if p then q
  – q if p
  – p is sufficient for q
  – p only if q

“If pigs can whistle then horses can fly”
“If you behave then I’ll buy you ice cream”

What if you don’t behave?

Converse, Contrapositive, Inverse

- Implication: $p \rightarrow q$
- Converse: $q \rightarrow p$
- Contrapositive: $\neg q \rightarrow \neg p$
- Inverse: $\neg p \rightarrow \neg q$

- Are these the same?

Biconditional $p \leftrightarrow q$

- $p$ iff $q$
- $p$ is equivalent to $q$
- $p$ implies $q$ and $q$ implies $p$

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \leftrightarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

English and Logic

- You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old
  - $q$: you can ride the roller coaster
  - $r$: you are under 4 feet tall
  - $s$: you are older than 16

Digital Circuits

- Computing with logic
  - $T$ corresponds to 1 or “high” voltage
  - $F$ corresponds to 0 or “low” voltage

- Gates
  - Take inputs and produce outputs = Functions
  - Several kinds of gates
  - Correspond to propositional connectives
    - Only symmetric ones (order of inputs irrelevant)

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \land q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Gates

- AND connective $p \land q$
- AND gate

```
<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

"block looks like D of AND"
Logical equivalence

- Terminology: A compound proposition is a
  - Tautology if it is always true
  - Contradiction if it is always false
  - Contingency if it can be either true or false

\[ p \lor \neg p \]
\[ p \equiv p \]
\[ (p \rightarrow q) \land p \]
\[ (p \land q) \lor (p \land \neg q) \lor (\neg p \land q) \lor (\neg p \land \neg q) \]
De Morgan’s Laws
\[ \neg (p \land q) \equiv \neg p \lor \neg q \]
\[ \neg (p \lor q) \equiv \neg p \land \neg q \]

What are the negations of:
- The Yankees and the Phillies will play in the World Series
- It will rain today or it will snow on New Year’s Day

Law of Implication
Example: \( (p \rightarrow q) \equiv (\neg p \lor q) \)

<table>
<thead>
<tr>
<th>( p )</th>
<th>( q )</th>
<th>( p \rightarrow q )</th>
<th>( \neg p )</th>
<th>( \neg p \lor q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Computing equivalence
- Describe an algorithm for computing if two logical expressions/circuits are equivalent
- What is the run time of the algorithm?

Understanding connectives
- Reflect basic rules of reasoning and logic
- Allow manipulation of logical formulas
  - Simplification
  - Testing for equivalence
- Applications
  - Query optimization
  - Search optimization and caching
  - Artificial Intelligence
  - Program verification

Properties of logical connectives
- Identity
- Domination
- Idempotent
- Commutative
- Associative
- Distributive
- Absorption
- Negation
Equivalences relating to implication
• \( p \rightarrow q \equiv \neg p \lor q \)
• \( p \rightarrow q \equiv \neg q \rightarrow \neg p \)
• \( p \lor q \equiv \neg p \rightarrow q \)
• \( p \land q \equiv \neg (p \rightarrow \neg q) \)
• \( p \leftrightarrow q \equiv (p \rightarrow q) \land (q \rightarrow p) \)
• \( p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q \)
• \( \neg (p \leftrightarrow q) \equiv p \leftrightarrow \neg q \)

Logical Proofs
• To show \( P \) is equivalent to \( Q \)
  – Apply a series of logical equivalences to subexpressions to convert \( P \) to \( Q \)
• To show \( P \) is a tautology
  – Apply a series of logical equivalences to subexpressions to convert \( P \) to \( T \)

Show \((p \land q) \rightarrow (p \lor q)\) is a tautology